Editores:

José María de Fuentes - Lorena González - Jose Carlos Sancho - Ana Ayerbe - María Luisa Escalante

Investigación en Ciberseguridad
Actas de las VII Jornadas Nacionales
(JNIC 2022)

27-29 de junio de 2022

Palacio Euskalduna, Bilbao

Editores:

José María de Fuentes

Lorena González

Jose Carlos Sancho

Ana Ayerbe

María Luisa Escalante

I.S.B.N : 978-84-88734-13-6

Tabla de contenidos

Bienvenida del comité organizador

Las VII Jornadas Nacionales de Investigación en Ciberseguridad (JNIC) se celebran en Bilbao, del 27 al 29 de junio de 2019, organizadas por TECNALIA junto con INCIBE. Esta edición de las JNIC es muy especial, dado que hemos podido volver a celebrarlas presencialmente tras haber tenido que suspender la edición del 2020 debido a la pandemia y de haber tenido que celebrar las del 2021 en modo remoto. Esto hace que las hayamos preparado con toda la ilusión del mundo y pensando en ese re-encuentro físico del ecosistema investigador en ciberseguridad nacional que nos permita consolidar y tejer nuevas relaciones que puedan materializarse en nuevos proyectos de investigación en ciberseguridad de especial transcendencia.

Durante estos tres días tenemos un amplio programa con sesiones sobre Seguridad Web e Internet, Formación e Innovación Educativa, Vulnerabilidades y Ciberamenazas, Seguridad IoT/IIoT e ICS, Mecanismos de protección del usuario, y Criptografía y herramientas matemáticas. En esta edición de las JNIC, se han recibido 95 trabajos, de los cuales finalmente se han admitido 77 para su presentación en las jornadas (43 en formato de comunicación oral y 34 en formato de póster), además de dos trabajos de estudiantes, premiados como mejor Tesis y Trabajo Fin de Master relacionados con la ciberseguridad. En esta edición, se cuenta con dos "Special Issues" de revistas indexadas en el JCR en posiciones relevantes, "Future Generations Computer Systems" (FGCS) (Q1) y "Wireless Networks" (WINET)(Q2) donde los mejores artículos serán invitados a enviar versiones extendidas. El programa cuenta además con dos conferencias invitadas y tres mesas redondas que esperamos disfrutéis.

En esta edición también se celebra el Capture The Flag JNIC, una competición virtual que pretende detectar y premiar a las nuevas promesas de la ciberseguridad del ámbito nacional. Los retos presentados tienen distintos niveles de dificultad y abordan diversas disciplinas de la ciberseguridad, entre ellas la Ingeniería Inversa, Exploiting. Análisis Forense, Hacking web, y Criptografía y esteganografía.

El programa se complementa con actividades sociales donde poder conocer mejor Bilbao desde los puntos de vista gastronómico, cultural y deportivo, como pequeña pero representativa parte del País Vasco.

Finalmente agradecer a todos los que habéis ayudado a que podamos celebrar esta edición, sin vuestra ayuda y colaboración no hubiese sido posible. Esperamos que sea de vuestro agrado, que disfrutéis las JNIC, como el gran evento de la I+D en ciberseguridad que es, y queden ganas de unas JNIC 2023 presenciales con todo su esplendor.

Ana Ayerbe y Oscar Lage
Presidentes del Comité Organizador
Lorena Gonzalez y Jose Maria de Fuentes
Presidentes del Comité de Programa Científico
Jose Carlos Sancho
Presidente del Comité de Programa de Formación e Innovación Educativa

Comité ejecutivo

Cristina Alcaraz Tello	Universidad de Málaga
Ana Ayerbe Fernandez-Cuesta	TECNALIA
Noemí de Castro García	Universidad de León
Juan Díez González	INCIBE
Rafael María Estepa Alonso	Universidad de Sevilla
Eduardo Fernández-Medina Patón	Universidad de Castilla-La Mancha
Pedro García Teodoro	Universidad de Granada. Representante de red RENIC
Pedro Peris López	Universidad Carlos III de Madrid
Jose Carlos Sancho	Universidad de Extremadura

Comité organizador

Presidentes del comité organizador

Ana Ayerbe Fernandez-Cuesta	TECNALIA
Oscar Lage	TECNALIA

Finanzas

Elena Urrutia Beldarrain	TECNALIA

Actas

Marisa Escalante Martínez	TECNALIA

Difusión

Eva Salgado Etxebarria	TECNALIA
Ana Diaz de Zugazua Salazar	TECNALIA

WEB

Alberto Berreteaga Barbero	TECNALIA
Marisa Escalante Martínez	TECNALIA

Logística y Organización

Eva Salgado Etxebarria	TECNALIA
Ana Diaz de Zugazua Salazar	TECNALIA

Comité de Programa Científico

Presidentes

Jose Maria de Fuentes García-Romero de Tejada	Universidad Carlos III de Madrid
Lorena González Manzano	Universidad Carlos III de Madrid

Miembros

Cristina Alcaraz Tello	Universidad de Málaga
Aitana Alonso Nogueira	INCIBE
Ana Ayerbe Fernández-Cuesta	TECNALIA
Marta Beltrán Pardo	Universidad Rey Juan Carlos
Carlos Blanco Bueno	Universidad de Cantabria
Jorge Blasco Alís	Royal Holloway, University of London
Pino Caballero Gil	Universidad de La Laguna
Andrés Caro Lindo	Universidad de Extremadura
Jordi Castellà Roca	Universitat Rovira i Virgili
Jesús Esteban Díaz Verdejo	Universidad de Granada
Josep Lluis Ferrer Gomila	Universitat de les Illes Balears
David García Rosado	Universidad de Castilla-La Mancha
Pedro García Teodoro	Universidad de Granada
Luis Javier García Villalba	Universidad Complutense de Madrid
Joaquín García-Alfaro	Instituto Politécnico de París
Manuel Gil Pérez	Universidad de Murcia
Félix Gómez Mármol	Universidad de Murcia
María Isabel González Vasco	Universidad Rey Juan Carlos I
Julio César Hernández Castro	University of Kent
Luis Hernández Encinas	Consejo Superior de Investigaciones Científicas
Jorge López Hernández-Ardieta	Verisure

Javier López Muñoz	Universidad de Málaga
Agustín Martín Muñoz	Consejo Superior de Investigaciones Científicas
Rafael Martínez Gasca	Universidad de Sevilla
Gregorio Martínez Pérez	Universidad de Murcia
David Megías Jiménez	Universitat Oberta de Cataluña
Raul Orduña Urrutia	Vicomtech
Luis Panizo Alonso	Universidad de León
Aljosa Pasic	ATOS
Cristina Regueiro Senderos	TECNALIA
Erkuden Rios Velasco	TECNALIA
Margarita Robles Carrillo	Universidad de Granada
Ricardo J Rodríguez	Universidad de Zaragoza
Luis Enrique Sánchez Crespo	Universidad de Castilla-La Mancha
José Soler	Technical University of Denmark - DTU
Victor A Villagrá González	Universidad Politécnica de Madrid
Urko Zurutuza Ortega	Mondragon Unibertsitatea

Comité de Programa de Formación e Innovación Educativa

Presidentes

Jose Carlos Sancho	Universidad de Extremadura

Miembros

Isaac Agudo Ruiz	Universidad de Málaga
Mar Ávila Vegas	Universidad de Extremadura
Noemí De Castro García	Universidad de León
David García Rosado	Universidad de Castilla - La Mancha
Iñaki Garitano Garitano	Mondragon Unibertsitatea
Ana Isabel González-Tablas	Universidad Carlos III de Madrid
Xavier Larriva	Universidad Politécnica de Madrid
Roberto Magán Carrión	Universidad de Granada
Óscar Mogollón Gutiérrez	Universidad de Extremadura
Ana Lucila Sandoval Orozco	Universidad Complutense de Madrid
Adriana Suárez Corona	Universidad de León
Ángel Jesús Varela Vaca	Universidad de Sevilla

Comunicaciones

Sesión I: Seguridad web e Internet

Extracción de variables para caracterización multi-clase de la severidad de IPs

David Escudero García
RIASC. Universidad de León
Campus de Vegazana s/n 24071
descg@unileon.es

Noemí DeCastro-García
Dpto. de Matemáticas. Universidad de León.
Campus de Vegazana s/n 24071
ncasg@unileon.es

Miguel V. Carriegos
Dpto. de Matemáticas. Universidad de León.
Campus de Vegazana s/n 24071
miguel.carriegos@unileon.es

***Resumen*—Determinar la severidad de un incidente de ciberseguridad es fundamental para establecer medidas efectivas contra el mismo. En este contexto, el aprendizaje automático es utilizado para crear modelos capaces de clasificar y predecir la peligrosidad de los eventos de ciberseguridad. Uno de los aspectos más importantes en el uso de este tipo de técnicas es la extracción de variables que permitan obtener modelos eficientes.**

El objetivo de este trabajo es construir un conjunto de variables o *features* que caracterice la maliciosidad de una dirección IP de manera multi-clase. La configuración final son 23 variables: 18 de ellas obtenidas mediante series temporales y listas de reputación, y 5 relacionadas con la geolocalización de la IP. No solo se han extraído las features, sino que se ha realizado un análisis estadístico para estudiar su adecuación y optimización. En el caso de las variables de geolocalización, por los posibles cambios que pueden sufrir en el tiempo. En el caso de las series temporales, por los hiper-parámetros inherentes a la construcción de las variables.

***Index Terms*—Severidad, aprendizaje automático, selección de variables, direcciones IP**

Tipo de contribución: *Investigación original*

I. Introducción

El objetivo de este trabajo es extraer un conjunto de variables o *features* estadísticamente significativo que caracterice, mediante aprendizaje automático, la peligrosidad o maliciosidad de una dirección IP de manera multi-clase. Se propone un conjunto de variables conformado por características que podamos encontrar en herramientas externas sobre localización geográfica de una dirección IP, y características construidas mediante series temporales y consulta a listas negras o de reputación de direcciones IP.

El conjunto final tiene 23 variables, 18 relacionadas con información temporal, y 5 con información geográfica. El análisis de la significancia de las mismas se basa en las limitaciones que presentan. Por un lado, las variables de geolocalización pueden sufrir cambios con el paso del tiempo. En este caso, las preguntas de investigación van dirigidas a determinar si existe deriva conceptual y, en caso afirmativo, si tiene efecto sobre los resultados de los modelos obtenidos. Por otro lado, las características temporales dependerán de dos hiper-parámetros que definen las ventanas temporales. Las preguntas de investigación se dirigen a determinar el efecto de los mismos en el ajuste de los modelos de clasificación.

El estudio se ha realizado sobre un conjunto de 99720 IPs proporcionado por INCIBE [1]. Se han llevado a cabo diferentes análisis descriptivos e inferenciales. Los resultados muestran que uno de los hiper-parámetros de los que depende la extracción de las variables temporales tiene un efecto significativo sobre los resultados alcanzados. Por otra parte, los cambios en la geolocalización no parecen implicar una degradación de los modelos.

Este artículo está organizado de la siguiente manera: en la sección 2, se desarrolla el trabajo relacionado. En la sección 3, se describe el proceso de construcción y obtención de las variables de caracterización. En la sección 4, se incluyen todos los detalles experimentales del estudio. En la sección 5, se desglosan y discuten los resultados obtenidos. Finalmente, se incluyen las conclusiones, los agradecimientos y las referencias.

II. Trabajo relacionado

Caracterizar la severidad de un evento de ciberseguridad, entendida como una medida del riesgo que supone, es fundamental para poder reaccionar de una manera eficiente ante el mismo. Actualmente, existen diferentes metodologías y estándares que asignan una puntuación para evaluar la severidad de eventos de ciberseguridad, y que se basan en taxonomías, o aplicación de consultas en informes internacionales (*Microsoft Security Bulletin Vulnerability Rating*, [1],*Common Vulnerability Scoring System* (CVSS)[2],*Open Web Application Security Project* (OWASP) *Risk Rating Methodology*, [3], *Cyber Incident Scoring System* , [4], entre otros). Recientemente, también se han utilizado técnicas de aprendizaje automático para esta tarea ([5]). En este último trabajo, la severidad de eventos de ciberseguridad de diferente naturaleza es caracterizada, de forma multi-clase, mediante 113 variables recogidas por un *Computer Emergency Response Team* (CERT).

En particular, si hablamos de la severidad de una dirección IP, suele ser habitual entender la misma como su reputación. El enfoque tradicional para determinar la maliciosidad de una IP se basa en el uso de *blacklists* que contienen conjuntos de IPs que han sido detectadas llevando a cabo comportamientos maliciosos. Muchas herramientas para la gestión de *firewalls*, como FireHOL [6], agrupan información de varias *blacklists* para bloquear eficientemente IPs sospechosas. Existen varios trabajos centrados en la extracción de información de *blacklists* para la predicción de la maliciosidad de una IP. En [7] se agregan las IPs en subredes usando un prefijo CIDR seleccionado y se construye una serie temporal usando como magnitud el número de IPs de la subred presentes en *blacklists* en diferentes intervalos de tiempo. Estas series temporales se dividen en 3 franjas ”buena”, ”normal”, y ”mala” según si el

[1]La publicación del *dataset* se encuentra en estudio legal en la fecha de envío de este trabajo

valor de la serie temporal está por debajo, en o por encima de la media. De estas series temporales se extraen características como el valor medio en cada franja, o la proporción de tiempo en la que la serie temporal permanece en una franja. Se alcanza un ratio de verdaderos positivos de 0.7. Otro enfoque se presenta en [8], en el que usando técnicas de clustering se determina qué partes del espacio de direcciones IP contienen una mayor frecuencia de IPs maliciosas. Se obtiene una tasa de acierto de 0.776. En [9] también se usa clustering, agrupando las IPs a clasificar de forma que maximice la dependencia entre la pertenencia de un IP a un cluster y su presencia en una *blacklist*. Este esquema alcanza precisiones de en torno a 0.8, pero requiere que se repita el proceso de clustering para actualizarse a cambios en la blacklist. En estos casos, el proceso se basa en la hipótesis de que direcciones IP de una misma subred o de redes contiguas tienen una mayor probabilidad de compartir un grado de maliciosidad, por lo que es necesario disponer de un volumen relativamente alto de IPs para poder crear un modelo que no sea demasiado local y sirva para extender las predicciones a un espectro amplio de IPs.

Por otro lado, existen enfoques basados en el uso de información adicional sobre la IP y su comportamiento como la geolocalización o registros DNS asociados para obtener predicciones más generalizables. Servicios como Maxmind [10] permiten obtener información sobre la geolocalización de la IP; otros como IPQualityScore [11] proporcionan un nivel de maliciosidad basándose en ciertas características de la IP como el contenido, registros DNS, etc. El trabajo en [12] propone una herramienta que utiliza información de geolocalización, además de la propia IP o dominio tratado, para predecir su maliciosidad. Se obtiene una tasa de acierto de 0.75 con el mejor modelo, que supera a las tasas de acierto obtenidas por otras fuentes como VirusTotal que se evalúan en el artículo. Otros trabajos como [13] se centran en la detección de IPs maliciosas a partir del tráfico web y de correo eléctrónico usando features como el volumen de peticiones, el número de correos de spam recibidos de una IP, etc. Las tasas de acierto alcanzadas son más altas que otros métodos basados en información contextual de la IP, pero el proceso de monitorización y análisis del tráfico es costoso. En [14] se propone un esquema más complejo que combina información de fuentes externas como una medida de fiabilidad de las predicciones, análisis de muestras de malware asociadas y análisis de las ocurrencias de cada IP en el tiempo. La tasa de acierto llega a alcanzar un 93 % en el mejor modelo, pero el procedimiento de análisis es complejo y la obtención de la información asociada conlleva un importante despliegue de recursos.

En general, métodos más simples alacanzan una tasa de acierto limitada, inferior a 0.8. Métodos como el propuesto en [14] son más eficaces pero todo el proceso de obtención y análisis de muestras de malware y análisis del tráfico requiere una infraestructura que limita su aplicabilidad.

Otra de las posibles limitaciones existentes en la caracterización de una dirección IP es el cambio a lo largo del tiempo de la misma. Su geolocalización puede cambiar, el dominio asociado puede ser diferente y, aunque una IP pueda estar asociada a actividad maliciosa en un instante de tiempo, puede no estarlo más tarde: quizás el equipo original fuese infectado pero se ha solventado el problema. En muchos trabajos se sugiere que es necesario mantener actualizados los modelos para ajustarse a estos cambios, pero esto conlleva un cierto consumo de recursos, así que sería deseable poder estimar el impacto que los cambios en la caracterización de las IPs tienen sobre los modelos. Estos cambios en los datos se pueden analizar bajo el marco teórico de deriva conceptual. La deriva conceptual es el cambio en la distribución de los datos en escenarios dinámicos de aprendizaje. Sea X el espacio de vectores de features o variables de una muestra de datos, y $P(X)$ la distribución de probabilidad marginal. Además, sea Y el espacio de etiquetas de X. En términos matemáticos, se define un *concepto* como la distribución conjunta de X e Y, $P(X, Y)$ ([15]). Si denotamos la distribución marginal de los datos en un instante t como $P_t(X)$, y la distribución condicional (posterior) de las etiquetas de los mismos mediante $P_t(Y \mid X)$, entonces la deriva conceptual ocurre cuando $P_t(y \mid X) \neq P_{t+\triangle t}(y \mid X)$ y/o $P_t(X) \neq P_{t+\triangle t}(X)$. Este puede ser el caso, por ejemplo, de la geolocalización de una dirección IP si, por ejemplo, esta cambia porque se asigna a un lugar diferente ($P_t(X)$ cambia) o se reciben muchas alertas de IPs maliciosas procedentes de un país particular ($P_t(Y \mid X)$ cambia). Aunque existen caracterizaciones más generales de la deriva conceptual [16], en términos generales esta puede ser de tres tipos ([15], [17]): real ($P_t(Y \mid X) \neq P_{t+\triangle t}(Y \mid X)$ pero $P_t(X) = P_{t+\triangle t}(X)$), virtual ($P_t(X) \neq P_{t+\triangle t}(X)$ pero $P_t(Y \mid X) = P_{t+\triangle t}(Y \mid X)$), o ambas ($P_t(Y \mid X) \neq P_{t+\triangle t}(Y \mid X)$ y $P_t(X) \neq P_{t+\triangle t}(X)$). Una de las líneas de investigación actuales sobre deriva conceptual está dirigida a encontrar técnicas y algoritmos que puedan detectarla. En este trabajo, se destaca el cálculo de la distancia entre los conceptos de los periodos t y $t + \triangle t$ dada en [18] mediante el concepto de Magnitud usando como distancia la variación total de Levin [19] y su versión corregida para el cálculo de la deriva condicional:

$$\sigma_{t,t+\triangle t}(Z) = \frac{1}{2} \sum_{\overline{z} \in Dom(Z)} \mid P_t(\overline{z}) - P_{t+\triangle t}(\overline{z}) \mid \text{ y} \qquad (1)$$

$$\sigma^{Y|X}_{t,t+\triangle t} = \sum [\frac{P_t(\overline{x}) + P_{t+\triangle t}(\overline{x})}{2} \frac{1}{2} \sum \mid P_t(y \mid \overline{x}) - P_{t+\triangle t}(y \mid \overline{x}) \mid] \qquad (2)$$

Otra limitación existente en la caracterización de la maliciosidad de las IPs está en que la mayoría de los trabajos tratan un problema biclase: distinguir entre IPs maliciosas y no maliciosas. En este trabajo, tratamos el problema de asignar un nivel de maliciosidad asociado a la IP.

En este trabajo, seguimos un enfoque mixto para la caracterización de las IPs por su severidad. Usamos features derivadas de *blacklists* como en [7], pero añadimos *features* relacionadas con la geolocalización de la IP. El conjunto de *features* resultantes es ligero y no requiere todo el procesado de apoyo de herramientas como la presentada en [14]. Además, realizamos un estudio del impacto de la deriva conceptual en la predicción de la severidad asociadas a IPs, así como de la posible optimización de los parámetros que afectan a la construcción de las variables extraídas mediante series temporales.

III. Variables de caracterización

Las variables de caracterización que se proponen en este trabajo son de diferente naturaleza. Por un lado, ciertas propiedades de una serie temporal que puede crearse a través de consultas a listas de reputación o listas negras. Por otro lado, la geolocalización de la IP. En total, tendremos 23 *features* o variables, que denotaremos mediante F_i con $i = 1, \ldots, 23$.

El esquema de extracción de las variables de series temporales está basado en el trabajo presentado en [7]. Sea una red $r = A.B.C.0$ de direcciones IP. Se puede crear una serie temporal $\mathfrak{X}_r(t)$ que asigna a cada instante de tiempo t, el número de direcciones IP pertenecientes a r contenidas en las listas de referencia (listas negras o de reputación) en ese instante. Este es un proceso que puede realizarse sobre cualquier lista de IPs con un *time stamp* asociado.

A partir de un fragmento de la serie temporal $\{X_r(t_1), X_r(t_2), \ldots, X_r(t_h)\}$ de tamaño *h*, vamos a extraer un vector de 9 *features* $(F_1, \ldots, F_9)$. Las primeras tres coordenadas del vector responden a la intensidad de la ventana temporal. Las tres siguientes a la duración, y las tres últimas a la frecuencia.

Para construirlas, seguiremos el siguiente procedimiento:

1. Fijamos un número real $\delta > 0$.
2. Se calcula el número medio de IPs de la red que aparecen en las listas negras o de reputación durante el fragmento de la serie temporal elegido:
$$\mu = \frac{\sum_{i=1}^{h} X_r(t_i)}{h} \tag{3}$$
3. Asignamos un nivel o rango a cada instante t_i del fragmento de la ventana temporal: diremos que un instante t_k estará en un nivel si cumple las siguientes condiciones:
$$\text{Nivel} = \begin{cases} \text{Bajo} & \text{si } t_k : X_r(t_k) \leq (1-\delta)\mu \\ \text{Medio} & \text{si } t_k : (1-\delta)\mu < X_r(t_k) < (1+\delta)\mu \\ \text{Alto} & \text{si } t_k : (1+\delta)\mu \leq X_r(t_k) \end{cases} \tag{4}$$
4. El primer trío de *features* se relaciona con la intensidad: para cada nivel (bajos, medios, altos), la intensidad es el valor medio de la serie temporal en los instantes de tiempo de esa nivel. Esto es,
$$i(nivel) = \frac{\sum_{t_k \in nivel} X_r(t_k)}{|nivel|} \tag{5}$$
donde $|nivel|$ es el número total de instantes del fragmento temporal que han sido asignados a ese nivel. Así, $F_1 = i(bajos), F_2 = i(medios), F_3 = i(altos)$.
5. El segundo trío de *features* se relaciona con la duración: para cada franja de valores la duración es el número medio de instantes k en los que la serie temporal permanece en el nivel concreta (varios k consecutivos permanecen en el mismo nivel). Así, $F_4 = d(bajos), F_5 = d(medios), F_6 = d(altos)$.
6. El tercer trío de *features* se relaciona con la frecuencia: para cada nivel, la frecuencia es la proporción de instantes que pertenece a ese nivel:. Esto es,
$$f(nivel) = \frac{|nivel|}{|h|} \tag{6}$$
Así, $F_7 = f(bajos), F_8 = f(medios), F_9 = f(altos)$.

En la figura 1 puede encontrarse un ejemplo de una serie temporal construida a lo largo de 20 días. Fijemos $h = 5, \delta = 0.001$.

En este caso $X_r(t_1) = 3, X_r(t_2) = 6, X_r(t_3) = 1, X_r(t_4) = 9, X_r(t_5) = 2$. El número medio de IPs de la red que aparecen en la lista durante esta ventana temporal de cinco días es $\mu = 4.2$. Por lo tanto, t_1, t_3 y t_5 son instantes del nivel bajo, mientras que t_2 y t_4 son instantes del nivel alto. Así, $(F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9) = (2, 0, 7.5, 0, 0, 0, 0.6, 0, 0.4)$.

Puede observarse que las features $F_1, \ldots, F_9$ se asocian a redes de IPs. Por lo que dos IPs que pertenezcan a la misma red, tendrán los mismos valores en estas *features*. Este hecho también puede verse como una medida de cómo de cercanas son las IPs.

Cabe destacar que h es el tamaño de la ventana temporal que elegimos. Cuanto más pequeña sea, menos recursos , en datos y computacionales, requerirá la extracción de *features*. Si necesitmos una h muy elevada, entonces es posible que la extracción de las variables no resulte eficiente.

En lo anterior se han agrupado las IPs en redes de la forma A.B.C,0 esto es redes que tienen un sentido físico como la red local de un hogar. Para añadir otras nueve caracteristicas, agrupamos en redes que no tienen un sentido real pero consituyen una relación de equivalencia como otra cualquiera. A saber, las redes de la forma A.B,0.D esto es redes donde A, B, D se mantienen fijos y 0 puede ser cualquier número. Lo que nos da otras nueve características. Por lo tanto, F_i de $i = 1, \ldots, 18$.

Por otro lado, las *features* de geolocalización son las siguientes (F_i de $i = 19, \ldots, 23$):

- Latitud y longitud se conservan como números decimales (F_{19} y F_{20}).
- El código de país se categoriza en forma de número entero, cuyo valor va de 0 a al número de países representados según el código ISO 3166-1 (F_{21}).
- La IP se transforma a un valor numérico entero con la siguiente fórmula:
$$A.B.C.D \rightarrow A * 256^3 + B * 256^2 + C * 256^1 + D * 256^0$$
(F_{22}).
- La fecha de ocurrencia se transforma al tiempo UNIX (número de segundos transcurridos desde el 01/01/1970 a las 00:00) (F_{23}).

Algunas de las cuestiones que debemos analizar al utilizar las *features* seleccionadas son las siguientes:

1. Las características temporales que se han extraído dependen de dos hiper-parámetros, h y δ. ¿Tienen influencia en los resultados obtenidos?
2. Cabe esperar que los datos de la geolocalización de la IP cambien en el tiempo, ya sea por la reasignación de las direcciones o por imprecisiones en la geolocalización. Este cambio podría causar que el modelo pierda capacidad predictiva. Esta cuestión es de particular importancia en el caso de tratar con datos de listas públicas, ya que nuevas IPs tienden a incorporarse o eliminarse de listas negras. Uno de los objetivos de este estudio es evaluar el cambio en la geolocalización implica un

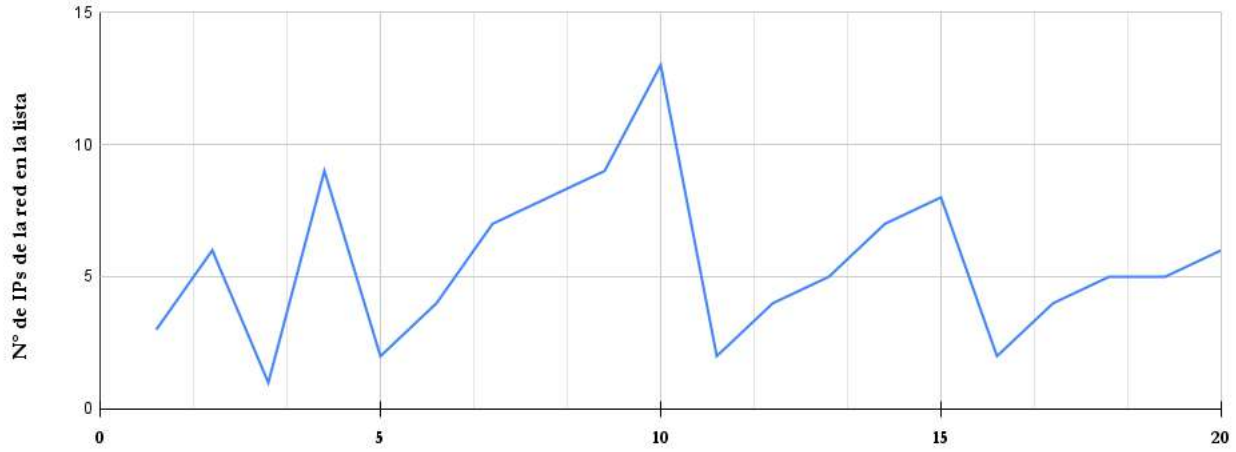


Figura 1. Ejemplo de serie temporal.Eje X: días. Eje Y: número de IPs de la red en la lista negra a cada instante

escenario de aprendizaje con deriva conceptual y, en ese caso, determinar si la capacidad predictiva disminuye . En caso positivo, habría que analizar en qué medida se degrada el modelo.

3. Si las cuestiones anteriores tienen efecto en la capacidad de los modelos de aprendizaje automático para clasificar IPs según su maliciosidad, habrá que determinar la configuración que nos aporte las mejores *features.*

IV. Sección experimental

En esta sección se describen los conjuntos de datos, las preguntas de investigación y los análisis realizados.

IV-A. Conjuntos de datos

Para la realización de este trabajo se ha utilizado un conjunto de datos de eventos de ciberseguridad aportado por INCIBE que corresponde al mes de mayo de 2021 [2]. Lo denotaremos por D. Se trata de un archivo CSV que contiene 99720 IPs. De cada IP se tiene la siguiente información:

1. IP.
2. Fecha del evento.
3. Latitud: extraída en el momento de recepción del evento.
4. Longitud: extraída en el momento de recepción del evento.
5. Código del país: extraída en el momento de recepción del evento.
6. Severidad asociada: el valor de severidad viene asignado por un experto de INCIBE. A su vez, se utiliza un modelo de aprendizaje automático para generar esta severidad [5], pero utilizando información de 113 variables que INCIBE recoge en su modelo de inteligencia. Es un valor multi-clase con 4 niveles (1, 3, 6 y 9, ordenados de menor a mayor severidad).

Trataremos este conjunto D como una lista de reputación o lista negra de IPs. El primer paso realizado ha sido curar el conjunto de datos. Para evitar el exceso de notación, lo volveremos a denotar D. Se eliminaron aquellos datos que contenían valores inválidos o incorrectos. El siguiente paso fue recalcular la variable Severidad para tener cuatro etiquetas o clases:

$$\text{Severidad} = \begin{cases} 1 & \text{si } Severidad_{antigua} \in 0, 1 \\ 3 & \text{si } Severidad_{antigua} \in 2, 3, 4, Low \\ 6 & \text{si } Severidad_{antigua} \in 5, 6, 7, Warning \\ 9 & \text{si } Severidad_{antigua} \in 8, 9, 10, High \end{cases} \tag{7}$$

A partir de D se van a extraer las variables $F_1, \ldots, F_{18}$ de las 99720 IPs. Para poder dar respuesta a las preguntas de investigación planteadas, para cada una de las IPs, realizamos una consulta a MaxMind para obtener la información de las columnas *latitud, longitud*, y *código del país*. La consulta se ha realizado en octubre de 2021 para comprobar si ha habido cambios y, por lo tanto, deriva conceptual. De estas 99720 IPs, 42677 de ellas tienen una geolocalización distinta de acuerdo con MaxMind [3]. Por lo tanto, se tendrán finalmente dos conjuntos de datos para crear los modelos de clasificación, D_1 y D_2. Ambos estarán formados por las 42677 IPs en las que ha habido cambios de geolocalización. Las features $F_1, \ldots, F_{18}$ serán iguales en ambos conjuntos de datos, así como la etiqueta asignada en la variable *Severidad*. Las variables $F_{19}, \ldots, F_{23}$ serán calculadas con la información contenida en las variables de geolocalización dadas por INCIBE en el conjunto D_1. Para D_2, las variables de geolocalización se calcularán con la información aportada por MaxMind. Cabe destacar, en términos de estudiar la deriva conceptual, que tomaremos $D_1 = D_t$ y $D_2 = D_{t+\triangle t}$.

La proporción de las clases (valores de severidad) en cada uno de los conjuntos se presenta en la tabla I.

IV-B. Preguntas de investigación

Las preguntas se dividen en los análisis sobre el conjunto de variables de geolocalización y las de series temporales.

PI1 ¿Hay deriva conceptual entre D_1 y D_2?

PI2 ¿Existen diferencias significativas entre los resultados obtenidos al utilizar D_1 y D_2? En caso positivo, ¿en qué conjunto se obtienen mejores resultados? ¿Hay degradación en los resultados?¿Es relevante?

[2]La publicación del *dataset* se encuentra en estudio legal en la fecha de envío de este trabajo

[3]https://www.maxmind.com/en/home

Tabla I
FRECUENCIAS DE LAS DIFERENTES CLASES EN LOS CONJUNTOS DE DATOS D_1, D_2 Y D.

Conjunto de datos	Severidad	Frecuencia	Proporción
D	1	8402	8.4255 %
	3	24943	25.0130 %
	6	54437	54.5898 %
	9	11938	11.9715 %
D_1 y D_2	1	2898	6.7907 %
	3	12317	28.8616 %
	6	22868	53.5851 %
	9	4593	10.7624 %

PI3 ¿Existen diferencias significativas entre los resultados obtenidos en las variables respuesta al variar el parámetro h en F_i con $i = 1, \ldots, 18$? En caso positivo, ¿con qué h se obtienen mejores resultados? ¿El efecto de h es relevante?

PI4 ¿Existen diferencias significativas entre los resultados obtenidos en las variables respuesta al variar el parámetro δ en F_i con $i = 1, \ldots, 18$? En caso positivo, ¿con qué δ se obtienen mejores resultados? ¿El efecto de δ es relevante?

IV-C. Análisis

Los modelos de clasificación se obtienen usando la herramienta *AutoSklearn* [20]. Se reserva un 75 % de los datos para el proceso de optimización de hiperparámetros y el 25 % restante se usa para evaluar el modelo con los hiperparámetros ya optimizados. Para optimizar los hiperparámetros utilizamos el método SMAC (Sequential Model-based Algorithm Configuration) [21]. Los experimentos se han realizado considerando los mejores resultados tras hacer *10-fold cross-validation.*

Los experimentos se llevan a cabo probando diferentes valores para los hiperparámetros h y d. Se fijan los valores de $h \in \mathbb{N}$ en el intervalo $[6, 10]$ y de $\delta = 0.001, 0.002, 0.003, 0.004]$. La muestra D es de 30 días, por lo que se toma un h máxima de 10.

Las variables respuesta analizadas son dos: el ajuste o *Accuracy*, y el coeficiente de correlación de Matthews o MCC [22] que se define de la siguiente manera:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{8}$$

donde TP, TN, FP, y FN denotan verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, respectivamente. Utilizamos esta métrica debido a que se considera más adecuada en el caso de conjuntos de datos no balanceados [23].

El resto de análisis llevados a cabo se enumeran a continuación:

1. Para analizar si las variables de geolocalización presentan deriva conceptual en los conjuntos de datos, se ha utilizado la aproximación dada en [18]. El grado de deriva se ha calculado para $P_{t,t+\triangle t}(X)$ y $P_{t,t+\triangle t}(Y \mid X)$, mediante el cálculo de $\sigma_{t,t+\triangle t}(Z)$ y $\sigma^{Y|X}_{t,t+\triangle t}$, véase Eq. (1) y Eq (2). Ambas toman valores entre 0 y 1, siendo más altas cuanta más deriva haya.
2. El primer análisis estadístico realizado es la prueba de normalidad de Kolmogorov–Smirnov con la corrección de Lilliefors. Al salir $\rho_{K-S}(D_1) = 0.002, \rho_{K-S}(D_2) = 0.000, \rho_{K-S}(D_1 \cup D_2) = 0.000$, los análisis inferenciales serán no paramétricos.
3. Para PI2, y debido a que los conjuntos de features D_1 y D_2 tienen valores diferentes en las variables de geolocalización, se ha aplicado el test U de Mann-Whitney para 2 muestras independientes. De esta manera, podremos determinar si existen diferencias estadísticamente significativas entre los resultados obtenidos. Si han existido diferencias, utilizamos un estudio descriptivo para determinar con qué conjunto obtenemos mejores resultados de clasificación de severidad.
4. En relación con la PI3, queremos determinar si existen diferencias significativas en las variables respuesta obtenidas cuando variamos el parámetro h en la construcción de las *features* correspondientes al bloque de series temporales. Al variar h, el valor de las features F_i con $i = 1, \ldots, 18$ cambia, por lo que lo conjuntos de datos con los que se construyen los modelos son diferentes y, por lo tanto, los análisis inferenciales utilizados han de ser para muestras independientes. Por otra parte, al utilizar cinco categorías para $h = 6, 7, 8, 9, 10$, se utilizará el test de Kruskall-Wallis para comparar los cinco grupos. En caso afirmativo, se realizarán comparaciones *Post Hoc* dos a dos y se utilizará la corrección de Bonferroni $\left(\overline{\alpha} = \frac{\alpha}{\text{número de combinaciones posibles}}\right)$.
5. Para PI4, se realizará un análisis similar al anterior comparando los resultados obtenidos cuando variamos el hiper-parámetro $\delta = 0.001, \ldots, 0.004$.
6. En las preguntas de investigación planteadas, no sólo se trata de analizar si hay diferencias estadísticamente significativas (ρ-valor < 0.05 sin corrección de Bonferroni), sino que también es necesario determinar la relevancia de las mismas. Para estudiar el efecto que tienen los grupos en aquellos casos en los que si aparecen diferencias, se ha utilizado la d de Cohen mediante la estandarización de las diferencias de medias. Este test se utiliza para medir la asociación entre variables cuantitativas (medidas en escala continua) y cualitativas (variables dicotómicas). Para interpretar el índice, se utiliza la escala descrita en Eq. 9 ([24]):

$$\text{Efecto} = \begin{cases} \text{Pequeño} & \text{si } d \in [0, 0.3] \\ \text{Medio} & \text{si } d \in [0.5, 0.8] \\ \text{Grande} & \text{si } d > 0.8 \end{cases} \tag{9}$$

7. Todos los análisis inferenciales se han hecho con $\alpha = 0.05$.

V. RESULTADOS

La sección de resultados está organizada en función de las preguntas de investigación planteadas.

V-A. PI1

En la tabla II, podemos encontrar los estadísticos descriptivos del cálculo de $\sigma_{t,t+\triangle t}(Z)$ y $\sigma^{Y|X}_{t,t+\triangle t}$. Los valores obtenidos no son muy elevados, por lo que podemos concluir que la deriva que se produce en las variables de geolocalización es baja. Además, hay más deriva virtual que real.

Tabla II
RESULTADOS DEL ANÁLISIS DE LA DERIVA CONCEPTUAL

Descriptivo	**Marginal** $\sigma_{t,u}(Z)$	**Posterior** $\sigma_{t,u}^{Y\|X}$
Media	0.2724	0.1362
Mediana	0.2746	0.1373
Desviación típica	0.3164	0.1582
Mínimo	0.2053	0.1027
Máximo	0.3371	0.1686

V-B. PI2

En la tabla III, podemos encontrar los resultados en las variables respuesta cuando comparamos los resultados de los modelos creados con D_1 y D_2.

Tabla III
RESULTADOS DEL TEST U DE MANN- WHITNEY PARA COMPARAR D_1 CON D_2

Variable respuesta	**Z**	ρ**-valor**
MCC	-2.929	0.003
Accuracy	-2.170	0.030

En ambos casos, hay diferencias significativas. Podemos ver en la tabla IV, los estadísticos descriptivos de ambas muestras.

Tabla IV
ESTADÍSTICOS DESCRIPTIVOS DE D_1 Y D_2

Variable respuesta		$\overline{X}$	σ	$Mediana$
MCC	D_1	0.7776	0.1168	0.7788
	D_2	0.7770	0.2732	0.7889
Accuracy	D_1	0.8640	0.0072	0.8648
	D_2	0.8626	0.1664	0.8696

La mediana en D_2 es mayor que en D_1 para ambos casos, en MCC y en *Accuracy*. Bien es cierto que la media en ambos casos es ligeramente superior para D_1, pero la desviación es menor. En general, se puede concluir que la presencia de desviación en la geolocalización de las IPs influye en el rendimiento de los modelos. En este experimento, los modelos no se ven demasiado perjudicados, probablemente porque la diferencia en la geolocalización no es muy elevada, solo 374 de las muestras ven modificadas su país de origen. Si estudiamos la relevancia de las diferencias, el efecto es pequeño ($d_{Cohen}(MCC) = 0.039, d_{Cohen}(Accuracy) = 0.10889$). Es probable que con cambios más bruscos de geolocalización, la capacidad predictiva del modelo varíe en un grado mayor, así que la bondad de ajuste de los resultados dependerá de cómo de representativa sea la magnitud de los cambios de la deriva conceptual de la geolocalización.

Por otra parte, podemos observar que el MCC en ambos casos es de, aproximadamente, 0.77, y el *Accuracy* se aproxima a 0.86. Aunque no sean tasas de ajuste superiores al $90\,\%$, sí que superan a las tasas conseguidas en otros trabajos de investigación. Cabe destacar que sería adecuado realizar los experimentos sobre los mismos conjuntos de datos para poder extraer conclusiones generales.

V-C. PI3

En la tabla V, podemos observar que existen diferencias significativas en el MCC cuando varíamos el parámetro h, tanto para D_1 como para D_2. Sin embargo, en la variable *Accuracy*, estas diferencias únicamente aparecen para D_1.

Tabla V
RESULTADOS TEST DE KRUSKALL-WALLIS ENTRE GRUPOS DADOS POR h

Variable respuesta		**H de Kruskall**	ρ**-valor**
MCC	D_1	16.433	0.002
	D_2	10.481	0.033
Accuracy	D_1	17.869	0.001
	D_2	9.441	0.051

Pasamos entonces a estudiar en detalle estas diferencias para extraer alguna conclusión sobre qué valores de h son los más óptimos. En la tabla VI, vemos las comparaciones *Post Hoc.*

Tabla VI
COMPARATIVA POST HOC CON CORRECCIÓN DE BONFERRONI

Variable respuesta	**Comparación h's**	ρ**-valor**
MCC(D_1)	$6-7$	0.136
	$6-8$	0.096
	$6-9$	1
	$6-10$	0.003
	$7-8$	1
	$7-9$	1
	$7-10$	1
	$8-9$	1
	$8-10$	1
	$9-10$	0.081
Accuracy(D_1)	$6-7$	0.540
	$6-8$	0.017
	$6-9$	1
	$6-10$	0.003
	$7-8$	1
	$7-9$	1
	$7-10$	0.918
	$8-9$	0.302
	$8-10$	1
	$9-10$	0.081
MCC(D_2)	$6-7$	1
	$6-8$	1
	$6-9$	0.155
	$6-10$	1
	$7-8$	11
	$7-9$	0.294
	$7-10$	1
	$8-9$	0.397
	$8-10$	1
	$9-10$	0.025

Como podemos observar, las diferencias no se dan entre todos los pares comparados. En el caso de D_1, las diferencias más significativas aparecen entre $h = 6$ y $h = 10$, tanto para el MCC como para el *Accuracy*. Y para el *Accuracy,* también para $h = 6$ versus $h = 8$. Los mejores ajustes de esas comparaciones se alcanzan con $h = 6$ (Mediana[MCC D_1]=0.7871, Mediana[*Accuracy* D_1]=0.8702 para $h = 6$.). Para D_2, las únicas diferencias aparecen cuando comparamos $h = 9$ y $h = 10$, debiéndose las mismas a que el MCC que se alcanza en D_2 con $h = 10$ es el más bajo de todos (Mediana[MCC D_2]=0.73226 para $h = 10$). Si estudiamos la relevancia del efecto de h, esta es elevada, véase tabla VII.

Tabla VII
VALOR DE d DE COHEN

Variable respuesta	**Comparación h's**	d**- Cohen**
MCC(D_1)	$6-7$	1.4158
Accuracy(D_1)	$6-8$	1.7586
	$6-10$	1.4586
MCC(D_2)	$9-10$	1.7057

Habría que profundizar entonces en el estudio de los resultados al combinar h con diferentes valores de δ, y de los recursos computacionales consumidos que requiere cada combinación.

V-D. PI4

En la tabla VIII, podemos observar que no existen diferencias significativas cuando varíamos el parámetro δ.

Tabla VIII
RESULTADOS TEST DE KRUSKALL-WALLIS ENTRE GRUPOS DADOS POR δ

Variable respuesta		H de Kruskall	ρ-valor
MCC	D_1	0.241	0.971
	D_2	0.439	0.637
Accuracy	D_1	0.241	0.971
	D_2	0.932	0.888

V-E. Discusión

Como conclusión, el mejor ajuste se alcanza con la siguiente combinación de hiper-paramétros:

1. MCC(D_1) = 0.7933 con $h = 9$, y $\delta = 0.001, 0.002, 0.003, 0.004$.
2. *Accuracy*(D_1) = 0.8729 con $h = 9$ y $\delta = 0.001, 0.002, 0.003, 0.004$ ó $h = 6$ y $\delta = 0.001$.
3. MCC(D_2) = 0.7871 con $h = 6$ y $\delta = .001, 0.002, 0.003, 0.004$.
4. *Accuracy*(D_2) = 0.8702 con $h = 6$ y $\delta = .001, 0.002, 0.003, 0.004$.

Para el caso de D_1, es evidente que el mejor resultado se consigue con $h = 9$, ya que obtiene el MCC y la *Accuracy* más altos para cualquier valor de δ . Sin embargo, habría que determinar si la pérdida de ajuste en el MCC es demasiada en relación con $h = 6$, ya que el *Accuracy* es el mismo para ambos valores de h, pero el consumo de recursos es menor para $h = 6$ por tener que gestionar una menor cantidad de datos al construir la serie temporal.

Para el caso de D_2, los mejores resultados se alcanzan claramente con $h = 6, \delta = 0.001, 0.002, 0.003, 0.004$ para ambas variables respuesta.

Por lo tanto, en general, el mejor resultado se obtiene con el valor de $h = 6$, el más bajo que se ha probado. Una posible pregunta de investigación futura sería determinar cuánto puede disminuir el valor de este hiper-parámetro manteniendo un buen ajuste.

VI. CONCLUSIONES

En este trabajo se ha extraído un conjunto de *features* que es significativo para categorizar, de manera multi-clase, la maliciosidad de una IP. Las variables son de doble naturaleza: temporal y de geolocalización. Los resultados alanzados tienen tasas de ajuste cercanas a 0.77 para el MCC, y a 0.86 para el *Accuracy.* Aunque estos valores no sean muy elevados, si que superan a otros trabajos de caracterización de reputación de IPs, teniendo en cuenta, además, que estos son para categorización bi-clase y suelen requerir una extracción de variables costosa.

Además, en el estudio hemos analizado el impacto de la deriva conceptual en las *features* relacionadas con la geolocalización, así como la influencia de los hiper-parámetros necesarios para calcular las variables que se extraen de las características temporales.

En los resultados se observa que no existe demasiado impacto en la capacidad predictiva de los modelos a causa de los cambios en la información de geolocalización; esto puede deberse a que la magnitud de los cambios es relativamente baja: solo 374 de las muestras tienen diferencias, por ejemplo, en el país de origen. Así, podría concluirse que los cambios en las *features* contextuales podría no ser tan relevante para los modelos como el tratamiento de las IPs cuya clasificación cambia, por ejemplo, al cesar la actividad maliciosa. Desde el punto de vista de las variables temporales, sí que encontramos que el efecto de uno de los hiper-parámetros es elevado, por lo que habrá que estudiar cuáles son los valores más óptimos del mismo.

Como trabajo futuro o de extensión, la investigación va dirigida a determinar si se puede realizar una selección de *features* del conjunto que aporte mejores resultados que los obtenidos. Además, trataremos de determinar el valor más óptimo de los hiper-parámetros que tienen influencia sobre los modelos. Por último, se estudiará la aplicación de más algoritmos de aprendizaje automático, y se estudiará la posible deriva conceptual con períodos más largos de tiempo. Todos los objetivos propuestos, se intentarán llevar a cabo sobre conjuntos de datos públicos para que puedan ser replicados y comparados con otras investigaciones.

AGRADECIMIENTOS

Este trabajo se enmarca dentro de os contratos art. 83 *Adenda 3: Machine learning para la calidad de los datos del modelo de inteligencia de INCIBE* y *Adenda 7: Prórroga de la Adenda 3* entre la Universidad de León e INCIBE en el periodo 2018-2022. Además, queremos agradecer a Diego Asterio de Zaballa el trabajo realizado en RIASC desde septiembre de 2020 a septiembre de 2021 y que forma parte de esta investigación.

REFERENCIAS

[1] Microsoft, "Security update severity rating system," Recuperado de https://www.microsoft.com/en-us/msrc/security-update-severity-rating-system.
[2] Forum of Incident Response and Security Teams (FIRST), "Common vulnerability scoring system," Recuperado de https://www.first.org/cvss/calculator/3.0.
[3] OWASP Foundation, "Owasp testing guide v4: Owasp risk rating methodology," Recuperado de https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology.
[4] Cybersecurity and Infrastructure Security Agency (CISA), "Nciss cyber incident scoring system," Recuperado de https://www.us-cert.gov/NCCIC-Cyber-Incident-Scoring-System.
[5] N. DeCastro-García, Á. L. Muñoz Castañeda, and M. Fernández-Rodríguez, "Machine learning for automatic assignment of the severity of cybersecurity events," *Computational and Mathematical Methods*, vol. 2, no. 1, p. e1072, 2020.
[6] "Firehol - linux firewalling and traffic shaping for humans," Recuperado de https://firehol.org/.
[7] Y. Liu, J. Zhang, A. Sarabi, M. Liu, M. Karir, and M. Bailey, "Predicting cyber security incidents using feature-based characterization of network-level malicious activities," in *Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics*, ser. IWSPA '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 3–9. [Online]. Available: https://doi.org/10.1145/2713579.2713582
[8] D. Likhomanov and V. Poliukh, "Predicting malicious hosts by blacklisted ipv4 address density estimation," in *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, 2020, pp. 102–109.

[9] B. Coskun, "(Un)wisdom of crowds: Accurately spotting malicious ip clusters using not-so-accurate ip blacklists," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, p. 1406–1417, jun 2017. [Online]. Available: https://doi.org/10.1109/TIFS.2017.2663333
[10] "Maxmind," Recuperado de https://www.maxmind.com/en/home.
[11] "Ipqualityscore," Recuperado de https://www.ipqualityscore.com/.
[12] J. L. Lewis, G. F. Tambaliuc, H. S. Narman, and W.-S. Yoo, "Ip reputation analysis of public databases and machine learning techniques," in *2020 International Conference on Computing, Networking and Communications (ICNC)*, 2020, pp. 181–186.
[13] Y. Huang, J. Negrete, A. Wosotowsky, J. Wagener, E. Peterson, A. Rodriguez, and C. Fralick, "Detect malicious ip addresses using cross-protocol analysis," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 664–672.
[14] N. Usman, S. Usman, F. Khan, M. A. Jan, A. Sajid, M. Alazab, and P. Watters, "Intelligent dynamic malware detection using machine learning in ip reputation for forensics data analytics," *Future Generation Computer Systems*, vol. 118, pp. 124–141, 2021.
[15] J. a. Gama, I. Žliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, 2014.
[16] G. Webb, R. Hyde, H. Cao, H. Nguyen, and F.Petitjean, "Characterizing concept drift," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964–994, 2016.
[17] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019.
[18] G. Webb, L. Lee, and B. Goethals, "Analyzing concept drift and shift from sample data," *Data Mining and Knowledge Discovery*, vol. 32, pp. 1179 – 1199, 2018.
[19] D. Levin, Y. Peres, and E. Wilmer, *Markov chains and mixing times*. American Mathematical Society, Providence, 2008.
[20] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proceedings of 28 Conference in Advances in Neural Information Processing Systems*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 2962–2970.
[21] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Proceedings of 5th Conference in Learning and Intellingent Optimization.*, C. A. C. Coello, Ed., 2011, pp. 507–523.
[22] B. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophys. Acta (BBA) - Protein Struct.*, vol. 405, no. 2, pp. 442–451, 1975.
[23] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, 2020.
[24] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences (2nd ed.).* Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers., 1988.

Ataques a servidores web: estudio experimental de la capacidad de detección de algunos SIDS gratuitos

Javier Muñoz [1], Felipe Bueno [1], Rafael Estepa [1], Antonio Estepa [1], Jesús E. Díaz-Verdejo [2]
[1] Dpto. Ingeniería Telemática, Escuela Superior de Ingenieros, Univ. de Sevilla
fmjc@us.es, felipebuenocarranza@gmail.com, rafaestepa@us.es, aestepa@us.es
[2] Dpto. Teoría de Señal, Telemática y Comunicaciones, CITIC, Univ. de Granada,
jedv@ugr.es

Resumen- **Este trabajo cuantifica de forma experimental la capacidad de detección de ataques a servidores web ofrecida por algunos de los detectores de intrusiones basados en firmas (SIDS) disponibles de forma gratuita. Para ello, se ha realizado una búsqueda y selección de 28 herramientas actuales para la generación de ataques y análisis de seguridad del servicio web. Con ellas, se han realizado casi 150 ataques a dos escenarios de uso de un servidor web (una web estática y una dinámica). Las peticiones HTTP registradas durante los ataques han sido utilizadas para crear un *dataset* de ataques que será utilizado como entrada a tres SIDS gratuitos seleccionados por su amplio uso, de forma que se podrá determinar la capacidad de detección de los mismos frente a los ataques generados. Este trabajo se encuentra aún en desarrollo, por lo que en esta contribución se muestran los primeros resultados relativos a la recolección y selección de herramientas para la generación de los ataques, la generación del *dataset* de ataques de forma que sea representativo de los ataques actuales y la evaluación preliminar de las capacidades de detección.**

***Index Terms*- IDS, firmas, WAF, ataques web, dataset**

Tipo de contribución: *Investigación en desarrollo*

I. INTRODUCCIÓN

Los servicios web han experimentado una gran expansión en los últimos años, tanto por el desarrollo de servicios cada vez más complejos y avanzados como por la fácil accesibilidad a los mismos mediante el uso de navegadores. Consecuentemente, los servidores web se han convertido en una de las dianas favoritas de los ciberataques [1]. Entre otros escenarios, los servidores comprometidos se utilizan habitualmente para la distribución de *malware*, para realizar *phishing* o como puerta de acceso a la red de una empresa. Una herramienta clave en la detección de ataques a servidores web son los sistemas de detección de intrusiones basados en firmas (SIDS) [2] o los WAF (*Web Application Firewall*). En este trabajo se pretende verificar la capacidad de detección de ataques basados en web que ofrecen algunos de los SIDS o WAF gratuitos más difundidos en la actualidad. En particular, en primera aproximación consideraremos tres de ellos: *Snort* [3], *ModSecurity* [4] y *Nemesida* [5] (con reglas gratuitas). Para ello, previamente, será necesario establecer *datasets* de ataques que sean representativos y se encuentren actualizados, lo cual no es una tarea fácil.

En este sentido, existen pocos datasets públicos disponibles para hacer experimentación sobre SIDS en general (p.ej., KDD'99 [6], CAIDA [7], UNSW-NB15 [8]) y aún menos que sean específicos para HTTP (p.ej. CICIDS2017 [9]). Los ataques que contienen estos *datasets* aparecen en número reducido, son antiguos y mayoritariamente de sistemas simulados [10]. Esta carencia de cercanía a la realidad y la falta de representatividad los hacen poco aplicables a entornos de producción [11] y, consecuentemente, los invalidan para desarrollar y/o evaluar sistemas de detección de ataques válidos en escenarios reales modernos [12], especialmente porque los servicios web y los ataques evolucionan rápidamente con el tiempo. Si bien es posible encontrar más de 80 bases de datos públicas [13] con peticiones reales a servidores web, un análisis de las mismas nos muestra que, o bien no están marcadas las peticiones de ataque, o bien se limitan a la actividad recopilada mediante *honeypots*, lo que no garantiza la representatividad ni la presencia de todos los tipos de ataques. Para evitar estas limitaciones se ha generado un nuevo *dataset* con tráfico de ataques web (teniendo en cuenta sólo aquellos basados en el contenido de las URI) a partir de varias herramientas de ataque actualizadas seleccionadas al efecto. Se han considerado dos escenarios como objetivos del ataque (una web estática y una web dinámica) y se ha seguido una metodología que permite su repetición y ampliación.

Este estudio se encuentra actualmente en desarrollo y en este artículo presentaremos sus resultados preliminares. Las principales contribuciones son: a) elaboración de una lista actualizada de herramientas susceptibles de ser utilizadas para la realización de ataques web, con sus principales características, b) generación de un *dataset* con ataques, y c) la evaluación inicial de la capacidad de detección de distintos SIDS gratuitos sobre el *dataset* anterior. Estas contribuciones permitirán la selección de las herramientas de ataque y de los SIDS más apropiados, así como la ejecución de nuevas pruebas que utilicen el *dataset* proporcionado con distintos SIDS o con nuevas firmas. Por otra parte, el *dataset* podría también utilizarse para evaluar sistemas basados en detección de anomalías.

II. HERRAMIENTAS DE GENERACIÓN DE ATAQUES WEB

Para que el *dataset* de ataques sea representativo de los diversos tipos de ataques que pueden llevarse a cabo se han buscado las herramientas de ataques web disponibles en la actualidad. Para ello se han empleado referencias encontradas en tres fuentes de información básicas: a) OWASP: *Dynamic*

Application Security Testing (DAST) [14], b) Mitre: técnicas *Exploit Public-Facing Application* [15] y Software [16] y, por último, c) Listado de Software del proyecto *Nmap* [17], que cuenta con una de las listas más actualizadas sobre herramientas de ciberseguridad. Se ha buscado en las categorías: *Web Vulnerabilitiy scanners* y *Vulnerability Exploitation tools.* Las distintas herramientas encontradas en las fuentes anteriores han sido revisadas de forma individual, encontrándose 22 de tipo *Opensource* y 23 con licencia comercial, para las que se ha solicitado al desarrollador una licencia de prueba gratuita temporal que ha sido concedida en 6 casos. Como resultado, han sido probadas e instaladas en el las herramientas que se detallan en la Tabla I, donde se muestra, para cada una, la licencia de uso, el sistema operativo, el tipo (columna *Tipo*: SIDS genérico – G- o específico de web -E), su funcionalidad (columna *Func*: sólo sondeo -S- o sondeo y explotación -E-) y los tipos de ataques que permite realizar (columna Tipos de *Ataque*, donde los 10 primeros corresponden con la clasificación *OWASP Top10 , 2021*).

Es posible encontrar dos grandes categorías de herramientas: aquellas específicas de web, que normalmente, permiten mayor granularidad en la especificación del ataque y suelen seguir los tipos de ataques web especificados en la clasificación de OWASP, y otras genéricas, orientadas a *pentesting,* que suelen ejecutar secuencialmente una lista de ataques ya preconfigurados y que permiten un menor control al usuario. Todas las herramientas tienen un funcionamiento similar, siendo preciso especificar la URL del sistema objetivo a atacar y, en su caso, el tipo de ataque correctamente parametrizado.

III. Generación del *dataset* de ataques web

A fin de que el *dataset* reproduzca la mayor diversidad de ataques posibles con las herramientas actuales, se han generado todos los tipos de ataques implementados en cada una de las 28 herramientas descritas en la Tabla I en dos escenarios web: una aplicación estática (*Apache* con un recurso html) y una aplicación dinámica (gestor de contenidos *Wordpress* con la instalación por defecto).

Se ha desplegado un escenario con dos máquinas: una para el atacante y otra para el servidor web correspondiente. En la máquina del atacante se ha instalado el software de ataque actualizado a fecha de abril de 2022, junto con el sistema operativo exigido por la herramienta. Desde dicha máquina se han lanzado todos los tipos de ataques posibles (ver Tabla I) contra los dos escenarios web previstos, capturando el tráfico recibido por el servidor *Apache* con *tcpdump*. Una vez guardado el tráfico de ataque en formato *pcap,* se han extraído las peticiones web en formato texto plano mediante la aplicación *tshark,* incluyendo la URI presente en las mismas. Adicionalmente, se han generado ficheros de resumen del tráfico (flujos) en formato IPFIX con las aplicaciones: *ipt-netflow* y *nfcapd*. De esta forma, para cada ataque realizado con cada herramienta, se han guardado los ficheros:*.pcap (paquetes)*, *.ipfix (flujos)*, *.csv* (paquetes en formato csv) y *.uri*. Estos ficheros conforman el cuerpo del *dataset* generado, que está disponible para la comunidad en

Tabla I
Herramientas para ataques web utilizadas

ID	Nombre	Licencia	Sistema Operativo [1]	Tipo [2]	Func [3]	Tipos de Ataque [4]											
						1	2	3	4	5	6	7	8	9	10	11	12
1	Havij	OpenSource	W	E	E												X
2	Wpscan	Comercial	L, M	E	S												X
3	Nuclei	OpenSource	W, M, L	E	E			X							X		
4	Sqlmap	OpenSource	W, M, L	E	E											X	X
5	OWASP-ZAP	OpenSource	W, M, L	E	S		X	X	X	X							
6	Grabber	OpenSource	W, M, L	E	S			X								X	
7	Openvas	OpenSource	L	G	S											X	X
8	Arachni	OpenSource	W, M, L	E	E			X									
9	Ironwasp	OpenSource	W, M, L	E	E	X	X	X								X	
10	W3af	OpenSource	L, M	E	S		X	X	X	X	X	X	X	X	X	X	
11	Nexpose	Comercial	W, L	E	S											X	X
12	SmartScanner	Comercial	W	E	S												X
13	Nessus	OpenSource	W	G	S												X
14	Golismero	OpenSource	W, M, L	E	S												X
15	Burpsuite	Comercial	W, M, L	E	S												X
16	Metasploit	OpenSource	W, M, L	G	E												X
17	Nikto	OpenSource	L	E	E	X	X	X									
18	Wapiti	OpenSource	W, M, L	E	S		X	X							X	X	
19	Grendel-Scan	OpenSource	W, M, L	E	S			X	X	X	X					X	
20	Webcruiser	Comercial	W	E	S											X	X
21	Nmap	OpenSource	W, M, L	G	E			X							X		
22	Nexploit	Comercial	SaaS	E	S											X	X
23	Xsser	OpenSource	W, M, L	E	E												X
24	Vega	OpenSource	W, M, L	E	S		X	X									
25	Skipfish	OpenSource	W, M, L	E	S											X	X
26	Watobo	OpenSource	W, M, L	E	S												X
27	Commix	OpenSource	W, M, L	E	E												X
28	Deepfence Threatmapper	OpenSource	L	G	S												X

[1] Sistema Operativo: W=Windows, M=MacOS, L=Linux, SaaS=Software as a service
[2] Tipo: G: Genérico, E: Específico web
[3] Función: S: sólo sondeo, E: incluye además explotación
[4] Ataques: 1: *Broken Access Control*, 2: *Cryptografic Failures*, 3: *Injection*, 4: *Insecure Design*, 5: *Security Misconfiguration*, 6: *Vulnerable and Outdated Components*, 7: *Identification and Authentication Failures*, 8: *Software and Data Integrity Failures*, 9: *Security Logging and Monitoring Failures*, 10: SSRF, 11: Otros, 12: Conjunto predefinido.

https://github.com/fbuenoc97/TFG/tree/main/capturas. En este repositorio podemos encontrar para cada escenario, un directorio por cada herramienta que contiene los ficheros del *dataset* cuyo nombre se corresponde con el tipo de ataque realizado. A modo de resumen, el *dataset* consta de un total de 148 ataques distintos realizados con las herramientas especificadas en el apartado anterior. Para el escenario estático los ficheros *pcap* contienen 317.433 peticiones web (URIs de sondeo o de explotación), mientras que para el escenario dinámico el número de peticiones HTTP recopiladas es de 391.006.

Cabe señalar que, se ha realizado una validación de la captura para evaluar si se ajusta al tráfico esperado, reajustando los parámetros del ataque en los casos en los que se consideró necesario. Esta inspección permitió verificar algunas peculiaridades, como que la mayoría de las herramientas realizan siempre una fase de escaneo previa al ataque en sí. Se observaron algunos casos muy residuales de comportamientos no esperados (con distintas tipologías de ataque, la herramienta lanzaba el mismo ataque –seguramente por un error de programación–) que fueron eliminados del *dataset*. La herramienta *wpscan* (ID=2) sólo se ha ejecutado sobre el escenario dinámico (es exclusiva para Wordpress), y, como incidencias significativas, podemos señalar que los resultados de dos herramientas han sido eliminados del *dataset*: *Deepfence Threatmapper* (ID=28) y *Commix* (ID=27) ya que todas las URI existentes para ambas son "/web".

El *dataset* generado con los ataques realizados por las 26 herramientas restantes ha sido utilizado para verificar la capacidad de detección de distintos SIDS, como se detallará en el siguiente apartado.

IV. Capacidad de detección de SIDS gratuitos

Las peticiones web incluidas en el *dataset* anterior han sido procesadas por sistemas de detección de intrusiones. En este trabajo nos centraremos en tres sistemas SIDS ampliamente extendidos y de uso gratuito, que ya han sido usados previamente por los autores:

a) Snort [3], software IDS genérico de amplia implantación. Utilizaremos las reglas de Talos [18] así como las reglas ETopen [19] actualizadas a 24/03/2022 con todas las reglas activas. Previamente, se han seleccionado las reglas que afectan únicamente al URI de peticiones HTTP, siguiendo el procedimiento indicado en [20].

b) ModSecurity [4]: módulo WAF también de amplio uso por su fácil integración con *Apache*. Utilizaremos las reglas OWASP *Core Ruleset* (CRS) en su versión 3.3.2 y el nivel de paranoia 2.

c) Nemesida: es un WAF completo que incluye un conjunto de firmas de uso gratuito. Dicho conjunto proporcionaría una mayor tasa de falsos positivos que la versión de pago, lo que no afecta a este estudio.

El fichero de traza de cada SIDS ha sido analizado a fin de determinar qué peticiones HTTP de entrada han sido detectadas como ataque. Los resultados se muestran en la Tabla II, que indica, por cada herramienta de ataque, el número de peticiones (N. URI) generadas por cada herramienta y el porcentaje de ellas que han sido detectadas como maliciosas por cada SIDS. Cabe señalar que muchas de las peticiones enviadas forman parte de una fase de escaneo. Esta puede ser considerada una etapa temprana del ataque, según la taxonomía de Mitre ATT&CK [21], por lo que resulta de interés verificar si los SIDS son capaces de detectar este tipo de peticiones. Para el escenario web estático la tasa media de detección de *Snort* es del 2,2%, subiendo al 13% y 14% para *ModSecurity* y *Nemesida,* respectivamente. En el caso del escenario dinámico, los valores son similares, quedando en torno al 20% para *ModSecurity* y *Nemesida*. Podemos inferir que la eficiencia en la detección para los SIDS específicos de web (*ModSecurity* y *Nemesida*) es

Tabla II
Capacidad de detección de los SIDS utilizados

			Web estática				Web dinámica			
ID	Herramienta	N. At.	N. URI	Snort	ModSec	Nemesida	N. URI	Snort	ModSec	Nemesida
1	Havij	1	294	14%	99%	84%	138	95%	98%	100%
2	Wpscan	1	-	-	-	-	166	2%	67%	69%
3	Nuclei	14	8362	21%	49%	51%	2076	22%	52%	52%
4	Sqlmap	1	98	38%	86%	64%	995	29%	46%	18%
5	OWASP-ZAP	8	2109	0%	0%	0%	9061	7%	50%	35%
6	Grabber	5	2108	17%	95%	60%	22358	3%	46%	35%
7	Openvas	1	10492	6%	19%	33%	50081	13%	27%	36%
8	Arachni	10	3385	0%	0%	0%	17044	4%	38%	33%
9	Ironwasp	13	12301	0%	7%	7%	582	2%	37%	22%
10	W3af	26	7896	2%	0%	0%	8879	5%	11%	43%
11	Nexpose	1	5344	13%	20%	24%	5344	13%	20%	24%
12	SmartScanner	1	1187	0%	12%	19%	2280	4%	20%	25%
13	Nessus	1	49472	3%	27%	15%	43508	5%	23%	19%
14	Golismero	1	8890	6%	44%	26%	290	3%	18%	24%
15	Burpsuite	1	254	6%	9%	12%	2274	9%	15%	14%
16	Metasploit	1	48039	0%	14%	22%	48039	0%	14%	22%
17	Nikto	12	4306	1%	10%	33%	2841	0%	16%	17%
18	Wapiti	14	21783	0%	1%	4%	44963	2%	11%	10%
19	Grendel-Scan	8	17835	4%	3%	14%	19249	4%	3%	14%
20	Webcruiser	1	1548	0%	0%	0%	4827	0%	10%	9%
21	Nmap	9	2531	2%	4%	5%	2946	2%	4%	11%
22	Nexploit	1	7335	1%	7%	3%	17063	2%	9%	6%
23	Xsser	1	24	0%	0%	0%	61	0%	3%	11%
24	Vega	13	23342	1%	1%	1%	53276	1%	5%	5%
25	Skipfish	1	74763	1%	5%	8%	31813	0%	3%	8%
26	Watobo	1	3734	0%	0%	0%	852	0%	0%	0%

superior a la del SIDS genérico (*Snort*). No se aprecian diferencias significativas en la tasa de detección para herramienta que realizan explotación, lo que resulta lógico, dado que estas herramientas también tienen fase de escaneo. Estos resultados sugieren que los SIDS probados muestran un nivel de detección bajo en la fase de escaneo a fin de reducir la tasa de falsos positivos. Sería necesario, en cualquier caso, una revisión más a fondo de los resultados para confirmar esta hipótesis. Finalmente, en la Figura 1 se muestra, para cada escenario, la media del porcentaje de peticiones realizadas por cada herramienta que han sido detectadas como ataques (valor medio de la capacidad de detección de los tres SIDS bajo estudio). Este indicador ha sido utilizado para ordenar las herramientas, y su complementario puede ofrecer una idea preliminar sobre la capacidad de cada herramienta de ataque para pasar inadvertida a los SIDS.

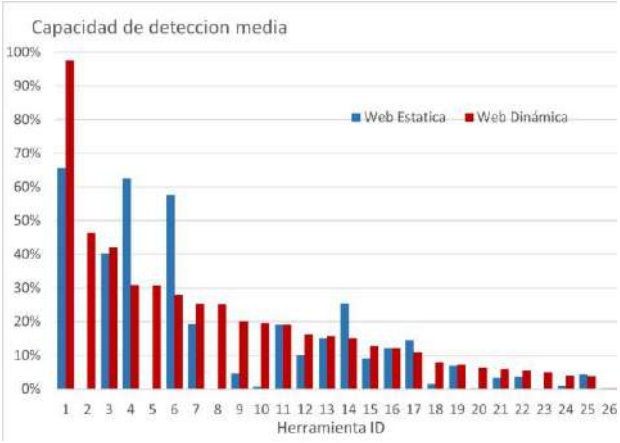


Fig. 1. Capacidad media de detección en ambos escenarios.

V. Conclusiones y líneas de avance

La disponibilidad de *datasets* adecuados es clave para acelerar la investigación en el campo de los IDS. En este trabajo se ha generado un *dataset* propio de ataques que se ha puesto a disposición de la comunidad investigadora. Este *dataset* incorpora todos los posibles tipos de ataques que ofrecen las principales herramientas gratuitas disponibles contra dos escenarios web: uno estático y otro dinámico.

La capacidad de detección de URIs maliciosas en los ataques web incluidos en el *dataset* por parte de los tres SIDS oscila entre un 5% y 20%. Este porcentaje no implica necesariamente un bajo rendimiento de los sistemas de detección, y podría responder a que la fase de escaneo de un ataque no es suficientemente detectada. Estos resultados son preliminares y el trabajo continúa en curso.

En las siguientes fases de nuestro trabajo se abordará el análisis de la eficiencia en la detección por tipo de ataque y por herramienta. También se desea discriminar en los resultados la capacidad de detección de la fase de escaneo e incrementar el número de herramientas utilizadas, incluyendo herramientas de pago. Por último, los resultados sobre la capacidad de detección de las distintas herramientas serán complementados con la tasa de falsos positivos que genera cada uno, lo que permitirá la estimación del rendimiento de manera fiable.

Como limitaciones de este trabajo pueden destacarse el uso de herramientas gratuitas y la limitación a dos escenarios de aplicación web sobre las que atacar, lo que restaría capacidad de generalización a los resultados. Aunque los dos escenarios utilizados son de amplia implantación, el uso de otros portales y aplicaciones/servicios web enriquecerían la aplicabilidad de los resultados. Por último, es importante reseñar que es posible que no todas las peticiones HTTP correspondan realmente a ataques, por lo que es necesario supervisar el *dataset*. Estos aspectos deberán ser tratados en posibles ampliaciones al trabajo.

Agradecimientos

Esta publicación es parte de los proyectos de I+D+i PID2020-115199RB-I00 financiado por MICIN/AEI/ 10.13039/501100011033 y PYC20-RE-087-USE y A-TIC224-UGR20 financiado por FEDER/Junta de Andalucía - Consejería de Transformación Económica, Industria, Conocimiento.

Referencias

[1] Acar, Gunes, et al. "Web-based attacks to discover and control local iot devices." Proceedings of the 2018 Workshop on IoT Security and Privacy. 2018.
[2] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, X. Bellekens, *A taxonomy and survey of intrusion detection system design techniques, network threats and datasets*, arXiv:1806.03517, 2018.
[3] Snort, *Snort: an open source network intrusion prevention and detection system*, Sourcefire, disponible en http://www.snort.org.
[4] *Modsecurity Open Source Web Application Firewall*. Disponible en https://github.com/SpiderLabs/ModSecurity.
[5] *Nemesida Web Application Firewall.* Disponible en: https://nemesida-waf.com.
[6] S. Hettich, S.D. Bay, *The UCI KDD Archive.* Univ. of California, Dep. of Information & Computer Science, http://kdd.ics.uci.edu, 1999.
[7] *Cooperative Association for Internet Data Analysis* (CAIDA) datasets, 2008.
[8] N. Moustafa, J. Slay, *UNSW-NB15: a comprehensive data set for network intrusion detection systems*, Military Communications and Information Systems Conference (MilCIS), 1-6, 2015.
[9] P. Ranjit,, S. Borah, *A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems*, International Journal of Engineering & Technology 7.3.24:479-482, 2018.
[10] Riera, T.S., Higuera, J.R.B., Higuera, J.B., Herraiz, J.J.M., Montalvo, J.A.S; *Prevention and fighting against web attacks through anomaly detection technology. A systematic review*, Sustainability 12:1-45, 2020.
[11] Sharafaldin I., Gharib A., Lashkari A.H., Ghorbani A.A., *Towards a reliable intrusion detection benchmark dataset*, Softw. Netw., 1:177-200, 2018.
[12] R. Sommer, V. Paxson; *Outside the closed world: On using machine learning for network intrusion detection*, Proc. IEEE Symp. Secur. Privacy, 305–316, 2010.
[13] https://datasetsearch.research.google.com/
[14] https://owasp.org/www-community/Vulnerability_Scanning_Tools
[15] https://attack.mitre.org/techniques/T1190/
[16] https://attack.mitre.org/software/S0390/
[17] https://sectools.org/
[18] https://www.snort.org/talos
[19] https://doc.emergingthreats.net/
[20] J.E. Díaz-Verdejo, A. Estepa, R. Estepa et al. / Future Generation Computer Systems 109:67–82, 2020.
[21] Strom, Blake E., et al., *Finding cyber threats with ATT&CK-based analytics*, The MITRE Corporation, Technical Report No. MTR170202 (2017).

Hunting@home: Plug and Play setup for intrusion detection in home networks

Lorena Mehavilla
Universidad de Zaragoza
María de Luna 1
lmehavilla@unizar.es

José García
Universidad de Zaragoza
María de Luna 1
jogarmo@unizar.es

Álvaro Alesanco
Universidad de Zaragoza
María de Luna 1
alesanco@unizar.es

Abstract- **Home networks cannot only rely on end-point antivirus and ISP router default configurations to provide cybersecurity to their users. In this work we present a plug and play low-cost setup leveraging Network Intrusion Detection System (NIDS) in home networks based on ARP spoofing in a Raspberry Pi computer. Zeek NIDS is used for testing purposes. We have developed a new Zeek plugin to enhance traffic log information. Early results have shown that our setup is reliable, transparent, and easy-to-use providing home users with the possibility of an extra security layer with no performance penalization.**

Index Terms- **ARP spoofing, Home networks, NIDS, Raspberry Pi, Zeek**

Contribution type: *Research in progress*

I. INTRODUCTION

The necessity of home network security has boosted. Home networks have evolved from scenarios where only a few devices where connected (desktop or laptop computer and a couple of smartphones) to scenarios where a complete ecosystem of (mostly) wireless devices are present (smartphones, computers, smart TVs, and a myriad of smart home appliances [1]). Nowadays, it can be said that a home network is basically a wireless network. Besides, due to the COVID19 pandemic situation, we have experienced an increase in the number of people working at home, transforming home spaces into the new office but lacking the security measures usually present at company spaces. The principal and (in many cases) only home network security measure is computer antivirus that, although necessary, is not enough to cope with all the new threats arisen in this new environment.

Traditional attack vectors in home environments were wireless unsecure technologies (e.g., WEP) or a faulty or insecure setup allowing attackers to hack access passwords and enter the wifi network. Thanks to evolution in wifi technologies (WPA2, WPA3) unsecure protocols such us WEP lie in the past. Also, Internet Service Provider (ISP) companies provide routers to home users with a much more secure configuration that years back, where many default misconfigurations allowed attackers to gain rapid access to home networks. Nevertheless, the number of cyberattacks affecting home networks has rapidly increased putting at risk home users and their new digital requirements and trends [2]. Traditional attack vectors have been replaced with more elaborated attacks, where malware is installed in end devices. This malware can rapidly expand within home networks since no internal security measures are implemented. It could be said that nowadays traffic in home networks is like a black hole where visibility is hard to gain. Thus, technologies such as network intrusion detection systems (NIDS) would be a valuable addition to increase home security, providing visibility to internal device traffic. One of the most deployed NIDS is Zeek [3]. Although Zeek is not as efficient as Suricata NIDS in binary pattern matching for rule triggering, outperforms any other NIDS in traffic connection analysis, generating very rich connection logs for different types of protocols and activities. Besides, it is possible to develop new scripts easily that enhances its logging capabilities enriching connection information at different levels. The use of Zeek not only provides home networks with intrusion detection capabilities but also opens the possibility of traffic classification, a feature that could be very interesting for home users.

The pillar of a typical home network is the router/firewall provided by the ISP (see Fig. 1a). This device acts also as an access point (AP) for wireless devices and in many cases includes up to 6 Gigabit Ethernet ports in a switch arrangement. With this setup, increasing home network security has been beyond regular users' capacities and required a monetary investment that has hampered its implementation. Any NIDS would need an external switch with port mirroring capacity connected to the primary router. All wired connected devices should be plugged into this switch and one of its ports should be configured as mirroring port connected to the NIDS device. Nevertheless, this setup is not enough since it only supports wired devices. To deploy a wireless setup, an AP device should be connected to the switch via ethernet cable or, more conveniently, replace the switch with a commercial router/firewall that has both AP and switch capabilities with port mirroring option (see Fig. 1b). As it can be seen, enabling home network scenarios with NIDS capabilities is neither easy nor cheap for regular home users.

To overcome all these setup drawbacks (complexity and monetary cost) we propose to use a very well-known attack to inspect network traffic: ARP spoofing. Thus, we apply an attacker technique for legitimate purposes, enabling to use traffic inspection in a transparent way with no performance penalization as we will show along this work.

The rest of the paper is organized as follows. Section II shows the architecture used and the ARP spoofing approach. Early results are presented along Section III and discussed in Section IV. Finally, conclusions are enumerated in Section V.

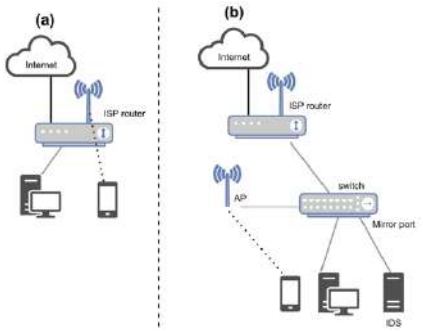


Fig. 1. Home network scenario. (a) Typical setup for home users. (b) Advanced setup for traffic analysis using IDS.

II. Home Network Setup Architecture

We use ARP spoofing for traffic inspection. Although this technique is used mainly by attackers to perform Man in The Middle (MiTM) attacks, the same principle enables to inspect traffic in a transparent way, overcoming the handicap of mirroring traffic without extra equipment. ARP protocol is essential for home networks. It is defined in RFC 826 and enables the mapping between ethernet addresses and IP addresses. Because in the early days of network communications development security was not considered as a design principle, ARP protocol is vulnerable to impersonation attacks, where the attacker can fool any machine on the network using crafted ARP packets pretending to be another machine. ARP spoofing attack is very well known and a detailed explanation of MiTM attack in local networks can be found elsewhere [4]. Although the attack is very well known and there exist defenses to avoid it in local networks [4], it is quite uncommon that home routers implement or use these defenses. As a result, ARP spoofing can be used in home networks as a mean to redirect all traffic to be inspected in a single spot and then rerouted to the original destination, being unnoticeable for network users. To this end, we use a small yet powerful device: a Raspberry Pi computer. Its reduced dimensions as well as its low price make the Raspberry Pi an ideal choice to be used as a home probe for hunting intruders. We use the Raspberry Pi 4 model with 8GB of RAM. The proposed architecture is shown in Fig. 2.

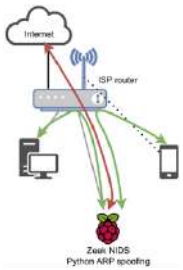


Fig. 2. Home network setup architecture.

It is a plug and play setup, where the only requirements are the Raspberry Pi device, the SD card with Raspbian 64-bit operating system, our custom scripts and NIDS software and a gigabit ethernet cable to plug the Raspberry Pi to one of the ethernet ports of the home router. As soon as the Raspberry Pi probe is turned on, it requests an IP address via DHCP protocol and starts the execution of the custom python software developed to identify all the network devices and perform the ARP spoofing attack.

A. ARP spoofing setup

We have developed a custom python script to enable the spoofing. It is included in the *rc* subsystem to be initiated at startup. This thread-based script is composed by 4 processes. Process 1 consists of a quick device discovery using ARP broadcast request packets. Devices that respond to these requests are stored in the *active_device* list. This process enables to discover devices that are already present in the home network. This process is periodically executed. Process 2, that starts at the same time as process 1, implements a DHCP discover/request listener enabling to discover devices as soon as they join the home network. This process is constantly running. Devices identified here are stored in the *active_device* list. Process 3 updates the *active_device* list by executing process 1 periodically and removing devices from the list if they are no longer seen. Process 4 starts ARP spoofing as soon as new devices are introduced into the *active_device* list.

B. Zeek setup

We have installed Zeek NIDS from source into the Raspbian 64-bit OS. Zeek is configured in a cluster setup, where zeek is divided in workers, proxy, and manager processes [5]. Although Zeek software can be configured to use more than one CPU core (one per worker), in this initial setup only one core, thus one worker, is used. Although Zeek generates logs about network connections and DNS resolutions, the information is not combined making it difficult to link connection IP destination addresses with the domain names resolved with DNS generated before establishing the connection. To improve network connection classification and anomaly detection, we have developed a custom Zeek script which goal is to log all connections and enhance their information with DNS resolution. Thus, the new logs generated (*connection_plus.log*) contain all the information related to network connections that could be found in the original *conn.log* zeek file plus destination IP name resolution information (*resp_name* and *server_resp_name* in Fig. 3) and, if the connection uses the TLS protocol, information related to TLS handshake phase *(ssl_** fields in Fig.3).

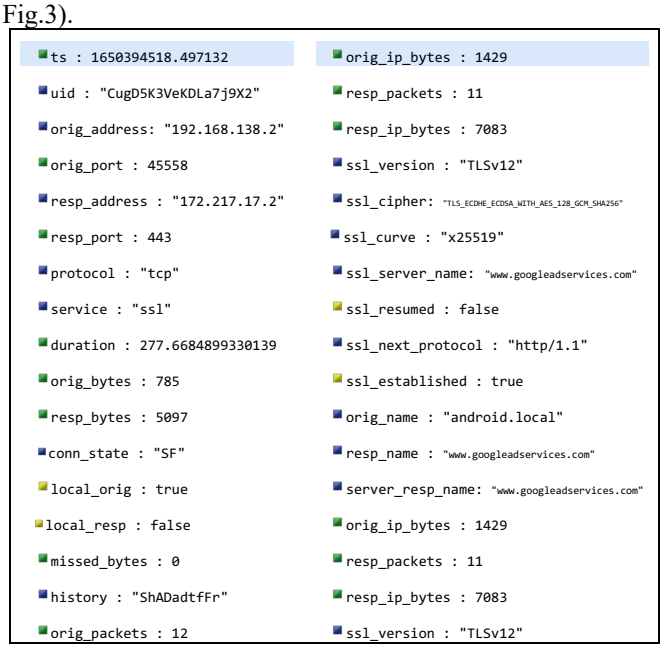


Fig. 3. Zeek connection_plus logs.

III. Results

To optimize and validate our proposal, we have performed three groups of tests: A) Raspberry Pi performance test to validate its capacity to act as traffic forwarder in the ARP spoof scenario. B) ARP spoofing tests to evaluate the capacity of our script to reroute all home network traffic through the Raspberry Pi. C) Zeek tests to evaluate the feasibility of using Zeek NIDS for traffic analysis.

A. Raspberry Pi performance

To evaluate the decrease (if any) in connection bandwidth due to the use of an extra element (Raspberry Pi) as a forwarder where all the connections must pass before being rerouted to its original destination, we have used the iPerf software considering the setup shown in Fig. 4, where the Raspberry Pi is performing an ARP spoofing attack to device 1 (iPerf server) and device 2 (iPerf client).

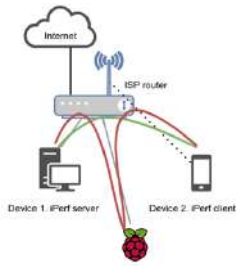


Fig. 4. Raspberry Pi performance test. In green direct connection path between device 1 (left) and device 2 (right). In red the Raspberry Pi forwarder path.

This setup allows us to measure the decrease in bandwidth in a Gigabit environment if the Raspberry Pi is acting as a traffic forwarder, the situation in which it will be working in our proposal. Table I shows the results when a direct cable connection and a 5GHz wireless connection are used.

Table I
RASPBERRY PI PERFORMANCE

	Direct connection	Raspberry Pi forwarder
Ethernet connection	910 Mbps	800 Mbps
5 GHz connection	320 Mbps	320 Mbps

B. ARP spoofing performance

To evaluate the feasibility of using ARP to perform the spoofing in home networks, we have tested the use of different ARP packets against ten different devices:

1. Vodafone CG7486E Wireless Router.
2. Vodafone Sercomm H500-s Router
3. Movistar 3505W Router
4. Yoigo Sagecom CS5001 Router
5. Mikrotik hAP ac Router.
6. MSI GL65 Leopard PC with Windows 7 Home Premium version 6.1
7. Lenovo Ideapad 320s with Windows 10 Home version 21H2
8. Samsung Galaxy-A7-2018 with Android 10
9. Samsung Galaxy-A32-5G with Android 11
10. iPhone 6 with IOS 12.5.5

Table II shows the six different ARP packet configurations (different option fields inside an ARP packet) setups, named from A to F. Spoofing attack only took effect on Device 1 when packets C and D were used. For devices 2, 3 and 4, all packets performed a successful attack except for packet A. For the rest of devices (5 to 10) all packets performed a successful attack. Considering these results, packet D was selected to be used in process 4 (see ARP spoofing setup).

Table II
ARP PACKET OPTIONS

Packet config.	A	B	C	D	E	F
Op.	Reply	Reply	Request	Request	Reply	Reply
S.IP	DevX	DevX	DevX	DevX	DevX	DevX
S.HA	Raspi	Raspi	Raspi	Raspi	Raspi	Raspi
T.IP	Broad.	DevY	DevY	DevY	DevX	DevX
T.HA	Broad.	DevY	Broad.	DevY	Broad.	DevY
E.D.A	Broad.	DevY	Broad.	DevY	Broad.	DevY

* Where Op. is Operation; S. IP is Source IP; S.Ha is Source Hardware; T.IP is Target IP; T. HA is Target Hardware. All this fields are ARP packet options. E.D.A is Ethernet packet destination address. DevY is the device suffering the attack and DevX is the device which identity is being spoofed. Raspi is the Raspberry Pi. Broad. mean broadcast.

To evaluate the efficiency in the spoofing process once the spoofing is started, we have measured the percentage of packets where the spoofing was active (Raspberry Pi Ethernet address as destination address) for different ARP spoofing time intervals. Fig. 5 shows the setup used to measure the efficiency. We have used a Mikrotik switch with port mirroring to see all network packets and thus calculate the efficiency.

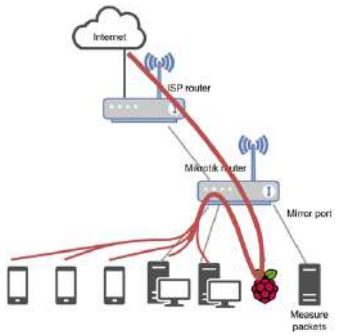


Fig. 5. ARP spoofing packets test scenario.

Table III shows the results when 5 active devices are connected to the home network and one of them is downloading a large file (701 MB, test A) and a small file (180 KB, test B) five consecutive times from the Internet using a bandwidth of 160 Mbps and 80 Mbps, respectively. Considering the results obtained, we have selected an interval of 10 sec. for our python script.

Table III
ARP SPOOFING TEST RESULTS

Test A Interval	Spoofed packets	CPU use	RAM use
5s	99.8%	<1%	240 MB
10s	99.8%	<1%	240 MB
30s	99.3%	<1%	240 MB
Test B Interval	**Spoofed packets**	**CPU use**	**RAM use**
5s	98.3%	<1%	240 MB
10s	98.5%	<1%	240 MB
30s	92.7%	<1%	240 MB

C. Zeek performance

Zeek is the NIDS used to perform traffic inspection. It is well known that depending on the traffic characteristics and the number of scripts/modules loaded in Zeek, the inspection performance (measured as the reported capture loss by Zeek) can be affected. To evaluate this possible performance degradation, in this early work we have tested several Zeek setups. Setup A uses default Zeek configuration. Setup B loads 1 extra module (the module we have developed to enrich connections with DNS information) and Setup C loads 2 extra modules (setup B plus one module that reports connection statistics every minute). Table IV shows the results for the different setups. We have performed 8 different tests downloading a large file (701 MB) form the Internet at 160 Mbps downloading speed for 1 minute and we provide the averaged results when the downloading device was connected to the ISP router using an ethernet cable and when using a 5GHz wifi connection.

Table IV
ZEEK TEST RESULTS

	Setup A		Setup B		Setup C	
	Loss	CPU	Loss	CPU	Loss	CPU
5Ghz	0%	28%	0%	30%	0%	31%
Ethernet	0%	28%	0%	33%	0%	35%

IV. DISCUSSION

As we have seen in the results, the Raspberry Pi setup limits the maximum bandwidth to 800 Mbps when used as single spot of traffic forwarder in the proposed scenario for home networks seamless traffic analysis. Considering the bandwidth that can be reached when this scenario is not in use, i.e., 910 Mbps, there is a drop of 12% in the maximum bandwidth when it is included. In real home network scenarios where the maximum Internet bandwidth is less than 1 Gbps in normal conditions, this drop would not event exist since the available bandwidth will be close to 800 Mbps and thus there will be no penalization.

Regarding ARP spoofing technique, it is interesting to see how any type of ARP packet used works with end devices (devices from 6 to 10). This would indicate that ARP implementation in end devices no matter what OS is not very restrictive. Tested routers in general are also very flexible with only one exception: device 1. This router seems to implement some kind of restriction since only packets C and D work on it. Nevertheless, using ARP requests instead of the commonly used ARP replays makes the deal. For our implementation, D class packet has been used.

The interval used to send the ARP spoofing packets is important to obtain the desired scenario, i.e., all the packets are sent to the Raspberry Pi meaning that ARP tables in all devices are permanently poisoned. Since ISP router and end devices continue sending legitimate ARP packets, it could be possible that ARP tables revert to their original state and for some time, the packets would not be sent to the Raspberry Pi. Our tests have shown that this effect is neglectable if the ARP spoofing interval is less than 10 seconds, being 10 seconds a very good interval choice. Running the python script that performs the spoofing with this time interval consumes very few resources, keeping the CPU below 5% and the RAM usage below 240 MByte.

Regarding Zeek usage, results show that at the tested level, traffic analysis can be done with no performance degradation (0% of uninspected packets) being completely seamless to home users, even when connection statistics were reported every minute. CPU usage levels are less than 35% in all cases, which is a low value. This CPU results could be improved if more than one worker would be used. Nevertheless, more tests must be done with different traffic characteristics and at different bandwidth levels to find the real limitations of Zeek usage with the Raspberry Pi setup.

V. CONCLUSIONS

We have presented a research-in-progress early results of using a low-cost hardware device (Raspberry Pi) with a plug and play configuration (using APR spoofing) for home networks traffic inspection, enabling anomaly detection and classification. Results have shown that our setup is reliable with no penalization for home networks, being completely invisible for their users. Thanks to this setup, there is no need to purchase extra devices to have access to traffic inspection (apart from the Raspberry Pi computer). Besides, it opens the door to evolve the setup into an Intrusion Prevention System since the Raspberry Pi probe is in the middle of any communication, with the capability of terminating any of them if a potential threat is detected. This architecture presents no bottleneck risk since if the probe crashes or has any misfunction, the ARP spoofing attack will end, and the ARP tables of the home network devices will be restored with the legitimate ARP packets information. Although it is a research-in-progress work, these early results are very promising and could provide home networks with an easy-to-implement NIDS security layer.

ACKNOWLEDGEMENTS

This research is funded by Gobierno de Aragón (reference group Cenit T31_20R) and Universidad de Zaragoza (project reference UZ2021-TEC-01).

REFERENCES

[1] https://www.pcmag.com/picks/the-best-smart-home-devices. Last visited April 2022

[2] https://www.kaspersky.com/resource-center/preemptive-safety/how-to-set-up-a-secure-home-network. Last visited April 2022

[3] https://zeek.org. Last visited April 2022

[4] https://en.wikipedia.org/wiki/ARP_spoofing. Last visited April 2022

[5] https://docs.zeek.org/en/master/cluster-setup.html. Last visited April 2022

Designing a platform for discovering TOR onion services

Javier Pastor-Galindo[1], Roberto Sáez Ruiz[2], Jorge Maestre Vidal[2], Marco Antonio Sotelo Monge[2], Félix Gómez Mármol[1], Gregorio Martínez Pérez[1]
[1]Department of Information and Communications Engineering, University of Murcia, 30100, Murcia, Spain
{javierpg, felixgm, gregorio}@um.es
[2] INDRA, 28108 Madrid, Spain
{rsaezr, jmaestre, masotelo}@indra.es

***Abstract*—Anonymous networks such as TOR have been a haven for cybercriminals to offer illicit activities. As a result, analysts, researchers and law enforcement agencies are interested in monitoring these sites, known as onion services. However, finding the random links that give access to them is a challenge, as there is no native mechanism to search or discover. In this paper, we propose a modular and extensible platform to actively launch automatic discovery processes across the surface and deep web to find .onion addresses. In particular, the solution incorporates pivoting techniques to move from public suspect resources to unindexed services, such as alternative web servers, transfer protocols, or invisible documents. Throughout the continuous execution of discoveries, both the identified onion services and the advertised resources where they have been found are recorded in a knowledge database, providing valuable insight into how these onion services are transferred and publicized.**

***Index Terms*—onion services, TOR, anonymity, privacy**

Type of contribution: *Short article*

I. Introduction

The TOR (The Onion Router) project implements one of the most important anonymous networks of today. The privacy, accessibility and operational properties of this anonymous network, which differentiate it from the commonly known network, make it a darknet. In conjunction with other similar networks such as Freenet, I2P or ZeroNet, they form the dark web, thus differentiating it from the content of the surface web or clearnet, which is accessed via traditional browsers [1].

The resources and web pages available on the TOR network are called "onion services", although they have traditionally been known as "hidden services". Users of TOR must employ a client, software or program capable of connecting and communicating through the TOR network, such as the browser *TOR browser*. On the server side, there is a set procedure for the administrator to open and publish a onion service in a secure manner [2].

Thanks to the TOR routing and encryption, the identity of any user is protected, as well as those of the server and administrators [3]. Unfortunately, due to the high level of anonymity it offers, this network is often exploited to offer illicit services openly. It is easy to find black markets for weapons, drugs, or stolen data, the sale of child pornography, or on-demand cybercrime services (cybercrime-as-a-service), among others [4].

In order to access the advertised content on the TOR network and analyze its content, it is necessary to discover the onion services hosted throughout the TOR network. In this sense, a researcher or analyst has to obtain the link that gives access to the onion service. In the current TOR version 3, the link consists of 56 random characters encoded in base 32 and followed by "*.onion*"[1]. It is important to note that those version 2 onion services that use 16 characters are no longer accessible from September 2021.

The discovery of onion services is a complex task. This is mainly due to the anonymous nature of this network, the random and unmemorable addresses, the absence of a Domain Name System (DNS) within TOR, or the low longevity of these services that frequently cause link changes [5]. This is compounded by the need for continuous monitoring of as much of the TOR network as possible. To face these limitations, in this article we propose a modular and extensible platform to identify onion services on the web, particularly reaching those unindexed ones transferred on the deep web.

II. Related work

Different approaches and scopes have been explored to search and discover onion services in TOR. Some target directly the general search for onion services while others focus the search on onion services that offer a specific type of service. Unlike TOR v2, there is not a wide variety of solutions based on TOR v3 (the latest version) as the changes brought about by the version upgrade have potentially turned TOR v2 solutions obsolete.

The classic search engines (Google, Bing, DuckDuckGo, etc.) are not able to directly index onion services as valid sites in response to queries, but they can be useful to identify onion addresses in surface content. The Darknet search engines, such as Ahmia, only return those onion services registered manually or crawled through the dark web. Nevertheless, the number of indexed onion services is much smaller than the nearly 900,000 currently in existence according to official figures[2]. Finding a TOR service is only possible after the publication has been made by the service owner either on Ahmia or in any sort of forum or website available on Tor, or even in the surface web by services aimed on collecting/grouping *.onion* domains as is the case of Tor Links, Pastebin, or The Onion Wiki.

[1]For example, the official homepage of the TOR project within its own network is *2gzyxa5ihm7nsggfxnu52rck2vv4rvmdlkiu3zzui5du4xyclen53wid.onion*

[2]https://metrics.torproject.org/hidserv-dir-v3-onions-seen.html

Research works addressing this topic are those as presented by Nair & Kannimoola [6] whom describe a bug in TOR directory nodes that allows the user to extract onion service names from the memory. Theoretically, this attack is applicable to TOR v2, although its applicability to TOR v3 should be verified. A different approach is presented by Oldenburg et. al. [7], where the foundations are laid to take advantage of the possibility of detecting the "Guard" node of TOR network entry by a client through honeypot and relay injection mechanisms. Although it does not addresses the discovery of onion services, it is of interest as it is applicable in TOR v3.

Based on the dynamic nature of TOR services, Höller et. al. [8] explore the possibility of using short-lived dynamic services in TOR running on distributed networks for the exchange of confidential information between peers. It gains in relevance in relation to the research problem and interest since couples with the potential future applications of the TOR network. A deeper dive on TOR V3 onion services is presented in [9] analyzing the changes brought about in TOR v3 addressing. This work also establishes certain parallels between both versions, clarifying the improvements that the new version implies. At the same time, an approximation to statistical information acquisition methods of the TOR network is carried out through the HSDir node injection. On the other hand, Meng & Fei focused on the identification of the descriptor publishing flow based where a profile-hidden Markov Model-based descriptor publishing flow correlation attack (DPFCA) is presented [10]. There, the circuit establishment and publishing is unveiled from a series of nodes with a certain presence in the network, to nodes trying to publish their service in the directory service.

Besides research initiatives concerning the analysis of TOR onion services, a set of tools aimed on autonomous service discovery have been proposed:

- *Tor-oriented Web Mining Toolkit* [11]: The National Research Council of Italy designed a tool that runs through the darkweb,it allows feeding other semantic services and building graphs for the representation of knowledge. This tool is capable of obtaining onion services and analyzing their content for monitoring tasks. An initial set of seeds is necessary to start the search processes.
- *MASSDEAL* [5]: Eindhoven University of Technology and Radboud University implemented a tool that automatically explores and analyzes TOR onion services. To discover new onion services, resources with more than 10 onion links are marked as listing. and they are used as input for the next crawling process. A blacklist is also made to discard those sites that stop working.
- *Darkweb Monitoring Application* [12]: In this application, the authors propose a methodology for analysis, classification, and visualization of onion services of the TOR network. Obtaining onion services is done through previously existing listings. Specifically, the Ahmia search engine is used.
- *Automated Tool for Onion Labeling (ATOL)* [13]: This framework was designed for the analysis and automatic categorization of onion services on the TOR network, it has a crawler that is activated daily and collects onions on pages from seed sets (onions datasets, DNS resolution collections, and generic repository web information) and web searches. On the other hand, it is capable of inspecting the pages with specific modules, extracting the main themes of the web page.
- *Analytical Framework for Darkweb scraping and analysis* [14]: In this application, the authors propose a methodology for scraping an objective onion service, specifically buying and selling services (marketplace). In practice, the scripts for the analysis of each target onion service have to be adapted to each. It includes a very interesting research module with Maltego to extend knowledge about vendors. In an experiment shown by the authors, the analyzed onion service is obtained after having visited the Reddit forum and the DeepDotWeb page.

On the other hand, patents approaching the discovery of onion services were recently published:

- *Onion service discovery method based on meta-search, 2020, China (Institute of Information Engineering, CAS)* [15]: The invention proposes a meta-search-based onion service discovery method so that new onion services can be discovered on anonymous networks. The algorithm has a series of steps, where taking some initial keywords, it performs a first search for onion addresses. These onion addresses will be subjected to predefined matching rules, and if they follow the rules, they will be scraped for new links.
- *Design method of novel dark net mining robot, 2020, China (Tianjin Tingge Network Technology)* [16]: This invention provides a method for deep mining a specified website or forum on a darknet. The method use a dark web crawling tool, deep mining is carried out on a specified website, and a correlation analysis is performed according to the keywords predefined, once it is found a web page containing the keywords is stored in a results file.

III. PASTOR: Platform for Analyzing Services in TOR

PASTOR is a platform for the analysis of indexed and non-indexed resources (on the surface and deep web, respectively) to collect onion addresses automatically. The system registers both the onion services found and the advertising resources where they have been found. Thus, in PASTOR, an advertiser is a resource (web page, forum, community, social network, document, etc.) where at least one onion address appears.

A. Architecture of PASTOR

PASTOR is built with a series of decoupled but interconnected components that together implement the necessary flow for the identification of onion services.

As shown in Figure 1, our proposal is not a monolithic and rigid application, but is conceived as a modular and extensible infrastructure with the following components connected via REST API:

- *Web Interface* and *API*: Permit the platform management through the configuration of parameters to model the discovery behaviour (begin date, end date, maximum

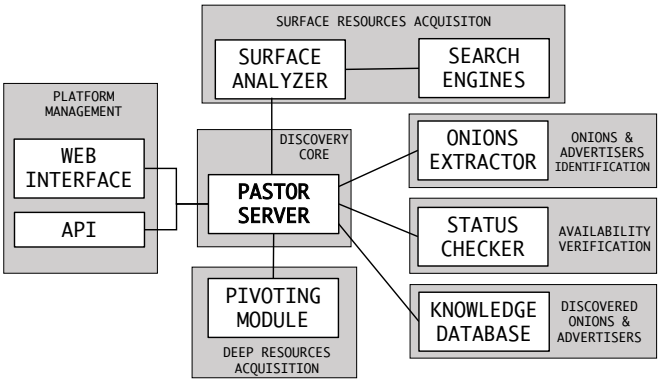


Fig. 1: Architecture of PASTOR

depth, number of threads, number of seeds to process, maximum resources to analyze, etc.) or to request data of discovered onions and advertisers (list of onions discovered, list of advertisers identified, and associated statistics). The *Web Interface* exposes user-friendly dashboard and *REST API endpoints* are exposed for programmatic write and read calls.

- *PASTOR Server*: Represents the core of the platform, keeping the state of the discovery process, handling user requests, and executing from a centralized point the distributed workflow (described in Section III-B).
- *Surface Analyzer* and *Search Engines Module*: Seek and identify indexed resources on the surface web potentially advertising onion services. The *Surface Analyzer* heuristically selects seeds to feed the *Search Engines Module* and group surface resources.
- *Onions Extractor*: Analyzes resources to identify onion addresses in the web content with a regular expression, registering the associated advertiser in positive cases.
- *Status Checker*: Verifies the active or inactive status of onion services. The status is checked when a new onion service is identified, as well as at a frequency that can be configured in the system. The latter allows the frequent intermittent status of onion services to be monitored.
- *Knowledge Database*: Saves onion services discovered (identificator, onion address, discovery date, state, and advertiser), associated advertisers (identificator, URL, discovery date, previous resource, and onion services), and configurations (set of platform management parameters).
- *Pivoting Module*: Executes intelligence techniques to move the discovery process from advertisers exposed on the surface web to correlated resources on the non-indexed deep web. For example, discover domains, subdomains and associated websites, explore correlated IP addresses and open ports, access to non-web protocols such as FTP, NNTP, or torrents, trace cryptocurrency transactions, visit chat-rooms through invitation links, etc.

In the following, we comment on the execution flow of a onion service discovery process.

B. Discovery process

The user of the platform can activate the process of onion discovery through the web interface or API calls. The engine is prepared to iteratively obtain seeds and fuel the discovery process, as presented in Figure 2:

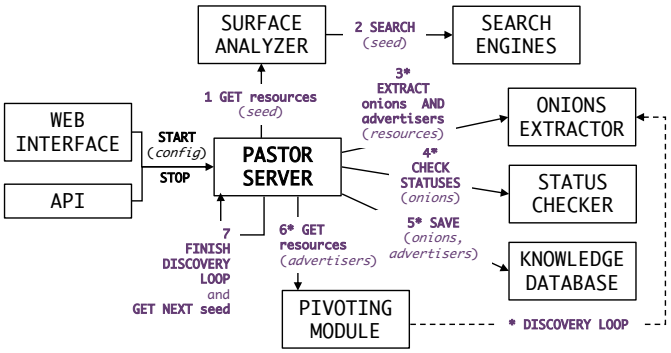


Fig. 2: Discovery flow of onion services for a given seed

1) The system selects and sends a new `seed` to the *Surface Analyzer*. Although we can use any seed, we propose to introduce an onion address to guide the discovery process around it.
2) The *Surface Analyzer* finds `resources` related to the seed through search techniques. Particularly, we rely on well-known and optimized search engines.
3) The *Onions Extractor* analyzes the aforementioned `resources` with a particular depth, compiling `onions` addresses if found and the `advertisers`.
4) The *Status Checker* visits the `onions` through the TOR network to verify the active or inactive status of them.
5) The system saves in the *Knowledge Database* the information regarding `onions` and `advertisers` identified. Therefore, this structure registers the onion services and the resources that are publishing them, thus keeping the relationship between them.
6) The *Pivoting Module* receives the list of `advertisers` that contained onion addresses and executes a set of pivoting techniques to reach other resources. In particular, this module seeks to jump to i) resources not indexed on the web, ii) pages without in-links, and iii) other data facilitators (chatrooms, file transmission protocols, torrents, etc.). The system sends the set of new pivoted `resources` to the *Onions Extractors*, thus conforming a *discovery loop* from step 3 to step 6.
7) The *discovery loop* for a given `seed` finishes when the set of resources is empty or the maximum number of resources to analyze per seed is reached. The platform selects a new seed to fuel a process of discovery again, going to step 1.

It is worth mentioning that these steps are decoupled and asynchronous, and can be executed with parallelism respecting the order of the calls. Additionally, the discovery flows can be executed in threads with different seeds at the same time to extend the capacity of the platform.

C. Properties of the platform

The design of this platform and the internal engine of onions discovery have a number of advantages:

- *Continuous update of the collection process*. The system is designed to be constantly fueled with seeds (onions), heuristically selecting new seeds to detect unexplored discovery paths.
- *Transparent unification of search surfaces*. The *Surface Analyzer*, *Onions Extractor* and *Pivoting Module* automatically launch the discovery process through the surface, deep, and dark web, respectively. This permit iterative jumps to extend the investigation beyond a human capacity, registering the relationship between these surfaces when onion addresses are identified.
- *Focused search to find onion services*. The platform does not launch a massive crawling, but guides the search to amplify the probability of encountering new onion services, discarding low-potential paths. The main features are i) the selection of goal-oriented seeds, ii) the correlation of identified resources through the pivoting techniques, and iii) the amplification of these benefits with the internal discovery loop through successful advertisers.
- *Execution of alternative discovery paths*. Apart from traditional crawling tasks based on links, the pivoting workflows of the platform enable the exploration of non-indexed resources (deep web). In this sense, the internal discovery loop being seed with non-indexed advertisers would power the analysis of the deep web, extending the investigation automatically towards remote, hard-to-reach sites.
- *Highly extensible*: The platform is composed by decoupled components with well-defined communication interfaces. Therefore, the integration of new functions is easy with the deployment of dedicated microservices and refinement the discovery flow in *PASTOR Server*.
- *Atemporal and evolvable*. The solution will work in the long term and does not rely on techniques or technologies with an expiry date, does not depend on the specific version of TOR, and does not exploit bugs or vulnerabilities that can be patched. In fact, the proposed discovery process can be enhanced extending, in type and number, i) the set of search techniques to find resources in the surface web (currently based on search engines), and ii) the group of pivoting techniques to reach the deep web.

IV. Conclusions and future work

Over the past few years, many efforts have been made to find onion services in TOR exploiting some vulnerabilities or leveraging potential inconsistencies in the TOR configuration. Therefore, as the TOR network itself has been updated, these onion services searching approaches became obsolete or simply stopped working. The solution proposed in this work aims to be version agnostic from the TOR standpoint, so that it can persist over time despite significant changes in technology, or updates that correct possible failures. With the proposed platform, an attempt has been made to solve the problem of discovering onion services in TOR, combining searching and pivoting techniques to work efficiently in the discovery process. At this early stage, the presented research lays on solid grounds with a promising outlook. Although the prototyping is evolving with good progress, its viability and efficiency for finding onion services are yet to be challenged in advanced validation scenarios. One of the most immediate lines of future work is to conclude the prototyping phase and to strengthen the discovery flow logic for a given seed, paving the way for a consistent framework validation.

Acknowledgments

This study was funded by the Spanish Government with grant FPU18/00304, and PASTOR (Platform of Analysis of Services in TOR) project, co-funded by the *Instituto para la Competitividad Empresarial* of the *Junta de Castilla y León* and the European Regional Development Fund.

References

[1] C. Cilleruelo, L. De-Marcos, J. Junquera-Sánchez, and J.-J. Martínez-Herráiz, "Interconnection Between Darknets," *IEEE Internet Computing*, vol. 25, no. 3, pp. 61–70, 2021.

[2] D. L. Huete Trujillo and A. Ruiz-Martínez, "Tor hidden services: A systematic literature review," *Journal of Cybersecurity and Privacy*, vol. 1, no. 3, pp. 496–518, 2021.

[3] M. Simioni, "Investigative Techniques for the De-Anonymization of Hidden Services," *IEEE Security and Privacy*, vol. 19, no. 2, pp. 60–64, 2021.

[4] I. Karunanayake, N. Ahmed, R. Malaney, R. Islam, and S. K. Jha, "De-anonymisation attacks on Tor: A Survey," *IEEE Communications Surveys Tutorials*, p. 1, 2021.

[5] P. Burda, C. Boot, and L. Allodi, "Characterizing the Redundancy of DarkWeb .Onion Services," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ser. ARES '19. New York, NY, USA: Association for Computing Machinery, 2019.

[6] V. Nair and J. M. Kannimoola, "A tool to extract onion links from tor hidden services and identify illegal activities," in *Inventive Computation and Information Technologies*, S. Smys, V. E. Balas, and R. Palanisamy, Eds. Singapore: Springer Singapore, 2022, pp. 29–37.

[7] L. Oldenburg, G. Acar, and C. Diaz, "From "onion not found" to guard discovery," *Proceedings on Privacy Enhancing Technologies*, vol. 2022, pp. 522–543, 01 2022.

[8] T. Höller, T. Raab, M. Roland, and R. Mayrhofer, "On the feasibility of short-lived dynamic onion services," in *2021 IEEE Security and Privacy Workshops (SPW)*, 2021, pp. 25–30.

[9] T. Hoeller, M. Roland, and R. Mayrhofer, "On the state of v3 onion services," in *Proceedings of the ACM SIGCOMM 2021 Workshop on Free and Open Communications on the Internet*, ser. FOCI '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 50–56. [Online]. Available: https://doi.org/10.1145/3473604.3474565

[10] Y. Meng and J. Fei, "Hidden service publishing flow homology comparison using profile-hidden markov model," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1081–1112, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22660

[11] A. Celestini and S. Guarino, "Design, Implementation and Test of a Flexible Tor-OrientedWeb Mining Toolkit," *ACM International Conference Proceeding Series*, vol. Part F1294, no. August 2018, 2017.

[12] N. Ferry, T. Hackenheimer, F. Herrmann, and A. Tourette, "Methodology of dark web monitoring," *Proceedings of the 11th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2019*, 2019.

[13] S. Ghosh, P. Porras, V. Yegneswaran, K. Nitz, and A. Das, "ATOL: A framework for automated analysis and categorization of the dark web ecosystem," in *AAAI Workshop - Technical Report*, vol. WS-17-01 -, 2017, pp. 170–178. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85046091132&partnerID=40&md5=97c23712172301caf6a2182866596ed2

[14] D. R. Hayes, F. Cappa, and J. Cardon, "A framework for more effective dark web marketplace investigations," *Information (Switzerland)*, vol. 9, no. 8, 2018.

[15] "Hidden service discovery method based on meta-search - patent cn110825950 - China," https://patentscope.wipo.int/search/es/detail.jsf?docId=CN289606995&_cid=P22-KZFD5U-78912-1.

[16] "Design method of novel dark net mining robot - patent cn112925966 - China," https://patentscope.wipo.int/search/es/detail.jsf?docId=CN327309219&_cid=P22-KZGT79-78024-1.

Generación sintética de topologías de red con Deep Learning: la *botnet* Neris como caso de estudio.

Francisco Álvarez-Terribas
Network Engineering & Security Group (NESG)
Dpto. Teoría de la Señal, Telemática y Comunicaciones,
ETSIIT,
Universidad de Granada
franciscoat@correo.ugr.es

Roberto Magán-Carrión
Network Engineering & Security Group (NESG)
Dpto. Teoría de la Señal, Telemática y Comunicaciones,
ETSIIT,
Universidad de Granada
rmagan@ugr.es

Gabriel Maciá-Fernández
Network Engineering & Security Group (NESG)
Dpto. Teoría de la Señal, Telemática y Comunicaciones,
ETSIIT,
Universidad de Granada
gmacia@ugr.es

Antonio M. Mora García
Free software for Optimisation, Search,
and Machine Learning (GeNeura)
Dpto. Teoría de la Señal, Telemática y Comunicaciones,
ETSIIT,
Universidad de Granada
amorag@ugr.es

***Resumen*—Uno de los problemas que afectan al rendimiento de sistemas de detección de intrusiones en red y en general a cualquier sistema de clasificación es el desbalanceo entre clases en los conjuntos de datos utilizados para su entrenamiento y validación. Esto ha sido abordado de forma recurrente en la literatura pero centrándose principalmente en la generación de muestras con variables continuas, obviando variables categóricas con claro interés para el problema de la detección de intrusiones, como pueden ser las direcciones IP o los puertos de un flujo de red. En este trabajo en curso se propone una metodología basada en la utilización de un VAE (Variational Autoencoder) para la generación de topologías de red sintéticas en una tipología de ataque concreto: la *botnet* Neris. Los resultados preliminares obtenidos, demuestran la viabilidad de esta propuesta.**

***Index Terms*—Generación sintética de datos, Deep Learning, Sistemas de detección de intrusos en red, Autoencoencoder Variacional.**

Tipo de contribución: *Investigación original, trabajo en curso.*

I. Introducción

En la actualidad es notable el incremento en el volumen de datos generados, la velocidad a la que se transmiten y su heterogeneidad debido a la alta conectividad entre dispositivos, sistemas, personas y cosas. Todo ello ha traído consigo un incremento de las amenazas a las que se enfrentan tanto individuos como organizaciones, ya sean estas empresas o gobiernos. Es por esto que se necesitan medidas de seguridad adicionales para hacer frente a todo tipo de amenazas de seguridad tanto conocidas como desconocidas (ataques *zero-day*). Para tal fin, tradicionalmente, se ha hecho uso de sistemas detección de intrusiones soportados por diferentes tecnologías, técnicas y algoritmos [1]. Es habitual que dichos sistemas se basen en la utilización de conjuntos de datos de tráfico de red para diferentes objetivos, normalmente para la clasificación de ataques o la detección de anomalías. Sin embargo, el principal inconveniente de estos sistemas, principalmente basados en técnicas de *Machine Learning*, es que precisan de conjuntos de datos adecuados y fiables para su entrenamiento, en los que es normal la existencia de diferencias notables entre la distribución de la clase positiva o ataque y la negativa o tráfico que sigue un comportamiento normal. Este hecho, junto con la utilización de conjuntos de datos no adecuados [1] en términos de representatividad, su actualización a nuevas muestras de ataque y su tipo, sintético o no, tienen un impacto notable en el rendimiento y viabilidad práctica de los NIDS (Network Intrusion Detection System).

Este trabajo sigue la siguiente estructura: primero se estudiará someramente el estado del arte en la Sección II para después describir el conjunto de datos utilizado y la metodología propuesta en la Sección III-B. A continuación exponemos de forma completa el entorno experimental del estudio junto con los resultados que hemos obtenido en la Sección IV-A. Posteriormente completaremos el documento con las conclusiones alcanzadas, así como posibles líneas de trabajo futuro en la Sección V.

II. Estado del arte

El desbalanceo de clases en conjuntos de datos es un problema recurrente en el desarrollo de sistemas de clasificación. Dicho problema cobra especial relevancia a la hora de implementar y evaluar NIDSs, ya que la clase minoritaria se hace muy complicada de detectar. Esto repercute directamente en la capacidad de los sistemas en producción para detectar anomalías. A continuación vamos a revisar algunos de los trabajos más relevantes en el campo de la generación de muestras sintéticas en el contexto de los NIDSs.

Los autores en [2] presentaron el algoritmo SMOTE (Synthetic Minority Oversampling Technique) que, junto a todos los algoritmos derivados de este [3], [4], ha sido el método más utilizado para la realización de *oversampling* o sobre-muestreo en conjuntos de datos desbalanceados. Principalmente se basan en seleccionar una instancia de la clase minoritaria, calcular cuales son sus vecinos más cercanos e interpolar muestras sintéticas entre dicha instancia y sus vecinos.

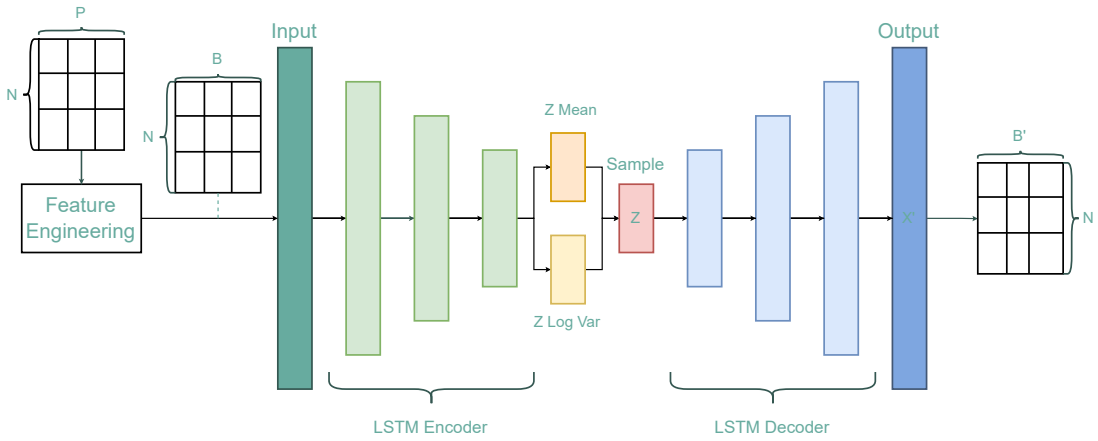


Figura 1. Arquitectura del VAE con capas LSTM.

Gracias a la irrupción del Deep Learning (DL) durante estos últimos años, se ha podido ver como diferentes autores empiezan a aplicar técnicas de esa rama para la generación de muestras sintéticas. Por ejemplo, Vu et al. [5] propusieron la aplicación de *deep generative adversarial models* sobre el dataset NIMS (Network Information Management and Security Group) [6], obteniendo ciertas mejoras en la clasificación de muestras respecto a algoritmos derivados de SMOTE. Posteriormente Engelmann et al. [7] plantearon una arquitectura de modelos generativos adversarios más robusta y capaz de trabajar con datos categóricos, aplicando técnicas de preprocesado *one-hot encoding* es decir, representando cada valor categórico como una *feature* con valor 0 (*low*) o 1 (*high*). Cabe destacar que dichos modelos, junto con los VAEs (Variational Autoencoders) [8] están siendo ampliamente utilizados para la generación de imágenes sintéticas [9].

Otra metodología a destacar, propuesta en los trabajos [10], [11], es la codificación de las muestras en un espacio latente. Una vez en este espacio, los autores aplican SMOTE u otro algoritmo derivado para, posteriormente, decodificarlas en el espacio de muestras original.

Todas estas soluciones funcionan muy bien a la hora de trabajar con variables continuas o aplicadas a la generación de imágenes sintéticas. Sin embargo, soluciones en las cuales se abordan problemas que involucran variables categóricas, como es el caso de los conjuntos de datos de tráfico de red, no siempre pueden aplicarse cuando existen variables que presentan un gran número de valores. En este contexto, la utilización de aproximaciones como *one-hot encoding* puede llegar a limitar su uso. Es por esto que en el presente trabajo se utilizarán modelos y arquitecturas VAE, por su capacidad de generación, para la obtención automática de muestras sintéticas de tráfico de red haciendo uso de conjuntos de datos predefinidos en donde, a partir de las variables categóricas, *i.e.* nodos (IP) y sus conexiones, se conformarán nuevos grafos dirigidos que representarán topologías de red sintéticas.

El problema que tratamos abordar en este trabajo es la ampliación de conjuntos de datos de red a través del empleo de técnicas DL, como los VAE, cuyo fin último es la mejora en el rendimiento y robustez de los NIDS en producción. De forma preliminar, para probar la viabilidad de nuestra propuesta, utilizaremos el conjunto de datos UGR'16 [12] y más concretamente las trazas de red relacionadas con el ataque etiquetado como *Botnet*, la Neris *botnet*. Aunque el objetivo final de nuestra propuesta es generar trazas de tráfico completas, en este trabajo preeliminar nos centraremos en la generación de direcciones IP que mantengan las características de las relaciones origen-destino de dichos datos. Estas relaciones entre direcciones IP conforman un mapa lógico el cual es denominado como topología de red.

III. Generación de topologías de red con Deep Learning

A continuación se introduce el conjunto de datos a utilizar así como la metodología propuesta para la generación sintética de topologías de red mediante la utilización de VAEs.

III-A. Conjunto de datos UGR'16: Neris botnet

El conjunto de datos UGR'16 [12] está formado por flujos de tráfico de red anonimizados (NetFlow) capturados durante 4 meses en las instalaciones de un ISP español de capa 3. Este se divide a su vez en dos: CAL y TEST. El primero de ellos solo contiene tráfico normal generado y visto en la red durante tres meses mientras que al segundo se le añaden ataques generados de forma sintética con herramientas actuales para su generación (*DoS (low y high rate)*, *Scan (Port Scanning)* o *Botnet*) y aquellos que fueron identificados por varios detectores (*UDP port scan*, *SSH scan* y campañas de *Spam*).

En el caso que nos atañe, nos centramos en uno solo de ellos: Botnet, hemos hecho esta elección en base a que presenta la tipología de ataque más compleja del dataset y, por tanto, la evaluación de las topologías de red generadas debe tener mucho más en cuenta el contexto. Este ataque, conformado por 2 millones de flujos, contiene el comportamiento y topología de la famosa *botnet* Neris [13]. Dicha *botnet* posee una estructura jerárquica en donde existe un *botmaster*, servidores C&C (*command and control* y los propios *bots* controlados por cada uno de ellos. Los *bots* se conectan a los servidores C&C HTTP para enviar *spam* y realizar ClickFraud (más detalles en [13]).

III-B. Metodología propuesta

En la *Fig. 1* se puede apreciar la arquitectura de nuestra solución. Esta está conformada por varias etapas o módulos

generales que son: una etapa previa de *Feature Engineering* y un VAE que hace uso de capas LSTM (long short-term memory) [14] principalmente.

En la etapa de *Feature Engineering* los flujos de red son pre-procesados para codificar las direcciones IP implicadas ($B = N \times P$, siendo N el número de observaciones y P el número de variables) en formato binario ($X = N \times P'$ con $P' > P$) adecuado para alimentar la siguiente etapa VAE.

La segunda etapa la conforma el VAE. Este codifica las muestras originales, en nuestro caso para la *botnet* Neris, en un espacio latente, siendo este una distribución gaussiana (representada por $ZMean$, la media de dicha distribución, y $ZLogVar$, la desviación estándar de esta). Partiendo de la distribución generada se extraen muestras aleatorias (representadas por Z). Una vez allí, dichas muestras son decodificadas presentando características similares a las muestras originales pero siempre con ciertas diferencias. Mediante el uso de capas LSTM (Long Short-Term Memory) se pretende que el modelo sea capaz de generar topologías de red teniendo en cuenta el contexto y temporalidad inherente de cualquier conjunto de datos de red para así replicar lo más fielmente posible el funcionamiento, topología, características y roles que definen el comportamiento de la *botnet* Neris.

IV. Experimentación y resultados

En esta sección se describe el entono de experimentación utilizado, así como la configuración de los experimentos para después evaluar su comportamiento.

IV-A. Entorno experimental y configuración.

Para llevar a cabo la implementación del modelo se ha hecho uso de la librería Tensorflow [15]. El modelo se ha configurado con tres capas LSTM tanto en el codificador como en el decodificador y como parámetros de entrenamiento se ha establecido un *learning rate* de 1e-5 durante 10 *epochs*.

El ajuste de este modelo lo realizamos con una partición de entrenamiento conformada por dos tercios del conjunto Botnet de UGR'16, que previamente hemos descrito. Por otra parte la generación de la realizamos utilizando una partición de test con el resto de flujos de red.

Actualmente agrupamos los flujos de red en *batches* de 75 flujos. Este parámetro es susceptible de ser configurado ya que se prevé que tenga un impacto relevante en el funcionamiento del modelo y debería de ser estudiado en mayor profundidad.

La máquina utilizada en este entorno experimental cuenta con un procesador Intel Xeon Silver 4208, a 2.10GHz y 32 núcleos; 32GB de memoria y tres Nvidia RTX 2080ti de 12GB.

IV-B. Resultados

Para evaluar el rendimiento del sistema propuesto de forma preliminar, hemos propuesto la comparación analítica y visual del conjunto original de datos con el generado por nuestro sistema. Podemos observar, en la *Fig. 2*, como los grados medios, de entrada y de salida de los nodos son menores en el conjunto de datos generado en comparación con el original. Por otro lado, la topología de red original y la generada, difieren principalmente en el número de nodos generados y no en tanta medida en el rol de estos, analíticamente podemos afirmar que la generación se realiza de forma correcta. Este

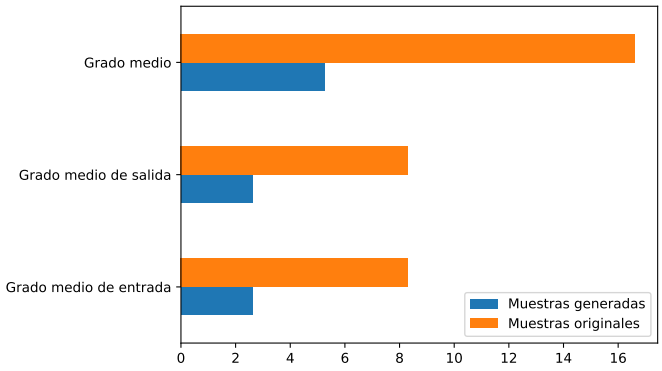


Figura 2. Grado medio de los nodos de red (tanto de la topología original como de la generada).

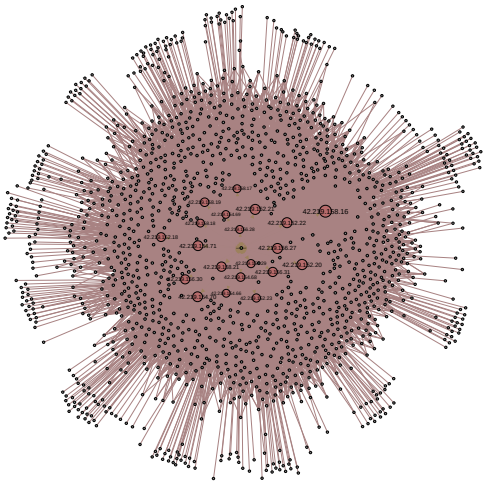


Figura 3. Topología completa de la *botnet* Neris.

hecho se observa en las *Figs. 3 y 4* para el conjunto de datos original y generado, respectivamente, dónde el número de *bots* (nodos coloreados en gris) y servidores C&C (coloreados en rojo) son claramente menores. Más en detalle, a través de las *Figs. 6 y 5*, vemos el papel *botmaster* (coloreado en verde) y sus conexiones principales con los nodos C&C. Concluimos así, como el conjunto de datos generado y, por ende el sistema en sí, es capaz de caracterizar los tres principales actores de la *botnet* Neris: *bots*, servidores C&C y el/los *botmasters*. Sin embargo, en estos resultados preliminares, no es capaz de representar la dimensión que abarca la *botnet* original en términos de número y conectividad entre nodos.

V. Conclusiones y trabajo futuro

En este trabajo se ve como de forma preliminar, nuestro sistema basado en VAE con capas LSTM es capaz de replicar en gran medida la estructura y topología de la red de la *botnet* Neris, en la que se caracterizan los roles y tipologías existentes en dicha *botnet*.

Como trabajo futuro, se tratará de generar observaciones completas de red que contemplen todas las variables originales y no sólo aquellas que conforman la topología de red como son las IP. Por otra parte sería también interesante la aplicación de modelos de Deep Learning más sofisticados, por ejemplo, la utilización de *transformers*, los cuales hacen uso

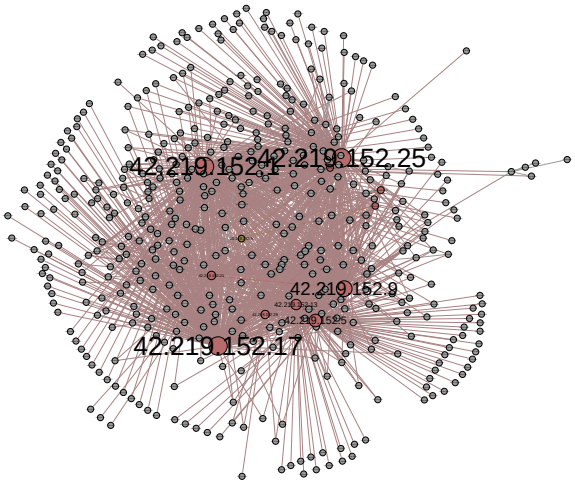


Figura 4. Topología generada completa.

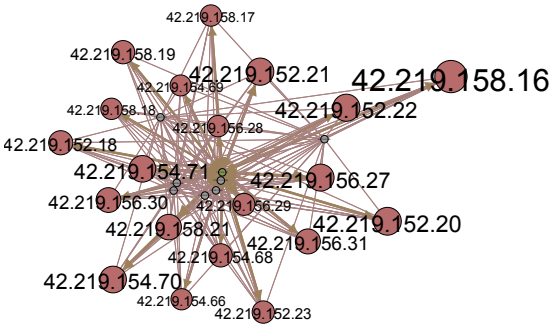


Figura 5. Detalle de la topología generada. Se muestra el nodo *botmaster* en verde rodeado por los nodos C&C coloreados en rojo. El tamaño de los nodos representa su grado, a mayor tamaño mayor grado.

de *attention layers* [16]. Estudios de *audio speech recognition* [17] han demostrado que los *transformers* presentan un rendimiento similar a los modelos con capas LSTM, si bien presentan entrenamientos más estables y eficientes en tiempo. Finalmente, se evaluará el impacto que tiene la generación sintética de datos propuesta en la mejora del rendimiento, robustez y fiabilidad de sistemas NIDS.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto SICRAC (PID2020-114495RB-I00) del Ministerio de Ciencia, Innovación y Universidades, así como los proyectos PID2020-113462RB-I00 (ANIMALICOS), financiado por el Ministerio de Economía y Competitividad, P18-RT-4830 y A-TIC-608-UGR20 financiados por la Junta de Andalucía, y el proyecto B-TIC-402-UGR18 (FEDER y Junta de Andalucía).

Referencias

[1] R. Magán-Carrión, D. Urda, I. Diaz-Cano, and B. Dorronsoro, "Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches," *Applied Sciences*, vol. 10, no. 5, 2020.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[3] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328.

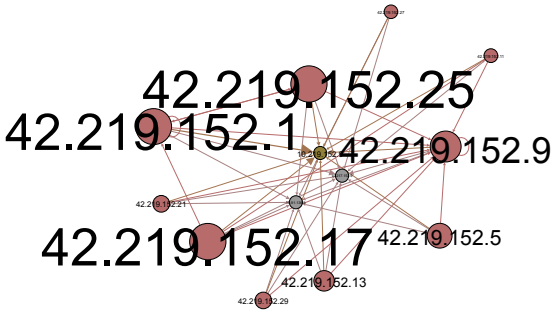


Figura 6. Detalle de la topología generada. Se muestra el nodo *botmaster* en verde rodeado por los nodos C&C coloreados en rojo. El tamaño de los nodos representa su grado, a mayor tamaño mayor grado.

[4] S. Barua, M. M. Islam, X. Yao, and K. Murase, "Mwmote–majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, 2014.

[5] L. Vu, C. T. Bui, and Q. U. Nguyen, "A deep learning based method for handling imbalanced problem in network traffic classification," in *Proceedings of the Eighth International Symposium on Information and Communication Technology*, ser. SoICT 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 333–339. [Online]. Available: https://doi.org/10.1145/3155133.3155175

[6] R. Alshammari and A. N. Zincir-Heywood, "Can encrypted traffic be identified without port numbers, ip addresses and payload inspection?" *Computer networks*, vol. 55, no. 6, pp. 1326–1350, 2011.

[7] J. Engelmann and S. Lessmann, "Conditional wasserstein gan-based oversampling of tabular data for imbalanced learning."

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[9] V. A. Fajardo, D. Findlay, C. Jaiswal, X. Yin, R. Houmanfar, H. Xie, J. Liang, X. She, and D. Emerson, "On oversampling imbalanced data with deep conditional generative models," *Expert Systems with Applications*, vol. 169, p. 114463, 2021.

[10] S. K. Lim, Y. Loo, N.-T. Tran, N.-M. Cheung, G. Roig, and Y. Elovici, "Doping: Generative data augmentation for unsupervised anomaly detection with gan," in *2018 IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 1122–1127.

[11] D. Dablain, B. Krawczyk, and N. V. Chawla, "Deepsmote: Fusing deep learning and smote for imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[12] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, "UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs," *Comput. Secur.*, vol. 73, pp. 411–424, 2018.

[13] "CTU-13 Dataset. Captura 42. Neris botnet." 2011.

[14] B. Bakker, "Reinforcement learning with long short-term memory," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. MIT Press, 2001.

[15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[17] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and lstm encoder decoder models for asr," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 8–15.

Análisis estadístico del tráfico DoH para la detección del uso malicioso de túneles

Marta Moure-Garrido, Celeste Campo, Carlos Garcia-Rubio
Dpto Ingeniería Telemática, Universidad Carlos III de Madrid, Avda. Universidad 30, E-28911 Leganés, Madrid, Spain
mamoureg@it.uc3m.es, celeste@it.uc3m.es, cgr@it.uc3m.es

***Resumen*—Las primeras versiones de DNS presentaban ciertos problemas de seguridad: integridad, autenticidad y privacidad. Para solventarlos se definió DNSSEC, pero esta versión seguía sin garantizar privacidad. Por ello, se definieron DNS sobre TLS (DoT) en 2016 y DNS sobre HTTPS (DoH) en 2018. En los últimos años se ha empleado la tunelización DNS para encapsular tráfico maligno. Las versiones DoT y DoH han complicado la detección de estos túneles dado que los datos van encriptados. En trabajos anteriores se emplean técnicas de aprendizaje automático para identificar túneles DoH, pero tienen limitaciones. En este trabajo realizamos un análisis estadístico para aprender el patrón del tráfico DoH y estudiar las diferencias entre el tráfico benigno y el tráfico creado con herramientas de tunelización. El análisis revela que ciertos parámetros estadísticos permiten diferenciar el tráfico. El siguiente paso de la investigación es aplicar técnicas más elaboradas basándonos en el análisis realizado.**

***Index Terms*—análisis estadístico, DoH, DoH maligno, tráfico, túneles DNS**

Tipo de contribución: *Investigación original (Artículos cortos (máximo 4 páginas))*

I. Introducción

La primera versión de DNS iba en claro y presentaba ciertos problemas de seguridad en relación a la integridad, la autenticidad y la privacidad, esta versión es conocida como Do53 (DNS sobre puerto 53) [1]. DNSSEC garantiza la integridad y autenticidad de las respuestas que recibimos, pero estas respuestas viajan sin cifrar por la red, dejando sin cubrir la privacidad. Con el propósito de resolver el problema de privacidad se definieron DNS sobre TLS (DoT) [2] en 2016 y DNS sobre HTTPS (DoH) [3] en 2018. DoH es la versión más extendida, en 2020 se introdujo en los principales navegadores web. En DoH, los mensajes DNS viajan encriptados por TLS a través del puerto 443.

La tunelización DNS permite explotar una conexión DNS como un canal de comunicación entre el cliente y el servidor, una forma encubierta de encapsular la transmisión de datos [4]. Estos túneles se pueden detectar analizando el contenido de los paquetes DNS. Sin embargo, en los túneles DoH [5], como el tráfico DNS está encriptado y no es perceptible para la infraestructura del cliente-servidor, estos métodos de detección pasan a ser obsoletos [6]. En investigaciones anteriores [7] se han utilizado técnicas de aprendizaje automático para detectar túneles DoH basándose en las características estadísticas del tráfico.

En la actualidad existen herramientas que generan túneles DoH, estos túneles permiten el envío de tráfico maligno dentro de las conexiones DoH. El objetivo de estas herramientas es crear túneles de datos para enviar tráfico encapsulado en consultas DNS que viajan a través de HTTPS [8]. Algunas de estas herramientas son: Iodine [9], dns2tcp [10] y dnscat2 [11]. El ataque de DoH malicioso más conocido se llama Godlua Backdoor, fue descubierto en 2019 por investigadores de NetLab [12]. Este ataque utiliza DoH como un canal de comunicación encubierto.

Como trabajo previo a un estudio que aplique técnicas de aprendizaje automático para detectar túneles DoH, en este trabajo analizamos las características estadísticas del tráfico DoH. El objetivo es estudiar si se puede diferenciar el tráfico DoH benigno y maligno a partir de parámetros estadísticos, siendo el tráfico maligno el tráfico creado con herramientas de tunelización. Además, este análisis permite obtener las características más significativas para aplicar técnicas de aprendizaje automático.

La estructura del artículo es la siguiente: la Sección II proporciona una visión general del estado del arte relacionado con el protocolo de comunicación DNS y la detección de túneles DNS. La Sección III describe el conjunto de datos utilizado en este estudio. A continuación, la Sección IV incluye el análisis realizado y se exponen los resultados obtenidos. Finalmente, las conclusiones de este trabajo y el trabajo futuro se presentan en la Sección V.

II. Estado del arte

II-A. Túneles DNS

Un túnel DNS permite encapsular datos en un paquete DNS para la comunicación entre el cliente y el servidor. Un método para detectar estos túneles frente a tráfico DoH benigno está basado en el análisis del contenido de los paquetes DNS. Se pueden analizar diferentes características de la carga útil y del flujo de los paquetes [4].

Las características extraídas de la carga útil son, por una parte, las relacionadas con el nombre de dominio (como longitud del nombre, número de caracteres especiales en el nombre del dominio o entropía de caracteres). Estas características tienen gran importancia en la detección de túneles ya que las diferencias entre los nombres de dominio del túnel y los legítimos son inevitables. Por otra parte, las relacionadas con el análisis estadístico del tamaño de los paquetes y de la proporción de los datos enviados y recibidos; y las relacionadas con los tipos de registros usados, las herramientas de tunelización tienden a utilizar tipos de registros poco comunes. Además, algunas herramientas de tunelización dejan una huella específica en el paquete DNS, un atributo en el encabezado DNS o contenido en la carga útil, por lo que también se puede utilizar esta característica.

Por otro lado, se analizan las características relacionadas con el flujo del tráfico DNS como el volumen de tráfico a

una determinada dirección IP o a un determinado dominio o el tiempo entre una consulta y una respuesta.

Parte de estas características, como el nombre del dominio o los tipos de registro usados, no son visibles cuando el tráfico DNS va sobre HTTPS porque el tráfico está encriptado, esto conduce a una mayor dificultad en la detección de los túneles DNS. Sin embargo, algunas de estas características siguen estando visibles, como las direcciones IP y los puertos, la longitud del paquete o el sello temporal [13], esto permite desarrollar investigaciones sobre técnicas automáticas para detectar este tipo de tráfico, como son los estudios estadísticos y las técnicas de aprendizaje automático.

II-B. Detección de túneles DoH

En la actualidad no hay mucha investigación sobre detección de túneles DoH. Una parte significativa de los trabajos que existen en esta línea se basan en un conjunto de datos de tráfico DoH tanto benigno, como maligno generado por MontazeriShatoori, Davidson, Kaur, et al. [7], llamado "CIRA-CIC-DoHBrw-2020". Los trabajos recogidos en la Tabla I se basan en este conjunto de datos con el objetivo de analizar y detectar túneles DoH usando diferentes técnicas de aprendizaje automático.

Tabla I
TRABAJOS RELACIONADOS

Referencia	Modelos	Objetivo
[7]	RF, DT, SVM, NB, DNN, CNN	2 capas (HTTPS/DoH, DoH benigno/ maligno)
[14]	DT, ET, RF, GB, LightGBM, XGBoost	2 capas (HTTPS/DoH, DoH benigno/ maligno)
[15]	DT, LDA, KNN, GNB, RF, AdaBoost, GB, ET, XGBoost, LightGBM	2 capas (HTTPS/DoH, DoH benigno/ maligno)
[8]	RF, DT, KNN, GB, XGBoost RNN, LSTM, GRU,	herramientas de tunelización
[16]	NB, LR, RF, KNN, GB	tráfico benigno/ maligno
[17]	KNN, SVM, DeepFM, RF	tráfico benigno/ maligno

En [7], [14], los autores presentan un enfoque basado en un clasificador de dos capas donde la primera capa clasifica el tráfico DoH y el tráfico no DoH (o HTTPS) y la segunda capa diferencia el tráfico benigno del maligno. Los resultados confirman que LightGBM y XGBoost superan a los demás algoritmos con una precisión del 100 % en [14].Alenezi y Ludwig [8] estudian si los clasificadores pueden clasificar las herramientas de tunelización, los clasificadores XGBoost y RF obtienen una precisión superior al 99 %. Singh y Roy [16] presentan varios clasificadores para detectar tráfico maligno, los clasificadores RF y GB son la mejor opción obteniendo una precisión del 100 %. En estos trabajos de clasificación se usan las 34 características del tráfico capturado como características de entrada: IP y puerto origen; IP y puerto destino; sello temporal; duración; número y tasa de bytes enviados y recibidos; media, mediana, moda, varianza, desviación estándar, coeficiente de variación, sesgo de la mediana y de la moda de la longitud del paquete, de la duración del paquete y de la diferencia de tiempo de solicitud/respuesta.

En [15], [17] se clasifica el tráfico sin tener en cuenta las direcciones IP ni los puertos. En [15], se introduce un método de selección de características que aumenta la precisión de la clasificación. Teniendo en cuenta la precisión y el tiempo de entrenamiento, LightGBM obtiene mejores prestaciones. El modelo de aprendizaje profundo presentado en [17] clasifica el tráfico con una precisión de 99,5 %.

Vekshin et al. [18] desarrollan un clasificador para diferenciar entre el tráfico HTTPS y el tráfico DoH y otro modelo para identificar el modelo de cliente DoH (Chrome, Cloudflare y Firefox) sin tener en cuenta la dirección IP ni los puertos. Los algoritmos de aprendizaje automático utilizados son KNN, DT, RF, NB y AdaBoost, AdaBoost obtiene la mejor precisión. Las características estadísticas con mayor importancia en la clasificación son la duración del flujo y el retraso medio ente paquetes y la varianza del tamaño de los paquetes recibidos.

En los trabajos anteriores se utilizan técnicas de aprendizaje automático supervisado. Nguyen y Park [19] proponen un sistema de detección de túneles DoH mediante una técnica de aprendizaje semi-supervisado basada en una arquitectura Transformer. Aunque se utiliza una técnica que no necesita que los datos estén etiquetados, la complejidad del modelo propuesto es mayor que los otros modelos.

En este trabajo estudiamos si se puede diferenciar el tráfico DoH benigno y maligno a partir de las características estadísticas del tráfico.

III. CONJUNTO DE DATOS

En nuestra investigación empleamos también el conjunto de datos "CIRA-CIC-DoHBrw-2020" [7]. Los autores proporcionan los archivos PCAP que contienen el tráfico sin procesar y el tráfico procesado mediante una herramienta que obtiene las características principales de los flujos de tráfico. Este conjunto de datos contiene tráfico HTTPS y tráfico DoH capturado a través de diferentes navegadores y herramientas.

El tráfico está clasificado en dos capas. La primera capa diferencia el tráfico HTTPS y DoH y en la segunda capa se clasifica tráfico DoH benigno y maligno. Este estudio se centra en el análisis del tráfico DoH benigno y DoH maligno, por lo tanto, nos interesa el tráfico de la segunda capa:

- Tráfico DoH benigno: tráfico generado mediante el acceso a sitios web usando los navegadores web Mozilla Firefox y Google Chrome.
- Tráfico DoH maligno: tráfico generado a partir de tres herramientas de tunelización (dns2tcp, dnscat2, iodine).

Los autores definieron el escenario de captura e implementaron la infraestructura necesaria para capturar el tráfico. En la Tabla II se pueden observar los detalles del tráfico capturado que constituyen este conjunto de datos.

Como hemos mencionado, el análisis de este estudio se enfoca en el tráfico DoH benigno y DoH maligno. Por lo tanto, se seleccionaron los archivos PCAP del tráfico DoH, tanto benigno como maligno.

En primer lugar, extraemos las características del tráfico de los archivos PCAP y las almacenamos para su posterior procesado. Obtenemos las direcciones IP (origen y destino), puertos (origen y destino), longitud (en bytes) y el sello temporal de cada paquete. Agrupamos los paquetes a nivel de conexión TCP teniendo en cuenta la apertura y el cierre de la conexión siguiendo la siguiente tupla:

$< IPorigen, IPdest, PuertoOrigen, PuertoDest >$

Las características de cada conexión (longitud y sello temporal) son almacenadas para realizar un análisis independiente

Tabla II
DETALLES DEL CONJUNTO DE DATOS [7]

Navegador/ herramienta	Servidor DoH	Num. paquetes	Num. total paquetes	Tipo
Google Chrome	AdGuard	5609K	2831K	HTTPS y DoH benigno
	Cloudflare	6117K		
	Google DNS	5878K		
	Quad9	10737K		
Mozilla Firefox	AdGuard	4943K	20611K	
	Cloudflare	4299K		
	Google DNS	6413K		
	Quad9	4956K		
dns2tcp	AdGuard	1281K	42436K	DoH maligno
	Cloudflare	3694K		
	Google DNS	28711K		
	Quad9	8750K		
DNSCat2	AdGuard	1301K	71025K	
	Cloudflare	12346K		
	Google DNS	48069K		
	Quad9	9309K		
Iodine	AdGuard	3938K	105997K	
	Cloudflare	5932K		
	Google DNS	73459K		
	Quad9	22668K		

de las direcciones IP y de los puertos. En la siguiente sección se presentarán los principales resultados obtenidos así como una breve discusión de los mismos.

IV. ANÁLISIS ESTADÍSTICO

En esta sección presentamos el análisis estadístico del tráfico DoH. El objetivo de este análisis es estudiar si existe una diferencia a nivel estadístico entre el tráfico benigno y el tráfico maligno. Las principales características del tráfico que vamos a estudiar son a nivel de conexión TCP, y son las siguientes:

- Número total de paquetes y de bytes enviados y recibidos durante la conexión.
- Duración de la conexión (la diferencia entre el sello temporal del último paquete y del primer paquete).
- Estadísticos de la longitud en bytes de los paquetes.
- Estadísticos del tiempo entre dos paquetes consecutivos, dos paquetes enviados y dos paquetes recibidos.

Se calculan las características estadísticas de cada conexión y después se promedian las que pertenecen a cada tipo de tráfico para obtener la media. Así se comparan las características de ambos tráficos. Las características del tráfico maligno se analizan teniendo en cuenta la herramienta por la que ha sido generado el tráfico, de esta manera podemos estudiar si existen diferencias entre estas herramientas de tunelización.

En los diagramas de caja presentados se pueden observar los estadísticos del tráfico maligno, capturado por cada una de las herramientas y del tráfico benigno. A continuación analizamos y comparamos los resultados obtenidos.

La duración media de las conexiones es similar en todos los tipos de tráfico, siendo las conexiones de las herramientas iodine y dnscat2 las más largas.

En primer lugar observamos el número de paquetes y el número total de bytes enviados. En la Fig. 1 se muestra la distribución del número de paquetes enviados. La media del tráfico generado por las herramientas dnscat2 e iodine es superior a la media de las conexiones del tráfico benigno.

Si nos fijamos en la longitud de los paquetes, en concreto en la media de los bytes enviados, el tráfico generado por las

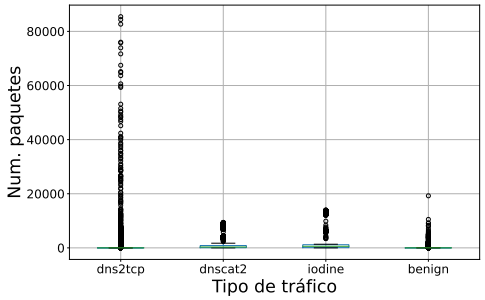


Figura 1. Boxplot (número de paquetes enviados).

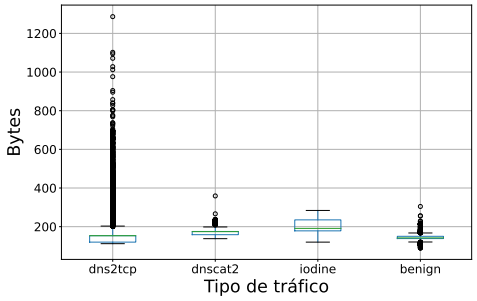


Figura 2. Boxplot (media de número de bytes enviados).

herramientas contiene mayor número de bytes, es decir, envían más datos. Esta diferencia se puede apreciar en la Fig. 2. En cuanto a la media de número de bytes recibidos por conexión se puede observar en la Fig. 3.

En cuanto al tiempo entre paquetes, se estudia el tiempo entre dos paquetes consecutivos, independientemente si son enviados o recibidos, el tiempo entre dos paquetes enviados y el tiempo entre dos paquetes recibidos. Destacamos el parámetro estadístico relativo al tiempo entre paquetes enviados. En la Fig. 4 se observa el diagrama de cajas sobre el tiempo medio entre dos paquetes enviados para los cuatro tipos de tráfico. Se puede observar una semejanza en la media del tráfico maligno, que es menor que el tiempo del tráfico benigno. Esto quiere decir que la frecuencia con la que se envían los paquetes en el tráfico generado por las herramientas de tunelización es mayor.

Si nos fijamos en ciertos parámetros estadísticos como la media del número de bytes enviados o el tiempo medio que transcurre entre dos paquetes enviados podríamos diferenciar el tráfico benigno del tráfico maligno. Sin embargo, hay otros parámetros estadísticos que no permiten esta diferenciación.

V. CONCLUSIONES

En este trabajo realizamos un análisis de las características estadísticas del tráfico DoH. Para realizar el estudio hemos seleccionado un conjunto de datos que contiene tráfico DoH de varios navegadores y de varias herramientas de tunelización. Hemos procesado el tráfico benigno y el tráfico maligno y hemos obtenido los parámetros estadísticos separando el tráfico en conexiones TCP.

Como ya se ha mencionado, este trabajo forma parte de una investigación del tráfico DoH con el objetivo de aprender los patrones de este tipo de tráfico y estudiar las diferencias entre

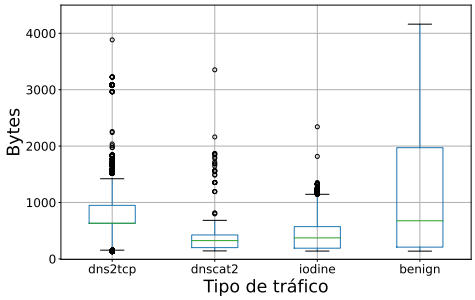


Figura 3. Boxplot (media de número de bytes recibidos).

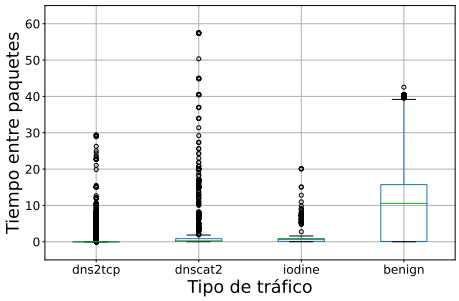


Figura 4. Boxplot (tiempo medio entre dos paquetes enviados).

el tráfico benigno y el tráfico maligno. Una vez analizados los parámetros estadísticos extraídos del tráfico se puede concluir con que hay ciertos parámetros que permiten diferenciar el tráfico maligno del benigno, como la media del número de bytes recibidos y, sobre todo, el tiempo medio que transcurre entre dos paquetes enviados, pero existen otros parámetros que no permiten esta diferenciación. Por otro lado, hay que tener en cuenta que el atacante podría conformar el tráfico maligno para parecerse al benigno, por ejemplo, el atacante puede reducir la cantidad de bytes enviados en cada petición, aumentando el número de peticiones. Una línea de investigación futura es estudiar si las herramientas de tunelización permiten realizar este conformado al tráfico.

El siguiente paso de la investigación es aplicar técnicas más elaboradas basándonos en el análisis realizado, es decir, explotando los parámetros estadísticos que permiten diferenciar el tráfico maligno del benigno.

Glosario

CNN	Convolutional Neural Network
DeepFM	Deep Factorization Machine
DNN	Deep Neural Network
DT	Decision Tree
ET	Extra Tree
GB	Gradient Boosting
GNB	Gaussian Naive Bayes
GRU	Gated Recurrent Unit
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
LSTM	Long Short Term Memory
LR	Logistic Regression
NB	Naive Bayes
RF	Random Forest
RNN	Recurrent Neural Network
SVM	Support Vector Machines

Agradecimientos

Este trabajo ha sido financiado por el proyecto COMPROMISE (PID2020-113795RB-C32) financiado por MCIN/AEI/10.13039/501100011033 y por la Comunidad de Madrid a través del proyecto CYNAMON (P2018/TCS-4566), co-financiado por los Fondos Estructurales de la Unión Europea (ESF and FEDER).

Referencias

[1] P. Mockapetris *et al.*, "Domain names-implementation and specification," 1987.
[2] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman, "Dns queries over https (doh)," 2016.
[3] P. Hoffman and P. McManus, "Dns queries over https (doh)," 2018.
[4] Y. Wang, A. Zhou, S. Liao, R. Zheng, R. Hu, and L. Zhang, "A comprehensive survey on dns tunnel detection," *Computer Networks*, vol. 197, p. 108322, 2021.
[5] D. A. E. Haddon and H. Alkhateeb, "Investigating data exfiltration in dns over https queries," in *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*, Jan 2019, pp. 212–212.
[6] N. Ishikura, D. Kondo, V. Vassiliades, I. Iordanov, and H. Tode, "Dns tunneling detection by cache-property-aware features," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1203–1217, 2021.
[7] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. Habibi Lashkari, "Detection of doh tunnels using time-series classification of encrypted traffic," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2020, pp. 63–70.
[8] R. Alenezi and S. A. Ludwig, "Classifying dns tunneling tools for malicious doh traffic," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021, pp. 1–9.
[9] iodine. [Online]. Available: https://github.com/yarrick/iodine
[10] dns2tcp. [Online]. Available: https://github.com/alex-sector/dns2tcp
[11] Dnscat2. [Online]. Available: https://github.com/iagox86/dnscat2
[12] A. Turing and G. Ye, "An analysis of godlua backdor," *360 Netlab Blog*, 2019.
[13] K. Bumanglag and H. Kettani, "On the impact of dns over https paradigm on cyber systems," in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*. IEEE, 2020, pp. 494–499.
[14] Y. M. Banadaki, "Detecting malicious dns over https traffic in domain name system using machine learning classifiers," *Journal of Computer Sciences and Applications*, vol. 8, no. 2, pp. 46–55, 2020.
[15] M. Behnke, N. Briner, D. Cullen, K. Schwerdtfeger, J. Warren, R. Basnet, and T. Doleck, "Feature engineering and machine learning model comparison for malicious activity detection in the dns-over-https protocol," *IEEE Access*, vol. 9, pp. 129 902–129 916, 2021.
[16] S. K. Singh and P. K. Roy, "Detecting malicious dns over https traffic using machine learning," in *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, 2020, pp. 1–6.
[17] H. Jha, I. Patel, G. Li, A. K. Cherukuri, and S. Thaseen, "Detection of tunneling in dns over https," in *2021 7th International Conference on Signal Processing and Communication (ICSC)*, 2021, pp. 42–47.
[18] D. Vekshin, K. Hynek, and T. Cejka, "Doh insight: Detecting dns over https by machine learning," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ser. ARES '20. New York, NY, USA: Association for Computing Machinery, 2020.
[19] T. A. Nguyen and M. Park, "Doh tunneling detection system for enterprise network using deep learning technique," *Applied Sciences*, vol. 12, no. 5, 2022.

FCTNLP: An architecture to fight cyberterrorism with natural language processing

Andrés Zapata Rozo[1], Daniel Díaz-López[1], Javier Pastor-Galindo[2], Félix Gómez Mármol[2]
[1]School of Engineering, Science and Technology, Universidad del Rosario, Bogotá, Colombia
{andresf.zapata, danielo.diaz}@urosario.edu.co
[2]Department of Information and Communications Engineering, University of Murcia, 30100, Murcia, Spain
{javierpg, felixgm}@um.es

***Abstract*—Law Enforcement Agencies (LEA) are everyday more and more concerned about illicit activities that may be found in cyberspace like cybercrimes, cyber espionage, cyberterrorism, cyber warfare, among others. In a cyberterrorism context, Hostile Social Manipulation (HSM) is a strategy that employs different manipulation methods mostly through social media to produce damage to a target state. The efforts to fight cyberterrorism could come along with new technologies that allow a faster and more effective control of offensive actions. For that reason, this paper proposes an artificial intelligence-based solution that processes posts in social networks using Natural Language Processing (NLP) techniques, applying the following three models: i) Sentiment Model to discriminate between threat and non-threat publications, ii) Similarity Model to identify suspects with similar intentions and iii) NER model that identifies entities in the text. Finally, the proposal was tested exhaustively to validate its functionality and feasibility, achieving an integrated and simple prototype.**

***Index Terms*—Cyberterrorism, Natural Language Processing, OSINT, Semantic Similarity, NER, Sentiment Analysis.**

Contribution type: *Consolidated scientific research*

I. Introduction

Cybercrimes are either committed against the integrity, availability, and confidentiality of computer systems and telecommunication networks or the use of such networks or their services to conduct traditional offenses [1]. Following the previous idea, we can consider cyberterrorism as a crime that aims to involve the generation of terror in the cyberspace with the aim of subverting the political order. Thus, it has a greater impact than conventional cybercrimes as it provokes a state of terror in the general public, a group of persons, or particular persons, for political, philosophical, ideological, racial, ethnic, religious or other nature purposes purposes[1].

One of the best-known cases of cyberterrorism began when ISIS posted a video on YouTube on August 19th 2014 titled "A Message to America", where the journalist James Foley was beheaded as a response to the authorization of offensive actions against this terrorist group by the Obama's government [2].

Another case of intimidation occurred on February 10th, 2020 when the Colombian guerrilla group ELN, through the squadron "Omar Gomez", announced an armed strike on the main roads of Colombia to be realized in the middle of that month[3]. Those announcements included publications on social networks as well. This armed strike intimidated the population forcing them to stay in their homes, and people who violated the restrictions could be victims of violence from this armed group. As a consequence, many Colombian towns and cities where ELN was present stopped most of their economic activities.

On the other hand, by applying Open Source Intelligence (OSINT), we can obtain knowledge that can be reached using publicly available data [2]. The Internet and especially social media have contributed to the growing importance of public information that can be extracted using OSINT tools. Also, these intelligence sources have been relevant for the defense enterprise due to their potential use in big data [3] [4].

The OSINT can be complemented using Natural language processing (NLP), which compounds computer science and linguistics to generate an approach to the understanding of the human language by a computer, this task is carried out through tools of artificial intelligence, statistics and grammar [5]. One of the most common examples of the application of NLP is a conversational agent that uses these techniques in order to understand the interlocutor language and emulates a functional conversation, taking into consideration variables such as the entities and the intention of the input text [6].

In this paper, we present an OSINT solution that extracts information from social networks and other resources. Then, such information is processed using NLP techniques including three models: i) a similarity model that relates text with similar semantic meaning, ii) a sentiment analysis model that estimates the polarity of a sentence, and iii) a Name Entity Recognition (NER) model that recognized relevant entities in the text and the type of these entities. All the results are integrated into a simple module that yields the output of this model generating a report that contributes actionable information to Law Enforcement Agencies.

[1]https://digitallibrary.un.org/record/631639?ln=es

[2]https://edition.cnn.com/2014/08/19/world/meast/isis-james-foley/index.html

[3]https://thecitypaperbogota.com/news/eln-announces-72-hour-armed-strike-warns-of-consequences-to-travelers/23849

II. State of the art

A compilation of remarkable works that use NLP to fight against cybercrimes is presented next. In [7] the authors use data from two datasets: one available online and another built with data from Twitter and Facebook and labeled manually to construct a cybercrime text classifier. In addition, they compare their different classifiers with sentiment analysis from the NLTK Python library.

The work proposed in [8] consists of a big data architecture that allows a real-time analysis of tweets to classify users and their respective followers as part of ISIS (a terrorist organization), according to parameters like level of activity, influence on other users, and post content. A graph was created where indicators of centrality were applied to identify the most influential users before applying analysis over the data. Finally, user profiles were obtained through Fuzzy clustering techniques.

Cyberterrorism may use hate speech as a technique to intimidate a specific target population. In [9] the detection of hate speech in the Arabic context was developed through the use of different comparative machine learning methods like Support Vector Machine (SVM), Naive Bayes (NB), Decision tree (DT) and Random Forest (RF). The data used in this work come from tweets related to racism, journalism, sports orientation, terrorism and Islam.

The detection of cyberterrorism vocabulary in web pages was proposed in [10] and in this work, the results of the following algorithms were evaluated: Random Forest, Boosting, SVM, Neural Network, K-Nearest Neighbor (KNN) and Naive Bayes, where a Random Forest approach gives the best results. In all the cases the percentage of accuracy was higher than 80% and in the case of the Random Forest approach was 95.62%. The vocabulary developed to detect the websites include information related to Al Qaeda, Supreme Truth, KKK and ETA groups.

An analysis of a historical dataset was presented in [11], where the data from Twitter messages of attacks between 2008 and 2019 by the terrorist group Boko Haram in Nigeria were analyzed using DynamicK-reference clustering algorithms. This work allowed the identification of various of strategies for Boko Haram attacks and pointed out weaknesses in security control in sectors of northern Nigeria.

The proposal presented at [12] tries to detect cyberterror and extremism in the text using Fuzzy sets-based weighting methods, Naive Bayes Multinomial (NBM) and SVM. The experimental analysis shows that the fuzzy set-based weighting method with SVM classifier gives the best classification with a 99.4% of accuracy.

As observed, the most recent works that make contributions in the fight against cyberterrorism through the use of NLP have the purpose of detecting terrorist behavior on the Internet.

III. Proposal of FCTNLP

This section describes the main aspects of the design of FCTNLP, covering the definition of requirements and the explanation of the main components. The development of this design follows the phases defined in a data science life cycle [13]: i) Business understanding, ii) Data acquisition and preprocessing, iii) Modeling, and iv) Deployment.

A. Business understanding

As seen in Section I, cyberterrorism is a real problem that affects the general population or a target state, and as a crime, it should be fought. Following this, one of the main problems in the fight against cyberterrorism is recognizing it in the middle of a large data flow, such as the one existing in social networks. In addition to the amount of data that must be analyzed, human-generated text may display different structures with different meanings [14]. Thus, one option to analyze sentences is to devote a person to extract key information from such human-generated text, however that analysis would be subjective and a very large group of people would be needed to analyze all content coming from social networks. This is why a solution like FCTNLP that automates the structuring and analysis of posts on social networks is vital in the fight against cyberterrorism. Thus, the architecture proposed for FCTNLP is expected to meet the following targets:

- Distinguish tweets: It should be capable of creating initial groups of tweets related by their semantic meaning.
- Evaluate polarity: It should contain a model that scores the polarity of each tweet.
- Recognize entities: It should be capable of extracting entities relevant to tracking cyberterrorism.
- Identify communities: It should use OSINT information related to the Twitter accounts that are generating content to identify existing relations between such actors.

B. Data acquisition and preprocessing

Amongst all social networks, Twitter has consolidated as an important source of data due to the relevance and diversity of data that can be obtained from it to be analyzed [15]. Thus, we consider Twitter as a social network with the possibility to generate a high impact beyond cyberspace and that is why such a social network was selected in this paper as the source of the raw data that feeds our proposal. In order to gather tweets, different OSINT tools and techniques may be used, which can be divided into two categories: i) Scrappers and ii) API-based tools. The first category contains scrappers that use bots, some of them emulating human behavior, to get specific data from Twitter, e.g. Octoparse[4]. These kinds of tools allow to personalize the type of data that will be extracted but, due to the policies of Twitter, most of these kinds of tools have a tweets extraction limit of 1000 tweets per day. The second category refers to tools that consume the Twitter API[5] and therefore run under the restrictions defined by such API, e.g. tweets can only be extracted from a short time window that may be up to the last 7 days previous to the date of the collection.

[4]https://www.octoparse.com/tutorial-7/scrape-tweets-from-twitter
[5]https://developer.twitter.com/en/docs/twitter-api

Once tweets are collected, many strategies of preprocessing can be used to prepare the text before feeding the NLP models. The principal objective of these strategies is to keep the meaning of the text while cleaning it from noise data that can influence in a bad way the performance of the models. The most common strategies applicable to tweets are: i) remove hashtags, mentions, URLs, strange characters and punctuation symbols, ii) normalize text that converts text into lower or upper case, iii) replace emojis for words that represent their meaning, iv) apply tokenization that divides the text in tokens that can be only words or words with punctuation symbols, and v) do lemmatization that replaces a word by their lemma, i.e the canonical form of the word. The different strategies used in the implementation will depend on the NLP model that will be fed with such preprocessed data.

C. Modeling

In this section three NLP models will be described. The first one is the sentiment analysis model that is used in the NLP tasks to determine the emotions that the author of a text expresses or the mood of the author at the moment of writing the text. Secondly, a NER model is used in NLP to extract relevant entities and their respective type from raw text. Finally, the similarity model is used to represent a text in a vector way so that the representation can contain the semantic meaning of the text. The inclusion of these models allows achieving the targets proposed in section III-A

1) Sentiment Analysis Model: A sentiment analysis model can be implemented as a classifier that discriminates text between classes according to the polarity of the text (positive, negative or neutral), classifies the subjectivity of the author (subjective, objective), or extracts the emotional state of the text (happy, angry, friendly, confident, etc.) [16].

Sentiment analysis models may also in a single function, e.x. regression function scores two or more aspects of the text like the polarity and subjectivity [17]. Finally, another way to implement a sentiment analyzer is using a rules-based algorithm using the knowledge about the language structure and the meaning of the words [18].

2) Name Entity Recognition Model: A Long Short Term Memory (LSTM) is a Recurrent Neural Network (RNN) that can take advantage of consecutive and non-consecutive terms to give the best results in the task of recognizing entities from human language. It may also offer an understanding of the relation between words and their grammatical meaning. Another architecture for this task is the Bidirectional LSTM (Bi-LSTM), where two LSTM models are concatenated in a way that the Bi-LSTM can receive information from the beginning of the text up to the end, and from the end up to the beginning [19]. Regardless of the RNN used for the implementation, a NER model is generally trained using a Begin Inside Outside (BIO) notation where a phrase is decomposed in beginning, inside and outside sections. Thus, a NER model adds actionable information to analyzed tweets such as the one that helps to describe where (location) and who (subject, organization) is involved in some specific actions being monitored.

Transformers are also used in NER tasks. They were proposed in [20] and consists of two stacks: one encoder and one decoder. Both inputs and outputs have embeddings and positional encoding. Each stack uses multi-head attention layers: a non-masked one for the encoder, and a masked one for the decoder. At the end of both stacks a fully connected feed-forward network is also placed. Finally, a linear layer and a softmax activation function are used to get the prediction.

3) Similarity Model: The similarity model uses word embeddings to represent the meaning of the words in a real space. Such representation is useful to get the relation between words that have a similar meaning as they will have a close vector representation. The metric used to calculate the similarity between words is the soft cosine distance, which is represented by Equation 1.

$$soft\_cosine(w, v) = \frac{\sum_{i,j}^{N} s_{ij} w_i v_j}{\sqrt{\sum_{i,j}^{N} s_{ij} w_i a_j} \sqrt{\sum_{i,j}^{N} s_{ij} v_i v_j}} \quad (1)$$

To generate the word embeddings two principal algorithms may be used: Word2Vec and FastText. In the case of Word2Vec, it creates a vector representation for each word in the text, keeping similar words close in the vector space [21]. This approach has the problem that words that are not included in the training set (new words included in tweets) will not be considered in the similarity calculus as they will not have a vector representation. FastText algorithm may help to solve this previous problem as it uses a vector representation that takes into account the n-grams of a word, i.e. the sequence of n characters. Thus, this last approach is capable of representing words that are composed of some n-grams contained in the training dataset, even if it losses the property of having semantic knowledge in the vector representation [21].

An illustration of how the previously described models (similarity model, sentiment analysis model and NER model) are integrated into the FCTNLP architecture is shown in Figure 1.

D. Deployment

The information generated by FCTNLP should be stored properly, so it can be recovered lately for additional processing. Amongst this information, we have certain diversity represented by: the gathered tweets, the outcomes from the polarity, the similarity and the NER models, and additional useful information that may be obtained by OSINT techniques. This scenario with heterogeneous and unstructured information suggests that graphs may be an adequate way to store the information. Additionally, graphs may be one of the most useful ways to show different types of information associated with the user accounts, the message polarity, the conformed clusters, and the identified entities, in the same space. For these previous reasons, FCTNLP incorporates in its architecture (Figure 1) a graph database that allows storing and representing nodes and relations.

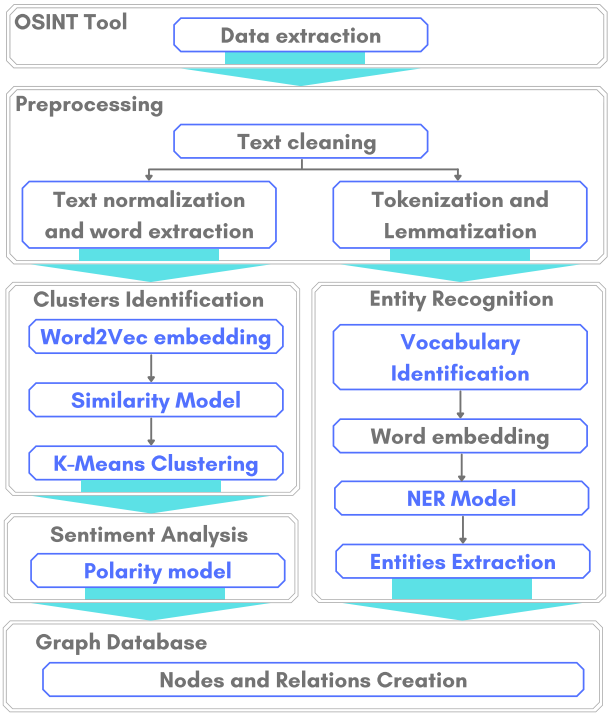


Figure 1: Components of the FCTNLP architecture

FCTNLP could be used by a Law enforcement Agency (LEA) to automatize the analysis of a human text obtained from open sources and help in the prevention of cyberterrorism. Such implementation can be a key tool for a cyber intelligence agent as it facilitates the search for spotlights of cyberterrorism. Results obtained from FCTNLP may also be enriched with the information provided by an already running commercial cyber intelligence solution.

Due to the modularity, the high cohesion, and the low coupling of the proposed architecture, each NLP model can be deployed as part of a scalable solution that allows the flow of a huge amount of data. For instance, to be deployed as a microservice. The extraction of tweets can also be set as a task to be executed in real-time or under demand. In both cases, extracted data can be saved in a data lake that will be processed by the NLP models that compose FCTNLP, and a cache solution can also be fed with the more recent and relevant results in order to have quick access to required data.

IV. Experiments

This section contains the results obtained from applying the proposal described in Section III in a scenario related to a protest that occurred on October 26, 2021, in Ecuador, being the data and code available in the project repository [6]. Twitter was the social network used to provide the raw information to be processed. The gathering was done using `TAGSv6.1`[7].

The embedding process in the English language was based on the use of Google News Embedding, which contains generic embeddings for 3,000,000 English words with a dimension of 300. On the other hand, the embedding of content in Spanish was done using the Spanish Billion Words Corpus and Embeddings[8] that contains a set of 1,000,653 words with a 300 dimensions vector representation. Both embeddings were built using the word2vec algorithm[9], as in section III. The extraction of information related to followers from Twitter accounts was made using tinfoleak[10] and the generation of the neighborhood graph for a selected cluster was made using Gephi[11].

A. Gathering tweets in the protest against economic policies in Ecuador

The straw that broke the camel's back was the announcement of the increase in fuel prices by the government of Guillermo Lasso[12], in addition to the economic crisis in which Ecuador finds itself and the fall in popularity of the Lasso government due to the investigation that he is facing due to he is appearing involved in the Pandora Papers[13].

This scenario implied the gathering of 10,608 tweets containing at least one of the following hashtags `#ParoNacional`, `#ParoNacionalEC`, `#LassoEsUnFracasso`, `#Quito`, `#LassoCorrupto`, `#LassoMentiroso` and in order to obtain only original tweets with text we use the following filters of the twitter API `-filter:images`, `-filter:videos`, `-filter:retweets`. Between October 24 and October 27 of 2021. The protest occurred between 26 October and October 27, 2021, for this reason the tweets before these dates were omitted so a total of 7,086 tweets remained.

The day of protests was marked by some acts of violence, the most serious of which was the confrontation between the demonstrators and the police in front of the presidential palace[14]. On the other hand, other disturbances occurred in various parts of Ecuador such as road blockades[15]. At the end of the protests, 37 people were arrested for acts of violence[16].

The data of this experiment is a unique collection from Twitter that uses TAGS. As we see in Section III, tools like TAGS that used the Twitter API to collect the information from Twitter have a limit to the tweets that can be collected in a window of time. Most of the time this limit is not reached, especially with specific topic queries like the use in this experiment.

[6]https://github.com/AndZapCod/NLP_Cybersecurity_Case
[7]https://tags.hawksey.info/
[8]https://crscardellino.ar/SBWCE/
[9]https://code.google.com/archive/p/word2vec/
[10]https://tinfoleak.com/
[11]https://gephi.org/
[12]https://www.argusmedia.com/en/news/2266703-ecuador-freezes-fuel-prices-update
[13]https://www.reuters.com/world/americas/ecuador-president-lasso-be-investigated-tax-fraud-after-pandora-papers-leak-2021-10-21/
[14]https://www.laprensalatina.com/quito-violence-marks-day-of-protests-against-ecuador-president/
[15]https://frontline.thehindu.com/dispatches/protesters-in-ecuador-block-roads-over-gasoline-price-hikes/article37195681.ece
[16]https://www.reuters.com/world/ecuador-demonstrators-block-some-roads-protests-over-gas-prices-2021-10-26/

Tweets were preprocessed and cleaned properly to be consumed by the models that will be used later in the pipeline. The first step in preprocessing was to construct a dictionary with the most used hashtags and mentions of the collected tweets and replace them with their meaning words. The second step was to remove URLs, mentions, hashtags, reserve words like RT and FAV, smilies and strange characters from the tweets using the python library `tweet-preprocessor`[17]. Then, emoticons were replaced by their meaning in words through the use of the Python library `emoji`[18]. Finally, empty and duplicated tweets were removed and a total of 7,077 tweets remained.

Additionally, preprocessing was required for each model. For the sentiment analysis model, punctuation symbols were removed from the text and the latter was normalized to lowercase. Regarding the similarity model, tweets were translated from Spanish to English using Google API Services[19], and punctuation symbols were also removed and the text was converted to lowercase. In the case of the NER model, cleaned tweets were tokenized and lemmatized. Using a python dictionary constructed in the training of the model, each token was converted to an integer to be used as input for the Bi-LSTM model see section III.

B. Application of the similarity model

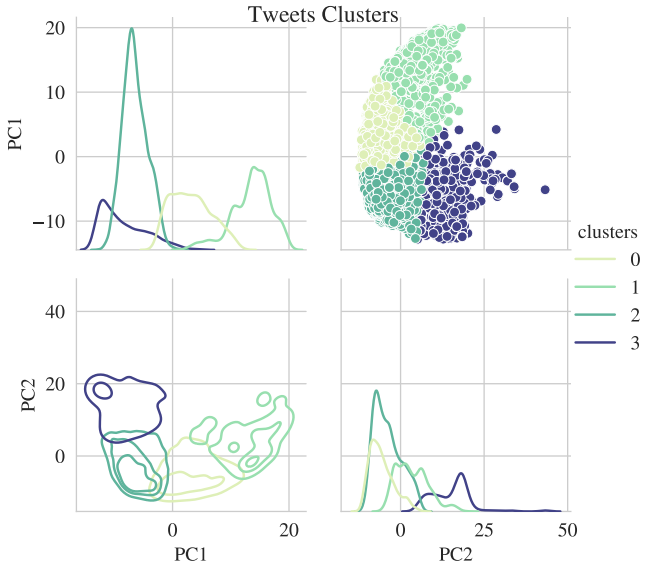


Figure 2: K-means clustering using Google News embeddings

For the analysis of the translated tweets, $(t_m = t_1, \ldots, t_n)$ were processed by the similarity model mentioned in Section III using the google news embedding to build a matrix of cosine distances between the different tweets. The entries of such a matrix were done by taking each tweet (t_i) and calculating their cosine distance against the remaining tweets. Afterward, The 7,077 tweets were split into four clusters according to the K-means algorithm using the similarity matrix in order to take advantage of the semantic representation of the tweets.

In Figure 2 we can see a representation of the tweets using the similarity matrix and the PCA algorithm to extract the two dimensions that represent the 86.4% of the variability of the data.

In Figure 3 we can see that the reasons for the protest are dominant in all of the clusters. In addition to this, the reason for the protest, Ecuador's president is more mentioned in clusters 0 and 1.

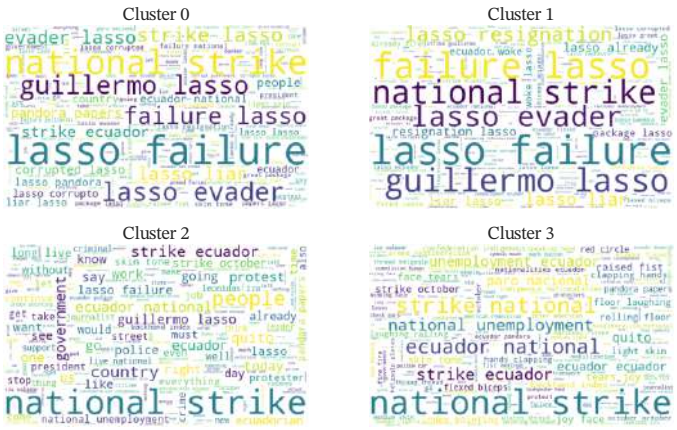


Figure 3: Word cloud for the clusters with google news embeddings

C. Application of the sentiment analysis model

For each cluster, sentiment analysis was conducted. The analysis used the `TextBlob` python library. that use a single perceptron to extract a score of the polarity between $[-1, 1]$ the values with a more negative score are also with a polarity more negative.

As we can see in the Figure 4 the cluster with a higher proportion of negative tweets is the cluster 1 with a 64% of negative tweets and as we can see in the Table I the cluster 1 has also the less polarity score, i.e we can consider a cluster of tweets that may contain cyberterrorism.

Cluster	Negative	Positive
0	-0.25	0.23
1*	**-0.33**	0.24
2	-0.21	0.22
3	-0.23	0.32

Table I: Polarity media of each cluster

D. Application of the NER model

In this case, a NER model was trained using the WikiNer dataset [22] with the Spanish corpus, this dataset contains $141,761$ sentences extracted from Wikipedia and annotated using the BIO format with three types of entities: i) *LOC*

[17]https://pypi.org/project/tweet-preprocessor/
[18]https://pypi.org/project/emoji/
[19]https://pypi.org/project/google-cloud-translate/

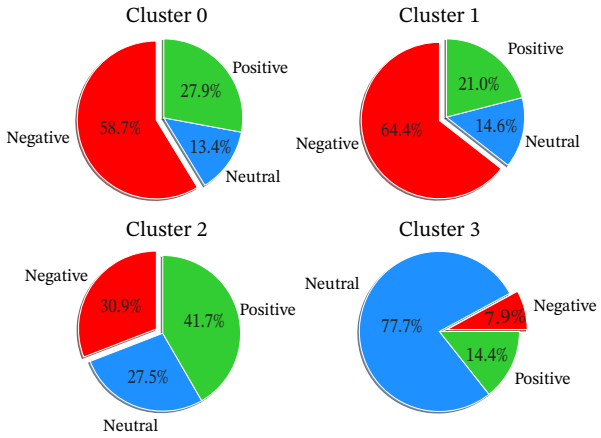


Figure 4: Polarity by cluster with TextBlob

Localization, ii) $MISC$ Miscellaneous iii) ORG Organization and iv) PER Person. The model was trained using the Bi-LSTM architecture see in the section III. The metrics obtained in the train set and the test are shown in the Table II.

In Table III we can see the eight entities with the highest frequency of appearance predicted by the NER model after removing bad predictions as stops words, punctuation symbols and names of emojis. The last name of the president of Ecuador is part of the set of entities with greater frequency in all categories of entities. In the case of entities such as person and organization, this would make sense since depending on the context this word can refer to the president as a human being or as a representation of an organization, in this case, the presidency. It must also be considered that this word was not found in the training set, so the model managed to label it in any case.

	Train Set Quality		
	Precision	***Recall***	***F1-Score***
General Performance	95.37%	95.98%	95.67%
LOC	94.73%	95.76%	95.24%
MISC	92.77%	93.05%	92.91%
ORG	93.88%	93.21%	93.55%
PER	98.11%	98.56%	98.34%
	Test Set Quality		
	Precision	***Recall***	***F1-Score***
General Performance	82.45%	84.93%	83.67%
LOC	82.90%	86.77%	84.79%
MISC	67.77%	68.94%	68.35%
ORG	77.48%	77.33%	77.41%
PER	89.91%	91.50%	90.70%

Table II: Tain and Test Quality

token	**ORG**	**LOC**	**PER**	**MISC**	**Total**	**Percentage**
Paro	3426	1801	196	74	5497	62.3% (ORG)
Lasso	353	3145	2125	264	5887	36.1% (PER)
Nacional	2661	2415	204	97	5377	49.5% (ORG)
Ecuador	724	2535	14	60	3333	76.1% (LOC)
Guillermo	21	36	1299	3	1359	95.6% (PER)
Fuera	115	1067	85	17	1284	1.3% (MISC)
Renuncia	10	593	56	3	662	0.5% (MISC)
Pandora	9	205	558	8	780	0.1% (MISC)

Table III: Principal NER Predictions and percentage of accuracy

On the other hand, An example of a correctly labeled location is the word “Ecuador”. In a similar way as the president’s Lastname. Ecuador also can be considered as a Location entity or Organization entity, this depends on the context in which the word is used. The model labeled the leaked documents known as “pandora papers” as a person most of the time due to the name “pandora”. Finally, we can see in the prediction of “Fuera” that most of the time was predicted as a Location entity. But in the context of protest, this word is used as an expression of rejection instead of a relative position.

Although the model successfully predicts several of the entities that were relevant within the context of the protests, errors also occur that are due to the difference between the training set, Wikipedia articles, and the analyzed tweets that also contain a language more informal in addition to the use of emojis.

E. Graph representation of the user network of contacts

Finally, the data of the Twitter users were collected using the `tinfoleak` tool in order to construct a network of contacts In the case of finding some type of suspicious activity. Intelligence agencies can consult a graph like the one shown in Figure 5. In this case, it is a sample of users who published tweets grouped using google news embeddings in cluster 1 (red nodes), which was the one with the most negative polarity and some of its contacts (blue nodes) from which tweets not were collected.

In this graph, we can see communities of users that follow big red nodes these communities are connected by internal nodes that are common contacts between these big nodes and other red nodes.

V. Conclusions and Future Work

Taking into consideration the influence that social networks have on society and their possible misuse of them to promote cyberterrorism, the application of NLP may be considered a way to automate the analysis of the large volume of data coming from social networks.

In this paper, we proposed FCTNLP, a solution that integrates three NLP models to process information extracted from open sources, i.e. social networks, with the purpose of identifying behaviors associated with HSM and in that way supporting LAWs agencies in the identification and prevention of cybercrimes. FCTNLP was tested through a set of experiments that process tweets related to a real scenario of

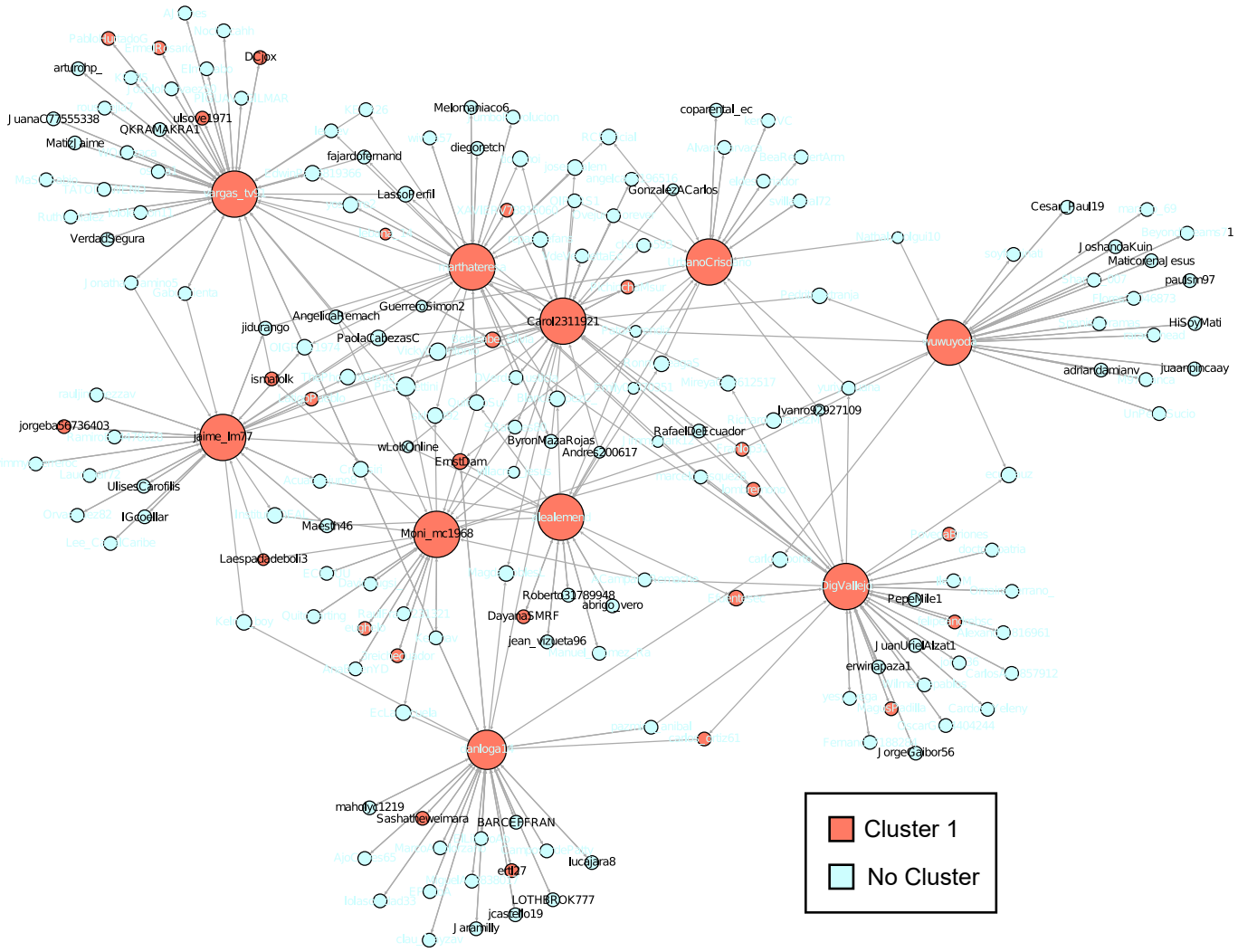


Figure 5: User network sample of user of the cluster 1

protests that occurred on October 26, 2021, in Ecuador. Such experiments demonstrated the feasibility of our proposal for a scenario of cyberterrorism and the usefulness that it may have for LAWs.

As future work, we plan to do a new set of experiments that include additional sources of information, like underground forums and other social networks, which allow enriching the information represented in the graph to discover more useful insights within cyberterrorism research. It may also be interesting to enlarge the windows of time employed for the gathering of data, including previous and subsequent moments of a protest, so it may be clearer to identify how changed the hostility in the content is generated. Finally, FCTNLP may be extended with new NLP models, e.g. one able to predict intentions so a response may also be designed with some automaticity as a way to contain a hostile campaign.

Acknowledgment

This work has been supported by Universidad del Rosario (Bogotá) through the project "IV-TFA043 - Developing Cyber Intelligence Capacities for the Prevention of Crime". This work has also been supported by an FPU contract (FPU18/00304) granted by the Spanish Ministry of Universities.

References

[1] C. of Europe, "Explanatory report to the convention on cybercrime," https://rm.coe.int/CoERMPublicCommonSearchServices/DisplayDCTMContent?documentId=09000016800cce5b, 2001.

[2] J. R. G. Evangelista, R. J. Sassi, M. Romero, and D. Napolitano, "Systematic literature review to investigate the application of open source intelligence (osint) with artificial intelligence," *Journal of Applied Security Research*, pp. 1–25, 2020.

[3] H. J. Williams and I. Blum, "Defining second generation open source intelligence (osint) for the defense enterprise," RAND Corporation Santa Monica United States, Tech. Rep., 2018.

[4] J. Pastor-Galindo, P. Nespoli, F. Gómez Mármol, and G. Martínez Pérez, "The not yet exploited goldmine of osint: Opportunities, open challenges and future trends," *IEEE Access*, vol. 8, pp. 10 282–10 304, 2020.

[5] A. Thomas, *Natural Language Processing with Spark NLP: Learning to Understand Text at Scale*. O'Reilly Media, 2020. [Online]. Available: https://books.google.com.co/books?id=sJw6zQEACAAJ

[6] L. Clark, N. Pantidi, O. Cooney, P. Doyle, D. Garaialde, J. Edwards, B. Spillane, E. Gilmartin, C. Murad, C. Munteanu, V. Wade, and B. R. Cowan, *What Makes a Good Conversation? Challenges in Designing Truly Conversational Agents*. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–12. [Online]. Available: 10.1145.3290605.3300705

[7] S. Kumari, Z. Saquib, and S. Pawar, "Machine learning approach for text classification in cybercrime," pp. 1–6, 2018.

[8] C. Sánchez-Rebollo, C. Puente, R. Palacios, C. Piriz, J. Fuentes, and J. Jarauta, "Detection of jihadism in social networks using big data techniques supported by graphs and fuzzy clustering," *Hindawi*, vol. 2019, no. 1238780, p. 13, 2019.

[9] I. Aljarah, M. Habib, N. Hijazi, H. Faris, R. Qaddoura, B. Hammo, M. Abushariah, and M. Alfawareh, "Intelligent detection of hate speech in arabic social network: A machine learning approach," *Journal of Information Science*, vol. 47, no. 4, pp. 483–501, 2021. [Online]. Available: https://doi.org/10.1177/0165551520917651

[10] I. Castillo-Zúñiga, F. Luna-Rosas, L. Rodríguez-Martínez, J. Muñoz-Arteaga, J. López-Veyna, and M. Rodríguez-Díaz, "Internet data analysis methodology for cyberterrorism vocabulary detection, combining techniques of big data analytics, nlp and semantic web," *International Journal on Semantic Web and Information Systems*, vol. 16, pp. 69–86, 01 2020.

[11] C. Oleji, N. Euphemia, G. Chukwudebe, and O. Chukwueneka Philips, "Big data analitic of boko haram insurgency attacks menace in nigeria using dynamick-reference clustering algorithm," vol. 7, pp. 1099–1107, 04 2020.

[12] V. N. Uzel, E. Saraç Eşsiz, and S. Ayşe Özel, "Using fuzzy sets for detecting cyber terrorism and extremism in the text," in *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2018, pp. 1–4.

[13] S. J. Wagh, M. S. Bhende, and A. D. Thakare, *Fundamentals of Data Science*. Chapman and Hall/CRC, 2021.

[14] S. Bazzaz Abkenar, M. Haghi Kashani, E. Mahdipour, and S. M. Jameii, "Big data analytics meets social media: A systematic review of techniques, open issues, and future directions," *Telematics and Informatics*, vol. 57, p. 101517, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736585320301763

[15] A. Karami, M. Lundy, F. Webb, and Y. K. Dwivedi, "Twitter and research: A systematic literature review through text mining," *IEEE Access*, vol. 8, pp. 67 698–67 717, 2020.

[16] Ankit and N. Saleena, "An ensemble classification system for twitter sentiment analysis," *Procedia Computer Science*, vol. 132, pp. 937–946, 2018, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705091830841X

[17] A. Poornima and K. S. Priya, "A comparative sentiment analysis of sentence embedding using machine learning techniques," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 493–496.

[18] S. Zahoor and R. Rohilla, "Twitter sentiment analysis using lexical or rule based approach: A case study," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020, pp. 537–542.

[19] B. Jang, M. Kim, G. Harerimana, S.-u. Kang, and J. W. Kim, "Bi-lstm model to increase accuracy in text classification: Combining word2vec cnn and attention mechanism," *Applied Sciences*, vol. 10, no. 17, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/17/5841

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[21] J. Choi and S.-W. Lee, "Improving fasttext with inverse document frequency of subwords," *Pattern Recognition Letters*, vol. 133, pp. 165–172, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865520300817

[22] J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran, "Learning multilingual named entity recognition from wikipedia," Oct 2017. [Online]. Available: https://figshare.com/articles/dataset/Learning_multilingual_named_entity_recognition_from_Wikipedia/5462500

Sesión II: Formación e Innovación Educativa

DIANA: Un asesor virtual inteligente para la transformación digital segura

Marta Fuentes-García
COSCYBER
FIDESOL
Granada, España
mfuentes@fidesol.org

Felipe Mirón
FIDESOL
Granada, España
fmiron@fidesol.org

***Resumen*—En los últimos años se han propuesto distintos tipos de ayudas para la transformación digital. Sin embargo, en muchas ocasiones los empresarios no pueden acceder a ellas, en su mayoría por desconocimiento de su existencia o incluso acerca de cómo solicitarlas. Según los propios empresarios, uno de los grandes inconvenientes para avanzar en la digitalización es la falta de formación y herramientas para llevarla a cabo. Este escenario, nos ha motivado a proponer el proyecto DIANA (DIgitalización ANdaluza Asistida).**

El objetivo de DIANA es ayudar a las pymes (pequeñas y medianas empresas) a llevar a cabo una transformación digital segura y eficiente. Para ello, se propone desarrollar un asesor virtual inteligente que facilite la adquisición de competencias digitales, proporcione recomendaciones adaptadas de actuación, y permita llevar a cabo acciones para facilitar la propia digitalización. Los ejes vertebradores del proyecto DIANA son los datos y la ciberseguridad, que se integrarán mediante la combinación de técnicas de aprendizaje automático y de visualización con disciplinas como la interacción persona-ordenador. En este artículo presentamos el proyecto, describimos sus principales características, objetivos y beneficios esperados.

***Index Terms*—transformación digital, digitalización, pymes, ciberseguridad, usabilidad, concienciación**

Tipo de contribución: *Investigación en desarrollo (límite 4 páginas)*

I. Introducción

La transformación digital es uno de los grandes retos de la sociedad para los próximos años. La situación derivada de la pandemia que sufrimos desde 2020 ha puesto de manifiesto más aún la necesidad y el desafío que ello supone a distintas escalas. En este contexto, las pymes (pequeñas y medianas empresas)[1] han tenido que invertir un mayor esfuerzo para acelerar este proceso y no quedarse atrás respecto a las grandes empresas [1]. Otro de los efectos de esta digitalización acelerada es que en los últimos años el número de ciberataques sufridos por las pymes ha incrementado significativamente, y aunque también se ha producido un aumento de la inversión en ciberseguridad, todavía es necesario avanzar en la concienciación y capacitación (tanto formativa como tecnológica) de las pymes en materia de ciberseguridad [2, 3].

Este escenario nos ha motivado a proponer el proyecto **DIANA** (DIgitalización ANdaluza Asistida), cuya piedra angular son los **datos** y la **ciberseguridad**. DIANA es un proyecto que pretende ir más allá de las actuales iniciativas para ayudar en la transformación digital de las pymes. El carácter innovador de este proyecto reside en varios aspectos:

1. Ayuda a la transformación digital en base a **cuatro pilares** fundamentales: *evaluación (P1)*, *formación (P2)*, *recomendación (P3)*, y *actuación (P4)*.
2. Evolución del concepto actual de asistente virtual inteligente hacia un ***asesor* virtual inteligente**[2] **no intrusivo** para combinar los pilares de ayuda a la digitalización.
3. Aplicación de técnicas de Inteligencia Artificial (**IA**), aprendizaje automático (**ML**, del inglés, *Machine Learning*), y **análisis y ciencia de datos** para dotar de valor y calidad todas las fases del proyecto.
4. Consideración de la **ciberseguridad como factor transversal y esencial** en la digitalización. Se integrará en todos los pilares de ayuda, con especial atención a la capacitación y la concienciación de la necesidad e importancia de la misma.
5. **Diseño centrado en el usuario**, considerando las necesidades reales de las pymes y autónomos, y utilizando técnicas de interacción persona-ordenador (HCI, del inglés, *Human-Computer Interaction*) para proporcionar una solución usable, cercana y sencilla.

En este artículo introducimos el proyecto DIANA, que se presentó en la convocatoria de 2021 de la línea de Ayudas para la realización de proyectos de I+D+i por Agentes del Sistema Andaluz del Conocimiento[3]. El resto de secciones se organiza como sigue: la Sección II muestra el trabajo relacionado con el proyecto, incluyendo otras iniciativas existentes, así como una breve introducción a los asistentes virtuales. La Sección III describe el propósito, objetivos y resultados esperados del proyecto. Finalmente, la Sección IV sintetiza la propuesta y muestra las principales conclusiones de la misma.

II. Trabajo relacionado

La Industria 4.0 supone una de las líneas de investigación e innovación más actuales. Sin embargo, existe un gran número de empresas que aún se encuentran en proceso de digitalización (Tercera Revolución Industrial) [4, 5]. Según el informe de digitalización de las pymes de 2021[4], una de las principales diferencias en cuanto al estado de digitalización se deriva del tamaño de la empresa.

Desde el punto de vista científico, existen numerosos trabajos que abordan la digitalización. Por ejemplo, en 2021 se presentó una tesis doctoral sobre "*Retos para la Transformación*

[1]**NOTA.** En este artículo nos referimos de forma general a las pymes, englobando con esto tanto a pymes como autónomos.

[2]En ocasiones, para facilitar la lectura, nos referiremos al asesor virtual inteligente simplemente como asesor o asesor virtual

[3]https://www.juntadeandalucia.es/boja/2021/239/7

[4]https://bit.ly/37kEaUh [Online, accedido el 02/02/2022]

Digital de las PYMES: Competencia Organizacional para la Transformación Digital" [5]. En varios artículos y libros se revisa la literatura para extraer las necesidades y retos de digitalización en las pymes [4, 6, 7]. También destaca la necesidad de adaptación y aplicación de disciplinas relacionadas con la interacción persona-ordenador para mejorar la experiencia de usuario [4]. Por otra parte, también se han llevado a cabo estudios sobre la ciberseguridad y la calidad de las pymes [8, 9]. En general, estos trabajos **destacan el reto que supone la transformación digital para las pymes**, así como la necesidad de formación, conocimiento y competencias para poder llevarla a cabo de forma satisfactoria [4–7, 10].

II-A. Iniciativas existentes

Existen distintas iniciativas y programas para ayudar a llevar a cabo la transformación digital. Se proponen tanto a nivel autonómico (p.e. *Plan de Acción de Empresa Digital de la Junta de Andalucía (PAED)*) como nacional (p.e. *España Digital 2025*[5] o *Kit Digital*[6]) y europeo (p.e. *Digital Europe Programme (DIGITAL)*[7]). Precisamente, uno de los factores condicionantes de la determinación del alcance en el presente proyecto ha sido el análisis de las iniciativas de apoyo desde el PAED de la Junta de Andalucía. Este plan pretende dar soporte al impulso de la Economía Digital en sectores estratégicos y fomentar la transformación digital en las pymes.

Por otra parte, INCIBE (Instituto Nacional de Ciberseguridad) creó el programa *Protege tu empresa* para ayudar a las pymes en materia de ciberseguridad. Dentro de este programa se ofrece información, talleres, y guías para concienciar a las empresas (especialmente pymes y autónomos) sobre la importancia de la ciberseguridad. También se proporcionan algunas herramientas (p.e. análisis de riesgos) y un catálogo de soluciones. Esta iniciativa es muy completa en cuanto a contenido y recursos.

Tanto el Observatorio Andalucía Conectada como INCIBE son ejemplos de cómo organizaciones públicas proponen distintas alternativas para ayudar a la digitalización. Ofrecen un muy buen punto de partida y con margen de maniobra suficiente para poder perfeccionar el modelo y hacerlo aún más cercano y adaptado a los empresarios. En este sentido, es importante tener en cuenta que no todos los usuarios tienen las mismas competencias digitales (brecha digital) y que, la mayoría dispone de tiempo limitado. **DIANA pretende complementar** las iniciativas existentes, **aprovechando el potencial de los datos** que gestiona la empresa. El objetivo es mostrar a las pymes el valioso conocimiento que se puede extraer de sus datos, así como **integrar y unificar** las herramientas y el conocimiento que pueden facilitar esta labor.

II-B. Tecnología previa

Un asistente virtual consiste en la aplicación de IA para procesar la voz, analizar solicitudes y proporcionar respuesta a búsquedas o comunicación. Se puede considerar que existen tres tipos de asistentes virtuales: *VEA* (del inglés, *Virtual Employee Assistant*), *VPA* (del inglés, *Virtual Personal Assistant*), y *VCA* (del inglés, *Virtual Customer Assistant*) [10]. Los VPA son los más utilizados y aceptados por la sociedad, aunque las empresas cada vez utilizan más los VCA para ayudar, asistir y asesorar a los clientes tanto durante la contratación de servicios como en la atención pos-venta de los mismos. Además, el uso de los VEA se está extendiendo para ayudar a los empleados a mejorar su rendimiento y como apoyo a la toma de decisiones [11].

Uno de los tipos de asistentes virtuales más conocidos son los *chatbots*. Suelen ser asistentes que utilizan IA, basados en texto en lugar de voz. Los *chatbots* son programas que permiten simular una conversación con seres humanos y que, con frecuencia, se utilizan para mejorar la experiencia de los clientes, proporcionándoles asistencia en su interacción con los servicios proporcionados [10, 11]. En la actualidad tienen un papel fundamental en la digitalización en dos vertientes: *i)* atención al cliente, y *ii)* ayuda al trabajador. Los primeros se enmarcan dentro de los VCA, ya que ayudan a dotar de personalidad los servicios con los que el usuario interactúa, haciéndolos más cercanos y reduciendo los costes que supondría tener más agentes humanos proporcionando dichos servicios de forma ininterrumpida [11]. Los segundos se podrían considerar VEA, pues están centrados en automatizar tareas repetitivas y en ayudar a añadir valor al negocio [12, 13]. **En DIANA proponemos** una evolución del concepto de asistente virtual inteligente y de *chatbot*: combinaremos características de VEA y VPA para obtener un **asesor virtual inteligente** que ayude a las pymes en su proceso de digitalización. El asesor estará **siempre presente** y podrá ser consultado cuando el usuario lo desee. También trabajará de manera proactiva, llevando a cabo **sugerencias y recomendaciones**, tanto de formación como de actuación. El asesor **no será intrusivo**, permitiendo incrementar el nivel de madurez digital de manera progresiva, adaptativa, e intuitiva. Para ello, se aplicará **diseño emocional**, centrado en mejorar la experiencia de usuario mediante la conexión con las emociones, favoreciendo la empatía, y basado en las necesidades particulares de cada usuario.

III. Proyecto DIANA

El objetivo del proyecto es **facilitar el proceso de transformación digital a las pymes**, abordándolo desde distintas perspectivas, a nivel tanto básico como específico. DIANA propone un enfoque innovador y multidisciplinar compuesto por cuatro pilares, que se enumeran a continuación:

- **[P1] Evaluación y diagnóstico interpretable** del estado de digitalización, desde el punto de vista de la pyme y del usuario.
- **[P2] Capacitación**. Ayuda a la adquisición de competencias digitales básicas y específicas que faciliten el uso y la adopción de la tecnología disponible.
- **[P3] Recomendación y asesoramiento adaptativo** de líneas de acción para la digitalización, basada en técnicas de IA y ML.
- **[P4] Actuación en líneas de mejora concretas**, tanto guiada como autónoma, basada en técnicas de IA y ML.

La ciberseguridad tiene un papel central en DIANA, afrontándose como factor transversal y esencial en la digitalización. Se integrará en todos los pilares de ayuda, con especial atención a la capacitación y concienciación sobre la

[5]https://bit.ly/3s20mJG y https://bit.ly/3rdvbf0
[6]https://www.red.es/es [online, accedido el 02/02/2022]
[7]https://bit.ly/3jdl8lh

necesidad e importancia de la misma. Entre otras, se facilitará la adquisición de competencias sobre prevención, detección y respuesta frente a incidentes. Así mismo, se dotará a las pymes de herramientas asequibles de última generación (p.e. basadas en ML e IA) y del conocimiento necesario para utilizarlas y proteger su negocio.

En la Figura 1 mostramos cómo los datos obtenidos de distintas fuentes (p.e. usuarios, empresas, organismos públicos y recursos disponibles) se explotarán para: *i)* recopilar requisitos y necesidades de los usuarios, así como de organismos públicos, *ii)* facilitar formación, información y contenido adaptado a los usuarios, y *iii)* proporcionar un sistema de asesoramiento, recomendación, predicción y actuación adaptado a las necesidades de las pymes, relacionadas con la transformación digital y la ciberseguridad. Estos datos se obtendrán tanto de forma pasiva (p.e. de dispositivos y bases de datos existentes) como activa (solicitados a usuarios y entidades de forma explícita). Todo lo anterior se conseguirá gracias a la aplicación combinada de IA, ML y HCI, ofreciendo además una experiencia inmersiva y adaptada al usuario. Esto se logrará gracias a la propuesta del concepto de **asesor virtual inteligente**, que consiste en la evolución de la idea actual de asistente virtual inteligente hacia un asesor no intrusivo, mediante: *i)* un cambio de perspectiva respecto a los asistentes virtuales tradicionales, **fusionando asistente personal y asistente virtual** para ayudar al usuario de forma simplificada, personalizada y pudiendo llevar a cabo acciones tanto sencillas como complejas; *ii)* una **integración centralizada de recursos y herramientas**; y *iii)* un **diseño centrado en el usuario**. Este asesor combinará los pilares de ayuda a la digitalización propuestos.

En la Tabla I se resumen los factores diferenciales de DIANA respecto al trabajo relacionado. Destacan la personalización y facilidad de uso gracias al diseño centrado en el usuario, así como la posibilidad de integrar distintos recursos y funcionalidades en un único asesor virtual, facilitando la ruptura de la brecha digital.

Tabla I
FACTORES DIFERENCIALES DE DIANA RESPECTO A OTRAS INICIATIVAS Y TECNOLOGÍA PREVIAS.

Factor	DIANA	Iniciativas previas	Tecnología (asistentes)
Contenido (P2,P3)	✓	✓	?
Ciberseguridad (P1-P4)	✓	✓	?
Adaptabilidad/Personalización (P1-P4)	✓	X	?
Integración (P1-P2)	✓	X	X
Facilidad de uso (P1-P4)	✓	X	✓
Diseño centrado en el usuario (P1-P4)	✓	X	X
Facilidad de actuación (P1, P3, P4)	✓	X	?

III-A. Retos relacionados

El proyecto DIANA contribuirá a resolver, entre otros, varios retos sociales definidos en el tercer objetivo general del Plan Andaluz de Investigación, Desarrollo e Innovación (PAIDI 2020):

- **Economía y sociedad digital y Sociedades inclusivas, innovadoras y reflexivas**. Ayudará a **reducir la brecha digital** existente en la sociedad a través de su contribución a la transformación digital de las pymes.
- Contribuirá a la **Acción por el clima, medioambiente, eficiencia de recursos y materias primas**, pues la digitalización conlleva en última instancia la optimización de recursos y materias primas, así como del uso de energía y transporte.

DIANA plantea su alcance en los sectores más estratégicos para la recuperación económica (como son el sector primario y el sector servicios), teniendo en cuenta también las acciones estratégicas definidas en el PEICTI[8] (Plan Estatal de Investigación Científica, Técnica y de Innovación). Las principales acciones en las que se centra son: **AE3 - seguridad para la sociedad** y **AE4 - mundo digital, industria, espacio y defensa**. De esta forma, se pone especial énfasis en la transformación digital de los sectores productivos y las cadenas de valor.

III-B. Beneficios esperados

DIANA ayudará a mejorar la sociedad mediante la **capacitación** en distintas competencias digitales, entre las que destacan las relacionadas con la **ciberseguridad**; **apoyo continuo a la transformación digital**, tanto de forma activa como pasiva; **empoderamiento**, gracias a la independencia y autonomía en la gestión empresarial de los propietarios de las pymes; **resiliencia** frente a situaciones adversas, como la crisis derivada de la pandemia; **rendimiento y fortalecimiento** del tejido empresarial, gracias al mejor aprovechamiento de los recursos y ayudas proporcionadas por distintos organismos; **mejora de la economía**; y **mejora de la sostenibilidad**.

En cuanto al potencial impacto científico y tecnológico, se prevén los siguientes hitos: **combinación de técnicas y disciplinas** (IA, ML, NLP, HCI, y ciberseguridad) de forma novedosa para mejorar la experiencia de usuario gracias al valor de los datos; y definición de un **nuevo concepto de asistente virtual inteligente: *asesor* virtual inteligente**, cuyo uso se propone para ayudar en la digitalización de las pymes, pero podría extenderse a otros verticales, como la Industria 4.0 y la formación en múltiples ámbitos y contextos.

IV. CONCLUSIONES

Las pymes y autónomos son el motor de Andalucía, España y Europa. Sin embargo, un elevado porcentaje de estas empresas aún se encuentra en una fase temprana de digitalización. Por otra parte, el número de ataques a pymes se ha visto incrementado en los dos últimos años. Todo ello, nos ha motivado a lanzar el proyecto DIANA, cuyo objetivo es ayudar a las pymes a llevar a cabo una transformación digital segura y eficiente. Con DIANA pretendemos obtener una visión más cercana a la realidad de las necesidades de las pymes y empoderarlas para mejorar su nivel de madurez digital. La espina dorsal de DIANA son los **datos** y la **ciberseguridad**, que se integrarán combinando técnicas de aprendizaje automático y de visualización con disciplinas como la interacción persona-ordenador.

Para materializar este proyecto se propone la evolución del concepto de asistente virtual inteligente hacia un **asesor virtual inteligente** que combine los pilares de ayuda a la digitalización propuestos: **evaluación y diagnóstico interpretable**,

[8]https://bit.ly/37YxuLQ

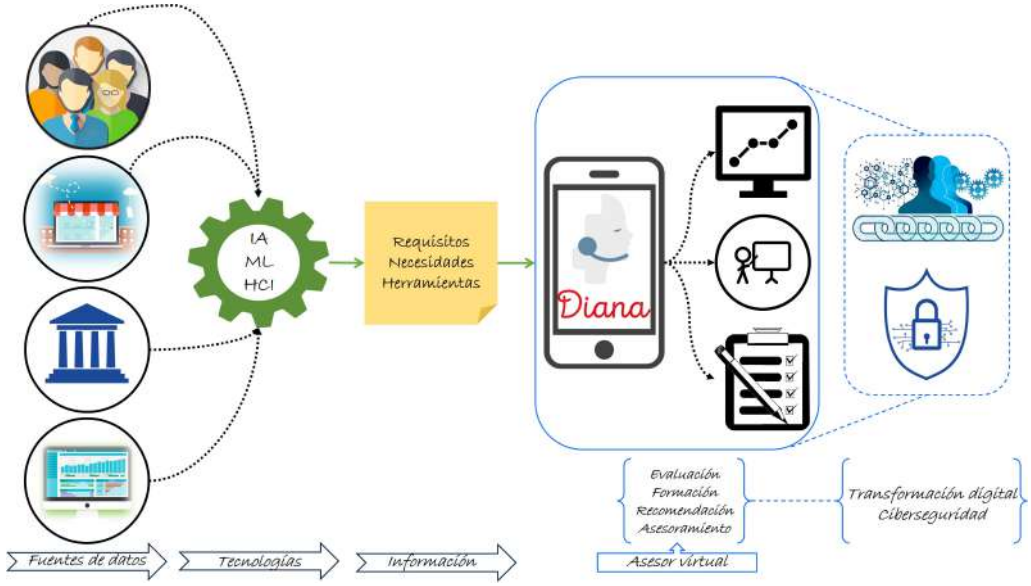


Figura 1. Funcionamiento de DIANA. En negro se representa el proceso de explotación de los datos, y en azul el resultado del mismo (asesor virtual).

capacitación, **recomendación y asesoramiento adaptativo**, y **actuación**. Este asesor será no intrusivo, brindando ayuda personalizada a los usuarios, y pudiendo llevar a cabo acciones tanto sencillas como complejas. Además, integrará recursos y herramientas, facilitando así el acceso a los mismos. Todo ello, siguiendo un diseño emocional y centrado en el usuario, simplificando así el uso y adopción del asesor.

La explotación de DIANA está prevista en forma de transferencia directa de los resultados del proyecto. Una vez finalizado, se analizará la posibilidad de establecer sinergias con las iniciativas de transformación digital que se encuentren activas en los distintos organismos oficiales. Se presentará un prototipo que seguirá evolucionando en función de lo identificado en los requisitos obtenidos y el diseño llevado a cabo, gracias al aprendizaje y retroalimentación de la interacción del asesor con los usuarios. El objetivo final es que se implante y se adopte masivamente para tener un impacto real, significativo y positivo en la digitalización de las pymes.

Agradecimientos

Este trabajo está financiado en parte por las Ayudas Cervera para Centros Tecnológicos del Centro Español para el Desarrollo de Tecnología Industrial (CDTI) en el marco del proyecto EGIDA (CER-20191012).

Referencias

[1] Cinco Días, "Las pymes españolas mejoran su nivel de digitalización," El País, Tech. Rep., [Online, accedido el 02/02/2022]. [Online]. Available: https://bit.ly/3gm5yCF

[2] ——, "Los ciberataques en España crecen un 125perjudicada," El País, Tech. Rep., [Online, accedido el 07/04/2022]. [Online]. Available: https://bit.ly/3DW8LDR

[3] A. R. Aguiar, "España puede convertirse en una superpotencia en ciberseguridad con una fórmula sencilla: más soberanía tecnológica, una visión estratégica y una apuesta decidida por el talento," Business Insider, Tech. Rep., [Online, accessed on 08/02/2021]. [Online]. Available: tinyurl.com/149u8kj1

[4] D. T. Matt, V. Modrák, and H. Zsifkovits, *Industry 4.0 for SMEs*, D. T. Matt, V. Modrák, and H. Zsifkovits, Eds. Springer International Publishing.

[5] J. M. González Varona, "Retos para la Transformación Digital de las PYMES: Competencia Organizacional para la Transformación Digital," phdthesis. [Online]. Available: https://uvadoc.uva.es/handle/10324/47767

[6] A. I. Canhoto, S. Quinton, R. Pera, S. Molinillo, and L. Simkin, "Digital strategy aligning in SMEs: A dynamic capabilities perspective," vol. 30, no. 3, p. 101682.

[7] C. Sassanelli, S. Terzi, H. Panetto, and G. Doumeingts, "Digital Innovation Hubs supporting SMEs digital transformation," in *2021 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. IEEE.

[8] A. Emer, M. Unterhofer, and E. Rauch, "A Cybersecurity Assessment Model for Small and Medium-Sized Enterprises," vol. 49, no. 2, pp. 98–109.

[9] D. Gaitero, M. Genero, and M. Piattini, "System quality and security certification in seven weeks: A multi-case study in spanish SMEs," vol. 178, p. 110960.

[10] G. Omale, "Gartner Predicts 25 Percent of Digital Workers Will Use Virtual Employee Assistants Daily by 2021," Gartner, Tech. Rep., [Online, accessed on 12/11/2020]. [Online]. Available: https://tinyurl.com/yy87r28g

[11] Bee Digital, "Diferencias entre los asistentes virtuales y un chatbot," Tech. Rep., [Online, 27/11/2020]. [Online]. Available: https://www.beedigital.es/tendencias-digitales/diferencias-entre-los-asistentes-virtuales-y-un-chatbot/

[12] J. A. Hernández, "Llega AVI, la Asistente Virtual Inteligente para las empresas," Movistar Empresas, Tech. Rep., [online, accedido el 03/02/2022]. [Online]. Available: https://bit.ly/34sVW6v

[13] Tracking Time, "Asistentes virtuales: los bots conquistan a startups y PyMEs," Tech. Rep., [online, accedido el 03/02/2022]. [Online]. Available: https://bit.ly/34t7R3W

Capacidades avanzadas de simulación y evaluación en Cyber Ranges con elementos de gamificación

Pantaleone Nespoli[1], José Antonio Pastor Valera[1], Mariano Albaladejo-González[1], José A. Ruipérez-Valiente[1], Félix Gómez Mármol[1]
[1]Departamento de Ingeniería de la Información y las Comunicaciones, Universidad de Murcia, 30100, Murcia, España
{pantaleone.nespoli, japv, mariano.albaladejog, jruiperez, felixgm}@um.es

***Resumen*—Sin duda alguna, estamos asistiendo a una revolución digital sin antecedentes, siendo la tecnología parte de nuestra vida cotidiana. Sin embargo, los últimos años han sido caracterizados por la explosión de ciberataques perpetrados por cibercriminales expertos y motivados. Por eso, la formación en ciberseguridad y ciberdefensa es cada día más esencial para proteger los activos del ciberespacio. En este contexto, es fácil entender como los Cyber Ranges representan una herramienta vital para ofrecer dicha formación a un público amplio. En este artículo se presenta una propuesta de Cyber Range totalmente funcional y virtualizada. Gracias a su arquitectura modular, nuestra propuesta es capaz de proporcionar escenarios realistas diferentes en cada iteración. Además, el Cyber Range posee varios componentes para mejorar la formación de los usuarios desde una perspectiva más educativa. En concreto, el módulo de gamificación que genera elementos para mejorar el compromiso de los usuarios, un sistema adaptativo para adaptar el ciberejercicio en función de su desempeño, y un sistema de recolección y procesado de datos que produce numerosas medidas de desempeño y da soporte al procesado de señales biométricas para la detección del estrés.**

***Index Terms*—Cyber Range, ciberseguridad, ciberdefensa, gamificación, formación práctica.**

Tipo de contribución: *Investigación científica consolidada*

I. Introducción

Las Tecnologías de la Información y la Comunicación (TIC) están teniendo un impacto muy relevante en nuestro día a día, mejorando de forma sustancial nuestra calidad de vida, a través del acceso a multitud de servicios on-line y al uso de productos que utilizan estas tecnologías. Además, el auge de nuevas tecnologías disruptivas (por ejemplo, Blockchain [1]) y de los paradigmas innovadores (por ejemplo, Internet of Things (IoT) [2]) han allanado el camino a importantes contribuciones tanto del mundo industrial como académico.

Sin embargo, la revolución digital tiene también un impacto negativo, dado que en este caso dichas tecnologías también se están convirtiendo en uno de los principales objetivos de actores con fines perniciosos que las atacan para causar daños operativos y/o sacar beneficios económicos de las empresas o los usuarios que hacen uso de las mismas. De hecho, el número de ataques cibernéticos a las TIC ha seguido creciendo de forma exponencial durante los últimos años, convirtiendo el cibercrimen en uno de los negocios más rentable para ellos. Por lo tanto, tanto el mundo académico como los sectores público y privado están trabajando activamente para construir planes de seguridad robustos y eficientes para enfrentarse a ciberamenazas. Esta demanda está intentando ser satisfecha también mediante programas educacionales específicos en materias de ciberseguridad, así como empresas dedicadas a la formación en estos aspectos.

Además, con la explosión de la guerra entre Rusia y Ucrania, se ha remarcado cómo los limites entre espacio físico y ciberespacio son cada vez menos tangibles. Efectivamente, el ciberespacio representa ya otro campo de maniobras donde las fuerzas militares se enfrentan para determinar su propia superioridad y obtener inteligencia que provea de beneficios. Es más, es posible encontrar siempre más iniciativas de las organizaciones militares para dotar sus efectivos con capacidades efectivas en materia de ciberdefensa para proteger dicho ciberespacio.

En este escenario tan alarmante, es fácil entender la importancia de dotar a los profesionales de todos los ámbitos con las capacidades de ciberseguridad y ciberdefensa necesarias para defender los activos pertenecientes al ciberespacio frente a potenciales ciberamenazas. Por eso, los Cyber Ranges representan una de las herramientas más vitales hoy en día para ofrecer una formación de calidad de profesionales con altas capacidades específicas y aplicadas en ciberseguridad [3]. Debido a lo anteriormente mencionado, en los últimos años los perfiles de potenciales usuarios se están diversificando, lo cual está ampliando sustancialmente el abanico de entornos que están apareciendo y la forma de utilizarlos. Esto hace que los Cyber Ranges están siendo utilizados por distintos colectivos como estudiantes, militares, profesionales de la seguridad TIC, empleados gubernamentales, profesionales de Pequeñas y Medianas Empresas (PYMEs), entre otros, haciendo uso de simulaciones realistas de escenarios propios de ciberseguridad y ciberdefensa, pudiendo así aprender a tomar decisiones claves y defender sistemas reales.

Sin embargo, el entrenamiento en dichas temáticas presenta aún varios desafíos. Por ejemplo, los Cyber Ranges actuales presentan dificultad a la hora de proporcionar escenarios realistas para efectuar entrenamiento o la imposibilidad de proporcionar servicios bajo demanda utilizando infraestructuras virtuales con características fundamentales como reuso, adaptabilidad y escalabilidad. Desde una perspectiva más educacional, una de las mayores complicaciones de los actuales Cyber Ranges es la pérdida de motivación y sensación de aburrimiento por parte de los usuarios del sistema [4]. Esta problemática puede verificarse por diversos motivos, como

por ejemplo el bajo interés por el contenido impartido, la sensación de apatía ante la materia si resulta muy fácil o, por el contrario, la sensación de incapacidad de entender contenidos o completar ejercicios debido a su alta dificultad, generando frustración.

Con el objetivo de solucionar los retos anteriormente mencionados en el marco del proyecto COBRA [5], este artículo presenta un Cyber Range totalmente funcional y virtualizado para entrenamiento en capacidades de ciberseguridad y ciberdefensa. A partir de una serie de parámetros, nuestra propuesta es capaz de generar una serie de escenarios aleatorios para efectuar cibermaniobras motivadoras y realistas. En concreto, el framework crea y maneja una serie de escenarios dinámicos y parametrizables con variables aleatorias, obteniendo una mayor flexibilidad a la hora de proporcionar ciberejercicios a los usuarios del sistema [6]. Además, las cibermaniobras ejecutadas en dichos escenarios son dinámicas y adaptativas al estudiante, en contraposición a las estáticas ofrecidas por los Cyber Ranges actuales, moviendo la educación en una dirección personalizada a las necesidades del usuario. Asimismo, la plataforma introduce elementos de gamificación en el sistema para generar una realimentación positiva en la motivación de los estudiantes [7].

Además, otra característica diferenciadora de nuestra propuesta es el uso de las trazas de datos telemétricas y las señales biométricas generadas por los usuarios del sistema mientras resuelven los retos presentes en las cibermaniobras. De esta manera, es posible evaluar las competencias clave en entornos duales (i.e., civil/militar), no solo relacionadas con contenidos en ciberdefensa y ciberseguridad, sino también las habilidades transversales como capacidad de trabajo bajo presión.

El resto de este artículo es el siguiente: la Sección II presenta una revisión de los principales trabajos relacionados con Cyber Range. Después, la Sección III describe la arquitectura de nuestra propuesta, detallando sus componentes. En la Sección IV se exponen una serie de casos de uso relacionados con el sistema propuesto. Luego, la Sección V proporciona una profunda discusión sobre la necesidad de utilizar el framework y su posible impacto social. Finalmente, la Sección VI concluye el artículo, proponiendo interesantes vías futuras.

II. Estado del arte

Recientemente, la demanda de expertos en ciberseguridad y ciberdefensa ha subido notablemente tanto en el sector público como en el privado, y se espera que la tendencia siga creciendo durante los próximos años. Consecuentemente, varias propuestas de Cyber Ranges han aparecido como respuesta a dicha demanda, intentando abarcar diferentes necesidades. Los trabajos en [8], [3], [9] revisan el estado del arte con enfoque a los Cyber Ranges, analizando varias características y diferentes perspectivas, incluyendo ventajas, potenciales inconvenientes y posibles vías futuras.

A nivel nacional, el ecosistema de Cyber Ranges y entrenamientos en ciberseguridad y ciberdefensa a través de estas infraestructuras están todavía en fases muy primarias. Eso es, existen pocos frameworks funcionales y que, al mismo tiempo, estén siendo utilizados. Un ejemplo es el *Indra Cyber Range* (ICR)[1], que se ha convertido en una referencia para el entrenamiento y formación en ciberseguridad. Otra solución de formación en ciberseguridad es el *Cyber Range del Centro Vasco de Ciberseguridad*[2], ubicado en el Parque Tecnológico de Álava.

Por otro lado, a nivel internacional, las plataformas Cyber Range son bastante más difusas, tanto en el sector privado como en el público. Por ejemplo, la Universidad de Masaryk (República Checa) lleva desde 2013 desarrollando la plataforma KYPO Cyber Range [10]. Dicha plataforma se basa en varios años de experiencia en el uso de ciberespacios en la educación, la formación y los ejercicios de ciberdefensa, incluidos los ejercicios técnicos checos de ciberseguridad, los Cyber Czech, que se organizaron en colaboración con la Agencia Nacional Checa de Ciberseguridad y Seguridad de la Información (NCISA). La plataforma ya se ha utilizado para la enseñanza de estudiantes en varios cursos de la Universidad de Masaryk y para la formación de profesionales de la ciberseguridad del sector energético. Una característica muy relevante de KYPO es que se trata de un Cyber Range de software abierto y gratuito, lo que ha permitido a otras instituciones de poder desplegar esa solución en sus propias instalaciones [11].

Con el objetivo de "democratizar el entrenamiento en ciberseguridad," los autores en [12] presentan *CyTrONE*, un sistema de formación en ciberseguridad que facilita las actividades de formación ofreciendo un ecosistema de código abierto[3] capaz de automatizar las tareas de generación y configuración de contenidos. Los autores afirman que CyTrONE tiene tres ventajas principales: i) mejorar la precisión de la configuración de la formación, ii) disminuir el tiempo y el coste global de la configuración, y iii) hacer que la formación sea instanciada para muchos usuarios. Además, el sistema viene evaluado desde dos perspectivas fundamentales (funcionalidades y de rendimiento), demostrando sus capacidades para un entrenamiento efectivo.

Asimismo, el Cyber Range *Alpaca* es propuesto en [13]. En particular, este framework de código abierto[4] pretende construir entrenamientos de acuerdo con las restricciones especificadas por el usuario. A diferencia de otros Cyber Range, Alpaca se apoya en una base de datos de vulnerabilidad y un motor de planificación para simular secuencias de exploits que permiten a un atacante alcanzar un objetivo específico. Todos los caminos que cumplen con las restricciones indicadas por el usuario (por ejemplo, el número mínimo de pasos o el uso de una vulnerabilidad concreta), son encontrados por Alpaca y recogidos en una red de vulnerabilidad. Dicha red muestra todas las formas posibles de explotar el sistema. Una vez encontrado el retículo, se construye un Cyber Range a través de secuencias de comandos automáticas para instanciar

[1] https://cyberrange.indracompany.com/
[2] https://www.basquecybersecurity.eus/es/cyberrange/
[3] https://github.com/crond-jaist/cytrone
[4] https://github.com/StetsonMathCS/alpaca

una máquina virtual que contenga todas las vulnerabilidades que componen el retículo. Luego, los autores presentan unos casos de uso iniciales así como una primera evaluación de la herramienta, especificando muchos trabajos futuros con la sensación que Alpaca está aún en plena fase de desarrollo.

Los proyectos mencionados anteriormente representan, sin duda, un avance importante hacía la posibilidad de ofrecer un entrenamiento real en temas de ciberseguridad y ciberdefensa para diferentes categorías de usuarios finales. Sin embargo, algunos retos siguen sin tener una respuesta concreta, obstaculizando una serie de potenciales desarrollos. Con el objetivo de solucionar dichos retos, nuestra propuesta pretende innovar con respecto a soluciones previas de Cyber Ranges sobre la generación de escenarios de entrenamiento. En particular, los escenarios generados en nuestro framework son parametrizables usando variables aleatorias, lo cual permite una mayor flexibilidad y, consecuentemente, unas repercusiones positivas por parte de los usuarios, tanto los instructores como los estudiantes.

Adicionalmente, los ejercicios (o cibermaniobras) en el Cyber Range han sido tradicionalmente estáticos, con una serie de parámetros y/o eventos codificados dentro de líneas específicas temporales, sin posibilidad alguna de adaptarse realmente a las capacidades del estudiante. Esta limitación conlleva implicaciones negativas, es decir, puede generar aburrimiento o frustración por parte del estudiante si el ciberejercicio planteado es demasiado fácil o difícil respectivamente. En este aspecto, nuestra solución mejora el estado del arte en el uso de las trazas telemétricas generadas por los estudiantes, así como señales biométricas capturadas por pulseras inteligentes para implementar un sistema adaptativo que pueda adaptar los recursos disponibles para los estudiantes en los entrenamientos, en función del desempeño del estudiante en los retos previos completados.

III. Arquitectura

La arquitectura propuesta destaca sobre el resto de Cyber Range citados anteriormente por la incorporación de un generador de escenarios aleatorios y parametrizables, un sistema de gamificación, un sistema adaptativo de los ciberejercicios y un sistema para el procesamiento y monitorización de medidas de desempeño que incluye un módulo de detección del estrés como puede visualizarse en la Figura 1.

III-A. Front-end, componentes software e interconexiones

Para interactuar con el framework propuesto, los usuarios se conectan al front-end. Dicha aplicación está compuesta por diversos componentes o microservicios, cada uno de ellos en un contenedor *Docker* donde se ejecutan de forma independiente del resto como se muestra en la figura 2. Dichos componentes son:

- **Nginx**: actúa como proxy inverso para redirigir las peticiones al *endpoint* correspondiente, permitiéndonos así aprovechar las características de seguridad y rendimiento del servicio. Las peticiones del usuario final son interceptadas por este componente y, en función del destino (*/* o */ws*), son enviadas a *Gunicorn* o *Daphne*, respectivamente. Además, también se encarga de servir el contenido estático (como las plantillas *HTML* o los scripts *JS*) y multimedia (imágenes de los dispositivos, avatares de los usuarios, etc.).
- **Gunicorn**: se encarga de servir las peticiones "tradicionales" de los usuarios, esto es, aquellas peticiones que hacen uso del protocolo HTTP o HTTPS.
- **Daphne**: se encarga de servir las peticiones WS/WSS dirigidas a los WebSockets, ya que estas siempre comienzan por el prefijo "/ws".
- **PostgreSQL**: se encarga de almacenar toda la información necesaria para el funcionamiento de la aplicación, como usuarios, definiciones de escenarios, retos y ciberejercicios, servicios, plantillas, entre otros.
- **Redis**: junto a *Django Channels* y *WebSockets* se encargan de ofrecer a la aplicación la posibilidad de obtener actualizaciones en tiempo real y de forma asíncrona sin necesidad de que el usuario final tenga que refrescar la página del navegador.
- **Celery**: se encarga del procesamiento y ejecución de tareas asíncronas en segundo plano, tales como el despliegue de escenarios o la liberación de los recursos asociados a un escenarios cuando éste es eliminado.

III-B. Funcionamiento del generador de escenarios aleatorios y parametrizables

La generación de escenarios de red se lleva a cabo gracias a un sistema adicional de la plataforma, el cual posee sus propios modelos internos e independientes del resto, con el objetivo de ser fácilmente extensible y mantenible. Este sistema toma como entrada dos parámetros: el *entorno* y la *complejidad* del escenario que se desea generar. A partir de estos se recupera la regla de generación correspondiente y, siguiendo la lógica interna del sistema, se origina como salida un *JSON* con la información del escenario concreto generado.

La lógica interna del generador consta de dos bucles: uno para entidades de red y otro para las entidades finales. El primero de ellos sirve para colocar las entidades de red en la topología del escenario de forma aleatoria. En particular, lo primero que se hace es crear una nueva entidad de red y, posteriormente, comprobar si ya se había generado alguna otra antes. En caso afirmativo se coge, de forma aleatoria, una de estas entidades de red previamente creadas para conectarla a la que se acaba de crear. Esta conexión será directa o indirecta (i.e., habrá una entidad final entre ambas) dependiendo de la probabilidad de conexión directa entre dos entidades. Para decidir esto, se genera un número aleatorio entre 0 y 1, y en caso de que este número sea menor que la probabilidad de conexión directa entre entidades de red, ambas entidades de red (la que se acaba de crear y la que se ha tomado de forma aleatoria de las que ya había creadas) quedarán conectadas de forma directa. En caso contrario, si este número aleatorio es igual o superior a la probabilidad de conexión directa, se crea una nueva entidad final que actuará como "puente" entre ambas entidades de red, ya que estará conectada a ambas. Por

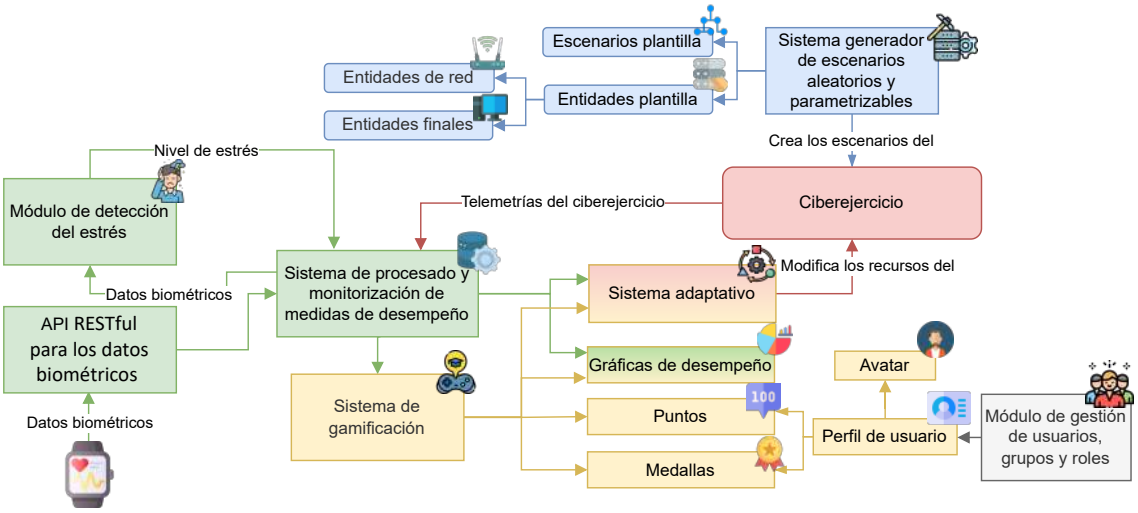


Figura 1. Arquitectura general del framework.

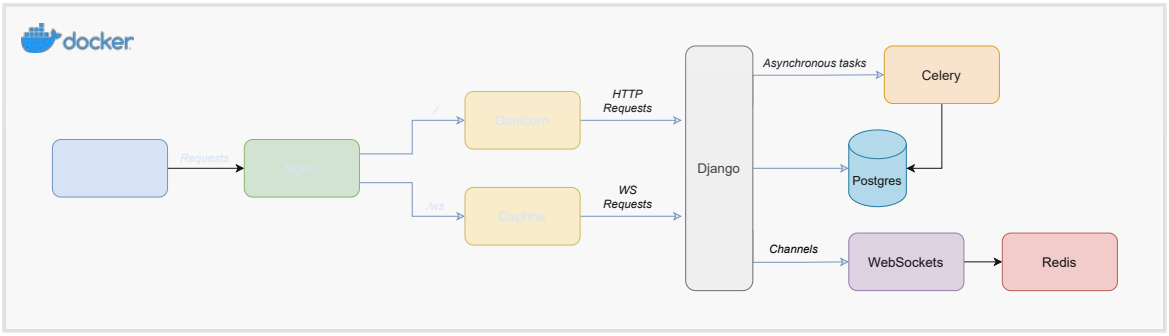


Figura 2. Arquitectura de la aplicación.

otra parte, dentro del segundo bucle, tan sólo se ha de crear una nueva entidad final en cada iteración y conectarla a una entidad de red seleccionada de forma aleatoria de entre las creadas previamente en el primer bucle.

Una vez que se han creado todas las entidades necesarias, el generador procede a realizar el direccionamiento IP de cada una de estas entidades, asignando a cada entidad de red un rango de direcciones y a cada entidad final conectada a dicha entidad de red una dirección única dentro de esa subred. Finalmente, el generador llama a una función auxiliar que se encarga, a partir de toda la información generada previamente (entidades, direccionamiento, etc.) de generar dos *JSON* con los nodos y enlaces presentes en la topología, los cuales se usan en el *front-end* para dibujar la topología para el usuario final.

Utilizando la lógica descrita previamente, el generador de escenarios aleatorios es capaz de construir modelos que representan topologías de entidades de red y finales realistas que serán diferentes en cada iteración y modificables para ser adaptadas a necesidades particulares.

III-C. Procesado y monitorización de medidas de desempeño

Otro sistema que destaca en el Cyber Range es el encargado del procesado y monitorización de medidas de desempeño, el cual recibe datos biométricos y diferentes telemetrías de los ciberejercicios y calcula a partir de estos una gran cantidad de medidas de desempeño. Dentro de estas medidas de desempeño es una novedad la inferencia del nivel de estrés del estudiante durante la realización del ciberejercicio. Para lograr esta detección del estrés el Cyber Range dispone de *API RESTful* que recibe datos biométricos y un módulo de detección del estrés. La *API* permite que diferentes pulseras inteligentes puedan enviar a través de *Wi-Fi* señales procedentes de electrocardiogramas, fotopletismografía (PPG), la distancia entre latidos o la frecuencia cardíaca, y sobre cualquiera de ellas se utiliza el módulo de detección de estrés. El módulo de detección del estrés divide estas señales en ventanas de longitud constante, sobre cada una de las ventanas extrae unas características que normaliza al rango [0,1] en función de un conjunto de mediciones del sujeto recopiladas en un periodo de no estrés. Finalmente, las características normalizadas se introducen en un modelo de detección de anomalías que clasifica cada una de las ventanas en estrés y no estrés.

III-D. Ciberejercicios dinámicos con gamificación

El sistema de gamificación y el sistema adaptativo reciben las medidas de desempeño procesadas por el sistema anterior

y las utilizan con el fin de mejorar la motivación de los estudiantes mediante el uso de elementos comunes de juegos en el contexto de los Cyber Ranges. El sistema adaptativo busca mantener al estudiante en una zona de desafío óptima, de forma que ni se sienta abrumado por la dificultad del ciberejercicio, ni se sienta aburrido por disponer de demasiados recursos para realizarlo. El sistema adaptativo establece los recursos disponibles según el desempeño previo, por lo tanto, manteniendo al estudiante en una zona óptima de desarrollo. Por ejemplo, el tiempo disponible, el número de intentos de bandera y las pistas consumibles se adaptan en función de su uso y del estrés en el reto anterior del ciberejercicio. El sistema adaptativo también es considerado parte del sistema de gamificación ya que es un tipo de herramienta que se encuentra frecuentemente en juegos, además el sistema de gamificación incluye los siguientes elementos:

- **Usuarios, grupos y roles**: un elemento básico de este Cyber Range es la separación por roles de los usuarios en *estudiantes*, *instructores* y *administradores*. Los estudiantes son los que realizan los ciberejercicios, generando diferentes telemetrías durante el ciberejercicio y de forma adicional sobre ellos se obtienen datos biométricos. Por otra parte, los instructores son los encargados de diseñar los ciberejercicios y de visualizar las diferentes medidas de desempeño recogidas durante estos. Finalmente, existe un rol de administrador centrado únicamente en el mantenimiento y gestión del Cyber Range. Asimismo, los estudiantes e instructores pueden pertenecer a diferentes grupos.
- **Perfil de usuario y avatar**: el perfil de usuario, además de almacenar el rol del usuario, contiene su avatar, nombre de usuario, nombre, apellidos y dirección de correo. En caso de que sea un estudiante, también almacena sus puntos y medallas obtenidas. Todos estos datos sólo son accesibles por el propio usuario, los instructores y los administradores.
- **Puntos**: los puntos son una moneda de cambio para promover un mayor rendimiento de los estudiantes en los ciberejercicios. Los instructores designan un número máximo de puntos otorgados por cada reto del ciberejercicio, y los puntos ganados por el estudiante se calculan en función de los recursos consumidos por el estudiante, siendo dichos recursos el tiempo utilizado, las pistas consumidas y los intentos de bandera (i.e., la solución del reto).
- **Medallas**: las medallas se usan como otro tipo de logro que pueden alcanzar los estudiantes, y tienen el objetivo de modelar el comportamiento de los estudiantes hacia acciones deseadas. Muchas de las acciones deseadas que modelan las medallas están relacionadas con los niveles de desempeño de los estudiantes, por ejemplo, para resolver retos sin uso de pistas o intentos fallidos. Este tipo de elemento de diseño permiten orientar el comportamiento del estudiante hacia direcciones deseadas, por ejemplo, si queremos que los alumnos espacien su aprendizaje y sean constantes, una medalla se generaría cuando un estudiante resolviera un ciberejercicio diferente durante cinco días consecutivos, motivando de forma externa mediante medallas que los estudiantes realicen un aprendizaje constante durante el tiempo.
- **Gráficas de desempeño**: permiten visualizar las medidas de desempeño, facilitando un análisis del rendimiento de los estudiantes y generar motivación competitiva entre estos cuando se comparan las medidas de desempeño de unos y otros. Sin embargo, el acceso a estas gráficas grupales de desempeño está restringido sólo para el instructor, y es el instructor el que debe decidir durante la realización de un ciberejercicio o al terminarlo, si quiere mostrar los resultados de las gráficas de desempeño grupales de toda la clase a los estudiantes.

IV. Casos de uso

En esta sección se describen los casos de uso más relevantes de un flujo de trabajo completo en la plataforma propuesta. Adicionalmente también se encuentra la gestión de usuarios y grupos, la personalización del avatar de los usuarios o la configuración del Cyber Range entre otros.

IV-A. Creación de un escenario

El primer paso en la creación de un ciberejercicio por parte de un instructor es la invención del escenario sobre el que se desarrolla. Para ello se ha de indicar un nombre, una descripción, un entorno y una complejidad. A partir de estos dos últimos parámetros y tomando la regla de generación correspondiente (la cual contiene una serie de parámetros que hacen que en cada iteración la generación sea diferente, tales como el máximo y mínimo de entidades finales, máximo y mínimo de entidades de red, etc.) se generará una topología, donde el instructor podrá personalizar cada una de las entidades de dicha topología haciendo clic sobre el nodo que desea modificar como se puede visualizar en la figura 3. Si el nodo se corresponde con una entidad de red, el instructor tan sólo podrá modificar el tipo de dispositivo. Por el contrario, si se pulsa sobre un nodo que se corresponde con una entidad final, el instructor podrá modificar el tipo de dispositivo, así como cambiar la plantilla que usará esa entidad, además de añadir o eliminar servicios y banderas. Una vez que el instructor ha diseñado el escenario y este se adecúa a sus requisitos, se guardará para ser usado posteriormente en futuros ciberejercicios.

IV-B. Creación de un reto

En nuestro framework los ciberejercicios son concebidos como una secuencia de retos adaptables desplegados en un mismo escenario. Por ello, el siguiente paso consiste en la creación de los diferentes retos que compondrán el ciberejercicio. Para cada reto es necesario establecer, su nombre, la competencia evaluada, el objetivo (atacar o defender), el tiempo máximo, el número de intentos, las pistas, el escenario sobre el que se ejecutará, una descripción, una explicación de la solución, la bandera del reto (la solución del reto)

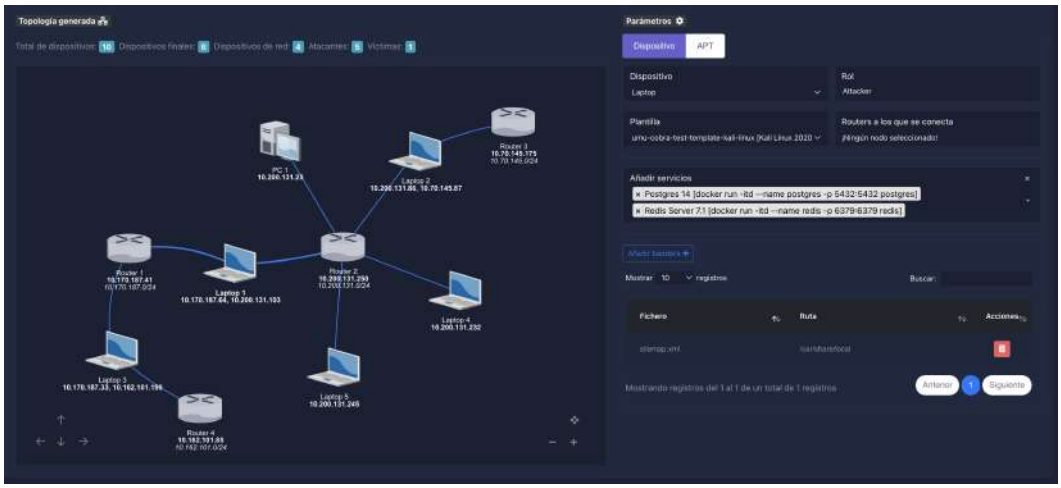


Figura 3. Topología en la creación de un escenario

y la dificultad que se calcula automáticamente en función de la puntuación del reto. Las pistas disponibles, el tiempo máximo y el número de intentos de bandera serán reducidos en función del rendimiento del estudiante en el reto anterior si el ciberejercicio es adaptativo.

IV-C. Creación de un ciberejercicio

Para crear un ciberejercicio en el framework propuesto, primero debe seleccionarse un escenario donde el ciberejercicio se desarrollará y, en función de éste, indicar la secuencia de retos entre los disponibles para ese escenario que compondrán el ciberejercicio estableciendo su competencia, puntuación, tiempo y dificultad. Además, el ciberejercicio tiene un nombre, una narrativa y unos objetivos.

IV-D. Activación de un ciberejercicio

Una vez seleccionado el ciberejercicio a activar, el instructor debe indicar los diferentes estudiantes o grupos de estudiantes que participarán en la simulación y establecer si será colaborativo y/o adaptativo. Posteriormente deberá asignar a cada usuario un despliegue del escenario o, si es un ciberejercicio colaborativo, a cada grupo de usuarios un despliegue. La generación de los despliegues de los escenarios puede realizarse mientras se activa el ciberejercicio o pueden haberse realizado previamente para evitar el tiempo de espera producido por la virtualización y la reserva de recursos en la generación de los despliegues. A continuación, el instructor puede iniciar, pausar y detener el ciberejercicio y, por otro lado, visualizar en tiempo real el progreso de los estudiantes como se muestra en la figura 4. Tras finalizar el ciberejercicio, el instructor puede observar diferentes gráficos que resumen la simulación del ciberejercicio como la puntuación de los estudiantes, el tiempo necesitado, las pistas utilizadas o la frecuencia cardíaca media de cada estudiante.

Si un estudiante dispone de una pulsera inteligente con la aplicación del framework propuesto, antes de comenzar el ciberejercicio deberá iniciar sesión a través de la pulsera y al menos cinco minutos antes de empezar el ciberejercicio

Figura 4. Gráfica de progreso en directo

deberá activar las mediciones en la pulsera con el objetivo de recolectar un estado base de los datos biométricos del estudiante antes de iniciar el ciberejercicio para el módulo de detección del estrés.

Una vez el estudiante accede al ciberejercicio, se le presenta la topología de las entidades visibles del escenario, permitiéndole conectarse a ellas. El front-end también le muestra al estudiante el nombre del reto, su descripción, el tiempo restante, la puntuación máxima y le permite realizar intentos de bandera y solicitar pistas. Tras finalizar cada reto, el estudiante avanza al reto siguiente, si el ciberejercicio es adaptativo los recursos disponibles en cada reto dependerán de su rendimiento en el reto anterior. Al finalizar el ciberejercicio, se le muestra al estudiante un resumen de sus estadísticas de la simulación, entre ellas su puntuación que se calcula en función de los recursos consumidos en cada uno de los retos.

IV-E. Visualización de medidas de desempeño

Los instructores pueden visualizar las medidas de desempeño de los ciberejercicios en tres niveles de agregación: a nivel de ciberejercicio, de simulación y de reto. Los instructores también pueden observar las medidas de desempeño globales o por competencias de los estudiantes para establecer rankings que pueden mostrarles a los estudiantes o utilizar para analizar el rendimiento de estos. Por otro lado, el estudiante

puede visualizar sus medidas de desempeño en cada una de las simulaciones que ha realizado, a nivel global y por competencias.

V. Discusión

La arquitectura del Cyber Range desarrollado incluye varios módulos que presentan una mejora con respecto al estado del arte. En primer lugar, después de analizar tres revisiones de la literatura sobre Cyber Ranges [8], [3], [9], observamos que en ningún caso se habla como características de la posibilidad de que el instructor configure sus propios escenarios a mano, y que estos puedan ser usados para crear nuevos ciberejercicios que posteriormente se podrán desplegar con facilidad en el entorno de virtualización. De esta forma, añadimos una gran versatilidad y funcionalidad con una herramienta de autoría que permite una infinidad de opciones de escenarios, y que a su vez es fácilmente escalable a otros entornos siempre y cuando se vayan generando las entidades plantilla necesarias para el tipo de escenario que se desee montar.

Además, otras de las características incluidas son novedades significativas dentro de la literatura. En primer lugar, mientras que la gamificación en general ha sido destacada como una potente aproximación para mejorar la motivación de los estudiantes [14], en el contexto de ciberseguridad ha sido aplicada principalmente en el formato de juegos serios [4] y encontramos escasos ejemplos de Cyber Ranges que incluyan estas características, con algunas excepciones como el Mizzou Cyber Range [15]. En nuestro caso, habilitamos un gran número de elementos en nuestro Cyber Range. En una taxonomía de elementos de gamificación en Cyber Ranges previa [8], encontramos que tenemos gran cantidad de los elementos y algunos que son la primera vez que se proponen, como el sistema adaptativo. En esta línea, en base a lo que hemos podido observar en las revisiones de la literatura [8], [3], [9], nuestro Cyber Range es el primero en considerar funcionalidades de aprendizaje adaptativo, a pesar de estar considerada como una de las tecnologías habilitadoras clave para proveer de una educación personalizada al estudiante [16].

Por último, destacar también el módulo para el procesado de señales multimodales habilitado mediante la recepción de datos con la API RESTful. Mientras que hasta ahora sólo hemos dado soporte a la recepción del ritmo cardiaco a través de un reloj inteligente y al uso de dicha señal para la detección del estrés, este módulo puede ser fácilmente extendido para contemplar una mayor cantidad de señales de otros dispotiivos y otras capacidades cognitivas. Por ejemplo, podríamos extenderlo con el uso de un casco de interacción cerebro-computador (BCI, de Brain-Computer Interaction), para obtener el electroencefalograma (EEG) y utilizarlo para medir la concentración y los niveles de actividad. Este tipo de aplicaciones entran dentro del área de evaluación de capacidades a través de sensores para dar soporte en la formación [17].

Dado que todo esto está unificado en una única plataforma, desde el punto de vista de tecnología educativa, el Cyber Range engloba lo que se conoce comúnmente como el sistema de gestión de contenidos educacionales (*LCMS*, de *Learning Content Management System*) que es con el que interactúan los instructores para crear los contenidos, y el sistema de gestión del aprendizaje (*LMS*, de *Learning Management System*), que es el lugar donde los estudiantes desarrollan el aprendizaje [18]. Además, con las funcionalidades incluidas en términos de analítica del aprendizaje con numerosas métricas incluso a tiempo real, los distintos elementos de gamificación, y la opción de activar el sistema adaptativo, podemos considerar nuestro Cyber Range como un sistema integral que incluye tres de las tecnologías educativas de vanguardia claves [19], unificadas en un sólo entorno para el entrenamiento de profesionales en ciberseguridad.

Además, la arquitectura presentada es modular, escalable y agnóstica del entorno de virtualización donde se vayan a desplegar los escenarios, lo cuál quiere decir, que se podría re-utilizar toda la arquitectura, cambiando sólo el entorno de virtualización de salida en el fichero Terraform generado por el sistema generador de escenarios. Sólo sería necesaria una pequeña adaptación para generar el nuevo fichero Terraform en el formato adecuado. Esto facilita en este sentido su reutilización en otros entornos de formación en ciberseguridad, que además tiene una clara aplicación dual (civil y militar). Las entidades que pueden estar interesadas este tipo de formación en ciberseguridad son variadas, incluyendo centros de educación superior, profesionales de las TIC en empresas y unidades de las fuerzas y cuerpos se seguridad que trabajen en ciberdefensa. Dada la infraestructura que tenemos y su fácil despliegue mediante contenedores, sería posible que la reutilizaran y la desplegaran en sus premisas. Sin embargo, también es cierto que en muchas ocasiones no tienen el personal necesario para mantener dicha infraestructura técnica, por lo tanto también podría ser de interés que se proporcionaran estos servicios de formación con Cyber Ranges como un servicio, siguiendo un paradigma que podríamos definir como *Cyber Range as a Service* (*CRaaS*). Este servicio podría ofrecer de forma transparente formación en ciberseguridad práctica y en escenarios realistas, abstrayéndose de toda la complejidad de infraestructura, virtualización, así como la gestión del aprendizaje y la creación de los contenidos.

Como principales limitaciones a destacar, es que el Cyber Range descrito aquí todavía no ha sido puesto a prueba en un entorno real con una gran cantidad de estudiantes y escenarios simultáneamente desplegados, por lo que aún no se conoce con exactitud las necesidades técnicas y los posibles problemas de escalabilidad que puedan surgir. Además, todavía hay trabajo en el que tenemos que ahondar y desarrollar, como capturar la telemetría de las máquinas de los escenarios desplegados, analizar la usabilidad del Cyber Range para mejorar algunos aspectos de diseño o incrementar el número de entidades plantilla disponibles en el repositorio de plantillas virtualizadas para mejorar las posibilidades de creación de escenarios. No obstante, dada la importancia que tiene la formación en ciberseguridad de alta calidad hoy en día, las contribuciones novedosas al estado del arte en el contexto de Cyber Ranges, y dada la escasa oferta de Cyber Ranges desarrollados por entidades españolas (especialmente por universidades),

consideramos que nuestro Cyber Range puede hacer una contribución significativa a la literatura y la sociedad.

VI. Conclusiones y trabajo futuro

En este artículo hemos presentado un resumen de la arquitectura del Cyber Range que está siendo desarrollado, que habilita funcionalidades clave y noveles dentro de la literatura. Algunos de los puntos clave de la arquitectura son los siguientes: 1) la generación de ejercicios aleatorios, que está soportada por un módulo específico que permite parametrizar, aleatorizar y configurar en detalle los escenarios y cada uno de sus nodos, 2) un módulo de gamificación que permite desplegar diversos elementos como puntuaciones, medallas, o gráficas de desempeño para mejorar el compromiso y la motivación de los estudiantes, 3) un sistema adaptativo que es capaz de adaptar los recursos disponibles de los estudiantes en cada reto de un ciberejercicio en función de su desempeño previo, y 4) un sistema de recolección y procesado de datos que genera numerosas medidas de desempeño en los ciberjercicios y también da soporte al procesado de señales biométricas, por ejemplo, para detectar el estrés de los estudiantes.

Este Cyber Range abre nuevas oportunidades de validación e investigación en el futuro. En primer lugar, aquellas relaciones con la validación técnica de la arquitectura para analizar su escalabilidad, efectividad, y usabilidad dentro de experiencias reales, ya sean en asignaturas de ciberseguridad de educación superior o en otros centros donde se interese esta formación. Estas experiencias también generarán grandes juegos de datos que permitirán analizar el comportamiento de los usuarios con el Cyber Range, facilitando la investigación con casos de estudio reales para evaluar el impacto de los distintos módulos del Cyber Range. Planeamos explorar como ofrecerse estos servicios a distintos actores interesados, de forma que puedan consumirlos de forma transparente, sin preocuparse de los requisitos de infraestructura o despliegue. Finalmente, también planeamos extender muchos de estos módulos para incorporar nuevas funcionalidades, como por ejemplo, recolectar señales biométricas adicionales o dar soporte al despliegue de más servicios en los escenarios.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto COBRA (10032/20/0035/00), concedido por el Ministerio de Defensa y en particular gestionado por DGAM/SDGPLATIN y MCCE bajo el proyecto COINCIDENTE, así como por el proyecto SCORPION (21661-PDC-21), concedido por la Fundación Séneca de la Región de Murcia.

Referencias

[1] J. V. Botello, A. P. Mesa, F. A. Rodríguez, D. Díaz-López, P. Nespoli, and F. G. Mármol, "BlockSIEM: Protecting smart city services through a Blockchain-based and distributed SIEM," *Sensors*, vol. 20, no. 16, 2020.

[2] P. Nespoli, D. Useche Peláez, D. Díaz López, and F. Gómez Mármol, "COSMOS: Collaborative, Seamless and Adaptive Sentinel for the Internet of Things," *Sensors*, vol. 19, no. 7, 2019.

[3] N. Chouliaras, G. Kittes, I. Kantzavelou, L. Maglaras, G. Pantziou, and M. A. Ferrag, "Cyber ranges and testbeds for education, training, and research," *Applied Sciences*, vol. 11, no. 4, pp. 1–23, feb 2021.

[4] J.-N. Tioh, M. Mina, and D. W. Jacobson, "Cyber security training a survey of serious games in cyber security," in *2017 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2017, pp. 1–5.

[5] F. Gómez Mármol, J. A. Ruipérez-Valiente, P. Nespoli, G. Martínez Pérez, D. Rivera Pinto, X. Larriva Novo, M. Álvarez Campana, V. Villagrá González, J. Maestre Vidal, F. A. Rodríguez Lopez, M. Páramo Castrillo, J. I. Rojo Lacal, and R. García-Abril Alonso, "COBRA: Cibermaniobras adaptativas y personalizables de simulación hiperrealista de APTs y entrenamiento en ciberdefensa usando gamificación," in *VI Jornadas Nacionales de Investigación en Ciberseguridad (JNIC '21)*, 2021.

[6] R. Beuran, K.-i. Chinen, Y. Tan, and Y. Shinoda, "Towards Effective Cybersecurity Education and Training," *Research report (School of Information Science, Graduate School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology)*, vol. IS-RR-2016, pp. 1–16, 2016.

[7] P. Blikstein and M. Worsley, "Multimodal learning analytics and education data mining: using computational technologies to measure complex learning tasks," *Journal of Learning Analytics*, vol. 3, no. 2, pp. 220–238, Sep. 2016. [Online]. Available: https://learning-analytics.info/index.php/JLA/article/view/4383

[8] E. Ukwandu, M. A. B. Farah, H. Hindy, D. Brosset, D. Kavallieros, R. Atkinson, C. Tachtatzis, M. Bures, I. Andonovic, and X. Bellekens, "A review of cyber-ranges and test-beds: Current and future trends," *Sensors (Switzerland)*, vol. 20, no. 24, pp. 1–36, 2020.

[9] M. M. Yamin, B. Katt, and V. Gkioulos, "Cyber ranges and security testbeds: Scenarios, functions, tools and architecture," *Computers and Security*, vol. 88, p. 101636, 2020. [Online]. Available: https://doi.org/10.1016/j.cose.2019.101636

[10] J. Vykopal, R. Ošlejšek, P. Čeleda, M. Vizváry, and D. Tovarňák, "KYPO cyber range: Design and use cases," *ICSOFT 2017 - Proceedings of the 12th International Conference on Software Technologies*, no. January, pp. 310–321, 2017.

[11] T. Lieskovan and J. Hajný, "Building open source cyber range to teach cyber security," in *The 16th International Conference on Availability, Reliability and Security*, ser. ARES 2021. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3465481.3469188

[12] R. Beuran, C. Pham, D. Tang, K.-i. Chinen, Y. Tan, and Y. Shinoda, "Cytrone: An integrated cybersecurity training framework," *Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017)*, pp. 157–166, 2017. [Online]. Available: https://ci.nii.ac.jp/naid/120007005345/en/

[13] J. Eckroth, K. Chen, H. Gatewood, and B. Belna, "Alpaca: Building dynamic cyber ranges with procedurally-generated vulnerability lattices," in *Proceedings of the 2019 ACM Southeast Conference*, ser. ACM SE '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 78–85. [Online]. Available: https://doi.org/10.1145/3299815.3314438

[14] E. D. Mekler, F. Brühlmann, K. Opwis, and A. N. Tuch, "Do points, levels and leaderboards harm intrinsic motivation? an empirical analysis of common gamification elements," in *Proceedings of the First International Conference on gameful design, research, and applications*, 2013, pp. 66–73.

[15] K. B. Vekaria, P. Calyam, S. Wang, R. Payyavula, M. Rockey, and N. Ahmed, "Cyber range for research-inspired learning of "attack defense by pretense" principle and practice," *IEEE Transactions on Learning Technologies*, vol. 14, no. 3, pp. 322–337, 2021.

[16] F. Martin, Y. Chen, R. L. Moore, and C. D. Westine, "Systematic review of adaptive learning research designs, context, strategies, and technologies from 2009 to 2018," *Educational Technology Research and Development*, vol. 68, no. 4, pp. 1903–1929, 2020.

[17] J. Schneider, D. Börner, P. Van Rosmalen, and M. Specht, "Augmenting the senses: a review on sensor-based learning support," *Sensors*, vol. 15, no. 2, pp. 4097–4133, 2015.

[18] S. Ninoriya, P. Chawan, and B. Meshram, "Cms, lms and lcms for elearning," *International Journal of Computer Science Issues (IJCSI)*, vol. 8, no. 2, p. 644, 2011.

[19] M. J. Gomez, J. A. Ruipérez-Valiente, and F. J. García Clemente, "Analyzing trends and patterns across the educational technology communities using fontana framework," *IEEE Access*, vol. 10, pp. 35 336–35 351, 2022.

Ejercicio de cyber-range avanzado en una subestación eléctrica

Cristina Regueiro
TECNALIA, Basque Research and Technology Alliance (BRTA)
Parque Científico y Tecnológico de Bizkaia
Astondo Bidea, Edificio 700
E-48160 Derio
Bizkaia - Spain
cristina.regueiro@tecnalia.com

Angel López
TECNALIA, Basque Research and Technology Alliance (BRTA)
Parque Científico y Tecnológico de Bizkaia
Astondo Bidea, Edificio 700
E-48160 Derio
Bizkaia - España
angel.lopez@tecnalia.com

Xabier Yurrebaso
TECNALIA, Basque Research and Technology Alliance (BRTA)
Parque Científico y Tecnológico de Bizkaia
Astondo Bidea, Edificio 700
E-48160 Derio
Bizkaia - España
xabier.yurrebaso@tecnalia.com

Elixabete Ostolaza
TECNALIA, Basque Research and Technology Alliance (BRTA)
Parque Científico y Tecnológico de Bizkaia
Astondo Bidea, Edificio 700
E-48160 Derio
Bizkaia - España
elixabete.ostolaza@tecnalia.com

Florian Gautier
Beware Cyberlabs
74 Rue Georges Bonnac
33000 Bordeaux - France
florian.gautier@beware-cyberlabs.eu

Julien Calvas
Beware Cyberlabs
74 Rue Georges Bonnac 33000
Bordeaux - France
julien.calvas@beware-cyberlabs.eu

Bernard Roussely
Beware Cyberlabs
74 Rue Georges Bonnac
33000 Bordeaux - France
bernard.roussely@beware-cyberlabs.eu

Resumen- **Las necesidades de formación en ciberseguridad en sectores críticos como el eléctrico se han incrementado exponencialmente según el número de ciberataques sufridos crece. En este contexto, las nuevas herramientas de formación como el cyber-range toman gran relevancia. Sin embargo, las infraestructuras tradicionales de cyber-range, basadas en elementos virtualizados, tienen algunas limitaciones de realismo cuando se consideran la Tecnologías de la Operación. Es por ello, que se ve la necesidad de avanzar y desarrollar nuevas infraestructuras que incluyan elementos reales. En este sentido, este trabajo presenta una nueva infraestructura de cyber-range avanzado para el sector energético que incluye sistemas reales de control industrial y, haciendo uso de ella, desarrolla un ejercicio de cyber-range con diferentes desafíos de ciberseguridad habituales en el sector. El cyber-range ha sido ejecutado en una sesión oficial con alumnos universitarios, obteniendo resultados muy positivos.**

***Index Terms*- Cyber-range, ciberseguridad, energía, formación, capacitación, OT, laboratorio**

Tipo de contribución: *Formación e innovación educativa*

I. INTRODUCCIÓN

En la actualidad, nos encontramos en una situación en la que la tecnología avanza a pasos agigantados. El sector energético está experimentando un proceso de digitalización donde tecnologías emergentes como, por ejemplo, la inteligencia artificial, Blockchain, la criptografía cuántica, 5G, el edge-computing, etc., están comenzando a ser desplegadas en los distintos sistemas e infraestructuras [1]. Aunque el uso de estas nuevas tecnologías abre la puerta a nuevas funcionalidades, también lo hacen a nuevas ciberamenazas. En los últimos años, el número de ciberamenazas existentes está creciendo exponencialmente y son cada vez más complejas y con efectos más relevantes [2] [3].

El papel de los trabajadores es esencial para garantizar la ciberseguridad en cualquier empresa. Es por ello por lo que se hace evidente la necesidad de que los trabajadores, de cualquier sector, dispongan de los conocimientos necesarios para tratar de prevenir, detectar y responder al creciente número de ciberataques que están comenzando a sufrir las

empresas del sector eléctrico. Es importante abordar la ciberseguridad dentro de las organizaciones de forma global. Por tanto, se necesita personal cualificado en ciberseguridad en todos los ámbitos de una organización, involucrando a ingenieros de sistemas, ingenieros software, ingenieros hardware, expertos en seguridad de redes, analistas de seguridad, consultores, etc. Los conocimientos y formación en materia de ciberseguridad deben ser, por tanto, adaptados a los distintos roles. Para que las acciones formativas sean eficaces, es preciso identificar tanto la actividad como el entorno de trabajo o los activos operados por cada uno de los trabajadores [4].

En la actualidad, sin embargo, la situación dista mucho de ser la deseada y existe una gran carencia de personal cualificado con amplios conocimientos en materia de ciberseguridad que contribuyan a aumentar la ciberresiliencia del sector eléctrico. A continuación, se enumeran las principales causas detrás de esta falta de personal:

- En primer lugar, las empresas del sector eléctrico están comenzando a darse cuenta de la importancia de la ciberseguridad a medida que abordan el proceso de digitalización de sus infraestructuras. No han abordado aún planes de concienciación y formación específicos para sus trabajadores.
- Además, la oferta educativa enfocada en la ciberseguridad en entornos de las Tecnologías de la Operación es escasa.
- Por otro lado, existe una alta demanda de personal cualificado en materia de ciberseguridad en todos los sectores, debido a que las organizaciones son conscientes de su importancia para sus negocios. Por este motivo la actual oferta existente en el mercado laboral no permite cubrir la alta demanda que se necesita.
- Además, aún hoy en día, siguen existiendo barreras de conocimiento en las personas que hacen que la formación sea un proceso bastante lento; hay trabajadores que no han trabajado nunca con temas de seguridad e, incluso, que no han tratado nunca con temas tecnológicos. En estos casos, formarles en aspectos tan avanzados puede resultar lento y costoso.
- Por último, la formación práctica, que habitualmente es mucho más beneficiosa, ágil y efectiva, suele ser preferible a una formación tradicional más teórica. Sin embargo, la formación práctica supone grandes costes para las organizaciones, por lo que no siempre es una opción.

En este contexto, el cyber-range surge como una herramienta práctica que permite ayudar a las organizaciones o, incluso, a los entes de formación (universidades, ciclos formativos, etc.) a elaborar sesiones formativas prácticas que permitan solucionar los problemas descritos anteriormente. El cyber-range incluye herramientas que ayudan a fortalecer la estabilidad, la seguridad y el rendimiento de los sistemas IT/OT. De hecho, la organización ECSO, a través de su grupo de trabajo SWG5.1 “Entornos y ejercicios técnicos de cyber-range”, ha declarado la importancia de avanzar en los cyber-ranges en Europa considerándolo uno de los mejores medios para permitir el crecimiento de la industria de la ciberseguridad y fortalecer la capacidad de ciberseguridad de Europa [5]. Los cyber-ranges se han identificado en la agenda estratégica de investigación de ECSO como uno de los constituyentes del Cyber Pillar (o pilar cibernético) para crear un ecosistema técnico de experimentación y capacitación en ciberseguridad.

Es cierto que, hasta hace poco, la ciberseguridad de las Tecnologías de la Operación (OT, por sus siglas en inglés, Operational Technologies) ha recibido mucha menos atención que la ciberseguridad de los sistemas tradicionales de Tecnologías de la Información (IT, por sus siglas en inglés, Information Technologies). Aunque esto ha comenzado a cambiar en los últimos años con el lanzamiento de normativas (por ejemplo, ISA/IEC 62443) y herramientas (por ejemplo, cortafuegos específicos como los FortiGate® Rugged Series [6]) específicas del sector industrial, todavía existe una brecha significativa entre el mundo IT y OT con respecto a la comprensión, el saber hacer, las soluciones y la formación de los profesionales al abordar los aspectos relacionados con la ciberseguridad.

En la actualidad, los sistemas de control industrial y automatización son fundamentales para garantizar el funcionamiento adecuado de diferentes sistemas críticos que las empresas y los ciudadanos utilizan habitualmente. Periódicamente, se han producido ataques que han afectado la operación de sistemas críticos de OT, lo que evidencia que estos sistemas son vulnerables y que, en muchos casos, los propietarios y los operadores no han aplicado las medidas básicas de protección necesarias. Uno de los sectores más críticos y que más ciberataques ha sufrido es el sector energético. Por ejemplo, Stuxnet el primer gusano conocido que espía y reprograma sistemas industriales que fue descubierto en 2010 y se usó contra la instalación nuclear iraní de Natanz para autodestruir centrifugadoras que enriquecían uranio [7]. Por otro lado, en el año 2012 la compañía petrolera Saudí Aramco sufrió uno de los peores ciberataques de la historia, con 35.000 ordenadores parcialmente borrados o incluso destruidos, lo que mantuvo a la empresa desconectada de la red durante meses [8]. Además, en el año 2016 hubo un ciberataque a la red eléctrica de Ucrania, que mantuvo a partes de la ciudad de Kiev sin luz durante horas [9].

Es por ello por lo que se hace evidente la necesidad de formación en ciberseguridad en un dominio como el energético. Precisamente, el objetivo de este trabajo es el de presentar una infraestructura de cyber-range avanzado en el sector de la energía, así como la definición de un ejercicio de formación que permita solventar, de cierta manera, las carencias de conocimientos en ciberseguridad industrial en el sector.

El resto del artículo está organizado de la siguiente manera: en la Sección II se introduce el concepto de un cyber-range tradicional para, en la Sección III, presentar una nueva infraestructura de cyber-range avanzado en el sector energético. En la Sección IV se presenta el ejercicio de formación propuesto haciendo uso de la nueva infraestructura, describiendo en la Sección V su ejecución y resultados. Finalmente, en la Sección VI se presentan las principales conclusiones y líneas futuras.

II. Cyber-ranges Tradicionales

El cyber-range es una infraestructura hardware y software que permite la creación de escenarios para facilitar la formación en ciberseguridad de forma práctica. Se presenta como una herramienta muy versátil con las siguientes características [10]:

- Estandarización: se suele usar un lenguaje único para todos los componentes del cyber-ranges para que sean fácilmente interoperables entre sí.
- Automatización: la creación de ejercicios o de nuevas

topologías de red, etc., o la monitorización o gestión del cyber-range suelen tener un cierto grado de automatización.

- Adaptabilidad: es una herramienta que se puede adaptar a cualquier tecnología, de forma que, no solo se practiquen herramientas tradicionales, sino que también se pueda, por ejemplo, practicar el uso de nuevas técnicas de ciberseguridad.
- Capacidad de simulación de servicios de Internet o de actividad de otros usuarios, que den un mayor realismo a los escenarios, ya que rara vez se opera de manera aislada.
- Capacidad de simulación de ataques controlados que permitan practicar la protección y seguridad o, viceversa, capacidad de simulación de defensa, que permita identificar vulnerabilidades y explotarlas.
- Gestión de competencias: los ejercicios se diseñan acordes a las competencias que se pretendan desarrollar; no es lo mismo un ejercicio enfocado a vulnerabilidades en la autenticación, que a virus.
- Recogida y análisis de datos: así se pueden determinar estadísticas o, por ejemplo, detectar "hábitos" de los usuarios.
- Puntuación e informe de resultados.
- Herramientas de instructor: supervisión y control de los progresos de los participantes, tablón de mensajes para comunicaciones generales, chat privado con los participantes para dar soporte o ayuda.

Actualmente, la mayoría de los cyber-ranges son plataformas a las que se accede remotamente a través de un navegador y que permiten a los usuarios definir sus propios escenarios virtualizados y realizar ejercicios de formación sobre redes simuladas [11]. CyberBit [12] o Cyrin [13] son algunos ejemplos de este tipo de herramientas. Sin embargo, también hay otros cyber-ranges que incluyen salas físicas, donde se encuentra todo el equipamiento necesario para que acudan los participantes a realizar la formación. Cybex [14], el Michigan Cyber Range [15] o el laboratorio de cyber-ranges de Tecnalia [16] son dos ejemplos de este tipo de cyber-range que no solo proveen la infraestructura cloud donde se diseñan y despliegan los ejercicios, sino que también cuentan con la infraestructura física necesaria para poder realizarlos. Este tipo de cyber-range ha sido ampliamente utilizado durante los últimos años con resultados muy prometedores [17] [18] [19], especialmente cuando únicamente se involucra al mundo IT. Sin embargo, cuando se comienzan a incluir sistemas de control industrial y automatización en los escenarios de cyber-range, la calidad de la formación va a depender enormemente de la calidad de la virtualización de los componentes industriales involucrados, es decir, en la exactitud del gemelo digital en "imitar" al componente OT real [20].

III. INFRAESTRUCTURA DEL CYBER-RANGE AVANZADO EN EL SECTOR ENERGÉTICO

Cuando se habla del sector energético, se involucra a numerosos componentes OT para monitorizar y controlar los procesos, dispositivos e infraestructura. Como se ha comentado en la sección II, algunos cyber-ranges han comenzado a emplear gemelos digitales de componentes industriales como es el caso, por ejemplo, de Cybex [14] que cuenta con gemelos digitales de algunos sistemas de control industrial (conmutadores inteligentes, enrutadores inalámbricos, etc.). Sin embargo, conseguir gemelos digitales de calidad de elementos OT específicos no es una tarea sencilla, lo que supone que, en ocasiones, no se tienen, o se emplean gemelos digitales "limitados" que restan realismo al escenario y, por tanto, a la formación del cyber-range.

Para evitarlo, se propone emplear directamente equipamiento real de control industrial y automatización, de forma que se consiga el máximo realismo posible, ya que se estarían usando los mismos equipos que se emplean en la realidad. En este trabajo se propone un nuevo concepto de cyber-range avanzado en el que se combina una infraestructura tradicional de cyber-range (Cybex, subsección A) con una infraestructura de laboratorio de ciberseguridad en redes eléctricas (Tecnalia, subsección B), permitiendo el acceso remoto de los participantes. La arquitectura de este cyber-range es la que se muestra en la Fig. 1. Todo el tráfico de red y la información de log de los equipos reales del laboratorio será enviada en tiempo real a la infraestructura de cyber-range para su análisis por parte de los participantes.

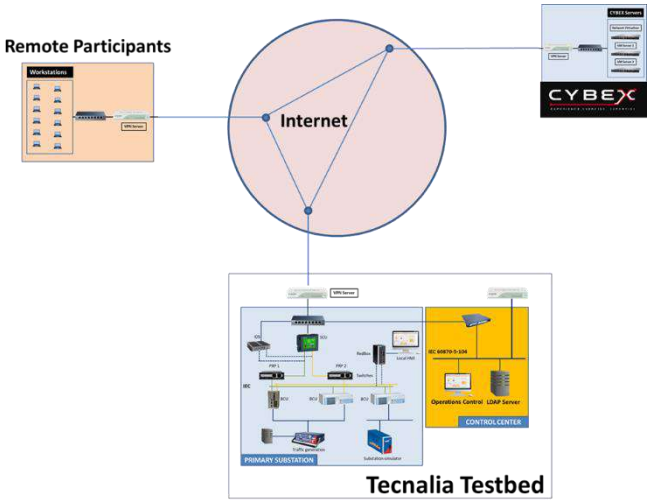


Fig. 1 Arquitectura del cyber-range avanzado en el sector energético

A. *Infraestructura de cyber-range: Cybex*

Cybex [14], de Beware Cyberlabs, es un sistema de entrenamiento en ciberdefensa, basado en un entorno virtual controlado y aislado, y que permite simular la red (parcial o totalmente) y los sistemas de una organización. Es un cyber-range muy versátil que se puede utilizar para entrenar a diferentes profesionales, para simular la actividad de IT, para llevar a cabo actividades de I+D y para probar diferentes soluciones de software o hardware, ya que puede ser hibridado combinando activos físicos y virtuales que trabajan conjuntamente.

Cybex proporciona un entorno de sandbox para realizar simulaciones de diferentes configuraciones replicadas, permitiendo el uso de soluciones de código abierto o patentadas en componentes virtuales. El cyber-range puede usarse de forma aislada o conectarse a Internet, a otra LAN (como es el caso del cyber-range avanzado propuesto) o a cualquier entorno físico con una interfaz ethernet adecuada.

No hay límite en el número de usuarios simultáneos de Cybex. Del mismo modo, no hay límite en el número de objetos virtuales que pueden ser administrados o utilizados y la única limitación con respecto a los recursos reside en el rendimiento total del servidor (núcleos de CPU) y la capacidad acumulada (RAM y espacio de disco). De esta forma, el

número máximo de usuarios simultáneos en un momento determinado puede estimarse fácilmente para cada escenario una vez conocido su consumo individual de recursos.

Cybex posee un generador de tráfico con muchos de los protocolos TCP/IP estándar, que se puede extender con protocolos o variantes específicas. Sin embargo, no se usó en este caso, ya que el tráfico se genera continuamente en el laboratorio de ciberseguridad de redes eléctricas que se describe en la subsección B.

B. Laboratorio de ciberseguridad en redes eléctricas

El laboratorio de ciberseguridad en redes eléctricas forma parte del Nodo de Ciberseguridad del Basque Digital Innovation Hub (BDIH) [21]. Se trata de un entorno seguro y controlado donde se pueden instalar y probar equipos con nuevas capacidades de ciberseguridad, y donde los incidentes de ciberseguridad pueden ser simulados para probar la eficiencia de soluciones avanzadas de detección y protección.

El laboratorio, que se observa en la Fig. 2, emula el comportamiento de una subestación primaria IEC 61850 [22] y un centro de control DSO (Distribution System Operator) que permite la generación de tráfico de los principales protocolos de comunicación industrial utilizados en el sector de redes eléctricas (e.g., IEC 60870-5-104, IEC 61850, DNP3, Modbus, LDAP, NTP), así como otros protocolos TCP/IP más generales, como HTTPS, SFTP, SSH o RDP. El laboratorio también dispone de un conjunto de herramientas que permiten simular diferentes tipos de ciberataques (e.g., denegación de servicio, Man-in-the-Middle, fuerza bruta, fraude de identidad, etc.), y herramientas de monitorización que permiten analizar lo que está sucediendo en la red de comunicaciones.

Fig. 2 Laboratorio de ciberseguridad en redes eléctricas

Por un lado, el Centro de Control contiene un SCADA IEC 60870-5-104 y servidores LDAP & NTP. Por otro lado, la subestación primaria IEC 61850 está compuesta por:

- Una Unidad de Control de Subestación (UCS).
- Tres Bay Control Units (BCU) que controlan los interruptores de media y alta tensión de la subestación simulada.
- Un SCADA IEC 61850.
- Un Bus de subestación redundante (PRP) donde los componentes anteriores están conectados.
- Un generador de señales analógicas y digitales (OMICRON CMC 256).
- Un generador de valores de muestra (OMICRON CMC 850).

En la Fig. 3 se muestra la arquitectura del laboratorio, en el que se ha configurado una red privada virtual (VPN) para comunicar la subestación primaria y el Centro de Control. El laboratorio permite:

- Generar una comunicación real entre los diferentes equipos y sistemas para obtener el tráfico real y continuo de la operación.
- Instalar equipos y sistemas con nuevas capacidades de ciberseguridad.
- Reproducir incidentes de ciberseguridad a través de herramientas de hacking ético.
- Comprobar la respuesta de los equipos y sistemas de información a estos ataques.
- Probar la eficacia de las herramientas de detección de ataques.

Cada uno de los equipos del laboratorio proporciona información de log que es recogida en un equipo que implementa la funcionalidad de "log collector", centralizando la información de log relativa a todo el equipamiento y permitiendo su visualización. Este equipo no se incluye en Fig. 3 porque no forma parte de la funcionalidad básica del laboratorio, y ha sido incluida únicamente para la realización del cyber-range avanzado.

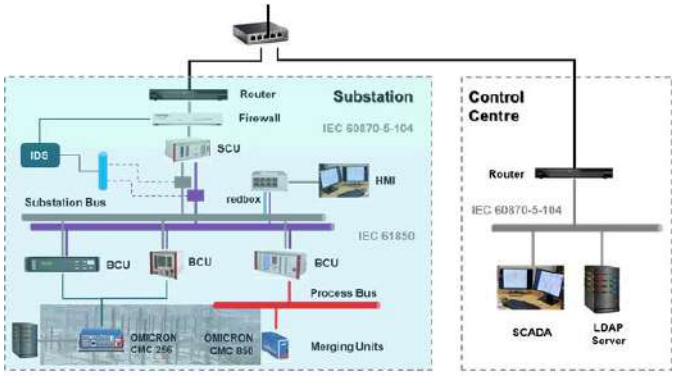


Fig. 3 Arquitectura del laboratorio de ciberseguridad en redes eléctricas

IV. EJERCICIO DE CYBER-RANGE AVANZADO

Haciendo uso de la infraestructura descrita en la sección III, se ha diseñado un ejercicio de cyber-range para potenciar la formación en defensa (ejercicio de blue team), en el que los participantes asumen el papel de operadores de un Centro de Operaciones de Seguridad (SOC) recientemente creado por una importante compañía eléctrica, para registrar, supervisar y analizar los eventos de ciberseguridad de los principales activos de control industrial de una subestación eléctrica de la empresa. El SOC recibe una gran cantidad de tráfico de red en bruto de los activos instalados en la subestación eléctrica (subsección III.B) que deben procesarse de forma adecuada para poder identificar posibles problemas de ciberseguridad.

Los participantes deben comprender cómo interpretar las comunicaciones de red entre los diferentes activos para identificar un posible incidente de seguridad que tenga lugar dentro de la infraestructura de la subestación. También deben identificar la mejor manera de automatizar su detección e identificación, así como las medidas adecuadas que permitan mitigar (o eliminar) esos incidentes de seguridad en el futuro.

Para ello, se han definido cinco grupos de desafíos técnicos diferentes que permitirán a los participantes identificar diferentes ataques habituales del sector en base a los logs de

los equipos y el tráfico de red. Alguno de estos desafíos requerirá de la ejecución de un ciberataque controlado por parte de los instructores del cyber-range, que provoque, en tiempo real, que los datos de log se actualicen consecuentemente. La puntuación máxima del ejercicio es de 550 puntos, que se consigue si todos los desafíos se resuelven correctamente.

A. Grupo de desafíos 1: Comprensión de los datos registrados

Este primer grupo tiene como objetivo mostrar a los estudiantes cómo interpretar los datos de log que normalmente se recogen en escenarios reales y contiene desafíos fáciles con la intención de familiarizar a los participantes con varios aspectos de la infraestructura de cyber-range Cybex utilizada en el ejercicio, con los diferentes protocolos de comunicaciones existentes en la infraestructura y con la arquitectura de red de una subestación eléctrica como la del laboratorio, ya que se considera crucial tener una buena comprensión de la arquitectura de red, junto con una adecuada interpretación de los datos de log disponibles, para poder avanzar hacia los próximos desafíos.

- *Objetivo:* A partir del análisis del tráfico de red, los participantes deben extraer información relacionada con las direcciones IP y MAC de cada dispositivo ubicado en la subestación. Además, deben identificar qué par IP/MAC corresponde al SCADA y, también, averiguar los nombres de los estándares que cumple la subestación eléctrica.
- *Instructor:* El instructor (o gestor de la sesión) no necesita ejecutar ningún ataque controlado ni realizar acciones especiales; el funcionamiento estándar de la subestación proporciona todos los datos necesarios para completar este grupo de desafíos.

B. Grupo de desafíos 2: Detección y protección contra un ataque de Fuerza Bruta

El segundo grupo de desafíos permite a los participantes enfrentarse a un ataque de fuerza bruta [23] procedente del exterior de la infraestructura de la subestación.

Un error de un operador de mantenimiento ha dejado abierto un puerto TCP de Internet en el cortafuegos, exponiéndolo indebidamente. Los atacantes han detectado esta vulnerabilidad y han lanzado un ataque de fuerza bruta contra la UCS (ataque ejecutado por el instructor).

- *Objetivo:* El principal objetivo de los participantes es detectar qué tipo de ataque se está produciendo, identificar la IP del activo que ha sido objeto del ataque, la IP del atacante y el servicio que se está atacando. Además, los participantes tienen que determinar si los atacantes han tenido éxito y, por tanto, han conseguido acceder de forma no autorizada a un dispositivo de dentro de la subestación y, en caso afirmativo, identificar qué cuenta de usuario se ha visto comprometida. Finalmente, los participantes deben identificar los problemas de seguridad que permitieron este ciberataque y proponer una respuesta adecuada para corregirlo.
- *Instructor:* El puerto pertinente debe abrirse en el cortafuegos para aceptar el tráfico procedente de la IP externa controlada por el instructor. El instructor lanza el ataque de fuerza bruta contra la UCS (atacante externo). Tras la ejecución controlada del ataque, los equipos involucrados habrán "actualizado" sus logs.

C. Grupo de desafíos 3: Detección y protección contra ataques de denegación de servicio

En este grupo de desafíos, los participantes necesitan detectar y responder a un ataque de denegación de servicio (DoS) [24], cuyo origen está dentro de la subestación (el atacante ha entrado a la subestación en la fase previa). Uno de los IEDs es vulnerable a un ataque TCP SYN Flood [25], quedando no disponible para responder a las órdenes del SCADA o proporcionar información de su estado. Cuando esto sucede, el SCADA comienza a utilizar protocolos ARP e ICMP para tratar de restablecer la conexión perdida con el IED.

Este ataque de DoS es el primer paso de un ataque más complejo que trata de secuestrar uno de los IED de la subestación (el próximo grupo de desafíos).

- *Objetivo:* Los participantes deben: detectar el tipo de ataque que está ocurriendo; determinar las direcciones IP, direcciones MAC y puertos atacados; identificar los protocolos utilizados por el SCADA para determinar si el IED está nuevamente en funcionamiento; y, finalmente, decidir cuáles son las acciones que deben ponerse en práctica para detener el ataque.
- *Instructor:* El instructor ejecuta el ataque TCP SYN Flood contra el IED vulnerable (ataque con información privilegiada desde dentro de la subestación).

D. Grupo de desafíos 4: Spoofing

En muchas ocasiones, los atacantes aprovechan el hecho de que los protocolos comunes de redes de IT no han sido diseñados teniendo en cuenta la ciberseguridad. Cuando los atacantes acceden a una LAN industrial, suelen utilizar diferentes técnicas como parte de un ataque más complejo que les permita interrumpir el funcionamiento normal de la subestación.

En este grupo de desafíos, los participantes tendrán que lidiar con ataques de spoofing [26] contra los protocolos ARP e ICMP, facilitados por el ataque de DoS del grupo de desafíos anterior, de forma que un atacante se va a hacer pasar por un dispositivo lícito de la subestación mediante falsificaciones en los datos de las comunicaciones (ARP e ICMP, en este caso).

- *Objetivo:* En este grupo de desafíos, los participantes necesitan identificar las técnicas de ataque utilizadas, las direcciones IP y MAC del dispositivo de ataque, y la mejor manera de identificar y mitigar estos ataques.
- *Instructor*: El instructor realizará los ataques de spoofing controlados desde un dispositivo bajo su control de la misma subred que el IED atacado en el grupo de desafíos 3. Para ello, necesita configurar las herramientas necesarias para realizar las siguientes acciones:
    - ARP Spoofing: Responder con la dirección MAC del dispositivo bajo su control a las solicitudes de ARP procedentes del SCADA buscando la dirección MAC del IED atacado en el grupo de desafíos anterior.
    - ICMP Spoofing: Responder con la dirección MAC del dispositivo bajo su control a los mensajes ICMP cuyo destino IP es el IED atacado.

E. Grupo de desafíos 5: IEC 61850

En este último grupo de desafíos, los participantes deben lidiar con las particularidades del protocolo industrial IEC 61850 [22]. Este desafío se basa en que un operador de la subestación ha enviado una orden desde el SCADA a un IED

solicitando la apertura del interruptor y deje de pasar tensión para que se realicen trabajos mantenimiento en la línea; sin embargo, el personal de mantenimiento se ha dado cuenta de que el IED se encuentra en un estado diferente del que se muestra en el SCADA y la línea tiene tensión.

- *Objetivo:* Los participantes deben identificar, a través del tráfico de red, el mensaje de comando IEC 61850 y analizarlo para identificar las direcciones IP y MAC del remitente y del receptor del comando y entender cuál es el problema que existe con el comando (la MAC del receptor es la MAC del atacante y no la misma MAC del IED destinatario de la orden que se obtuvo en el primer grupo de desafíos). Al igual que en los desafíos anteriores, deben identificar la mejor acción a realizar para detener el ataque.
- *Instructor:* El instructor debe ejecutar una orden de apertura operando el SCADA subestación, que dará como resultado un comando IEC 61850 enviado al IED comprometido en el grupo de desafíos 3 solicitando la apertura del interruptor que controla. Sin embargo, este mensaje de comando llegará al atacante en lugar del IED legítimo gracias a los ataques de spoofing del grupo de desafíos 4. El mensaje será respondido por un segundo software que estará en ejecución en el equipo atacante usado por el instructor. Este software es un emulador que ha sido específicamente creado para este escenario e implementa el protocolo IEC 61850, simulando el comportamiento de IED legitimo secuestrado.

V. Ejecución del cyber-range avanzado

El 21 de enero de 2022 tuvo lugar la primera sesión oficial de cyber-range avanzado en el sector de la energía con 20 estudiantes (agrupados en parejas) de la universidad ENSEIRB MATMECA de Burdeos, Francia. Sin embargo, se realizaron dos eventos de validación previos, tal y como se muestra en la Fig. 4.

- Un evento de validación interna entre los propios equipos de desarrollo de Beware (Cybex) y de Tecnalia (laboratorio del sector energético) para verificar técnicamente el correcto funcionamiento del ejercicio de cyber-range avanzado (cadena de ataques, datos recogidos, verificación del tráfico de red, cuadros de mando, determinar el tiempo que tardan los logs en mostrarse en Cybex desde su generación en el laboratorio de Tecnalia, etc.).
- Una prueba piloto de validación con trabajadores de ambas empresas (Beware y Tecnalia) con conocimientos limitados sobre el ejercicio, para validar todo el contexto (requisitos, contextualización, explicaciones para los participantes, refinar la dificultad de cada desafío y afinación sus puntos, tiempo para cada desafío, pistas, duración total, etc.).

A. *Prueba piloto*

La prueba piloto de validación se programó con el fin de validar la correcta ejecución del ejercicio de cyber-range avanzado. Algunos participantes accedieron a distancia, mientras que otros estuvieron físicamente en la misma sala que los instructores. Se creó un canal de "Microsoft Teams" para comunicarse con los participantes remotos.

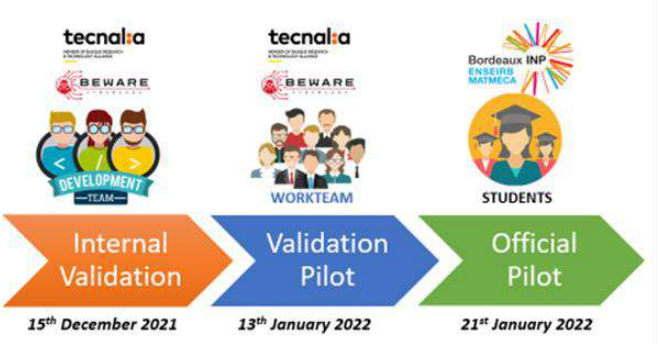


Fig. 4 Planificación de las ejecuciones del cyber-range

Las primeras horas se dedicaron a la instalación y configuración de las herramientas necesarias para acceder al cyber-range. Una vez que los participantes tenían sus sistemas y conexiones preparadas, comenzaron con la resolución de los 5 grupos de desafíos. Durante la ejecución, los participantes tuvieron varios problemas para entender correctamente algunos de los desafíos y cómo utilizar la herramienta Cybex; los instructores trataron de ayudarles, tanto con explicaciones sobre las particularidades del sector energético como con detalles adicionales útiles para resolver los desafíos. Estas explicaciones extra se incluirían posteriormente en la sesión oficial de cyber-range.

Durante esta prueba, se detectaron algunos errores menores en la validación de las respuestas de los usuarios que, aun siendo correctas, no supusieron la concesión de puntos.

Todos los desafíos, a excepción del primero, requieren de un ataque previo que provoque un cambio de estado en la subestación. Durante la prueba piloto de validación, los ataques se realizaron en función de las demandas de los participantes para identificar el tiempo requerido para cada uno de ellos y poder "programar" los diferentes ataques/desafíos en la sesión oficial.

Después de unas 5 horas, la mayoría de los participantes habían finalizado con éxito el ejercicio de cyber-range avanzado en el sector de la energía. Sin embargo, como se puede ver en la Fig. 5 donde se muestra la tabla de puntuación final de los participantes recogida en Cybex, ningún participante pudo alcanzar la puntuación máxima del ejercicio (550 puntos) debido a los errores mencionados sobre la validación de las respuestas que fueron solventados para la ejecución de la sesión oficial.

Fig. 5 Resultados de la prueba piloto de validación

B. *Sesión oficial*

Se programó una "sesión preliminar in situ" con los estudiantes, dos días antes del piloto propiamente dicho (19 de enero de 2022), y uno de los instructores con el objetivo de verificar el correcto acceso al ejercicio de todos ellos y evitar dedicar tiempo de la sesión oficial para la instalación y configuración de las herramientas necesarias como sucedió en el piloto de validación. Esta sesión previa además se usó para dar una formación básica a los alumnos sobre la herramienta Cybex y la herramienta de monitorización de los logs que

utilizarían durante la sesión oficial. Además, se contextualizó ligeramente el sector energético, para que en la sesión oficial ya tuvieran un conocimiento previo de lo que podían esperar del cyber-range avanzado en el sector energético.

En la sesión oficial, hubo un instructor in situ con los alumnos, mientras que otros dos se conectaron a distancia. Por esta razón, al igual que en la prueba piloto, se creó un canal de Teams para la sincronización y la resolución de problemas. En base a la experiencia de la prueba piloto de validación, se definieron franjas horarias (se puede filtrar el tráfico de red en base a una fecha y hora de inicio y otra de fin) para cada grupo de desafíos: dos horas, una hora, una hora, media hora y media hora, por orden de ejecución. Por ello, los instructores ejecutaron automáticamente los ataques requeridos para cada grupo de desafíos cuando se alcanzaba la franja horaria correspondiente.

- Grupo de desafíos 1 (entender los datos de log): En este caso solo se requiere la actividad "normal" de la subestación, por lo que aún no se ha ejecuta ningún ataque. La Fig. 6 muestra la actividad normal en el SCADA.

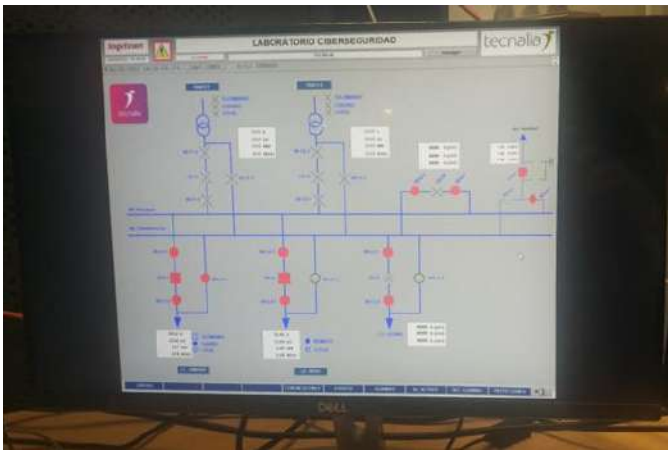


Fig. 6 Pantalla del SCADA mostrando actividad “normal”

- Grupo de desafíos 2 (detección y protección contra ataques de fuerza bruta): Estos desafíos requieren que se produzca un ataque de fuerza bruta en la subestación. La Fig. 7 muestra la ejecución del ataque en el que se realizan varios intentos de autenticación antes de acceder correctamente al sistema.

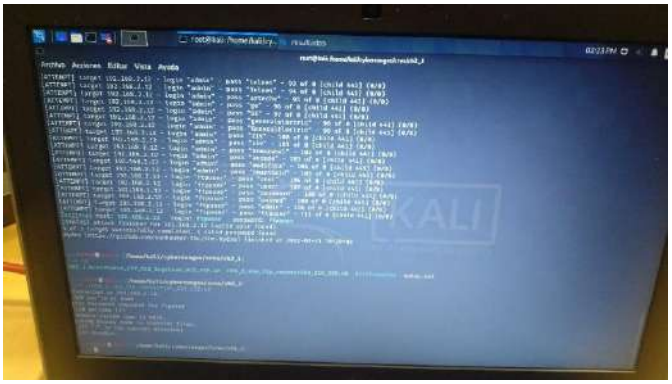


Fig. 7 Ataque de fuerza bruta

- Grupo de desafíos 3 (ataque de denegación de servicio): Estos desafíos requieren que se produzca un ataque de denegación de servicio (DoS) en la subestación. La Fig. 8 muestra en la pantalla del SCADA que el IED está caído mientras el ataque DoS está activo.

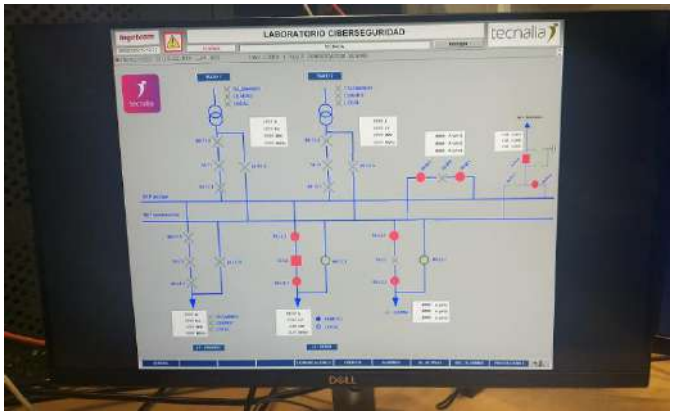


Fig. 8 Ataque de denegación de servicio

- Grupo de desafíos 4 (spoofing, suplantación): Estos desafíos requieren que un dispositivo atacante conectado a la red del IED caído responda a mensajes ARP e ICMP, lo que da lugar a ataques de spoofing ARP e ICMP. La Fig. 9 muestra el equipo atacante, con el ataque DoS activo, el emulador del IED en ejecución y los ataques de suplantación ARP e ICMP para recibir el tráfico 61850 del SCADA.

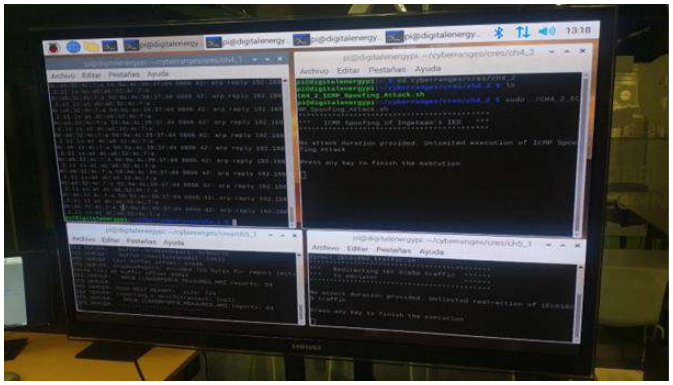


Fig. 9 IED falso ejecutando ataques de ARP e ICMP spoofing

- Grupo de desafíos 5 (IEC 61850): La Fig. 10 muestra el SCADA de la subestación en funcionamiento conectado al IED falso gracias a los ataques de los desafíos anteriores 3 y 4. En consecuencia, recibirá los comandos IEC 61850 en lugar del legítimo.

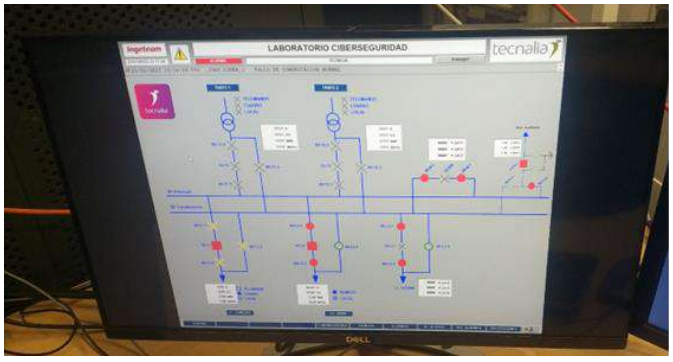


Fig. 10 Subestación en operación con un IED falso

La sesión oficial se celebró con éxito y sin grandes problemas. La Fig. 11 recoge la puntuación final de los

participantes. En este caso se observa como, a diferencia de en la prueba piloto, varios de los equipos participantes consiguieron resolver el ejercicio en su totalidad, obteniendo la puntuación máxima (550 puntos), ya que los errores de la prueba piloto fueron correctamente solventados, y la duración, la formación básica ofrecida y las pistas adecuadas han adecuadas.

Fig. 11 Resultados de la sesión oficial de cyber-range

VI. Conclusiones

Esta investigación ha propuesto y validado una nueva infraestructura de cyber-range avanzado en el sector energético, utilizando equipos reales de automatización y control industrial en operación de un laboratorio, en lugar de gemelos digitales con capacidades limitadas. Para ello, se ha conectado el laboratorio de ciberseguridad en redes eléctricas de Tecnalia a la infraestructura de cyber-range Cybex.

Además, se ha desarrollado un ejercicio de cyber-range enfocado a los operadores de compañías eléctricas y/o estudiantes especializados en ciberseguridad de cursos de educación superior haciendo uso de la nueva infraestructura. Con este ejercicio se ha demostrado que es posible conectar un cyber-range con una infraestructura real, como es el laboratorio de ciberseguridad en redes eléctricas de Tecnalia, para desarrollar escenarios de formación más realistas. Además, se podrían añadir nuevos dispositivos al laboratorio de Tecnalia que permitirían desarrollar nuevos ejercicios de cyber-range empleando la misma infraestructura para proporcionar escenarios que se adapten perfectamente a las necesidades de los usuarios finales.

Finalmente, este escenario de cyber-range podría replicarse, o podrían desarrollarse nuevos escenarios para otras infraestructuras críticas (por ejemplo, la del agua, el petróleo y el gas), así como en otros ámbitos (por ejemplo, cualquier entorno industrial) mediante la conexión de los respectivos laboratorios a la plataforma de formación Cybex.

Agradecimientos

Este trabajo ha sido realizado dentro del proyecto CRES (Cyber Range for Electrical Substations), financiado por el Consejo Regional de Nueva Aquitania (CRNA).

Referencias

[1] Borowski, P.F. Digitization, Digital Twins, Blockchain, and Industry 4.0 as Elements of Management Process in Enterprises in the Energy Sector. Energies 2021, 14, 1885. https://doi.org/10.3390/en14071885.

[2] Becerril Gil, A. A. (2021). Retos para la regulación jurídica de la Inteligencia Artificial en el ámbito de la Ciberseguridad. Revista IUS, 15(48), 9-34.

[3] Hodar, J. P. N. (2021). DESAFÍOS DE LA TECNOLOGÍA 5G EN EL ÁMBITO DE LA CIBERSEGURIDAD. Cuadernos de Difusión, (45), 79-102.

[4] Angulo, I., Modelo de concienciación y buenas prácticas en ciberseguridad para empresas del sector eléctrico, VII Congreso Smart Grids, Madrid, Diciembre 2020.

[5] Yamin, M. M., Katt, B., & Gkioulos, V. (2020). Cyber ranges and security testbeds: Scenarios, functions, tools and architecture. Computers & Security, 88, 101636.

[6] FortiGate® Rugged Serie datasheet; disponible en:https://www.fortinet.com/content/dam/fortinet/assets/data-sheets/FortiGate_Rugged_Series.pdf (Accedido: marzo 2022).

[7] Albright, D., Brannan, P., & Walrond, C. (2010). Did Stuxnet take out 1,000 centrifuges at the Natanz enrichment plant?. Institute for Science and International Security.

[8] Bronk, C., & Tikk-Ringas, E. (2013). The cyber attack on Saudi Aramco. Survival, 55(2), 81-96.

[9] Whitehead, D. E., Owens, K., Gammel, D., & Smith, J. (2017, April). Ukraine cyber-induced power outage: Analysis and practical mitigation strategies. In 2017 70th Annual Conference for Protective Relay Engineers (CPRE) (pp. 1-8). IEEE.

[10] European Cyber Security Organization (ECSO). (2020). Understanding Cyber Ranges: From Hype to Reality. Brussels, Belgium: ECS.

[11] Davis, J., & Magrath, S. (2013). A survey of cyber ranges and testbeds.

[12] Hyper-Realistic Cyber Range Attack Simulation; disponible en: https://www.cyberbit.com/ (Accedido: marzo 2022).

[13] Cyrin Cyber Range; disponible en: https://cyrin.atcorp.com/ (Accedido: marzo 2022).

[14] Cybex: experience, exercise, expertise; disponible en: https://beware-cyberlabs.eu/what-is-cybex/ (Accedido: marzo 2022).

[15] Cyber Range Hubs. Training & Development; disponible en: https://www.merit.edu/security/training/hubs/ (Accedido: marzo 2022).

[16] Tecnalia Cyber range laboratory; disponible en: https://www.tecnalia.com/en/infrastructure/cyber-range-laboratory (Accedido: marzo 2022).

[17] Gustafsson, T., & Almroth, J. (2020, November). Cyber range automation overview with a case study of CRATE. In Nordic Conference on Secure IT Systems (pp. 192-209). Springer, Cham.

[18] Jiang, H., Choi, T., & Ko, R. K. (2020, October). Pandora: A cyber range environment for the safe testing and deployment of autonomous cyber attack tools. In International Symposium on Security in Computing and Communication (pp. 1-20). Springer, Singapore.

[19] Vykopal, J., Vizváry, M., Oslejsek, R., Celeda, P., & Tovarnak, D. (2017, October). Lessons learned from complex hands-on defence exercises in a cyber range. In 2017 IEEE Frontiers in Education Conference (FIE) (pp. 1-8). IEEE.

[20] Olivares-Rojas, J. C., Reyes-Archundia, E., Gutiérrez-Gnecchi, J. A., Molina-Moreno, I., Cerda-Jacobo, J., & Méndez-Patiño, A. (2021). Towards Cybersecurity of the Smart Grid using Digital Twins. IEEE Internet Computing.

[21] Basque Digital Innovation Hub; disponible en: https://basqueindustry.spri.eus/es/ (Accedido: marzo 2022).

[22] Brunner, C. (2008, April). IEC 61850 for power system communication. In 2008 IEEE/PES Transmission and Distribution Conference and Exposition (pp. 1-6). IEEE.

[23] Najafabadi, M. M., Khoshgoftaar, T. M., Kemp, C., Seliya, N., & Zuech, R. (2014, November). Machine learning for detecting brute force attacks at the network level. In 2014 IEEE International Conference on Bioinformatics and Bioengineering (pp. 379-385). IEEE.

[24] Chao-Yang, Z. (2011, August). DOS attack analysis and study of new measures to prevent. In 2011 International Conference on Intelligence Science and Information Engineering (pp. 426-429). IEEE.

[25] Eddy, W. (2007). TCP SYN flooding attacks and common mitigations (pp. 2070-1721). RFC 4987, August.

[26] Schuckers, S. A. (2002). Spoofing and anti-spoofing measures. Information Security technical report, 7(4), 56-62.

Herramienta de generación de tráfico realista basado en comportamiento de usuario para entornos Cyber Range

Mario Sanz-Rodrigo, Manuel Álvarez-Campana, Sonia Solera-Cotanilla, Diego Rivera-Pinto, Xavier Larriva-Novo
ETSI Telecomunicación, Universidad Politécnica de Madrid
mario.sanz@upm.es, manuel.alvarez-campana@upm.es, sonia.solera@upm.es, diego.rivera@upm.es, xavier.larriva.novo@upm.es

Resumen- **La generación de tráfico realista en una red es un problema complejo, que habitualmente se aborda mediante el establecimiento previo de un modelo de tráfico. Un modelo de tráfico es una representación, lo más cercana a la realidad, del comportamiento de los paquetes y mensajes que recorren una red en un escenario concreto. Por ello, en este artículo se parte del análisis del estado del arte de todas las tecnologías que permiten la generación de tráfico, para posteriormente introducir el diseño de un sistema cuyo objetivo es la generación de tráfico realista basado en comportamiento de usuario, haciendo uso de usuarios simulados (NPC – Non-Playable Characters), totalmente automatizado y auto adaptable a escenarios virtualizados para su uso en plataformas de tipo Cyber Range utilizadas en el ámbito del entrenamiento en ciberseguridad.**

***Index Terms*- Generador de tráfico, comportamiento de usuario en red, Cyber Range, ciberseguridad, ciber ejercicios, NPC.**

Tipo de contribución: *Formación e innovación educativa en ciberseguridad*

I. Introducción

La definición de entornos simulados o virtuales para la prueba de características de una red o para el entrenamiento tienen entre sus principales necesidades la generación de topologías de red y la generación de tráfico realistas que permita asemejar el comportamiento de la red a sus contrapartidas reales.

Estos dos elementos son fundamentales para la generación de escenarios en los que se puedan realizar pruebas o ejercicios de entrenamiento en ciberseguridad sin comprometer sistemas reales, pero permitiendo una alta correlación entre el comportamiento del sistema simulado y el real.

El modelado de tráfico es a su vez un paso previo para la generación de tráfico real que pueda ser inyectado en una red (simulada, virtual o real). Para la generación de ese tráfico se pueden utilizar una serie de mecanismos y herramientas que transforman el modelo teórico en una serie de flujos o paquetes que pueden ser insertados en una red, cumpliendo con los protocolos adecuados existentes en ella. La generación de tráfico se puede llevar a cabo mediante la replicación del tráfico capturado anteriormente, con las modificaciones necesarias para adecuarlo al escenario donde se va a inyectar, o mediante la generación de tráfico sintético en base a parámetros de configuración. La generación de tráfico puede a su vez tener varios niveles, desde la construcción de cada paquete byte a byte hasta la definición de flujos o tráfico de aplicaciones de alto nivel. También es posible la definición de tráfico en base a modelos más o menos detallados, dejando la generación de los paquetes a las herramientas que permiten esta forma de funcionamiento.

El objetivo principal de esta herramienta es la generación de tráfico realista basado en comportamiento de usuario, auto configurable, modular, escalable y multiplataforma. Que sea capaz de ejecutarse en entornos Cyber Range sin necesidad de que los escenarios cuenten con acceso a Internet. A lo largo del documento se presenta un análisis de tecnologías facilitadoras, una descripción de los elementos y funcionamiento del sistema propuesto, para finalmente acabar con el apartado de evaluación de la herramienta, conclusiones y líneas futuras.

II. Análisis de Tecnologías

En este apartado se analizan los principales mecanismos y herramientas para la generación de tráfico y modelado de comportamiento utilizados en la mayoría de los sistemas de red actuales. Dado el gran número de propuestas para la generación de tráfico existentes, en este artículo se presenta una clasificación de estas en base a su funcionamiento. Para ello, se ha seguido el modelo de clasificación presentado en los trabajos de Abhishek et al. [1] y Molnar et al. [2]. En estos trabajos, se propone una clasificación de las herramientas de generación de tráfico atendiendo al modo en el que este tráfico se genera. Pese a que en este artículo se recogen herramientas software, también existen soluciones basadas en hardware, como por ejemplo SmartBits [3]. Adicionalmente, aunque existen propuestas tanto comerciales como de código abierto, la clasificación presentada se centra especialmente en las segundas, incluyendo aun así algunos ejemplos de las primeras, junto con herramientas experimentales, normalmente presentadas en artículos científicos.

Siguiendo el sistema de clasificación planteado, se tienen herramientas que se basan en replicar capturas de tráfico real previamente obtenidas mediante la monitorización de una red.

Otras generan de forma sintética los paquetes a inyectar en una red, permitiendo la construcción campo a campo o incluso byte a byte de los paquetes a enviar. Este tipo de herramientas, en general, tienen como objetivo la medición del rendimiento de las redes, buscando generar altas cargas de tráfico que permitan comprobar los límites de la red, lo cual suele contrastar con las herramientas del primer tipo, que intentan evaluar el comportamiento de la red cuando se enfrenta a tráfico real, por ejemplo, para evaluar sistemas de seguridad o para probar nuevos protocolos. Más allá de estas dos categorías, a la que pertenecen la gran mayoría de las herramientas descritas aquí, se pueden encontrar propuestas que basan la generación del tráfico en la utilización de modelos descriptivos o estadísticos del tráfico, así como aquellas que, siguiendo un esquema similar, se centran en escenarios mucho más concretos (modelado de tráfico web o de Streaming de video, por ejemplo). Finalmente, se tienen herramientas que, sin llegar a basarse en el modelado del tráfico general, se pueden considerar de alto nivel, al basarse en la generación de tráfico de aplicaciones o flujos de tráfico en lugar de centrarse en los paquetes y en la distribución de su envío.

Si bien es clara la clasificación de las herramientas en una categoría u otra, en algunos casos, las herramientas permiten la generación del tráfico en más de uno de los modos anteriormente descritos. En esos casos, se opta por clasificar la herramienta en ambos apartados.

Finalmente, se concluye este apartado con una descripción de otras herramientas que, aunque no se pueden considerar propiamente generadores de tráfico, se han utilizado como herramientas de este tipo en algunas propuestas relacionadas con entornos Cyber Range y otros entornos de pruebas.

Tabla I.
HERRAMIENTAS DE GENERACIÓN DE TRÁFICO BASADAS EN REPLICACIÓN DE TRÁFICO REAL

Herramienta	Licencia	Referencia	Última versión
TCPReplay	GLPv3	[4]	4.3.3
TCPivo	-	[5]	-
Divide & Conquer	-	[6]	-
Bit-twist	GPLv2	[7]	2.0
Ostinato	GPLv3+	[8]	1.1
RewriteCAP	Apache 2.0	[9]	1.41
Netsniff-NG	GPLv2	[10]	0.6.8
TRex	Apache 2.0	[11]	2.87
TCPopera	-	[12]	-
TCPTransform	-	[13]	-

A. Generación basada en la replicación de tráfico.

En esta categoría se incluyen aquellas herramientas y propuestas que utilizan, para la generación de los paquetes que componen el tráfico, trazas previamente capturadas por la propia herramienta o por otras herramientas de captura de tráfico. Normalmente, este tipo de software permite la modificación en mayor o menor medida del tráfico capturado, ya sea modificando campos en las cabeceras o mediante la temporización en el envío o recepción del tráfico. Las herramientas analizadas se pueden ver en la Tabla I

Tabla II.
HERRAMIENTAS DE GENERACIÓN DE TRÁFICO SINTÉTICO

Herramienta	Licencia	Referencia	Última versión
Iperf3	BSD	[14]	3.9
Iperf2	Open Source	[15]	2.1.0-rc2
BRUNO	-	[16]	-
BRUTE	GPLv2	[17]	1.14-legacy
KUTE	-	[18]	1.4
RUDE	GPLv2	[19]	0.70
Ostinato	GPLv3+	[8]	1.1
Bit-twist	GPLv2	[7]	2.0
Trafgen	GPLv2	[10]	0.6.8
PackETH	GPLv3	[20]	2.1
WARP17	BSD 3	[21]	1.7
Netperf	-	[22]	2.7.0
Uperf	GPLv3	[23]	1.07
Seagull	GPLv2	[24]	1.8.2
MoonGen	MIT	[25]	-
Cat Karat Packet Builder	Propia	[26]	1.51
Colasoft Packet Builder	Propia	[27]	2.0
Nemesis	BSD	[28]	1.4
Packet Sender	GPLv2	[29]	7.0.6
Pierf	Propia	[30]	0.137.0

B. Generación de tráfico sintético mediante la creación de paquetes.

En esta segunda categoría de herramientas se han revisado todas aquellas que permiten la construcción de paquetes de forma completa y sintética, ya sea mediante el uso de plantillas que indican la construcción de las cabeceras de cada protocolo, o mediante la manipulación byte a byte del paquete a enviar. La mayoría de estas herramientas tienen como objetivo la realización de pruebas de rendimiento, pruebas de carga en la red, etc. Es por ello por lo que muchas de estas herramientas se centran especialmente en la flexibilidad de creación de paquetes y la posibilidad de generar altos volúmenes de tráficos. Esta categoría es la más numerosa. Las herramientas analizadas se pueden ver en la Tabla II

Tabla III.
HERRAMIENTAS DE GENERACIÓN DE TRÁFICO BASADAS EN MODELOS

Herramienta	Licencia	Referencia	Última versión
Postel TG	-	[31]	-
MGEN	Open Source	[32]	5.02c
BRUTE	GPLv2	[17]	1.14-legacy

C. Generación de tráfico basada en modelos

Las herramientas de generación de tráfico basadas en modelos se diferencian de las analizadas en las secciones anteriores en que se centra específicamente en la utilización de modelos estadísticos para acercar lo más posible la distribución del tráfico al tráfico que habría en redes reales. Las herramientas analizadas se pueden ver en la Tabla III.

D. Generación de tráfico de alto nivel y auto configurable

En esta categoría se analizan herramientas que, en lugar de centrarse en la flexibilidad a la hora de generar paquetes de

Tabla IV.
HERRAMIENTAS DE GENERACIÓN DE TRÁFICO DE ALTO NIVEL Y AUTO CONFIGURABLES

Herramienta	Licencia	Referencia	Última versión
HARPOON	GPLv2	[33]	-
SWING	GPLv2	[34]	0.3.1
LiTGen	-	[35]	-
D-ITG	GPLv3	[36]	2.8.1r
TMIX	Open Source	[37]	-
Scapy	GPLv2	[38]	V2.4.4
Mausezahn	GPLv2	[10]	0.6.8
TRex	Apache 2.0	[11]	2.87

Tabla V.
HERRAMIENTAS DE GENERACIÓN DE TRÁFICO PARA ESCENARIOS ESPECÍFICOS

Herramienta	Licencia	Referencia	Última versión
EAR	GPLv3	[39]	4.3.3
ParaSynTG	-	[40]	-
Youtube Workload Generator	-	[41]	-
Graph-based generator	GPLv2	[42]	2.9
ProWGen	GPLv3+	[43]	1.1

distintos niveles de red, se centran en la generación de tráfico basado en la definición de flujos o aplicaciones, dejando que sea la propia aplicación la que genere el tráfico específico a partir de estas definiciones. Las herramientas analizadas se pueden ver en la Tabla IV.

E. Generación de tráfico para escenarios específicos

Por último, se analizan herramientas de generación de tráfico, que de forma similar a las de la categoría anterior, se centran en la definición de elementos de alto nivel para la construcción de tráfico que se asemeje lo más posible al tráfico real. En el caso de estas propuestas, se han diseñado para la generación de tráfico en un entorno muy específico, ya sean redes inalámbricas, tráfico web o Streaming de video. La mayoría de estas herramientas son experimentales. Las herramientas analizadas se pueden ver en la Tabla V.

F. Herramienta para el modelado de usuarios y tráfico realista

GHOSTS (General HOSTS) [44] es una herramienta de generación de tráfico desarrollada por el Instituto de Ingeniería de Software (SEI) de la Universidad Carnegie Mellon, basada en el modelado del comportamiento de usuarios y orientada a la simulación de ejercicios de ciberseguridad realistas.

Esta herramienta tiene como objetivo automatizar y orquestar las actividades de los llamados "personajes no jugadores" (NPC – Non-Playable Characters). Creando usuarios sintéticos en el entorno, permite la simulación avanzada de actividades de usuarios para enriquecer el realismo de los ejercicios de ciberseguridad aplicados a entornos Cyber Range. Los actores simulados con los que interactúan los participantes pueden realizar funciones como la navegación en Internet, ejecución de comandos de terminal, envío de correos, o simplemente ejecutar software que genere actividad de red.

GHOSTS permite simular la existencia de usuarios realizando tareas habituales en todas o en algunas de las máquinas que forman parte de un escenario de red de una ciber maniobra. Por ello, GHOSTS no es simplemente un mecanismo de creación de tráfico, sino que crea tráfico de red realista en forma de actividad de usuarios de una red basada en el contexto de la propia red.

Tabla VI.
FUNCIONALIDADES DISPONIBLES EN LOS AGENTES GHOSTS

Capacidad	Acción del usuario	Métodos
Navegación Web	Navegación Introducción de texto Clic en botones o enlaces	Aleatorio, específico, en bucle
Comandos en terminal	Ejecución de comandos en terminal Ejecución de comandos en PowerShell	Aleatorio, específico, en bucle
Comunicación entre usuarios NPC	Creación y gestión de correo electrónico	Específico, en bucle
Gestión de documentos	Formatos comunes para procesadores de texto. Posibilidad de creación y almacenamiento de documentos	Aleatorio, específico, en bucle

Los agentes o clientes GHOSTS son programas que se instalan y ejecutan en las máquinas virtuales presentes en el ciber escenario y que se encargan de ejecutar, conforme a la programación (en instantes aleatorios, específicos o de manera cíclica) configurada en la máquina orquestadora de tráfico, distintos tipos de aplicaciones como puede ser la navegación web, uso de hojas de cálculo, procesadores de texto, lectura y envío de mensajes, etc. En la Tabla VI se muestran las aplicaciones disponibles en los clientes GHOSTS.

El tráfico generado aplicado a escenarios Cyber Range, depende en gran medida de las aplicaciones utilizadas y de la utilización de estas por parte de los usuarios simulados. Tras analizar las aplicaciones de red utilizadas habitualmente, se obtienen cuatro modelos de tráfico básicos: elástico, interactivo, Streaming y conversacional. A continuación, se presenta una descripción de estos modelos de tráfico.

- *Tráfico de tipo elástico.* Este modelo de tráfico engloba a todo el tráfico generado por servicios de almacenamiento y envío de información. Por ejemplo, un servicio FTP, un servicio de correo electrónico, mensajería instantánea, etc. Las tasas de transferencia de este tipo de tráfico son aleatorias, y tiene una serie de requisitos de calidad como puede ser la minimización del retardo y el jitter. Además, en este tipo de tráfico no se presenta la pérdida de datos y se requiere un alto ancho de banda.

- *Tráfico de tipo interactivo*. Este tipo de tráfico engloba a servicios de tipo petición y respuesta. Típicamente, tráfico relacionado con la navegación web, el intercambio interactivo de mensajes, etc. Este tipo de tráfico es especialmente sensible a retardos y pérdidas y

no requiere tanto ancho de banda.

- *Tráfico de tipo Streaming*. Tráfico principalmente unidireccional que utiliza la red de forma continua y constante, formando un flujo de información entre dos entidades. Se trata habitualmente de tráfico en tiempo real donde no se requiere un mantenimiento estricto del retardo o jitter, al igual que se pueden producir pérdidas de datos. El Streaming de video es el ejemplo más característico de este tipo de tráfico.

- *Tráfico de tipo conversacional*. Se trata de tráfico generado por servicios interactivos en tiempo real, como, por ejemplo, audio y video conferencias, VoIP, juego online, etc. En general, cualquier servicio que implica la interacción entre usuarios en tiempo real. Este tipo de tráfico requiere un retardo bajo para el funcionamiento de los servicios, con una pérdida de datos lo menor posible.

III. Tras el análisis de herramientas presentado, y en base a los requisitos específicos del proyecto en el cual se enmarca este trabajo, se opta por la utilización y desarrollo de una herramienta basada en comportamiento realista de usuario. Principalmente por dos motivos, en primer lugar, debido a requisitos del sistema se descarta la inyección de pcaps estáticos, o la creación de paquetes de forma programática, ya que el sistema debe auto adaptarse tanto al escenario como a la tipología de servicios definidos en el desde el Framework COBRA y en segundo lugar, al utilizar software real para la generación de los distintos tipos de tráfico, se deja traza en el equipo host, algo que puede aportar valor en ejercicios tipo red team, ya que los equipos atacados no serán máquinas estáticas sin interacción por parte del resto de elementos del escenario.Sistema propuesto

Como se ha indicado anteriormente, el objetivo de este trabajo se basa principalmente en la emulación de tráfico generado por actividades de usuarios, sin entrar en la parte de generación de tráfico malicioso debido a que esa funcionalidad se complementa en el ecosistema general del Cyber Range con un simulador basado en Mitre Caldera, cuyo comportamiento será automatizado y generará de forma indirecta el tráfico que se va a inyectar en la red del ciber ejercicio. El sistema de generación de tráfico, mostrado en la Figura 1, es totalmente integrable en cualquier tipo de escenario virtualizado, componiéndose de dos elementos principales, clientes GHOSTS presentes en las máquinas virtuales del escenario, del cual solo se utiliza la funcionalidad de ejecución de comandos y acciones, y una máquina orquestadora que se es la encargada de analizar la topología de red del escenario en concreto, generar los ficheros de comportamiento en base a la tipología de escenario y orquestar las acciones a realizar por los usuarios NPC. A continuación, se describen los elementos del sistema en profundidad.

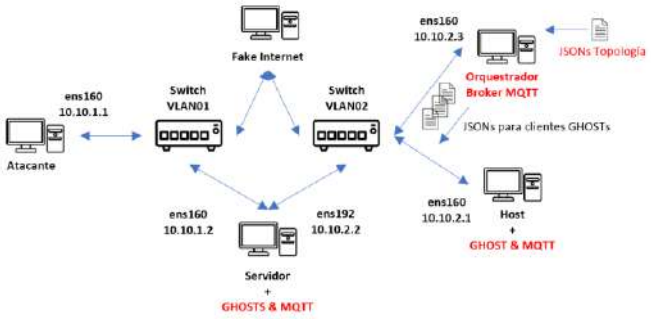


Fig. 1 Sistema generación de tráfico en escenario virtual de ejemplo

A. *Clientes GHOSTS*

Como se ha comentado anteriormente, una de las piezas más importantes de este sistema, aplicado a entornos Cyber Range, son los clientes GHOSTs. Estos clientes han de estar presentes en cada una de las distintas máquinas virtuales desplegadas en el escenario, ya que serán los encargados de realizar las tareas asignadas en los modelos de comportamiento en nombre de los usuarios NPC, y por tanto generar tráfico en la red virtual. Con el fin de realizar el sistema lo más escalable y auto configurable, se han implementado los clientes GHOSTs, tanto para Windows como para Linux, en contenedores Docker. Dichos contenedores facilitan la ejecución de GHOSTs en las máquinas finales, haciendo uso de un fichero de configuración en el cual se establecen los siguientes campos para su funcionamiento autónomo:

- **BROKER_ADDR**. Dirección IP del broker MQTT presente en la máquina orquestadora. Dicha IP está reservada en cada subred, de forma que cuando el cliente arranca, consulta su segmento de red y autoconfigura la dirección del broker con la IP reservada en ese segmento.
- **MQTT_TOPIC**. Topic al que se suscribe tras el arranque para recibir los ficheros que modelan el comportamiento NPC para esa máquina específica. Con el objetivo de ser auto configurable, el cliente consulta su dirección IP y establece el Topic siguiendo el formato preestablecido en el orquestador, MQTT/TOPIC/IP.
- **PATH_GHOSTS**. Path donde se depositarán los ficheros *timeline.json* enviados desde el orquestador, de forma que el cliente GHOSTs pueda ejecutar las acciones reflejadas en el.
- **RUN_GHOSTS**. Path donde se encuentra el binario de ejecución de GHOSTs. La ejecución de este binario está condicionada con la recepción del fichero *timeline.json*. En el primer arranque, el cliente se mantiene suscrito vía MQTT al orquestador a la espera de recibir el fichero de modelado de comportamiento y ejecutar GHOSTs para realizar las acciones NPC.

B. *Protocolo MQTT*

Con el objetivo de comunicar las distintas máquinas virtuales que albergarán los clientes GHOSTS NPC con la máquina orquestadora, se hace uso del protocolo de mensajería Message Queue Telemetry Transport (MQTT) [45] basado en publicación/suscripción. Mediante el uso de este protocolo, es posible realizar una diferenciación en

los flujos de mensajes, de manera que el orquestador sea capaz de mandar de manera unívoca los ficheros de comportamiento específicos a cada una de las máquinas presentes en el escenario. Con el fin de que el sistema sea auto configurable, se establece una política de asignación de topics basados en las direcciones IP asignadas a cada una de las máquinas, dejando la dirección IP del orquestador, como dirección reservada dentro del rango de cada VLAN. A través de la publicación de mensajes el orquestador es capaz de enviar los ficheros de comportamiento para iniciar los flujos de tráfico, o mandar un mensaje broadcast a todo el escenario para parar la comunicación en un instante concreto.

C. *Fichero de topología*

Debido a la aplicación del sistema de generación de tráfico en entornos Cyber Range, se hace uso de un fichero de topología facilitado por el generador de escenarios virtualizados del Framework COBRA, en el cual se refleja en formato JSON la infraestructura de red completa a alto nivel del escenario desplegado. Este fichero contiene el identificador general del escenario, así como las diferentes VLANs presentes en el y las máquinas virtuales asociadas a cada segmento de red. A continuación, se muestran los campos relevantes contenidos en dicho fichero asociados a las máquinas virtuales para su posterior uso por parte de la lógica presente en la máquina orquestadora.

- **name_in_place**. Nombre establecido para la máquina virtual en el entorno de virtualización. En este caso concreto los escenarios podrían estar disponibles en proveedores tanto Cloud como AWS, Azure, Google Cloud, etc o en sistemas propios.
- **vm_name**. Nombre asociado a la máquina virtual como hostname.
- **type_of_scenario**. Tipo de escenario desplegado, en los desarrollos realizados se toman tres tipologías de escenarios aplicables a entornos Cyber Range, Smart home, Smart industry y Smart office.
- **type_of_OS**. Tipo de Sistema operativo instalado en la máquina virtual, Windows o Linux. Este campo condiciona los métodos de ejecución del NPC en función de la sintaxis específica para cada sistema operativo.
- **ip_dir**. Dirección IP asignada a la máquina en el escenario desplegado.
- **type_of_vm**. Identifica la tipología general de la máquina virtual desplegada, diferenciando entre máquinas servidoras o máquinas hosts. Para el sistema propuesto, la lógica de generación de tráfico solo hace uso de máquinas tipo hosts, ya que son las que podrán desplegar los clientes GHOSTS para la emulación NPC. En el caso de que la máquina sea de tipo servidor, se almacena la dirección IP para ser utilizada por el resto de las máquinas hosts que tengan acceso a él.
- **services_in_vm**. Array de servicios presentes en la máquina virtual, tanto si es máquina de tipo host o server. Estos servicios serán consultados por lógica del generador de tráfico para poder establecer los modelos de comportamiento basado en los distintos servicios disponibles.
- **accesibles_vms**. Array de máquinas a las que una máquina virtual específica es capaz de llegar. Este campo, junto con el anterior, es utilizado para la correcta formación de comandos a ejecutar por los NPC de manera que los flujos de tráfico sean realizables.

D. *Orquestador*

La principal diferencia con el sistema GHOSTs original, es que en esta herramienta, se prescinde de la comunicación con el GHOSTs API Server, generando una máquina orquestadora, la cual será la encargada de generar de forma aleatoria los ficheros de comportamiento que ejecutará el cliente GHOSTs, consiguiendo una aportación adicional al sistema, ya que esos ficheros generados por orquestador se adaptan automáticamente a los recursos y topología del escenario virtualizado en la plataforma Cyber Range.

El orquestador de tráfico se trata de una máquina virtual portable, que puede y debe ser desplegada en cada uno de los escenarios virtualizados en los cuales se quiera generar tráfico basado en el comportamiento de usuario. Esta máquina se debe bastionar correctamente para evitar que los alumnos que utilicen el Cyber Range interactúen con ella. Este elemento, aparte de contener un Broker MQTT para la organización de los flujos de mensajes hacia las distintas máquinas objetivo en el escenario, contiene dos funciones que se encargan de auto configurar y generar todo lo necesario para que comience la generación de tráfico en el escenario sin necesidad de interoperar directamente con ella.

La primera de estas funciones es una función *parser*, la cual se activa una vez el orquestador recibe el fichero de topología anteriormente descrito. Con el fin de diferenciar las distintas máquinas presentes en el escenario, inicialmente se crea un árbol de directorios en base al identificador de escenario, los identificadores de VLAN y los identificadores de máquina virtual. Seguidamente el programa comienza a analizar la información asociada a las distintas máquinas virtuales del escenario, y en base a la lógica programada comienza a generar los ficheros de comportamiento *timeline.json*. El número de ficheros a generar y los parámetros a reflejar en los distintos *timelines* están definidos en un fichero de variables tal y como se ilustra en la Figura 2.

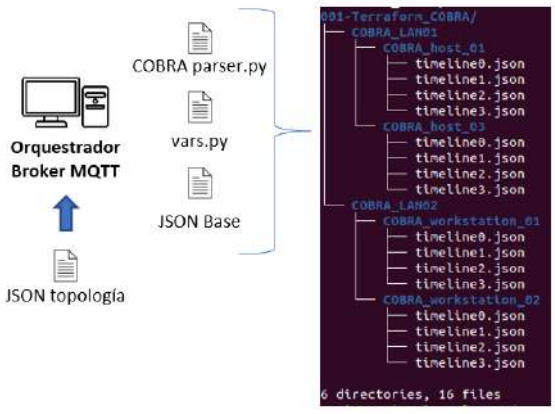


Fig. 2 Función parser del fichero de topología y creación de los ficheros de modelado de comportamiento NPC timeline

Tabla VII.
TAREAS Y MODELOS DE TRÁFICO PARA USUARIOS NPC

Tarea	Modelo de tráfico	Ejemplo	Ejemplo en GHOSTs	Porcentaje de uso
Navegación web	Tráfico interactivo	Acceso a páginas o aplicaciones web a través de navegador	"HandlerType": "BrowserFirefox", "TimeLineEvents": [{ "Command": "random", "CommandArgs": ["http://www.google.com",	25%
Correo electrónico	Tráfico elástico	Envío de mensajes a correos	"HandlerType": "Outlook", "TimeLineEvents": [{ "Command": "create",	15%
Creación y envío/recepción de documentos	Tráfico elástico	Generación de documentos ofimáticos y envío	"HandlerType": "Word", "TimeLineEvents": [{ "Command": "create",	5%
Streaming de video	Tráfico de Streaming	Presentaciones remotas, video de plataformas Streaming	"HandlerType": "BrowserChrome", "TimeLineEvents": [{ "Command": "random", "CommandArgs": "http://www.netflix.com",	10%
Llamadas de voz / video	Tráfico conversacional	Llamadas mediante VoIP, video conferencias	"HandlerType": "BrowserChrome", "TimeLineEvents": [{ "Command": "random", "CommandArgs": "https://meet.jit.si/test-room",	15%
Ejecución de comandos	Tráfico interactivo / elástico	Ejecución de comandos en terminal	"HandlerType": "Command", "TimeLineEvents": [{ "Command": "cd %homedrive %\\Downloads",	10%
Ejecución de programas de red	Tráfico elástico	Ejecución de programas que hacen uso de la red	"HandlerType": "Command", "TimeLineEvents": [{ "Command": "./dropbox",	10%
Mensajería instantánea	Tráfico interactivo	Uso de programas de mensajería	"HandlerType": "Command", "TimeLineEvents": [{ "Command": "./chat.py",	10%

Una vez termina esta primera fase, el sistema está listo para comenzar a publicar en el broker MQTT los distintos ficheros de comportamiento para que sean obtenidos por los clientes GHOSTs que están suscritos a los distintos topics generados.

La segunda función se encarga de generar un fichero JSON, tal y como se puede observar en la Figura 3, en el cual se establecen las rutas absolutas de cada uno de los ficheros de comportamiento asociados a las máquinas, así como el topic a utilizar para poder publicarlos correctamente.

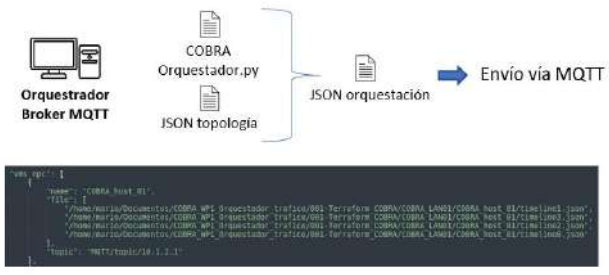


Fig. 3 Función orquestadora para el envío de los ficheros timeline a los clientes GHOSTs

Como se ha comentado anteriormente, el comportamiento a modelar, haciendo uso de GHOSTs, se basa en una serie de estructuras, contenidas en los ficheros *timeline.json*. Estos ficheros definen periodos de tiempo en los que el usuario realiza cada una de las tareas disponibles con cierta probabilidad. Estas tareas se pueden repetir o no durante el periodo de duración de estas. A partir de esto, es posible definir un modelo de usuario NPC mediante las tareas que se muestran en la Tabla VII a modo de ejemplo, concretamente GHOST trabaja con distintos *HandlerType* encargados de manejar diferentes acciones a realizar por los NPC, en ellos a su vez se pueden definir eventos en el array denominado *TimeLineEvents*. El modelo de comportamiento se compondrá de una combinación de estas tareas de forma estática o dinámica, aplicando aleatorización en el caso que se desee, en base a los servicios disponibles en la máquina que ejecutará el usuario NPC. Dicha información se obtiene desde el campo de servicios del fichero de topología anteriormente mencionado.

IV. EVALUACIÓN

Para evaluar el correcto funcionamiento de la herramienta desarrollada, tanto en la fase de despliegue, auto configuración, como en la fase de generación de tráfico por parte de los equipos host que albergan el cliente NPC, se diseña un pequeño escenario de red, tal y como se puede observar en la Figura 4. Este escenario forma parte del catálogo de escenarios manejados en la plataforma Cyber Range en la que se integra la herramienta.

El punto de partida inicial es el fichero JSON facilitado a la máquina orquestadora (PP7_Orquestador), el cual describe a alto nivel las VLANs presentes en el escenario (VLAN01 y VLAN02), así como las máquinas desplegadas, obviando la máquina orquestadora (PP7_Host, PP7_Servidor y PP7_Kali). Adicionalmente, se observan dos máquinas DHCP, que únicamente se encargan de dar direccionamiento IP a cada VLAN. En este escenario de validación, en primer lugar, tanto la máquina PP7_Host, como la máquina PP7_Servidor, tienen instalado el cliente dockerizado de GHOSTs con la lógica para conectar automáticamente al topic MQTT asignado a cada una de ellas. Como se vio en la sección anterior, se utiliza como identificador de topic la dirección IP de la máquina, evitando que el operador de la plataforma tenga que introducir esta información de manera explícita. En segundo lugar, la máquina PP7_Kali, se reserva como máquina a utilizar por alumno encargado de realizar el ciber ejercicio, por lo que es una máquina que no albergará un usuario NPC.

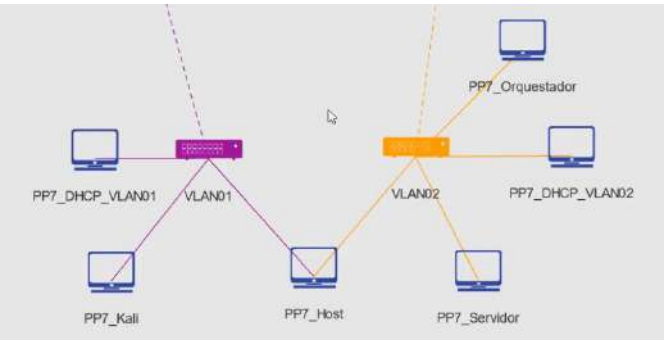


Fig. 4 Escenario desplegado para la validación del funcionamiento del sistema de generación de tráfico

Una vez desplegado el escenario, y recibido el fichero JSON por parte de la máquina PP7_Orquestadora, automáticamente comienza la generación de los ficheros de comportamiento *timeline.json* auto adaptado a las condiciones de este escenario en concreto. En caso de introducir modificaciones, o desplegar un escenario más complejo, el orquestador volvería a generar los nuevos ficheros en base a la topología y contexto asociado al escenario. Tras la generación de los ficheros que modelarán el comportamiento NPC por parte del orquestador, automáticamente se comienzan a publicar vía MQTT ficheros en los topics asignados a las máquinas objetivo. En la Figura 5 se puede observar el log de la máquina host, la cual en cuanto recibe el fichero de comportamiento *timeline.json*, arranca el servicio GHOSTs y comienza a ejecutar los comandos asociados a los distintos servicios disponibles en la máquina PP7_Hosts.

Fig. 5 Captura del comportamiento ejecutado por el cliente GHOSTs del equipo PP7_Host

Con el objetivo de verificar que se está generando correctamente tráfico entre las máquinas PP7_Host y PP7_Servidor, se habilita la opción de port-mirroring en la VLAN02, de manera que desde la máquina PP7_Orquestador, a modo de verificación, se lanza el analizador wireshark asociado a la interfaz correspondiente para capturar todo el tráfico que se está generando en la VLAN02. Como se puede ver en la Figura 6, las máquinas comienzan a generar el tráfico de forma autónoma, cambiando la tipología y frecuencia de tráfico en base a los ficheros *timeline.json* que va publicando el orquestador, y a los eventos individuales reflejados en dichos ficheros. Debido a que el proyecto en el que se enmarca aun no ha finalizado, queda pendiente la fase de validación en la cual se compararan los *pcaps* generados en los escenarios virtuales con *pcaps* obtenidos en escenarios reales para obtener la tasa de correlación entre ambos escenarios.

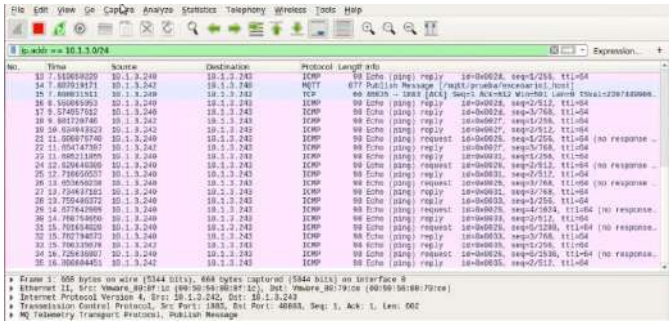


Fig. 6 Captura del tráfico generado a través del cliente GHOSTs en la máquina PP7_Host

V. CONCLUSIONES Y LÍNEAS FUTURAS

Este trabajo presenta la investigación y el despliegue de un sistema para la generación de tráfico realista basado en comportamiento de usuarios, aplicable a entornos de ciber entrenamiento, el cual permite auto adaptarse a cualquier tipología de escenario virtualizado.

Se ha tratado de desarrollar una arquitectura simple y modular que evita dependencia de una gestión externa para su correcto funcionamiento. El sistema puede ser ampliable a través de las funciones definidas en la máquina orquestadora, permitiendo el uso de nuevos servicios, protocolos específicos, o acciones concretas con el fin de acercarse cada

vez más a los comportamientos reales de las redes que se están simulando en plataformas de tipo Cyber Range.

Como líneas futuras se propone en primer lugar, la mejora de la lógica de aleatorización encargada de la generación de los modelos de comportamiento timeline, mediante el análisis de flujos reales asociados a entornos de red similares a los desplegados en los ciber ejercicios, con el objetivo de mantener la generación por parte de usuarios NPC capaces de dotar de realismo a este tipo de aplicaciones. Por otra parte, se plantea la integración en el orquestador de algunas de las herramientas *open source* analizadas en el estado del arte, las cuales podrían complementar el flujo final de tráfico generado en el escenario en base a casos de entrenamiento más específicos. Adicionalmente, de cara a la fase final del proyecto en el cual se enmarca esta herramienta, se propone la utilización de sistemas UBA (User Behavior Analytics) con el fin de obtener datasets que permitan mejorar los modelos de comportamiento generados por la máquina orquestador. Por último, se contempla la integración en un único sistema del elemento mencionado al inicio de la sección III, encargado de la simulación de APT (Advanced Persistent Threat) para poder controlar tanto la generación de tráfico malicioso como tráfico legítimo desde una única herramienta.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto COBRA (10032/20/0035/00) del programa COINCIDENTE 2019 trabajando en colaboración con el MCCE, DGAM/SDG. PLATIN. Este proyecto ha sido concedido por el Ministerio de Defensa, así como por las ayudas FJCI-2017-34926 y RYC2015-18210, concedidas por el Gobierno de España y con-financiadas por el Fondo Social Europeo.

REFERENCIAS

[1] M. A. G. Patil, A. R. Surve, A. K. Gupta, A. Sharma, y S. Anmulwar, «Survey of synthetic traffic generators», presentado en Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, ago. 2016. doi: 10.1109/INVENTIVE.2016.7823282.

[2] S. Molnar, P. Megyesi, y G. Szabo, «How to validate traffic generators?», en *2013 IEEE International Conference on Communications Workshops (ICC)*, Budapest, Hungary, jun. 2013, pp. 1340-1344. doi: 10.1109/ICCW.2013.6649445.

[3] «Spirent SmartBits», *Data Edge*. https://dataedge.ie/product/spirent-smartbits/ (accedido feb. 08, 2021).

[4] F. Klassen, «Tcpreplay - Pcap editing and replaying utilities». https://tcpreplay.appneta.com/ (accedido feb. 08, 2021).

[5] W. Feng, A. Goel, A. Bezzaz, W. Feng, y J. Walpole, «TCPivo: a high-performance packet replay engine», en *Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*, New York, NY, USA, ago. 2003, pp. 57-64. doi: 10.1145/944773.944783.

[6] T. Ye, D. Veitch, G. Iannaccone, y S. Bhattacharya, «Divide and conquer: PC-based packet trace replay at OC-48 speeds», en *First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities*, feb. 2005, pp. 262-271. doi: 10.1109/TRIDNT.2005.18.

[7] A. Yeow Chin Heng, «Bit-Twist: Libpcap-based Ethernet packet generator». http://bittwist.sourceforge.net/index.html (accedido feb. 08, 2021).

[8] B. R. Patil, M. Moharir, P. K. Mohanty, G. Shobha, y S. Sajeev, «Ostinato - A Powerful Traffic Generator», en *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, dic. 2017, pp. 1-5. doi: 10.1109/CSITSS.2017.8447596.

[9] «rewritecap · pkg.go.dev». https://pkg.go.dev/github.com/jordan2175/rewritecap (accedido feb. 08, 2021).

[10] «netsniff-ng toolkit». http://netsniff-ng.org/ (accedido feb. 08, 2021).

[11] «TRex». https://trex-tgn.cisco.com/ (accedido feb. 08, 2021).

[12] S.-S. Hong y S. F. Wu, «On Interactive Internet Traffic Replay», en *Recent Advances in Intrusion Detection*, Berlin, Heidelberg, 2006, pp. 247-264. doi: 10.1007/11663812_13.

[13] S.-S. Hong *et al.*, «TCPtransform: Property-Oriented TCP Traffic Transformation», en *Detection of Intrusions and Malware, and Vulnerability Assessment*, Berlin, Heidelberg, 2005, pp. 222-240. doi: 10.1007/11506881_14.

[14] «iPerf - The TCP, UDP and SCTP network bandwidth measurement tool». https://iperf.fr/ (accedido feb. 08, 2021).

[15] «IPerf2», *SourceForge*. https://sourceforge.net/projects/iperf2/ (accedido feb. 08, 2021).

[16] G. Antichi, A. D. Pietro, D. Ficara, S. Giordano, G. Procissi, y F. Vitucci, «BRUNO: A high performance traffic generator for network processor», en *2008 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, jun. 2008, pp. 526-533.

[17] B. Nicola, G. Stefano, G. Procissi, y S. Raffaello, «BRUTE: A High Performance and Extensibile Traffic Generator», *undefined*, 2005. /paper/BRUTE%3A-A-High-Performance-and-Extensibile-Traffic-Nicola-Stefano/3b45e4d64e356820064247b6a356963366edf160 (accedido feb. 08, 2021).

[18] S. Zander, D. Kennedy, y G. Armitage, «KUTE A high performance Kernel-based UDP traffic engine», Swinburne University of Technology. Centre for Advanced Internet Architectures, Melbourne, VIC, Report, 2005. Accedido: feb. 09, 2021. [En línea]. Disponible en: https://researchrepository.murdoch.edu.au/id/eprint/36419/

[19] J. Laine, S. Saaristo, y R. Prior, «(C)RUDE - RUDE & CRUDE». http://rude.sourceforge.net/ (accedido feb. 09, 2021).

[20] «PACKEth». http://packeth.sourceforge.net/packeth/Home.html (accedido feb. 09, 2021).

[21] «WARP17». http://warp17.net/ (accedido feb. 09, 2021).

[22] «The Netperf Homepage». https://hewlettpackard.github.io/netperf/ (accedido feb. 09, 2021).

[23] «Uperf - A network performance tool». http://uperf.org/ (accedido feb. 09, 2021).

[24] «Seagull - Open Source tool for IMS testing». Hewlett-Packard, 2007. [En línea]. Disponible en: http://gull.sourceforge.net/doc/WP_Seagull_Open_Source_tool_for_IMS_testing.pdf

[25] P. Emmerich, S. Gallenmüller, D. Raumer, F. Wohlfart, y G. Carle, «MoonGen: A Scriptable High-Speed Packet Generator», en *Proceedings of the 2015 Internet Measurement Conference*, New York, NY, USA, oct. 2015, pp. 275-287. doi: 10.1145/2815675.2815692.

[26] «Cat Karat Packet Builder - reliable tool for reliable tests». https://packetbuilder.net/ (accedido feb. 09, 2021).

[27] «Packet Builder for Network Engineer - Colasoft». https://www.colasoft.com/packet_builder/ (accedido feb. 09, 2021).

[28] J. Nathan, «Nemesis - Packet injection tool suite». http://nemesis.sourceforge.net/ (accedido feb. 09, 2021).

[29] D. Nagle, «Packet Sender - Free utility to for sending / receiving of network packets. TCP, UDP, SSL.» https://PacketSender.com/ (accedido feb. 09, 2021).

[30] P. Blommaert, «Pierf packet generator/analyser». http://pierf.sourceforge.net/ (accedido feb. 09, 2021).

[31] P. E. McKenney, D. Y. Lee, y Denny, Barbara A., «Traffic Generator Software Release Notes». SRI International and USC/ISI Postel Center for Experimental Networking, 2002. [En línea]. Disponible en: https://www.postel.org/tg/tg2002.pdf

[32] «MGEN User's and Reference Guide». http://cpham.perso.univ-pau.fr/ENSEIGNEMENT/QOS/mgen.html (accedido feb. 09, 2021).

[33] J. Sommers, H. Kim, y P. Barford, «Harpoon: a flow-level traffic generator for router and network tests», *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 32, n.º 1, p. 392, jun. 2004, doi: 10.1145/1012888.1005733.

[34] K. V. Vishwanath y A. Vahdat, «Swing: Realistic and Responsive Network Traffic Generation», *IEEEACM Trans. Netw.*, vol. 17, n.º 3, pp. 712-725, jun. 2009, doi: 10.1109/TNET.2009.2020830.

[35] C. Rolland, J. Ridoux, y B. Baynat, «LiTGen, a Lightweight Traffic Generator: Application to P2P and Mail Wireless Traffic», en *Passive and Active Network Measurement*, vol. 4427, S. Uhlig, K. Papagiannaki, y O.

Bonaventure, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 52-62. doi: 10.1007/978-3-540-71617-4_6.

[36] S. Avallone, S. Guadagno, D. Emma, A. Pescape, y G. Ventre, «D-ITG distributed Internet traffic generator», en *First International Conference on the Quantitative Evaluation of Systems, 2004. QEST 2004. Proceedings.*, sep. 2004, pp. 316-317. doi: 10.1109/QEST.2004.1348045.

[37] M. C. Weigle, P. Adurthi, F. Hernández-Campos, K. Jeffay, y F. D. Smith, «Tmix: a tool for generating realistic TCP application workloads in ns-2», *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, n.º 3, pp. 65-76, jul. 2006, doi: 10.1145/1140086.1140094.

[38] P. Biondi, «Scapy», *Scapy*. https://secdev.github.io/ (accedido feb. 09, 2021).

[39] C. Ku, Y. Lin, Y. Lai, P. Li, y K. C. Lin, «Real traffic replay over WLAN with environment emulation», en *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, abr. 2012, pp. 2406-2411. doi: 10.1109/WCNC.2012.6214199.

[40] R. E. A. Khayari, M. Rucker, A. Lehmann, y A. Musovic, «ParaSynTG: A parameterized synthetic trace generator for representation of WWW traffic», en *2008 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, jun. 2008, pp. 317-323.

[41] A. Abhari y M. Soraya, «Workload generation for YouTube», *Multimed. Tools Appl.*, vol. 46, n.º 1, p. 91, jun. 2009, doi: 10.1007/s11042-009-0309-5.

[42] P. Siska, M. Ph. Stoecklin, A. Kind, y T. Braun, «A flow trace generator using graph-based traffic classification techniques», en *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, New York, NY, USA, jun. 2010, pp. 457-462. doi: 10.1145/1815396.1815503.

[43] M. Busari y C. Williamson, «Prowgen: A synthetic workload generation tool for simulation evaluation of web proxy caches», *Comput. Netw.*, vol. 38, pp. 779-794, 2002.

[44] D. Updyke, G. Dobson, T. Podnar, L. Osterritter, B. Earl, y A. Cerini, *GHOSTS in the Machine: A Framework for Cyber-Warfare Exercise NPC Simulation*. 2018.

[45]. Karagiannis, V.; Chatzimisios, P.; Vázquez-Gallego, F.; Alonso-Zarate, J. A Survey on Application Layer Protocols for the Internet of Things. Trans. IoT Cloud Comput. 2015, 1, 11–17.

Sesión III: Vulnerabilidades y ciber amenazas

Estrategias de inmunización selectiva para la mitigación del movimiento lateral

David Herranz Oliveros
Universidad de Alcalá,
28805 Alcalá de Henares, España,
david.herranz@edu.uah.es

Iván Marsá Maestre
Universidad de Alcalá,
28805 Alcalá de Henares, España,
ivan.marsa@uah.es

José Manuel Giménez Guzmán
Universitat Politècnica de València,
46022 Valencia, España,
jmgimenez@upv.es

***Resumen*—Esta investigación tiene por objetivo el análisis de las rutas de ataque que, en redes *Windows* administradas por Directorio Activo, permiten que actores malintencionados alcancen activos críticos mediante movimiento lateral. Modelamos la topología de las redes como grafos, considerando sus elementos como vértices, y las relaciones de confianza del Directorio Activo como los enlaces que los interconectan. Emplearemos un modelo epidemiológico para trazar el movimiento lateral como una sucesión de saltos a través del grafo que representarán el avance de un atacante desde el exterior hacia los elementos críticos de la red. El objetivo es identificar las áreas más estratégicas donde, de aplicar salvaguardas, se minimice el impacto de una intrusión, facilitando la priorización y evaluación de esas salvaguardas de manera más precisa. Este análisis será de aplicación preventivamente a nivel de diseño de la red, y como mejora futura de manera reactiva cuando sea detectada una intrusión en curso.**

***Index Terms*—Directorio Activo, grafos, movimiento lateral, resiliencia, modelos epidemiológicos, centralidad**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Los ciberataques sobre infraestructuras TI que manejan información crítica han proliferado mucho durante los últimos años, dado que dichas infraestructuras están más interconectadas que nunca [1]. Además, la heterogeneidad de esas redes también está aumentando radicalmente. Por ello a día de hoy podemos encontrar complejos ecosistemas *Internet of Everything* (IoE) [2] compuestos de ordenadores portátiles, *smartphones*, sensores conectados y demás equipamiento informático que bien acceden o generan datos críticos. Esta heterogénea combinación de dispositivos junto con la deseada ubicuidad y alta conectividad del entorno en que se encuentran, supone un complejo problema desde el punto de vista de la ciberseguridad. Por ello, nuestra investigación se centra en obtener nuevas morfologías de red que mejoren la seguridad y resiliencia mediante la búsqueda de nuevas técnicas de mitigación del riesgo tanto preventiva como reactivamente.

Caracterizaremos las redes como grafos dirigidos donde los elementos que los componen (usuarios, ordenadores, dominios, grupos de seguridad, etc...) serán sus vértices, y las relaciones de confianza, es decir, las relaciones administrativas y de pertenencia a grupos de permisos comunes en el entorno de Directorio Activo, las aristas que unan esos vértices entre sí. La amenaza a mitigar nacerá de la aparición de rutas de ataque desde los elementos más expuestos hacia los activos críticos de la red siguiendo los caminos trazados a través de esas aristas [3].

Modelaremos así un *snowball attack* [4], [5] donde el vector de ataque potencial será la explotación de diversas brechas de seguridad en cuanto a la gestión y conservación de credenciales de equipos y usuarios tanto de manera local como remota. Esto permitirá a los atacantes, una vez comprometido un primer elemento de la red, realizar escaladas de autenticación encadenando credenciales de un equipo a otro con privilegios cada vez más elevados a medida que la intrusión se acerca al controlador de dominio de la red infectada.

Este artículo consta de tres partes. En la primera de ellas se detallan los principales aspectos de otros trabajos relacionados con el que nosotros venimos realizando. Posteriormente se desarrollan los fundamentos, métodos experimentales y resultados de nuestro análisis; y, finalmente, se incluyen las conclusiones extraídas hasta el momento de nuestras aportaciones sobre este campo.

II. Trabajos Relacionados

II-A. Modelado de la red como un grafo

Para poder determinar cómo ser capaces de mitigar el impacto de un ataque sobre una red orquestada por Directorio Activo, nuestro estudio se apoyará fundamentalmente en los análisis realizados en [6], donde el autor propone modelar la red como un grafo dirigido. Así, a la vista del rol que desempeña cada vértice en la conectividad del grafo, espera poder encontrar aquellos que serían de mayor utilidad a los atacantes para, una vez comprometidos, moverse lateralmente a gran escala a lo largo de la red.

En ciberseguridad, encontramos la propuesta de abstraer la información de una red modelándola como un grafo también en otras investigaciones. En [7], los autores emplean esta metodología para construir un modelo de amenazas que, basado en grafos multicapa, permita reducir el riesgo al que se exponen los activos de una red. En [8], los autores modelan como un grafo el tráfico web que tiene lugar entre las distintas entidades que componen una red para proponer (de manera preliminar) un análisis conductual que permita detectar anomalías que supongan comportamientos malintencionados.

Como se señala en [3], actualmente los atacantes tienden a concebir cada vez más la red como un grafo de ataque donde existen rutas a través de las que avanzar hacia sus objetivos. Sin embargo, defensivamente, la concepción típica de la red se acerca más a un inventario de activos a proteger, por lo que la percepción de la red como una estructura relacional modelada como un grafo será la piedra angular de nuestra ventaja frente a las amenazas. Así, para proteger un elemento de la red debemos fijarnos no sólo en él, sino en todos esos otros que, sin ser necesariamente valiosos en un principio, lo exponen directa o indirectamente.

II-B. Valores atípicos y modelos epidemiológicos

Volviendo nuevamente a lo detallado en [6], para determinar aquellos vértices o zonas de la red cuya vulneración será de interés para los atacantes en términos de movimiento lateral, se debe analizar el rol que cada vértice desempeña para con los demás. Esto se puede lograr extrayendo métricas a través del estudio de la centralidad de los vértices del grafo. Estas métricas permitirán poner en relación la posición en que cada vértice se encuentra en la red frente al poder o capacidad de influencia que tiene sobre la misma [9].

Modelando las infraestructuras TI como grafos, normalmente encontramos un componente gigante de baja conectividad conformado por los elementos situados en la zona de la red más expuesta hacia el exterior, es decir, en la periferia de dicha red (equipos de trabajo, usuarios no administrativos, etc...). Esta morfología suele dar lugar a una distribución homogénea de los valores de centralidad a lo largo de la mayoría del grafo. Sin embargo, se presta a su vez a que algunos vértices tengan una conectividad muy alta en la red, distinta del resto. Debido a estas condiciones, identificar estos vértices resulta de gran valor para poder determinar sobre qué zonas de la red hay que actuar y cómo hacerlo de cara a la mitigación del riesgo de manera eficiente. Así, en [6] se propone la identificación de aquellos vértices con valores de centralidad atípicos mediante la aplicación de algoritmos de agrupamiento basado en densidad como DBSCAN [10]. Con ello, se puede lograr una reducción sustancial del área de estudio de la red localizando las zonas de mayor influencia en lo referente a movimiento lateral.

Otra forma de reducir el área de estudio de la red para nuestro análisis será descomponiendo el grafo en capas o estratos mediante el cálculo de la métrica *K-shell*, ya empleada en epidemiología anteriormente [11]. En este trabajo los autores señalan que, dado un proceso infeccioso o de transmisión en red cualquiera, en nuestro caso de movimiento lateral, el alcance de la propagación tenderá a ser similar siempre que esta comience o transcurra comprometiendo vértices pertenecientes a la misma *K-shell*.

Se define *K-shell* como el mayor subgrafo que puede conformarse con al menos grado K en todos sus vértices. Eso permite estratificar el grafo de acuerdo con lo nuclear que es cada vértice respecto al conjunto, o lo que es lo mismo, lo alejado de la periferia que se encuentra [12]. Serán así aquellos con mayor valor de *K-shell* los localizados en los estratos más centrales y que por tanto tendrán mayor potencial como grandes transmisores de un proceso de infección o movimiento lateral.

Haremos uso, además, de modelos de contagio epidemiológico, empleados con frecuencia en ciberseguridad [13]. En nuestro caso, siguiendo el enfoque de [14], modelaremos el movimiento lateral como un proceso de infección probabilístico. Esto se hará aplicando un modelo *Susceptible-Infected* (SI) [15], donde las simulaciones comenzarán con uno o más vértices inicialmente comprometidos, y se prolongarán hasta que la infección alcance todos los vértices posibles dadas las condiciones iniciales de simulación. La aplicación de estos modelos permitirá así evaluar los beneficios de la inmunización completa o selectiva de los vértices atípicos que se logren identificar mediante las técnicas descritas anteriormente.

III. Estrategias de inmunización selectiva para la mitigación del movimiento lateral en entornos Windows orquestados por Directorio Activo

Nuestra hipótesis es que el enfoque definido en [6], basado en la identificación de vértices de centralidad atípica, puede ampliarse aprovechando otras técnicas ya empleadas en epidemiología como es el uso de *K-shell*. A continuación desarrollamos estos enfoques en sendos epígrafes.

III-A. Agrupamiento basado en densidad: DBSCAN

Procedemos a la identificación de vértices con valores de centralidad atípicos. Encontramos que en ocasiones el volumen que los grafos llegan a alcanzar puede ser muy grande. Esto puede resultar excesivo a la hora de calcular las métricas de centralidad de todos sus vértices y posteriormente ejecutar DBSCAN eficientemente para la identificación. En [6] se propone un mecanismo de síntesis previo al proceso de cálculo que puede minimizar este problema. Se define así una versión compactada del grafo de red denominada grafo de autenticación, donde pueden verse fácilmente las rutas existentes entre los diferentes elementos de la red de un tipo determinado (equipos, usuarios, etc...), omitiendo buena parte de las relaciones intermedias existentes entre ellos a la vez que se preserva su direccionalidad.

Para cada vértice del grafo de autenticación se calculan las siguientes métricas: número de descendientes, excentricidad, centralidad de intermediación, centralidad de vector propio (*eigenvector*), centralidad de grado y cercanía (*closeness*). [16].

Aplicamos el agrupamiento sobre dos grafos distintos, ambos procedentes de infraestructuras reales:

- **Grafo I**: se trata del grafo de autenticación analizado en [6], por lo que no disponemos de la versión original (sin compactar) del mismo.
- **Grafo II**: grafo de red completo recolectado mediante la herramienta BloodHound de una infraestructura real y anónima. Procedemos a compactarlo para su análisis obteniendo como resultado dos grafos de autenticación paralelos. Estos contendrán, representados a modo de vértices, los equipos y los usuarios de la red original respectivamente. En el caso del grafo "compactado por usuarios" (CPU), las dimensiones del mismo hacen imposible una aplicación eficiente de DBSCAN, por lo que la identificación mediante esta vía solo será de aplicación sobre el grafo "compactado por equipos" (CPE).

En el caso del grafo I, donde el número de enlaces respecto al total de vértices es muy elevado ($\approx$ 150 veces mayor), observamos un número muy significativo de vértices atípicos. Por otro lado, en el caso del grafo II CPE, encontramos un subconjunto atípico muy pequeño dado que hablamos de un grafo de conectividad centralizada donde el componente central es mayormente inaccesible desde la periferia de la red (véase *Tabla I*).

Aplicamos ahora un *modelo de contagio SI* sobre ambos grafos. Denominamos $\boldsymbol{\rho}$ al porcentaje inicial de vértices infectados, y $\boldsymbol{\tau}$ a la tasa de transmisión de la infección. Empleamos distintos valores aleatoriamente seleccionados en los intervalos $\rho \in [0.05, 0.35]$ y $\tau \in [0.05, 0.5]$ como parámetros de entrada durante los experimentos, obteniendo así resultados

con el suficiente grado de diversidad. El objetivo será realizar una comparativa del alcance del contagio enfrentando la situación inicial a una donde el subconjunto atípico ha sido inmunizado. Para conocer el alcance de la infección una vez se inmunizan esos vértices, debemos trabajar sobre el grafo original (sin compactar). No obstante, al no disponer por cuestiones de privacidad de la versión original del grafo I, en este caso evaluamos el resultado aplicando el modelo de contagio y simulando sobre el propio grafo compactado en lugar de hacerlo sobre el grafo de red completo. Dado que inmunizar el 100 % del subconjunto atípico no da lugar a que la infección a simular pueda sustentarse si se trabaja sobre el grafo compactado, inmunizamos tan solo 200 vértices que constituyen en torno a la mitad del subconjunto atípico identificado (aproximadamente el 6 % del total del grafo compactado), para así valorar la mitigación en el alcance del contagio que acarrea.

Contrastamos el impacto enfrentándolo a otras técnicas empleadas en epidemiología como, dado un conjunto vértices aleatoriamente seleccionados del grafo, inmunizar un vecino de cada uno de ellos [17]. Cada curva observada será resultado de la ejecución y promedio de 20 simulaciones independientes.

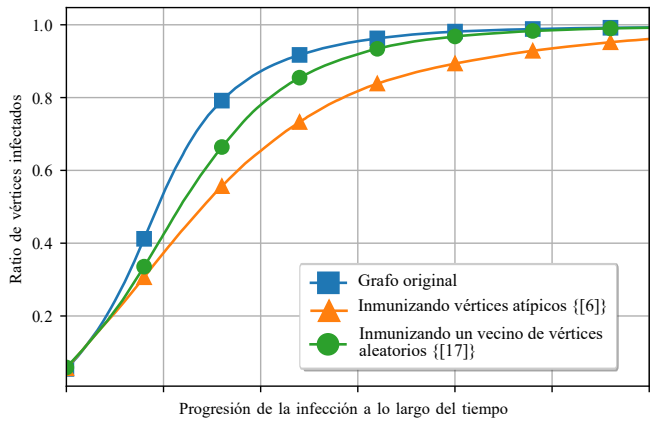


Figura 1. Resultado mostrado para ρ **= 0.05,** τ**= 0.2.** Modelo SI promedio sobre el grafo I antes y después de aplicar varias técnicas de inmunización.

Atendiendo a la *Fig. 1*, observamos una visible mitigación en el curso de la infección en caso de inmunizar vértices tanto aleatoriamente como selectivamente mediante la identificación con DBSCAN. Mediante la inmunización selectiva de esos 200 vértices del subconjunto atípico encontramos una reducción mayor a la obtenida en caso de inmunizar un vecino de un número igual de vértices aleatorios. Esto es indicador de la efectividad del proceso de detección realizado, que hacemos extensible ahora al caso del grafo II, donde sí podremos evaluar el efecto de la inmunización sobre el grafo de red original y así extraer resultados concluyentes.

III-B. Descomposición de la red: K-shell's

Recordamos que para el caso del grafo II solo hemos podido ejecutar DBSCAN sobre su versión CPE, y no hemos identificado un número suficiente de vértices cuya inmunización suponga resultados sustanciales en el desarrollo de una infección sobre el grafo original de red. Por esto, utilizamos ahora la métrica *K-shell* para localizar en el grafo de autenticación aquellos vértices con mayor conectividad y capacidad de transmisión de malware. Para ello descomponemos la red en varias capas o *K-shell's*, buscando los vértices pertenecientes a los estratos más profundos y nucleares de la misma. Típicamente encontraremos que, en redes TI, existe una primera capa superficial en los grafos que actúa a modo de componente gigante aislado, cuyos vértices no implican la formación de subgrafos de alto grado entre sí. Gracias a esto podremos aligerar en gran medida el volumen de vértices susceptibles de análisis.

Así, descomponiendo la red correspondiente al grafo II CPE, encontramos un gigantesco componente aislado correspondiente a la *6-shell* del grafo, y hasta 6 vértices concretos pertenecientes a la *10-shell* del mismo. Estos últimos conforman la parte más interna del grafo a modo de componente fuertemente conectado, dotando de conectividad a su vez a todos los demás. Este análisis se muestra consistente con lo estudiado hasta ahora puesto que estos 6 vértices coinciden con los identificados mediante el algoritmo DBSCAN anteriormente. Del mismo modo, aplicamos este análisis sobre la versión CPU del grafo II, obteniendo hasta 32 vértices que componen la *kshell* más interna del mismo (véase *Tabla I*).

Tabla I
RESUMEN DE RESULTADOS DEL PROCESO DE IDENTIFICACIÓN DE VÉRTICES ATÍPICOS MEDIANTE DBSCAN Y K-SHELL

Fuente	Número vértices	Número vértices atípicos	Relación de vértices atípicos
Grafo I	3335	353	10.58 %
Grafo II (CPE)	3236	6	0.19 %
Grafo II (CPU)	102588	32	0.03 %

Nuevamente, aplicamos un *modelo de contagio SI* sobre el grafo II siguiendo la filosofía empleada anteriormente, y analizando el efecto de la inmunización de los vértices identificados tanto únicamente mediante *K-shell*, como en conjunción con los encontrados a través de DBSCAN. En este caso las curvas se obtendrán iterando 100 veces las simulaciones, y los resultados analíticos expuestos serán el promedio a su vez de 100 casos de parametrización distinta del modelo empleado (véase *Fig. 2*).

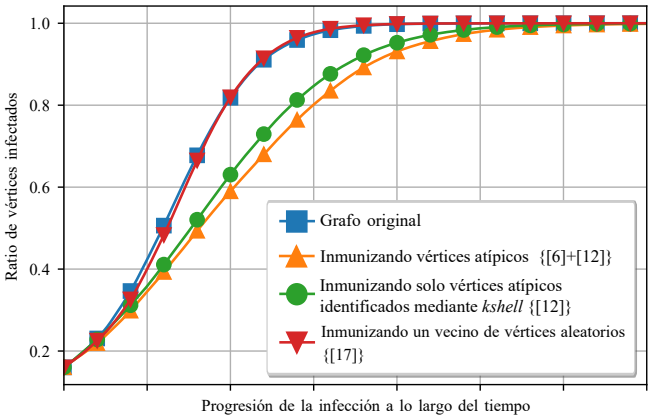


Figura 2. Resultado mostrado para ρ **= 0.16,** τ**= 0.37.** Modelo SI promedio sobre el grafo II antes y después de aplicar varias técnicas de inmunización.

Atendemos al resultado de la inmunización de un vecino de tantos vértices aleatoriamente seleccionados como los

identificados mediante DBSCAN y *K-shell* conjuntamente. Observamos que esta inmunización no tiene efecto alguno en cuanto a mitigación de la infección. Por el contrario, observamos que la sola inmunización de los vértices del grafo identificados mediante *K-shell* sí reporta notables beneficios, más aún si incluimos también los equipos identificados por DBSCAN en el subconjunto inmunizado. En este último caso hablamos de una reducción promedio del 2.826 % del área bajo la curva de infección (AUC) respecto al grafo original. De igual manera, atendiendo a la *Tabla II*, observamos una ralentización sustancial de la infección a lo largo del tiempo a medida que esta abarca el total de vértices susceptibles de ser vulnerados.

Tabla II
GRAFO II: DEMORA TEMPORAL DEL PROCESO DE INFECCIÓN DE LOS CUARTILES DEL CONJUNTO TOTAL DE VÉRTICES VULNERABLES

Fuente	25 %	50 %	75 %	100 %
Grafo original	–	–	–	–
[12]	26.11 %	24.31 %	31.13 %	34.50 %
[6] + [12]	48.35 %	41.12 %	50.83 %	35.93 %

IV. CONCLUSIONES

La concepción de la topología de las redes TI como grafos, junto con el enfoque basado en la búsqueda de rutas de ataque, son los ejes que fundamentan nuestro estudio. A lo largo de nuestro análisis proponemos diferentes maneras de reducir el área de estudio de la red que, sin perder efectividad durante el proceso, nos permitan identificar en qué puntos de la misma aplicar salvaguardas previamente a valorar cuáles deben implantarse. Encontramos que el estudio de la centralidad de los vértices del grafo tanto mediante DBSCAN como mediante *K-shell* son maneras prometedoras de identificar esas zonas vulnerables de los entornos que tratamos de proteger. Gracias a la aplicación conjunta de ambas técnicas hemos sido capaces de obtener un subconjunto de vértices muy relevantes en cuanto a conectividad en la red, y cuya inmunización afecta visiblemente al número de elementos de la misma que, en un tiempo limitado, puede llegar a alcanzar una infección en curso. A pesar de ello, existen varias mejoras en la aproximación analítica realizada pendientes de abordar.

Valoraremos el modelado de la red como un grafo dirigido donde las aristas del mismo poseerán diferentes pesos. Estos se asignarán en función de la probabilidad de éxito que un atacante tenga para la explotación de las relaciones que esas aristas representan. También ampliaremos las vías de identificación de vértices atípicos expuestas, tratando de categorizar cuantitativamente la prioridad con que cada uno de esos vértices ha de ser inmunizado. Todos estos avances se harán a su vez analizando un número mayor de infraestructuras de red de distinta naturaleza para observar como las diferencias entre ellas afectan a los resultados alcanzados.

La investigación que llevamos a cabo tiene como objetivo por tanto el ser capaces de identificar subconjuntos de vértices que, siendo lo más reducidos posibles, sean a su vez lo suficientemente amplios como para que su inmunización pueda tener una relevancia sustancial en el progreso de una infección. Esa identificación a su vez deberá ser además lo suficientemente granular como para que los vértices identificados se encuentren distribuidos a lo largo de la red. Eso permitirá obtener el mayor beneficio posible en caso de priorizar salvaguardas sobre esos puntos de la red, evitando además la identificación de elementos categorizables como críticos simplemente mediante un inventariado de activos al uso. Así, a través de un método analítico ligero y eficiente, esperamos encontrar esos puntos críticos que pueden desempeñar un papel clave para que el movimiento lateral pueda llegar a tener éxito o fracasar.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto PID2019-104855RB-I00/AEI/10.13039/501100011033, del Ministerio de Ciencia e Innovación, y por el proyecto SBPLY/19/180501/000171, de la Junta de Comunidades de Castilla-La Mancha. David Herranz también está financiado por una beca FPU del programa propio de la UAH.

REFERENCIAS

[1] S. Safavi, A. M. Meer, E. Keneth Joel Melanie, Z. Shukur: “Cyber Vulnerabilities on Smart Healthcare, Review and Solutions”, en *2018 Cyber Resilience Conference (CRC)*, pp. 1-5, 2018.

[2] D. Evans: “The internet of everything: How more relevant and valuable connections will change the world”, en *Cisco IBSG*, pp. 1-9, 2012.

[3] J. Lambert: “Defenders think in lists. Attackers think in graphs.”, 2015. [Online]. Disponible: https://github.com/JohnLaTwC/Shared/blob/master/Defendersthinkinlists.Attackersthinkingraphs.Aslongasthisistrue,attackerswin.md [Accedido: 14-Sep-2021].

[4] M. Guo, J. Li, A. Neumann, F. Neumann, H. Nguyen: “Practical Fixed-Parameter Algorithms for Defending Active Directory Style Attack Graphs”, en *arXiv preprint arXiv:2112. 13175*, 2021.

[5] J. Dunagan, A. X. Zheng, D. R. Simon: “Heat-ray: combating identity snowball attacks using machine learning, combinatorial optimization and attack graphs”, en *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pp. 305-320, 2009.

[6] B. A. Powell: “The epidemiology of lateral movement: exposures and countermeasures with network contagion models”, en *Journal of Cyber Security Technology*, vol. 4, n. 2, pp. 67-105, 2020.

[7] I. Marsa-Maestre, J. M. Gimenez-Guzman, D. Orden, E. de la Hoz, M. Klein: “REACT: Reactive resilience for critical infrastructures using graph-Coloring Techniques”, en *Journal of Network and Computer Applications*, vol. 145, pp. 102402, 2019.

[8] F. Zola, L. Segurola, J. L. Bruse,, M. G. Idoate: “Temporal graph-based approach for behavioural entity classification”, en *Investigación en ciberseguridad: Actas de las VI Jornadas Nacionales (JNIC2021 LIVE)*, vol. 34, pp.77-80, 2021.

[9] P. Bonacich: “Power and centrality: A family of measures”, en *American journal of sociology*, vol. 92, n. 5, pp. 1170-1182, 1987.

[10] M. Ester, H.-P. Kriegel, J. Sander, X. Xu et al.: “A density-based algorithm for discovering clusters in large spatial databases with noise”, en *kdd*, vol. 96, n. 34, pp. 226-231, 1996.

[11] M. Kitsak et al.: “Identification of influential spreaders in complex networks”, en *Nature physics*, vol. 6, n. 11, pp. 888-893, 2010.

[12] B. Pittel, J. Spencer, N. Wormald: “Sudden Emergence of a Giant k-Core in a Random Graph”, en *Journal of Combinatorial Theory, Series B*, vol. 67, n. 1, pp. 111-151, 1996.

[13] J. O. Kephart, S. R. White: “Directed-graph epidemiological models of computer viruses”, en *Computation: the micro and the macro view*, pp. 71-102, 1992.

[14] D. Acemoglu, A. Malekian, A. Ozdaglar: “Network security and contagion”, en *Journal of Economic Theory*, vol. 166, pp. 536-585, 2016.

[15] I. Z. Kiss, J. C. Miller, P. L. Simon, et al.: “Mathematics of epidemics on networks”, en *Cham: Springer*, vol. 598, 2017.

[16] A. Hagberg, D. Chult, P. Swart: “Exploring network structure, dynamics, and function using NetworkX”, en *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pp. 11-15, 2008.

[17] R. Cohen, S. Havlin, D. Ben-Avraham: “Efficient immunization strategies for computer networks and populations”, en *Physical review letters*, vol. 91, n. 24, pp. 247901, 2003.

Revisión sistemática de técnicas de aprendizaje automático para la detección de Amenazas Persistentes Avanzadas (APT) utilizando flujos de red en formato NetFlow

Christian Vega González
Instituto Nacional de Ciberseguridad
Avenida de José Aguado, 41; 24005 León
christian.vega@externos.incibe.es

Adrián Campazas Vega
Universidad de León
Campus de Vegazana, 24071 León
acamv@unileon.es

Ignacio Samuel Crespo Martínez
Universidad de León
Campus de Vegazana, 24071 León
icrem@unileon.es

Ángel Manuel Guerrero-Higueras
Universidad de León
Campus de Vegazana, 24071 León
agueh@unileon.es

Vicente Matellán Olivera
Universidad de León
Campus de Vegazana, 24071 León
vmato@unileon.es

***Resumen*—Las amenazas persistentes avanzadas son uno de los problemas de seguridad más complejos y preocupantes que afectan a un gran número de empresas y entidades públicas. Este tipo de amenazas emplean multitud de técnicas para atacar a objetivos de alto valor y gran parte de estas técnicas generan tráfico malicioso. Debido al notable incremento de los ataques por APT, los investigadores han desarrollado herramientas encargadas de detectar este tipo de tráfico malicioso mediante el uso de modelos de aprendizaje automático entrenados con paquetes de red. Sin embargo, existen infraestructuras de red que manejan una gran cantidad de tráfico, siendo imposible el análisis de todos los paquetes gestionados por la red. Estas infraestructuras hacen uso protocolos ligeros basadas en flujos, siendo NetFlow uno de los más utilizados. El objetivo de este trabajo es llevar a cabo una revisión de la literatura para conocer qué algoritmos de clasificación y conjuntos de datos ofrecen mejores resultados en la detección de tráfico malicioso, similares a los producidos por una APT, utilizando flujos en formato NetFlow. Los resultados obtenidos muestran que el conjunto de datos más utilizado es CIC-IDS-2017. Asimismo, los algoritmos que mayor exactitud ofrecen son regresiones logísticas, árboles de decisión y *Naive Bayes*.**

***Index Terms*—Amenaza persistente avanzada, Aprendizaje automático, Conjunto de datos, NetFlow, Revisión de literatura, Sistema de detección de intrusos.**

I. Introducción

Las amenazas persistentes avanzadas, en inglés *Advanced Persistent Threat* (en adelante APT) son uno de los retos más importantes a los que se deben enfrentar las entidades, los gobiernos y las empresas en la actualidad. Una APT se define en [1] como ataques dirigidos contra organizaciones concretas, sustentados en mecanismos muy sofisticados de ocultación, anonimato y persistencia. Los objetivos de una APT son específicos y por lo general suelen estar centrados en el robo de información. Este tipo de amenazas trata de persistir en su objetivo el mayor tiempo posible, por lo que utilizan técnicas de ocultación, lo que aumenta considerablemente la dificultad de detección de dichas amenazas. Para lograr su objetivo, una APT utiliza multitud de técnicas como escaneo de puertos, distribución de malware, movimientos laterales, escalada de privilegios, ataques de día cero o ingeniería social. La complejidad de los ataques y el sigilo empleado provocan que la detección de APTs sea un problema no resuelto en la actualidad.

Para poder detectar una amenaza de tipo APT es importante conocer su ciclo de vida. En la literatura podemos encontrar numerosos trabajos que tratan de definir dichas fases sin un criterio unificado, en este trabajo se utilizara como referencia el modelo propuesto por [2] en el cual los autores proponen las siguientes fases como el ciclo de vida de una APT:

1. **Reconocimiento y armamento**: Esta primera fase consiste en el estudio y la recopilación de información del objetivo. Con la información recopilada, los actores involucrados en una APT construyen un plan de ataque y se preparan para su ejecución. En esta fase del ciclo de vida, las APT generan tráfico malicioso tratando de obtener información de los equipos y servicios que la víctima tiene disponible en Internet.
2. **Distribución**: En esta etapa los atacantes distribuyen sus *exploits* a los objetivos. La distribución se puede llevar a cabo de forma directa o indirecta. En la distribución directa, los atacantes utilizan técnicas de ingeniería social, como el *phishing* dirigido para engañar a sus víctimas e introducir sus *exploits* en la red objetivo. La entrega indirecta consiste en comprometer a un tercero que sea de confianza de la víctima del ataque.
3. **Intrusión inicial**: La intrusión inicial ocurre cuando el atacante obtiene acceso a la computadora o la red del objetivo por primera vez. Las APT utilizan dos mecanismos diferentes. Por un lado, utilizando credenciales de un usuario legítimo que previamente habían sido robadas mediante el uso de técnicas de ingeniería social. Por otro lado, utilizando código malicioso y explotando una vulnerabilidad en la máquina del objetivo.
4. **Comando y control**: Una vez establecida una puerta trasera para poder conectarse a las máquinas infectadas,

los actores de la APT utilizan C&C, lo que les permite tener una mayor capacidad de control de la red.

5. **Movimientos laterales**: Una vez dentro de la red de la organización objetivo, la amenaza trata de expandir su control a lo largo de la organización para poder recopilar la mayor cantidad de datos posible.
6. **Exfiltración de datos**: Es el principal objetivo de una APT. Para evitar que el objetivo detecte la fuga de datos, las APT suelen cifrar el tráfico saliente utilizando protocolos seguros como TLS o aprovechar las funciones de anonimato que ofrece la Red Tor.

Uno de los métodos más utilizados en la detección de APTs es la detección del tráfico malicioso generado en las diferentes fases de su ciclo de vida. En la literatura encontramos diferentes propuestas para detectar tráfico malicioso. Entre los métodos más populares está el uso de modelos de detección basados en aprendizaje automático, específicamente modelos basados en aprendizaje supervisado. En [3], los autores obtuvieron una exactitud (accuracy) del 94,36 % utilizando el modelo *Averaged One-Dependence Estimator* (AODE), una exactitud del 92,70 % utilizando un modelo basado en redes bayesianas y una exactitud del 75,73 % utilizando un modelo basado en *Naive Bayes* (NB). Se llevó a cabo un trabajo similar en [4]. Aquí, los autores no solo se centraron en la eficacia del modelo, sino también en su eficiencia. Los autores concluyeron que AODE es el mejor algoritmo, con una exactitud del 97,26 % y un tiempo de ejecución de aproximadamente 7 segundos. Utilizando un enfoque más complejo, los autores en [5] propusieron un modelo híbrido utilizando técnicas de *bagging* y rotación de arboles, obteniendo una exactitud del 85,80 %. Finalmente en [6], los autores propusieron un método basado en características gráficas, obteniendo una exactitud del 98,54 % con el modelo *K-Nearest Neighbors* (KNN).

En los trabajos anteriores, los conjuntos de datos utilizados almacenan paquetes de red. Un paquete contienen toda la información relativa a una comunicación de red. El estudio de paquetes de red es un escenario ideal pero poco realista para redes que manejan una gran cantidad de tráfico. Este tipo de redes se ven obligadas a utilizar protocolos ligeros basados en flujos para disminuir la carga computacional de sus enrutadores a la hora de analizar el tráfico que gestiona la red.

Un flujo se define como un conjunto de paquetes que pasan por un punto de observación en la red durante un intervalo de tiempo específico. Todos los paquetes que pertenecen al mismo flujo tienen un conjunto de propiedades comunes, como la dirección IP de origen y destino y el número de puerto de origen y destino [7]. Uno de los protocolos ligeros basado en flujos más utilizado es NetFlow. NetFlow es un protocolo de red diseñado por Cisco que tiene como objetivo recopilar estadísticas del tráfico de red [8]. NetFlow recopila características como la cantidad de paquetes que forma un flujo, el tipo de protocolo IP, las banderas del protocolo y una marca de tiempo. NetFlow no almacena las cargas útiles de los paquetes, por lo que se pierde la mayor parte de la información en la comunicación pero a cambio, se reduce el coste computacional de analizarla.

En la actualidad, el aprendizaje automático se utiliza habitualmente en el desarrollo de sistemas de detección de intrusos (IDS). Estos sistemas permiten detectar ataques analizando paquetes de red para posteriormente lanzar alertas. Debido al potencial de esta tecnología, existe un gran número de investigaciones que aplican aprendizaje automático a la detección de tráfico malicioso utilizando paquetes completos de red. Por ejemplo, en [9], los autores llevaron a cabo un mapa de revisión de la literatura con el fin de establecer los conjuntos de datos y algoritmos más utilizados en la detección de ataques de red utilizando paquetes de red completos.

Sin embargo, la literatura actual no proporciona suficiente información sobre la detección de APTs mediante técnicas de ML utilizando flujos de red. La detección de tráfico malicioso en redes que utilizan flujos de red, puede ser una ruta de estudio importante con respecto a la detección de APTs, por lo que se considera útil contar con revisiones de literatura que aborden este problema. Esta razón es la principal motivación de este artículo. El estudio que se ha llevado a cabo tiene como objetivo proporcionar una visión objetiva del contexto tecnológico actual, en relación con la detección del tráfico malicioso que puede generar una APT utilizando flujos de red en formato NetFlow. Por lo tanto, este artículo pretender dar respuesta a la siguiente pregunta de investigación:

- **PI1** ¿Cuáles son las tendencias actuales tanto en el uso de algoritmos de aprendizaje automático como en conjuntos de datos para la detección de intrusos, a partir de los flujos de red maliciosos en formato NetFlow generados por una APT?

El resto del artículo está organizado de la siguiente forma: La Sección II define el procedimiento seguido para la recolección y filtrado del universo de estudio en el que se basa la investigación. En la Sección III se ha llevado a cabo la presentación de los resultados obtenidos en la búsqueda y construcción del conjunto de artículos de estudio, así como el análisis de la información obtenida de cada uno de los artículos seleccionados. Finalmente en la Sección IV se presentan las conclusiones y las aplicaciones futuras de la investigación realizada.

II. Metodología

Para la realización de esta investigación se ha seguido la metodología PRISMA [14] y las recomendaciones de Kitchenham [15]. Esta metodología se divide en cuatro fases: Planificación de la búsqueda (1); Proceso de búsqueda (2); Selección de muestras (3); Extracción de datos (4).

II-A. Planificación de la búsqueda

Tras comprobar que hasta la fecha no existe en la literatura una revisión sistemática que de respuesta a la pregunta de investigación planteada en la Sección I, se ha procedido con la búsqueda de artículos que constituirán el universo de estudio.

Las fuentes de información que se van a emplear en la investigación son: IEEE *Digital Library*, Scopus, *Science Direct* y *Web of Science*.

II-B. Proceso de búsqueda

Definidas las bases de datos en las que se realizarán las búsquedas de la investigación, se han construido las cadenas de búsqueda que permitirán obtener resultados relevantes para esta investigación. Se han elaborado dos cadenas que han sido

aplicadas en cada una de las bases de datos anteriormente citadas. Estas cadenas han sido construidas usando las palabras clave extraídas después de aplicar la estrategia PICOC [31], [32] a la pregunta de investigación planteada. Las cadenas de búsqueda seleccionadas para realizar esta investigación son las siguientes:

- **SS1** ('APT' OR '*Advanced Persistent Threat*' OR '*Network Anomaly Detection*') AND ('*dataset generation*' OR '*malicious traffic generation*' OR 'NetFlow *traffic generators*') AND ('*dataset*') AND ('*machine learning*' OR '*classification*' OR '*intrusion detection system*' OR 'IDS'))
- **SS2** ('APT' OR '*Advanced Persistent Threat*' OR '*multi-stage attack*') AND ('*machine learning*' OR '*training*') AND ('*intrusion detection system*' OR 'IDS' OR '*attack detection*') AND ('NetFlow *traffic*')

Utilizando las cadenas de búsqueda anteriores, se recopilaron los artículos objeto de estudio en este trabajo. Para ello se aplicaron dos filtros al proceso de búsqueda. Por un lado, debido a la constante evolución del campo de la ciberseguridad en general y de las APTs en particular solo se han considerado artículos publicados entre los años 2017 y 2021. Por otro lado, se han seleccionado artículos de acceso libre y de acceso privado.

II-C. Selección de muestras

Para llevar a cabo la selección de muestras, se ha realizado un proceso de filtrado. Primero, se eliminaron los artículos duplicados, eliminando 8 de los 163 artículos iniciales. A continuación, se realizó un filtrado en base a la lectura del resumen de cada uno de los artículos restantes. Para decidir si un artículo es seleccionado, se han establecido una serie de limitaciones y criterios de inclusión y exclusión. En caso de que un artículo cumpla con las limitaciones, se analizan los criterios de inclusión y exclusión. Si un artículo cumple simultáneamente un criterio de exclusión y un criterio de inclusión, se le dará más peso al criterio de exclusión. De esta forma, se eliminan aquellos artículos que cumplan al menos un criterio de exclusión o que no cumplan alguno de los criterios de inclusión. Los criterios y limitaciones aplicados son los siguientes:

- Limitaciones:
  1. Fecha de publicación de los artículos seleccionados. Únicamente se han considerado artículos comprendidos entre 01-01-2017 y 11-11-2021.
  2. Formato del tráfico de red. Únicamente se admiten artículos que utilizan flujos de red en formato NetFlow.
- Criterios de inclusión:
  - **CI1** El artículo nombra las APT (haciendo un estudio o referenciando el término).
  - **CI2** El artículo nombra el protocolo NetFlow (haciendo un estudio o referenciando el término).
  - **CI3** El artículo nombra el término IDS (realizando un estudio o referenciando el término).
  - **CI4** El artículo emplea técnicas de aprendizaje automático en el análisis de tráfico de red.
  - **CI5** El artículo trabaja con conjuntos de datos basados en flujos de red.
- Criterios de exclusión:
  - **CE1** El artículo no está escrito en inglés ni en castellano.
  - **CE2** El artículo no emplea técnicas de aprendizaje automático.
  - **CE3** El artículo emplea técnicas de aprendizaje automático, pero no pertenece al campo de la ciberseguridad.
  - **CE4** El artículo emplea técnicas de aprendizaje automático aplicadas a la ciberseguridad, pero no las aplica al análisis de tráfico de red.
  - **CE5** El artículo es una revisión de literatura.
  - **CE6** El artículo no está publicado entre los años 2017 y 2021.

Una vez aplicadas las limitactiones y los criterios de inclusión y exclusión mencionados en el apartado anterior, se ha elaborado un cuestionario el cual permite evaluar la calidad de cada uno de los artículos considerados. Este cuestionario fue construido gracias a la información recopilada tras la lectura de los artículos. El cuestionario utilizado es el siguiente:

- **P1** ¿Se trata el problema de las APT?
- **P2** ¿Se utilizan flujos de red en formato NetFlow?
- **P3** ¿Se mencionan/usan algoritmos de aprendizaje automático?
- **P4** ¿Se especifican los conjuntos de datos con los que trabajan?
- **P5** ¿Se hace uso de gráficos y/o métricas para tratar los distintos tipos de variables?
- **P6** ¿Se utilizan flujos de red para detectar intrusiones?

Las preguntas planteadas permiten seleccionar los artículos de mayor relevancia para la investigación realizada en este trabajo. Estos artículos nos permitirán conocer los algoritmos de aprendizaje automático más utilizados y que mejores resultados aportan en la detección de tráfico de red malicioso – similar al generado por una APT –, sobre conjuntos de datos que utilizan flujos de red en formato NetFlow.

Una vez definidas las preguntas del cuestionario, se han establecido los criterio de evaluación de las mismas, otorgando un valor de 1 punto por pregunta contestada con "Sí" y un valor de 0 si la pregunta se responde con "No". De esta forma, un artículo puede obtener un máximo de seis puntos. Finalmente, para asegurar la calidad de los artículos seleccionados, se ha establecido que la puntuación mínima para que un artículo sea admitido es de 4 puntos.

II-D. Extracción de datos

Una vez reducido el universo de estudio, se ha establecido un formulario de extracción de datos. Las variables que se extraerán de cada uno de los artículos se han obtenido de las preguntas de investigación planteadas en la Sección I. Los datos se extraerán de la lectura completa y detallada de cada uno de los artículos. Las variables que se desean obtener de cada artículo se han establecido respetando el objetivo de realizar una revisión sistemática y por tanto preservando la posibilidad de que esta extracción sea replicable y objetiva. De esta forma las variables que se extraerán son las siguientes:

- **V1** Conjuntos de datos utilizados y fecha de creación de los mismos.

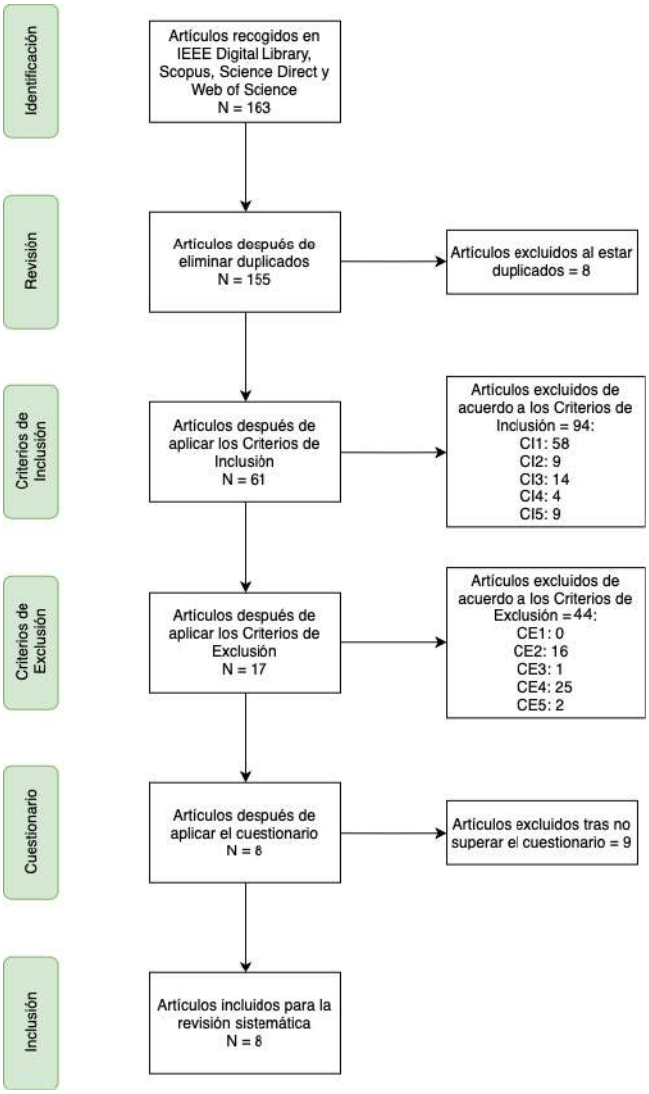


Figura 1. Diagrama PRISMA.

- **V2** Utilización de conjuntos de datos de terceros o generación propia
- **V3** Conjuntos de datos públicos o privados.
- **V4** Porcentaje de balanceo de los conjuntos de datos.
- **V4** Número de características de los conjuntos de datos.
- **V5** Exactitud del algoritmo o algoritmos utilizados.
- **V6** Tipo de algoritmo de aprendizaje utilizado (supervisado, no supervisado, semi-supervisado o reforzado).

III. Resultados

Tras aplicar las cadenas de búsqueda, se obtuvieron un total de 163 artículos los cuales se reparten de la siguiente manera: IEEE *Digital Library* (99), Scopus (27), *Science Direct* (36) y *Web of Science* (1). De los 163 artículos totales, se eliminaron 8 artículos que estaban duplicados.

Posteriormente, se aplicaron los criterios de inclusión (CI) y exclusión (CE) definidos en la Sección II. Después de aplicar estos criterios a los 155 artículos restantes, el número de artículos se redujo a 17. Este proceso de filtrado se puede observar en el diagrama PRISMA de la Fig. 1.

Aplicando el cuestionario elaborado en la sección anterior, sobre los 17 artículos seleccionados, se han rechazado 9 artículos. De esta forma, el número de artículos propuestos para su análisis es de 8 [16], [17], [18], [19], [20], [21], [22] y [23]. No habiendo alcanzado ninguno de ellos la puntuación máxima de seis puntos en el cuestionario establecido.

Una vez agrupados los 8 artículos que se desean analizar, se han extraído los datos de cada uno de ellos. Con el objetivo de dar respuesta a la pregunta de investigación planteada anteriormente, se ha puesto especial interés en los conjuntos de datos y en los algoritmos que se emplean. Por ello, los resultados que se obtengan de los algoritmos a analizar van a depender de los conjuntos de datos sobre los que se apliquen dichos algoritmos.

Tras realizar el análisis de los artículos seleccionados, se han obtenido 10 conjuntos de datos diferentes, todos ellos de distribución pública, siendo CIC-IDS-2017 el conjunto de datos más utilizado. La Tabla I muestra la fecha de creación, el número de características y la frecuencia de aparición de los conjuntos de datos en los artículos estudiados. Además, la Tabla II muestra el porcentaje de tráfico malicioso y benigno de cada uno de los conjuntos de datos.

Tabla I
Fecha, Número de características y Referencias de los conjuntos de datos considerados.

Id.	Fecha	Nº de Características	Ref.
UNSW-NB15 [26]	2015	49	[17]
Sperotto [24]	2009	10	[18] [21]
UNB ISCX [25]	2012	18	[18]
CTU-13	2011	13	[19] [21]
PBDCIC	2014	3	[20]
THPISWB	2018	3	[20]
TU intrusion dataset	2012	11	[21]
CIC-IDS-2017	2017	80	[22] [18] [17]
KDDcup99	1999	41	[23] [17]
NSL-KDD	2015	41	[16] [17]

Tabla II
Porcentaje de tráfico malicioso y tráfico benigno de los conjuntos de datos analizados.

Conjunto de datos	Tráfico normal	Tráfico malicioso	Tipo
Sperotto	41 %	44 %	Simulado
UNB ISCX	41 %	44 %	Simulado
UNSW-NB15	31,93 %	68,07 %	Simulado
CTU-13	70 %	30 %	Simulado
PBDCIC	50,75 %	49,25 %	Simulado
THPISWB	50,75 %	49,25 %	Simulado
TU dataset	43,70 %	56,30 %	Simulado
CIC-IDS-2017	41 %	44 %	Simulado
KDDCup99	65-75 %	35-25 %	Simulado
NSL-KDD	54,80 %	45,20 %	Simulado

Respecto a los algoritmos utilizados, de los 8 artículos estudiados 4 utilizan más de 1 algoritmo en sus estudios siendo todos ellos algoritmos basados en aprendizaje supervisado. La Tabla III muestra la frecuencia de aparición de cada uno de los algoritmos y la exactitud que proporcionan en función del conjunto de datos sobre el que se ha aplicado dicho algoritmo. Cabe destacar que se han omitido los algoritmos que únicamente se utilizan en un artículo.

IV. Conclusiones

Una de las técnicas más utilizadas en la detección del tráfico de red generado por una APT es el uso de modelos de

Tabla III
EXACTITUD Y FRECUENCIA DE APARICIÓN DE LOS ALGORITMOS ESTUDIADOS.

Algoritmo	Conjunto de datos	Exactitud	Frecuencia
CNN	KDDcup99	99,41 % 95,50 %	12,50 %
LR	Sperotto UNB ISCX UNSW-NB15 CIC-IDS-2017	99 % 99 % 99 % 99 %	12,50 %
J48 DT	*Public Botnet dataset* *TheHoneynet Project* NSL-KDD UNSW-NB15	98,83 % 98,83 % 96 % 98,54 % 98,54 %	50 %
NB	*Public Botnet dataset* *TheHoneynet Project* NSL-KDD UNSW-NB15 CTU-13	98,83 % 98,83 % 96 % 98,54 % 98,54 % 85,02 %	12,50 %
DNN	CIC-IDS-2017	96,20 %	12,50 %
KNN	KDDcup99 NSL-KDD UNSW-NB15	80,60 % 98.54 % 98.54 %	25 %

aprendizaje automático. En la literatura se ha demostrado la eficacia de estas técnicas cuando utilizan conjuntos de datos que contienen paquetes de red. Las redes que manejan una gran cantidad de tráfico no pueden permitirse computacionalmente analizar todos los paquetes de red que encaminan. Para evitar saturar sus enrutadores, este tipo de redes emplean protocolos ligeros basados en flujos, siendo NetFlow uno de los protocolos más utilizados.

En este trabajo se ha realizado una revisión sistemática para conocer la tendencia actual de las técnicas de aprendizaje automático en la detección del tráfico de red generado por APTs en flujos NetFlow. Los resultados obtenidos muestran que el conjunto de datos CIC-IDS-2017 es el más utilizado y el que mejores resultados ofrece a la hora de clasificar flujos de red, por lo que este conjunto de datos puede ser de utilidad en el desarrollo de un IDS para redes que manejen una gran cantidad de tráfico. Respecto a los algoritmos empleados en los artículos objeto de este estudio, se ha concluido que el mejor algoritmo es la Regresión Logística (LR), con una exactitud del 99 %. Además, cabe destacar que los algoritmos basados en Árboles de Decisión (DT) y NB también son válidos en la detección de tráfico malicioso en flujos de red, con una exactitud superior al 98 %,

Es relevante tener en cuenta que los resultados que se han detallado en este artículo únicamente han tenido en cuenta la exactitud de los algoritmos, y no se han estudiado los fenómenos de sobre ajuste y desajuste de cada uno de ellos, cuya verificación y estudio son susceptibles de futuras investigaciones.

Esta investigación pretende poner a disposición de la comunidad científica un sumario de las tendencias actuales en aprendizaje automático aplicado a la detección de tráfico malicioso en flujos de red. La naturaleza de las APT hace que sea necesario tener en cuenta muchos tipos de ataques a la hora de intentar detectar este tipo de amenazas y, por tanto, es útil conocer las herramientas y algoritmos más utilizados para detectarlos individualmente. Los resultados obtenidos durante esta investigación pueden ser un punto de partida para nuevos estudios que intenten desarrollar sistemas para detectar ataques de APTs, en redes que manejan una gran cantidad de tráfico y utilizan protocolos ligeros basados en flujos para aliviar la carga computacional de sus enrutadores.

ABREVIATURAS

En este artículo se utilizan las abreviaturas mostradas en la Tabla IV.

Tabla IV
ABREVIATURAS.

AODE	*Averaged One-Dependence Estimator*
APT	*Advanced Persistent Threat*
CE	Criterio de Exclusión
CI	Criterio de Inclusión
CNN	*Convolutional Neural Networks*
DNN	*Deep Neural Networks*
DT	*Decision Tree*
IDS	*Intrusion Detection System*
IP	*Internet Protocol*
KNN	*K-Nearest Neighbors*
LR	*Logistic Regresion*
ML	*Machine Learning*
NB	*Naive Bayes*
P	Pregunta
PBDCIC	*Public Botnet Dataset Canadian Institude Cybersecurity*
PI	Pregunta de Investigación
PLC	*Programmable Logic Controller*
SS	*Search Strings*
THPISWB	*TheHoneynet Project Involving Storm Waledac Botnets*
V	Variable

REFERENCIAS

[1] Incibe-cert.es: "Guía nacional de notificación y gestión de ciberincidentes"{Archivo PDF}. Recuperado de https://www.incibe-cert.es/sites/default/files/contenidos/guias/doc/guia_nacional_notificacion_gestion_ciberincidentes.pdf, 2020.

[2] Hutchins, E. M., Cloppert, M. J., & Amin, R. M.: " Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains", en *Leading Issues in Information Warfare & Security Research*, vol. 1, 1, pp. 80, 2011.

[3] Nawir, M., Amir, A., Lynn, O.B., Yaakob, N., Badlishah Ahmad, R.: "Performances of Machine Learning Algorithms for Binary Classification of Network Anomaly Detection System", en *Journal of Physics: Conference Series 1018*, pp. 1-8, 2018.

[4] Nawir, M., Amir, A., Yaakob, N., & Bi Lynn, O.: " Effective and efficient network anomaly detection system using machine learning algorithm", en *Bulletin of Electrical Engineering and Informatics*, vol. 8, 1, pp. 46–51, 2019.

[5] Tama, B. A., Comuzzi, M., & Rhee, K. H.: "TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System", en *IEEE Access*, vol. 7, pp. 94497–94507, 2019.

[6] Chen, S., Zuo, Z., Huang, Z. P., & Guo, X. J.: " A graphical feature generation approach for intrusion detection", en *MATEC Web of Conferences*, vol. 44, pp. 02041, 2016.

[7] Claise, B., Trammell, B., & Aitiken, P.: "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information". {Archivo PDF}. Recuperado de https://www.rfc-editor.org/rfc/pdfrfc/rfc7011.txt.pdf, 2013.

[8] Dreijer, J.: "NetFlow Anomaly Detection; finding covert channels on the network". {Archivo PDF}. Recuperado de https://rp.os3.nl/2013-2014/p74/report.pdf, 2014.

[9] Sobrín-Hidalgo, D., Campazas Vega, A., Guerrero Higueras, N. M., Rodríguez Lera, F. J., & Fernández-Llamas, C.: "Systematic Mapping of Detection Techniques for Advanced Persistent Threats", en *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*, vol. 1, 3, pp. 426–435, 2020.

[10] S. Al-Rabiaah.: ”The “Stuxnet” Virus of 2010 As an Example of A “APT” and Its “Recent” Variances”, en *2018 21st Saudi Computer Society National Computer Conference (NCC)*, pp. 1-5, 2018.
[11] Holguín, J. M., Moreno, M., Merino, B.: ”Detección de APTs”. {Archivo PDF}. Recuperado de https://www.incibe-cert.es/sites/default/files/contenidos/estudios/doc/deteccion_apt.pdf. 2013.
[12] Quittek, J., Zseby, T., Claise, B., Zander, S.: ” Requirements for IP Flow Information Export (IPFIX)”. {Archivo PDF}. Recuperado de https://www.rfc-editor.org/rfc/pdfrfc/rfc3917.txt.pdf, 2004.
[13] Luh, R., Marschalek, S., Kaiser, M., Janicke, H., & Schrittwieser, S.: ”Semantics-aware detection of targeted attacks: a survey”, en *Journal of Computer Virology and Hacking Techniques*, vol. 13, 1, pp. 47–85, 2016.
[14] Moher, D.: ”Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement”, en *Annals of Internal Medicine*, vol. 151, 4, pp. 264-269, 2009.
[15] Kitchenham,B.A.,Budgen,D.,Brereton,P.: ” Evidence-Based Software Engineering and Systematic Reviews”, vol. 4. CRC Press, 2016.
[16] Campazas-Vega, A., Crespo-Martínez, I. S., Guerrero-Higueras, Á. M., & Fernández-Llamas, C.: ”Flow-Data Gathering Using NetFlow Sensors for Fitting Malicious-Traffic Detection Models”, en *Sensors*, vol. 20, 24, pp. 1-13, 2020.
[17] Marir, N., Wang, H., Feng, G., Li, B., & Jia, M.: ”Distributed Abnormal Behavior Detection Approach Based on Deep Belief Network and Ensemble SVM Using Spark”, en *IEEE Access*, vol. 6, 59657–59671, 2018.
[18] Nkongolo, M., van Deventer, J. P., & Kasongo, S. M.: ” UGRansome1819: A Novel Dataset for Anomaly Detection and Zero-Day Threats”, en *Information*, vol. 12, 10, pp. 1-34, 2021.
[19] Radoglou-Grammatikis, P. I., & Sarigiannidis, P. G.: ”Flow anomaly based intrusion detection system for Android mobile devices”, en *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2017.
[20] Li, X. J., Ma, M., & Yen, Y. L.: ”Detecting IRC-based Botnets by Network Traffic Analysis Through Machine Learning”, en *In 2019 29th International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 1-6, 2019.
[21] Umer, M. F., Sher, M., & Bi, Y.: ”Flow-based intrusion detection: Techniques and challenges”, en *Computers & Security*, vol. 70, pp. 238–254, 2017.
[22] Gamage, S., & Samarabandu, J.: ”Deep learning methods in network intrusion detection: A survey and an objective comparison”, en *Journal of Network and Computer Applications*, vol. 169, pp. 1-21, 2020.
[23] Peng, Y.: ” Application of Convolutional Neural Network in Intrusion Detection”, en *2020 International Conference on Advance in Ambient Computing and Intelligence (ICAACI)*. pp. 1-4, 2020.
[24] Noorbehbahani, F., Fanian, A., Mousavi, R., & Hasannejad, H.: ” An incremental intrusion detection system using a new semi-supervised stream classification method”, en *International Journal of Communication Systems*, vol. 30, 4, 2015.
[25] Mighan, S. N., & Kahani, M.: ” A novel scalable intrusion detection system based on deep learning”, en *International Journal of Information Security*, vol. 20, 3, pp. 387–403, 2020.
[26] Kasongo, S. M., & Sun, Y.: ”Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset”, en *Journal of Big Data*, vol. 7, 1, 2020.
[27] Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A.: ”Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”, en *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, pp. 108–116, 2018.
[28] Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., & Garant, D.: ”Botnet detection based on traffic behavior analysis and flow intervals”, en *Computers & Security*, vol. 39, pp. 2–16, 2013.
[29] Kurniabudi, Stiawan, D., Darmawijoyo, bin Idris, M. Y., Bamhdi, A. M., & Budiarto, R.: ” CIC-IDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection”, en *IEEE Access*, vol. 8, pp. 132911–132921, 2020.
[30] Lincoln laboratory.: ”1998 DARPA Intrusion Detection Evaluation Dataset”. Recuperado de http://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset, 1998.
[31] Roehrs, A., da Costa, C. A., Righi, R. D. R., & de Oliveira, K. S. F.: ”Personal Health Records: A Systematic Literature Review”, en *Journal of Medical Internet Research*, vol. 19, 1, 2017.
[32] Schardt, C., Adams, M. B., Owens, T., Keitz, S., & Fontelo, P.: ” Utilization of the PICO framework to improve searching PubMed for clinical questions”, en *BMC Medical Informatics and Decision Making*, vol. 7, 1, 2007.

Sistema de conciencia cibersituacional y gestión dinámica de riesgos basado en ontologías

Carmen Sánchez-Zas, Víctor A. Villagrá, Mario Vega-Barbas,
Xavier Larriva-Novo, José Ignacio Moreno, Julio Berrocal
Universidad Politécnica de Madrid (UPM). DIT, ETSI Telecomunicaciones. Avda. Complutense 30, 28040 Madrid
{carmen.szas, victor.villagra, mario.vega, xavier.larriva.novo, joseignacio.moreno, julio.berrocal}@upm.es

***Resumen*—Actualmente, la gestión de riesgos es vital en el entorno de ciberseguridad para cualquier empresa, ya que están constantemente bajo amenaza. Por ello, este proceso debe ser dinámico en tiempo real, con el objetivo de definir estrategias de actuación ante ataques, y debe abarcar fuentes heterogéneas, aprovechando sistemas basados en sensores para detectar anomalías. De esta forma se puede obtener una visión conjunta del riesgo de un sistema en todo momento, mientras se trabaja con grandes volúmenes de información. En esta situación, el uso de ontologías proporciona una ventaja en cuanto a representación semántica de manera uniforme y extracción de nuevos conceptos y comportamientos. Este estudio presenta una ontología para describir distintos tipos de anomalías, partiendo de un trabajo previo de modelado de ciberamenazas, convirtiéndose en una propuesta para definir la gestión de riesgos en tiempo real en un entorno seguro convergente, mediante el uso de reglas de razonamiento.**

***Index Terms*—Ontología, Ciberseguridad, SPARQL, SPIN, OWL, Anomalía, Inteligencia de Amenazas, Gestión de Riesgo**

Tipo de contribución: *Investigación original*

I. Introducción

En los entornos corporativos es esencial conocer en todo momento el estado del sistema, especialmente en términos de riesgo. Por ello, idealmente el objetivo es una gestión dinámica del mismo, de forma que tenga la capacidad de adaptar su respuesta a los distintos datos de entrada recibidos en tiempo real.

Según la normativa ISO 31000 [1], que proporciona las guías para la gestión del riesgo en organizaciones, el objetivo principal de la evaluación de los riesgos es crear y proteger valor. Debe ser integral, personalizada, inclusiva y dinámica, y debe considerar la mejor información disponible, factores humanos y culturales y la mejora constante a través del aprendizaje y la experiencia. Siguiendo esta guía, para monitorizar el sistema, es necesario estudiar un entorno heterogéneo y obtener un resultado adaptativo dinámicamente.

El uso de ontologías se ha popularizado para administrar este tipo de tareas debido al desarrollo de la Web Semántica y los datos enlazados, para automatizar la obtención de conocimiento relacionado. De hecho, utilizando estas técnicas, se puede alcanzar una visión general de todo el conjunto automáticamente, gracias a la inferencia del conocimiento necesario, y así poder estimar el nivel de riesgo al que está sometido el sistema analizado.

Para la adquisición automática de nueva información, es necesario establecer unas guías de comportamiento o reglas que definan el objetivo que se desea alcanzar, y cómo las distintas clases deben interactuar entre ellas [2]. Para cumplir este objetivo, el lenguaje más utilizado es *Semantic Web Rule Language* (SWRL) [3], que combina *Web Ontology Language* (OWL) y *Rule Markup Language*. En este enfoque, las reglas se componen de un antecedente y un consecuente, de forma que, al cumplirse el primero, se infiere el segundo. Otra posibilidad que está ganando importancia en los últimos años es *SPARQL Inference Notation* (SPIN) [4], una recomendación de W3C que combina conceptos de lenguajes orientados a objetos, consultas y sistemas basados en reglas para describir el comportamiento de los datos.

Siguiendo la metodología propuesta por Riesco et al. [5] para conseguir un *framework* para la gestión dinámica del riesgo basada en información de Inteligencia de Amenazas y el uso de razonadores y reglas SWRL, nuestro objetivo es ampliar las posibles fuentes de amenazas existentes en un entorno real. En este sentido, debemos considerar anomalías en comunicaciones, tanto individualmente como tras analizar su correlación. Este modelado implica un estudio previo del entorno del sistema, para definir las clases relevantes, atributos y relaciones. En cuanto a las reglas de inferencia, proponemos un enfoque distinto al de la ontología base, reemplazando SWRL por la recomendación de W3C, que presenta mayor eficacia en casos donde se debe procesar un volumen muy alto de individuos, como el siguiente. En este sentido, nuestro objetivo es conseguir un sistema de gestión dinámica de riesgo y conciencia cibersituacional, más exhaustivo que el anterior, mediante la inclusión de distintas fuentes y reglas SPIN.

Para ello, a lo largo del documento discutiremos los conceptos anteriores, y definiremos una metodología de trabajo, presentando los resultados obtenidos para un caso de uso específico. En la primera sección presentamos la visión general del proyecto y el objetivo principal. A continuación describiremos otras propuestas y trabajos relacionados con este campo: distintos enfoques de la gestión de riesgos mediante ontologías y otras aplicaciones de SWRL y SPIN para inferir conocimiento, y finalmente presentar el proyecto base del que surge éste. A lo largo de la siguiente sección detallaremos la propuesta y el escenario en el que tiene lugar el desarrollo. Para continuar, presentaremos el sistema en su conjunto, la ontología creada para modelar distintos tipos de anomalías y las reglas SPIN para inferir relaciones. Además, presentaremos la validación de la metodología propuesta y finalmente, definiremos las conclusiones y líneas futuras.

II. Trabajos Relacionados

En los últimos años el uso de ontologías ha crecido significativamente, debido a su impacto en distintas áreas de trabajo, lo que ha llamado la atención de muchos autores, que han descrito sus desarrollos aplicando ontologías con propósitos muy variados. Por ello, hemos analizado los estudios previos

relacionados con la temática de este proyecto, en cualquiera de sus aspectos.

Uno de los objetivos principales de nuestro trabajo es la detección, análisis y reacción frente a ataques mediante la selección de estrategias de respuesta en función del nivel de riesgo. En línea con este enfoque, C. Onwubiko [6] describe una ontología para mapear la información del proceso de detección de los ciberincidentes, estudiando las fuentes, el uso de sensores y cómo responder y recuperarse de ellos. Las fortalezas principales del *framework* son el disparo de alertas y el análisis para encontrar el origen del problema, así como las posibilidades de recuperación o mitigación. En [7], Yuan et al. describen la estructura y pasos para definir un *framework* basado en el estudio del entorno, para establecer diferentes clases y relaciones que permitan abordar el problema del razonamiento, creando reglas SWRL con el objetivo de modelar el riesgo residual de un sistema.

En cuanto al análisis de los datos de sensores, el trabajo presentado en [8] utiliza los activos para controlar sistemas *ciber-físicos*. Con la aparición de tecnologías inalámbricas, también han surgido nuevos *exploits*, especialmente dirigidos al campo de Internet de las Cosas. En ese sentido, Mozzaquatro et al. aplican reglas SWRL y un razonador Pellet para la inferencia de conocimiento, y realizan consultas SPARQL para verificar la información. Para validar la ontología propuesta, aplican la metodología *Software Product Quality Requirements and Evaluation* (SQuaRE). En su desarrollo, para gestionar las vulnerabilidades que surgen de las distintas fuentes, y con el objetivo de proteger al usuario, Syed R. [9] también propone la construcción de una ontología y un sistema de razonamiento basado en reglas SWRL y consultas SPARQL para evaluarlo.

En cuanto al lenguaje de definición de las reglas, estudios previos y trabajos de investigación han utilizado en general SWRL como base. Sin embargo, en los últimos años, algunos artículos introducen el desarrollo de consultas SPARQL o incluso reglas SPIN para la generación de relaciones o nuevos individuos. En [10], los autores describen un modelo de riesgo para la gestión y evaluación de posibles amenazas con el objetivo de considerar las respuestas adecuadas. El artículo se enfoca en incluir el factor humano en el cálculo del riesgo, utilizando reglas SPIN para inferir relaciones y establecer el perfil del factor humano (experiencia, entrenamiento, etc.).

Sobre esta base, parece válido que las ontologías proporcionan una herramienta primaria para almacenar y procesar información en cualquier dominio, así como la ventaja de usar reglas de inferencia en estos entornos, especialmente si el número de instancias que deben ser procesadas crece.

El desarrollo de este artículo parte de un proyecto anterior, descrito en [5] por Riesco et al., donde los autores proponen una solución integrando ontologías OWL y reglas SWRL, junto a un razonador Pellet, en un modelo por capas basado en Inteligencia de Amenazas. Los autores incluyen en la ontología información de *Cyber Threat Intelligence* (CTI), activos, y medidas para estimar las amenazas existentes sobre el sistema, y así poder inferir los riesgos a los que se exponen, y poder gestionarlos.

III. Escenario

Basándonos en la ontología descrita anteriormente [5], nuestra intención es expandir las fuentes de amenaza, que ya contaban con activos e información CTI, incluyendo distintas anomalías detectadas mediante sensores físicos y lógicos. Para describir la nueva ontología, por tanto, es importante entender tanto el entorno de trabajo como las distintas tecnologías y herramientas aplicadas.

Este proyecto se construye sobre los principios de la Web Semántica [11], que proporciona tecnologías para definir vocabularios y reglas para manejarlos. También conocida como la web de los datos enlazados [12], uno de los requisitos principales es contar con grandes cantidades de datos en formato estándar, para poder acceder a ellos y generar relaciones entre las clases. Nuestra ontología se construye usando OWL [13], recomendación de W3C aplicada cuando los datos recogidos se procesan en aplicaciones.

Las ontologías que se desarrollan en el artículo del que parte este proyecto contienen información para describir el entorno de inteligencia de amenazas y los datos *Disaster Risk Management* (DRM) que rodean el sistema. En la ontología que representa CTI tratan de describir los objetos de dominio (Patrones de ataque, Campañas, *Course of Action*, Indicadores, Vulnerabilidades, Herramientas o Actores) siguiendo la especificación OASIS [14]. Además, en [5], se presenta la ontología DRM, que incluye Activos, junto con su valoración, Contexto, Incidentes, Amenazas, Salvaguardas, y un conjunto de términos relacionados con el riesgo: tipo, evaluación, impacto, gestión, probabilidad o severidad. Los conceptos de ambas ontologías se relacionan mediante reglas SWRL, con la intención de establecer las condiciones que determinan que el sistema monitorizado está siendo amenazado, y en ese caso, el nivel de riesgo asociado, y la utilidad de las salvaguardas, o las posibles acciones a tener en cuenta. Por ello, y debido a las características del entorno donde el desarrollo se despliega, equipado con distintos tipos de sensores, creamos una ontología para definir términos relacionados con anomalías en las comunicaciones (sistemas que monitorizan redes Wi-Fi o Bluetooth, controlan Firewalls o el tráfico de una red, o analizan el comportamiento del usuario). Para conseguir una gestión dinámica real del riesgo y tener conciencia sobre el sistema, es necesario analizar los registros procedentes de estos sensores desplegados, mediante técnicas de aprendizaje automático, para decidir si representan o no anomalías, o si distintos eventos tienen relación entre ellos, representando una secuencia de ataques en cadena. Esta información se almacena en forma de instancias de las clases correspondientes de la ontología, con propiedades y relaciones entre ellas. Además, el conjunto de activos presente en el sistema también se incluye en la ontología, como los datos de inteligencia de amenazas que afectan al entorno. Estos procesos se representan en la Fig. 1.

La mayor desventaja que supone incluir las anomalías es el incremento de individuos que se deben procesar. Para enfrentar esta situación, dado que las reglas SWRL ralentizan el procesamiento debido a la ejecución del razonador sobre todas las instancias, y para completar la interconexión entre estas clases, las reglas aplicadas a las ontologías de CTI y DRM se traducen a lenguaje SPIN, que será el utilizado para definir las nuevas reglas para anomalías.

De esta forma, obtenemos un catálogo de instancias de amenazas generado a partir de anomalías o vulnerabilidades,

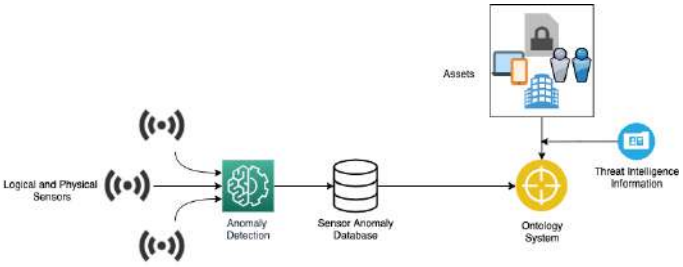


Figura 1. Entorno del proyecto, compuesto de las distintas fuentes que alimentan la ontología

con un valor asociado de probabilidad e impacto. Además, las amenazas se asocian con un tipo de riesgo, del que se calculará el valor medio de probabilidad e impacto heredados, obteniendo así el nivel de riesgo.

Con toda esta información, obtenemos una vista genérica de las fuentes de amenazas a las que se expone el sistema, modelando los diferentes tipos, y valores de probabilidad e impacto de cada uno de ellos. De cada instancia de amenaza se realiza una inferencia para generar el riesgo que afecta al sistema, definiendo relaciones entre las distintas categorías. El nivel de riesgo total se obtiene a partir de una serie de cálculos de la siguiente forma: Cada instancia de amenaza tiene un valor de probabilidad P_{Ti} y de impacto I_{Ti}. Estos individuos se agrupan según las subclases a las que pertenecen para calcular el valor total de probabilidad P_T e impacto I_T para cada tipo de riesgo, donde n_T representa el número de instancias que pertenecen a cada categoría (Ec. (1)):

$$P_T = \frac{\sum P_{Ti}}{n_T}; \quad I_T = \frac{\sum I_{Ti}}{n_T}. \tag{1}$$

Posteriormente, se agrupan las amenazas que generan cada tipo de riesgo obteniendo, de igual forma, una probabilidad P_R y un impacto I_R para todas las clases de riesgo que, al multiplicarse, proporcionan el nivel de riesgo potencial para cada sub-categoría (PR_R) (Ec. (2) y Ec. (3)):

$$P_R = \frac{\sum n_T \cdot P_T}{\sum n_T}; \quad I_R = \frac{\sum n_T \cdot I_T}{\sum n_T}, \tag{2}$$

$$PR_R = P_R \cdot I_R. \tag{3}$$

Además, para considerar el efecto de las salvaguardas (S_{Ri}) que actúan sobre cada tipo, el riesgo residual de cada categoría también se calcula (RR_R) (Ec. (4)):

$$RR_R = PR_R - \sum S_{Ri}. \tag{4}$$

Finalmente, el riesgo del sistema en global se calcula en términos de riesgo potencial y residual (PR_{Total} y RR_{Total}), siendo n_R el número de tipos de riesgo incluidos (Ec. (5) y Ec. (6)):

$$PR_{Total} = \frac{\sum PR_R}{n_R}, \tag{5}$$

$$RR_{Total} = \frac{\sum RR_R}{n_R}. \tag{6}$$

IV. PROPUESTA: MODELO DE INFORMACIÓN

IV-A. Modelado de Ontologías - Anomalías

Como se ha mencionado anteriormente, el punto de partida eran las ontologías [5] previamente modeladas para definir los dominios de Inteligencia de Amenazas y DRM. Por lo tanto, para conseguir el objetivo de expandir el sistema incluyendo otras fuentes de amenaza, debemos desarrollar otra para definir estos conceptos y posteriormente unir las tres. El primer paso es modelar el concepto de *Detected Anomaly*. Se puede subdividir según el origen de la anomalía (sensor), ya que cada tipo debe tener distintas propiedades, según los datos que se obtienen de los sensores: potencia, direcciones IP o MAC, frecuencia, número de bytes o paquetes, números de puertos, etc. Además, una combinación de anomalías simples puede generar una correlación, incrementando la probabilidad o impacto de la amenaza relacionada. Todos los sub-tipos de anomalías se presentan en la Fig. 2.

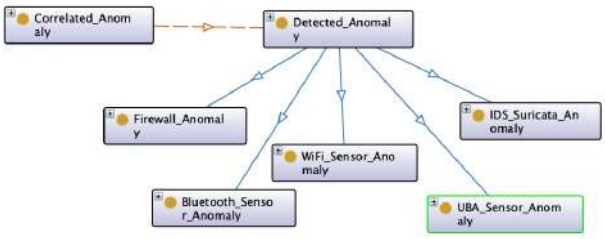


Figura 2. Subclases definidas en la ontología para la clase *"Detected_Anomaly"*

En una visión más amplia, en cuanto a las relaciones con las otras ontologías, establecemos que las anomalías (*Detected_Anomalies*) podrían generar correlaciones (*Correlated_Anomalies*), y en ambos casos se generan amenazas. Las salvaguardas mitigan todo tipo de anomalías o vulnerabilidades y protegen a los activos. En el otro lado del gráfico, las amenazas podrían ser generadas por activos, al explotar sus vulnerabilidades y, por último, esas amenazas producen riesgos, que dañan a los activos. La Fig. 3 muestra las relaciones principales en la ontología conjunta.

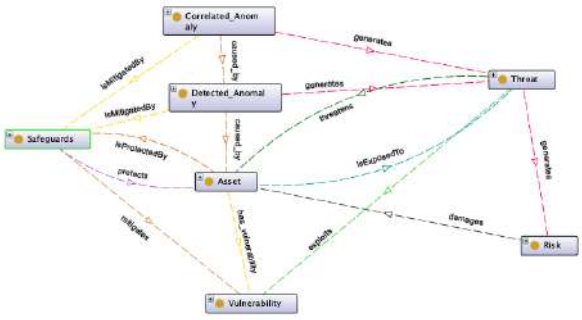


Figura 3. Grafo relacional con los principales conceptos de la ontología conjunta

IV-B. Reglas de Inferencia

OWL es un lenguaje estándar que puede usarse para definir clases, propiedades y relaciones entre estas clases. Su objetivo es dar definiciones axiomáticas, sin establecer el comportamiento computacional. Para ello se desarrolla SPIN [4], que combina conceptos de lenguajes orientados a objetos, lenguajes de consulta y sistemas basados en reglas para describir el comportamiento de los datos de la web. Una de las ideas básicas es enlazar definiciones de clase con consultas SPARQL para capturar reglas que formalizan el comportamiento esperado de esas clases.

Una vez que la ontología está poblada con individuos de las clases definidas, la generación de nuevo conocimiento se automatiza si definimos correctamente las reglas. Para el desarrollo presentado aquí, la ontología ya tiene individuos de varias clases de anomalías e Inteligencia de Amenazas. Sobre esto es necesario crear amenazas relacionadas y definir el riesgo que suponen estableciendo las reglas mediante este lenguaje. Se construyen sobre la estructura de una consulta CONSTRUCT.

```
CONSTRUCT {
# Tripletas de individuos o relaciones
# construidas a partir de la regla
} WHERE {
# Tripletas de condición.
# Cuando se cumplen, las definiciones
# de la parte superior se construyen }
```

Las inferencias planteadas en este desarrollo son muy diversas. En primer lugar, las reglas definidas en [5] para Inteligencia de Amenazas se traducen manualmente de SWRL a SPIN. Además, en relación con la ontología para modelar anomalías también se implementan reglas propias. Para ejemplificar los distintos procesos de razonamiento, la estructura de una de las reglas de cada grupo se detallan a continuación:

Reglas de Enriquecimiento para generar más información sobre el entorno de Inteligencia de Amenazas en el que se opera, realizando inferencias sobre los datos originales. Este es uno de los tipos de reglas que estaban escritas en SWRL, así que procedemos a traducirlas al nuevo estándar. En este caso, el propósito de la regla es relacionar el Dominio con las IPs para obtener enriquecimiento de DNS:

```
CONSTRUCT {
?ref CTI:resolves_to_refs ?ip.
} WHERE {
?ip a CTI:IPV4Addr.
?ref a CTI:Domain_Name.
?ip DRM:belongToRefs ?ref. }
```

Reglas de Inventario de Amenazas para generar amenazas (nuevos individuos) a partir de algunos activos y sus interrelaciones. En este caso, las reglas también se traducen a SPIN. Para su desarrollo estudiamos todos los tipos de activos que podrían estar presentes en el sistema, y cómo influyen entre ellas o en conjunto. En el ejemplo presentado a continuación, el objetivo es generar una amenaza de distribución de Software Malicioso con probabilidad 4 e impacto 3 cuando el entorno cumple unas condiciones de activos e Información de Inteligencia de Amenazas (Usuarios, Datos, Software, su valoración y el alcance del riesgo).

```
CONSTRUCT {
?x a DRM:DelibMaliciousSWDistrib.
?x DRM:probability 4.
?x DRM:impact 3.
?x DRM:threatens ?sw.
} WHERE {
?u a DRM:Users.
?rs a DRM:Risk_Scope.
?av a DRM:Asset_Valuation.
?av DRM:evaluates ?rs.
?new_data a DRM:Data.
?sw a CTI:Software.
?rs DRM:dependsOn ?new_data.
?new_data DRM:dependsOn ?sw.
?new_data DRM:dependsOn ?u.
BIND(URI(DRM:SWD1) as ?x) }
```

Reglas de Inteligencia de Amenazas para detectar eventos maliciosos relacionados con un tipo de amenaza. Existe una regla de este tipo para cada categoría de CTI incluida en la ontología. También traducimos este grupo de reglas. El objetivo en este caso era seguir redirecciones hasta encontrar un ejecutable, lo que implicaría un evento de seguridad.

```
CONSTRUCT {
?x a DRM:Security_Events.
?x CTI:src_ref ?sr.
?x CTI:dst_ref ?red.
?x CTI:related-to ?nt2.
} WHERE {
?red a CTI:URL.
?sr a CTI:URL.
?nt a CTI:Network_Traffic.
?nt2 a CTI:Network_Traffic.
?pl CTI:Artifact.
?pl2 CTI:Artifact.
?pl CTI:redirection ?red.
?nt CTI:dst_payload_ref ?pl.
?nt2 CTI:dst_payload_ref ?pl2.
?nt2 CTI:src_ref ?sr.
?nt2 CTI:dst_ref ?red.
?pl2 CTI:extensions ”windows”.
?pl CTI:mime “javascript”.
BIND(URI(DRM:SE1) as ?x) }
```

Anomalías generan Amenazas. Este es el primer tipo de reglas definido específicamente para la extensión del proyecto al añadir datos de los sensores. Se crea al menos una regla para cada sensor, definiendo la relación entre anomalía y la amenaza asociada, que se infiere a partir del entorno del sistema y los posibles ataques que se llevarían a cabo a través de la tecnología monitorizada por cada sensor presente en esta ontología (ONA). Para el siguiente ejemplo se asume que una anomalía en redes Wi-Fi genera amenazas de Denegación de Servicio con una probabilidad e impacto dados.

```
CONSTRUCT {
?x a DRM:DenialOfService.
?x a DRM:Threat.
?x DRM:isGeneratedBy ?a.
?a DRM:generates ?x.
?x DRM:probability 5.
?x DRM:impact 8.
} WHERE {
?a a ONA:WiFiSensorAnomaly.
BIND(URI(DRM:DoS1) as ?x) }
```

Amenazas generan Riesgo. Este conjunto de reglas es parte del núcleo de razonamiento propuesto en este proyecto, ya que establece la relación entre amenazas y riesgos. Por lo tanto, será necesario definir al menos una regla para cada tipo de amenaza considerada, siendo posible relacionar una de estas instancias con varios tipos de riesgo. Uno de los casos considerados son las amenazas de Denegación de Servicio, que pueden implicar riesgos de reclamación de los usuarios (*User Complaints*).

```
CONSTRUCT {
?r DRM:isGeneratedBy ?t.
?t DRM:generates ?r.
} WHERE {
?t a DRM:DenialOfService.
?r a DRM:UsersComplaintsRisk }
```

Reglas de Severidad de Riesgo para evaluar el nivel de riesgo y clasificarlos. Se generan reglas a partir de las anteriores en SWRL para todas las posibles categorías: bajo, medio, alto y extremo.

```
CONSTRUCT {
?x a DRM:HighRiskSeverity.
?x DRM:drm_value ?ar.
?x DRM:evaluates ?risk.
} WHERE {
?risk a DRM:Risk.
?rr a DRM:ResidualRisk.
?rr DRM:actualRisk ?ar.
?rr DRM:evaluates ?risk.
FILTER(?ar > 6 AND ?ar < 8)
BIND(URI(DRM:HRS1) as ?x) }
```

Reglas de Evaluación de Riesgos. Mediante estas reglas generamos instancias para los subtipos de riesgos potenciales y residuales, con el objetivo de diferenciar en cada categoría el nivel de riesgo antes y después de considerar el efecto de las salvaguardas. Estas reglas se utilizan para relacionar las instancias de riesgo con sus correspondientes valores potenciales y residuales.

```
CONSTRUCT {
?rr a DRM:ResidualRisk.
?pr a DRM:PotentialRisk.
?r DRM:hasAssessmentOf ?rr.
?r DRM:hasAssessmentOf ?pr
} WHERE {
?r a DRM:Risk.
BIND(URI(DRM:RR_1) as ?rr)
BIND(URI(DRM:PR_1) as ?pr) }
```

Los valores de riesgo se incluyen en cada instancia de riesgo potencial/residual después de el cálculo descrito en secciones anteriores.

Reglas de Gestión de Riesgos para elegir una respuesta como reacción ante los riesgos. En este caso, las opciones implementadas en las reglas serán Mitigación (severidad Extrema y Alta de riesgos), Investigación (severidad media del riesgo) y Monitorización (severidad baja del riesgo). Este caso se podrá extender, tras estudiar el entorno en profundidad, para definir medidas más específicas dependiendo del riesgo y del contexto.

```
CONSTRUCT {
?x a DRM:RiskMitigation.
?x DRM:drm_value ?sev.
?x DRM:manages ?risk.
} WHERE {
?risk a DRM:Risk.
?er a DRM:HighRiskSeverity.
?er DRM:drm_value ?sev.
?ex DRM:evaluates ?risk.
BIND(URI(DRM:RMS1) as ?x) }
```

El objetivo último de estas reglas es la inferencia de relaciones entre instancias incluidas en la ontología (anomalías en sensores, inteligencia de amenazas y activos) y así obtener una visión general del sistema y del nivel de riesgo en tiempo real, al igual que las medidas a implementar y la efectividad de las salvaguardas dispuestas.

V. Validación de la Ontología

Para validar la ontología construida se utilizan consultas SPARQL y la aplicación *Protégé* para visualizar la inferencia de relaciones y el cálculo del nivel de riesgo. Para demostrar la eficiencia de las reglas, a modo de ejemplo consultamos la ontología para verificar la inferencia "Anomalías generan Amenazas", obteniendo las anomalías introducidas en el sistema (anomalías del sensor de ciberseguridad), las amenazas generadas y, al elegir una de estas instancias (Denegación de Servicio), los valores de impacto y probabilidad (Fig. (4)).

```
SELECT ?anomaly ?threat ?impact ?prob
WHERE {
?anomaly a ONA:Detected_Anomaly.
?anomaly DRM:generates ?threat.
?threat a DRM:DenialOfService.
?threat DRM:impact ?impact.
?threat DRM:probability ?prob. }
```

anomaly	threat	impact	probability
Anomaly_CS_Type_0	DenialOfService_CS_Type_0	"5.0"^^	"3.0"^^<ht
Anomaly_CS_Type_1	DenialOfService_CS_Type_1	"5.0"^^	"3.0"^^<ht

Figura 4. Resultado de la regla Anomalía-Amenaza (Consulta SPARQL)

Esta información también se comprueba en la aplicación (Fig. (5)).

Figura 5. Resultado de la regla Anomalía-Amenaza (Protégé)

Siguiendo este ejemplo, validamos las reglas creadas para el proyecto, por ejemplo, el cálculo de los riesgos del sistema (Fig. (6)).

```
SELECT ?risk ?riskvalue
WHERE {
?risk a DRM:Risk.
?risk DRM:hasAssessmentOf ?pr.
?pr a DRM:PotentialRisk.
?pr co:potentialRisk ?riskvalue. }
```

Otro enfoque de validación que queremos afrontar es la aplicación de metodologías para evaluar la calidad de la ontología, utilizando procedimientos estandarizados, como *Ontology Quality Requirements and Evaluation Method and Metrics* (OQuaRE) [8], mediante la herramienta SQuaRE (*Software Product Quality Requirements and Evaluations*).

Los resultados obtenidos al aplicar la herramienta sobre la ontología creada se estandarizan, convirtiéndolos en valores comprendidos entre 1 y 5, siguiendo la guía definida en [15], y obteniendo una puntuación media considerando todos lo indicadores de 3,83/5, que se considera aceptable. Para continuar, analizamos los resultados individuales conseguidos para cada categoría. Desde una perspectiva estructural, este procedimiento contempla los factores de calidad presentes en la ontología. La métricas obtenidas se agrupan dependiendo del indicador al que se refieren, en cuatro categorías, y consiguiendo un valor medio de 2,5 sobre 5 puntos. La ontología puntuó alto en cohesión, mientras que los resultados fueron

risk	riskvalue
PressNegativeImpactRisk_Risk	"10.222222"
TechnicalComplexityDerivedRisk_Risk	"10.54273"
NetworkOutageRisk_Risk	"10.468878"
CorporateBrandImageDamageRisk_Risk	"9.551021"
UntrustworthyRisk_Risk	"13.142858"
DataProtectionComplianceRisk_Risk	"9.540741"
NonIntentionalInformationTamperingRisk_Risk	"12.0"
NonIntentionalInformationDestructionRisk_Risk	"12.0"
DenialOfServiceRisk_Risk	"13.485207"
DeviceTheftRisk_Risk	"4.964286"
DeliberatedRegistersTamperingRisk_Risk	"9.173778"
UsersComplaintsRisk_Risk	"11.298731"
SocialEngineeringRisk_Risk	"6.0"
DeliberatedHWTamperingRisk_Risk	"9.25"
PhysicalFailureRisk_Risk	"10.0"
StrategicObjectiveRisk_Risk	"4.0"
NonIntentionalUserErrorRisk_Risk	"6.0"
DeliberatedMaliciousSWDistributionRisk_Risk	"16.866667"
NonIntentionalMaliciousSWDistributionRisk_Risk	"18.307692"
DeliberatedSWTamperingRisk_Risk	"11.910289"
BadReputationRisk_Risk	"12.108473"
DeliberatedConfigFilesTamperingRisk_Risk	"9.173778"
FireRisk_Risk	"4.0"
DeviceLostRisk_Risk	"4.964286"

Figura 6. Resultado del cálculo de valores de riesgo

más bajos en redundancia y *tangledness*, que se convertirán en áreas futuras de mejora. De esta información, representada en la Fig. (7), extraemos la necesidad de identificar los conceptos que no son informativos y diseñar una mejor distribución de las categorías principales.

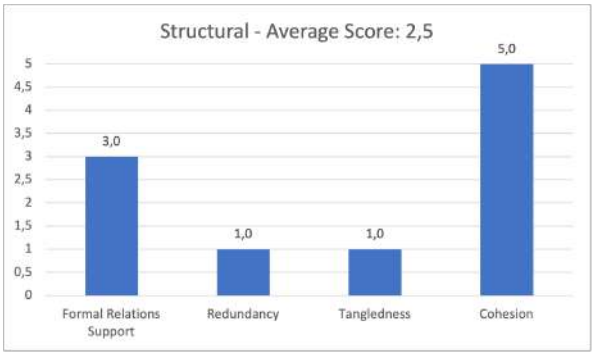


Figura 7. Resultados obtenidos con SQuaRE para las métricas estructurales

El indicador de adecuación funcional, o la capacidad de proporcionar determinadas funciones, mejora la puntuación media de la perspectiva anterior (3,225/5), mostrando buenos resultados en orientación, indexación y enlace, entre otros. Sin embargo, los resultados de vocabulario controlado y agrupación se deben mejorar, por lo que una de las siguientes acciones a realizar será unificar términos similares, como se muestra en la Fig. (8).

Finalmente, existen otras métricas relacionadas con el entorno y el esfuerzo: Compatibilidad, cuando dos componentes pueden intercambiar información o realizar sus funciones mientras comparten el mismo entorno, la Posibilidad de Transferencia, que representa el grado en el cual el entorno se podría cambiar, o la Operabilidad como el esfuerzo necesario para utilizar el software. En estos aspectos se obtuvieron resultados muy positivos, puntuando por encima de 4 en todos

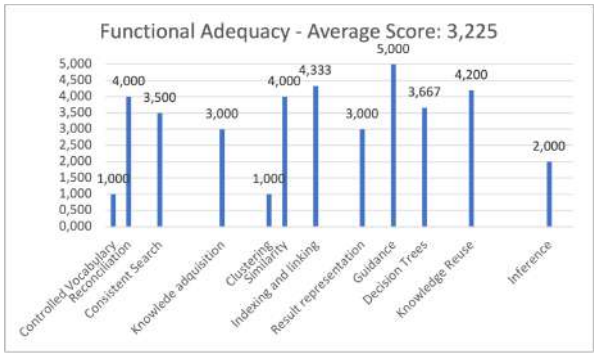


Figura 8. Métricas de Adecuación Funcional obtenidas con SQuaRE

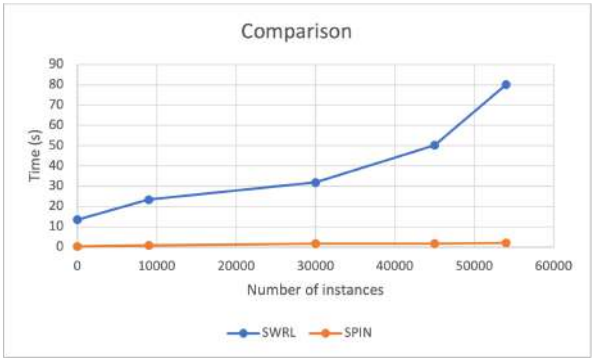


Figura 10. Comparación entre SWRL y SPIN. Representa el número de instancias frente al tiempo requerido para la ejecución

los casos (Fig. (9)).

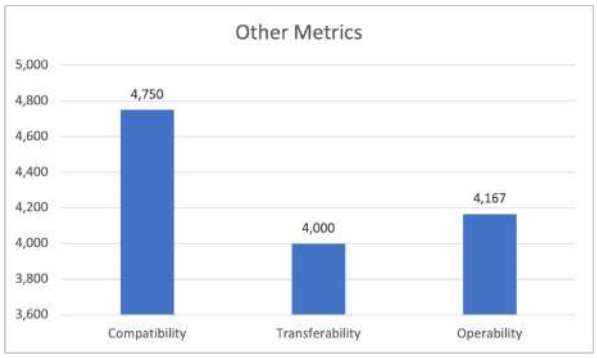


Figura 9. Otras métricas de la Ontología obtenidas con SQuaRE

V-A. *Comparación entre SWRL y SPIN*

A lo largo del desarrollo del proyecto, la ontología con la que trabajamos empezó a crecer sustancialmente en cuanto al número de individuos de anomalías, alcanzando el rango de miles de instancias rápidamente y, utilizando SWRL, el tiempo de ejecución se disparó. Por ello, uno de los principales objetivos del proyecto es la evolución de reglas SWRL a SPIN, para verificar la eficiencia de la propuesta.

En este aspecto, cabe mencionar que las reglas SWRL se suponían más lentas de antemano, ya que deben estar todas las instancias guardadas en la ontología, aplicar sobre ellas el razonador y finalmente ejecutar las reglas, mientras que el caso de SPIN permite crear instancias e inferir sobre ellas sin necesidad de almacenarlas y sin el requisito del razonador. Por ello, el procedimiento de guardado de las últimas inferencias en SWRL no se tiene en cuenta, al igual que la acción de guardado de la ontología SPIN en un documento. Además, la Fig. (10) muestra el número de instancias de anomalías que se crean al principio, aquellas que disparan las reglas y, en algunos casos, crean más individuos, por lo que el número de instancias de la ontología al final es significativamente mayor. Los resultados obtenidos en SPIN son mucho más bajos, y por tanto más eficientes, por lo que, para ontologías con número de individuos alrededor de los millares, este lenguaje muestra mejor rendimiento.

VI. Conclusiones

Dada la importancia que están ganando actualmente las ontologías, es esencial tener un conocimiento profundo y de primera mano de la tecnología y las diferentes posibilidades que ofrece. Por ello, decidir cuando es más eficiente usar cada lenguaje o estructura es vital para el desarrollo de cualquier proyecto. Tratando de satisfacer la necesidad de mantener el riesgo bajo control mediante una gestión continua del riesgo, en este artículo presentamos una ontología conjunta para gestionar dinámicamente el riesgo de un sistema en el cual las amenazas proceden desde fuentes lógicas y físicas, considerando anomalías, activos e inteligencia de amenazas. Con ese propósito, comenzamos a partir de una ontología previamente desarrollada sobre Inteligencia de Amenazas (CTI), Activos, Amenazas y Riesgos (DRM), y el objetivo principal es extenderlo para añadir anomalías (ONA) y nuevas reglas en formato SPIN, traduciendo las anteriores, descritas en SWRL, para calcular el nivel de riesgo y realizar soporte a la toma de decisiones.

La ontología obtenida se analizó utilizando el framework OQuaRE, con una puntuación media de 3.83 sobre 5 entre todas las métricas. Los mejores resultados se obtuvieron en las métricas de compatibilidad (4.75/5), operabilidad (4.167/5) y posibilidad de transferencia (4.0/5), mientras que la métrica de adecuación funcional consiguió un valor intermedio en la escala (3.225/5). Los resultados más bajos se obtienen en términos estructurales (2.5/5), debido a que las métricas que influyen en este cálculo tienen valores divergentes, debido a la unión de tres ontologías distintas, desde 1.0/5 a 5.0/5, convirtiendo este aspecto en un punto de acción muy importante de cara a futuro.

También estudiamos la diferencia en los tiempos de ejecución entre ambos tipos de reglas, consiguiendo resultados significativamente mejores para el lenguaje SPIN en el caso que se detalla en este desarrollo. Este cálculo se representa bajo condiciones específicas, debido a que SWRL requiere tiempo adicional para almacenar todos los individuos inferidos, al contrario de SPIN. Por ese motivo, se considera únicamente el número de individuos generados manualmente al principio de la ejecución. El tiempo requerido por las reglas SWRL es considerablemente mayor que en el caso de SPIN, presentando un crecimiento no lineal con el número de instancias. También cabe mencionar que, en aquellos escenarios con más de 6000 individuos, no se han podido obtener resultados concluyentes para SWRL, lo que apoya el uso de SPIN en este proyecto, donde el número de individuos a procesar es alto. A fin de reforzar esta decisión, nuevos

tipos de métricas se podrían implementar para realizar la comparación objetiva entre ambos planteamientos.

Como hemos mencionado anteriormente, la validación llevada a cabo con el framework OQuaRE revela los puntos débiles del desarrollo, como la redundancia y el *tangledness* entre conceptos. Con el objetivo de mejorar la construcción de la ontología, debemos identificar los conceptos que no son informativos y diseñar mejor la distribución de las categorías, unificando términos similares. Así, para continuar trabajando en este proyecto, proponemos analizar y corregir las métricas OQuaRE que obtuvieron valores más bajos en primer lugar y, para continuar, extender la variedad de reglas a aplicar, especialmente en el caso de soporte a la toma de decisiones, o incluir otras fuentes de anomalías.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente apoyado por el Ministerio de Defensa del Gobierno Español en el marco del proyecto PLICA (Ref: 1003219004900-Coincidente)

REFERENCIAS

[1] ”ISO 31000:2018, Risk management — Guidelines”, [Online]. Disponible: https://www.iso.org/obp/ui/#iso:std:iso:31000:ed-2:v1:en, (2021-11-16).

[2] Mercier, Chloe and Roux, Lisa and Romero, Margarida and Alexandre, Frederic and Vieville, Thierry: ”Formalizing Problem Solving in Computational Thinking : an Ontology approach”, en *2021 IEEE International Conference on Development and Learning (ICDL)*,pp.1-8, 2021.

[3] ”SWRL: A Semantic Web Rule Language Combining OWL and RuleML”, [Online], Disponible: https://www.w3.org/Submission/SWRL/, (2021-05-27).

[4] ”SPIN - Overview and Motivation”, [Online], Disponible: https://www.w3.org/Submission/spin-overview/, (2021-05-20)

[5] Riesco, R. and Villagrá, V. A.: ”Leveraging cyber threat intelligence for a dynamic risk framework: Automation by using a semantic reasoner and a new combination of standards (STIX™, SWRL and OWL)”, en *International Journal of Information Security*, n.6, pp.715-739, 2019.

[6] Onwubiko, Cyril:”CoCoa: An Ontology for Cybersecurity Operations Centre Analysis Process”, en *2018 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*, pp.1-8, 2018.

[7] Yuan, Jingfeng and Li, Xuan and Chen, Kaiwen and Skibniewski, Mirosław J.: ”Modelling residual value risk through ontology to address vulnerability of PPP project system”, en *Advanced Engineering Informatics*, vol. 38, pp. 776-793, 2018.

[8] Mozzaquatro, Bruno and Agostinho, Carlos and Goncalves, Diogo and Martins, João and Jardim-Goncalves, Ricardo: ”An Ontology-Based Cybersecurity Framework for the Internet of Things”, en *Sensors*, vol. 18, n. 9, pp. 3053, 2018.

[9] Syed, Romilla: ”Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system”, en *Information & Management*, vol. 57, n.6, pp. 103334, 2020.

[10] Williams, Sophie and Marriot, Damien: ”Human Factors in a Computable Cybersecurity Risk Model”, en *Twelfth International Symposium on Human Aspects of InformationSecurity & Assurance (HAISA)*, pp. 214-224, 2018.

[11] ”Semantic Web - W3C”, [Online], Disponible: https://www.w3.org/standards/semanticweb/, (2021-06-04)

[12] ”Data - W3C”, [Online], Disponible: https://www.w3.org/standards/semanticweb/data, (2021-06-04)

[13] , ”OWL - Web Ontology Language Overview”, [Online], Disponible: https://www.w3.org/TR/owl-features/, (2021-06-04).

[14] ”Introduction to STIX”, [Online], Disponible: https://oasis-open.github.io/cti-documentation/stix/intro, (2021-06-04).

[15] ”The Quality metrics of OQuaRE”, [Online], Disponible: http://miuras.inf.um.es/evaluation/oquare/Metrics.html, (2021-07-17).

Model-Based Analysis of Race Condition Vulnerabilities in Source Code

Razvan Raducu, Ricardo J. Rodríguez, and Pedro Álvarez
Dept. of Computer Science and Systems Engineering, University of Zaragoza, Spain
{razvan, rjrodriguez, alvaper}@unizar.es

***Abstract*—Formal methods have been commonly used to understand the behavior of systems in the field of security. The application of these methods requires an accurate modeling of the source code and the subsequent analysis of the behavior of that code during execution. Most of the research proposals are very specific solutions, restricting the type of code analyzed (programming language, the level of concurrency, or the level of parallelism, among others), the vulnerabilities studied, and the formal analysis techniques applied. In this paper, we propose a framework that facilitates the integration of different formal models and analysis techniques for the detection of vulnerabilities in the source code of complex programs. Our proposal is based on a model-to-model transformation approach, relying on translating the source code into abstract syntax trees and then into different modeling formalisms that will be analyzed to detect vulnerabilities. The choice of a certain formalism and its corresponding analysis techniques is conditioned by the type of vulnerability to study. In particular, here we briefly describe the application of Petri nets and model verification to detect a type of race conditions vulnerabilities. The complete integration of this formalism and its automatic analysis in the framework is still a work in progress.**

***Index Terms*—vulnerability, analysis, source code, model-based**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

Formal methods are applied in the field of Information Systems to improve their security, analyzing the behavior of software systems and detecting different types of vulnerabilities. These methods have gained popularity because they help developers identify bugs, inconsistencies, or unwanted code executions at an early stage. Their main advantage over more traditional testing approaches is that they use mathematical proofs as a complement to ensure that the system behaves as expected.

Formal methods have also proven useful for analyzing the code of individual components, especially when dealing with complex software systems that internally execute parallel or concurrent programs [1]. Thus, formal method-based code analysis requires a mathematical-based language to model the system to be tested and a set of tools to check the behavior of the modeled system. In the case of vulnerability analysis, the model provides a representation of the events, activities, or programming primitives that can be attacked.

With respect to analyzing source code and formal methods, the Abstract Syntax Tree (AST) [2] is a tree representation of the structure of a source code written in a formal language, used primarily by compilers to read code and generate the target programs. This graphical representation is also used in program analysis and program transformation systems [3].

AST[1] are interesting models for analyzing source code, since they are abstract (they do not contain superfluous elements of the source code, such as spacing, comments, parentheses, etc.) and allow any ambiguity in the code to be resolved. However, they do not contemplate relevant aspects such as the possibility of concurrence or parallelism in the programs. The existence of concurrent or parallel programs in operating systems where the operations are not atomic are a breeding ground for the appearance of race condition vulnerabilities, which can go unnoticed in an AST-based analysis.

Since concurrent or parallel programs are very difficult to write, test, and debug [4], [5], it is easy to inadvertently create vulnerabilities in shared resource accesses between threads and processes without proper synchronization, called race condition vulnerabilities. The exploitation of these vulnerabilities can cause a much broader impact on security, such as bypassing security checks, breaking the integrity of databases [6], hijacking the vulnerable program control flow, or escalating privileges [7], among others.

Concurrency and parallelism of programs can be captured with other formal models, such as Petri nets. Petri nets [8] are a mathematical formalism language that easily represent common features of computer systems, such as concurrency, synchronization, conditional branches, looping, and sequencing, to name a few. The advantage of Petri nets over other modeling languages is that their execution semantics are well defined and supported by mathematical theory, allowing for formal analysis of the model.

Therefore, using model-to-model transformation and model-based analysis, in this paper we propose a framework for detecting race condition vulnerabilities in source code. Our ultimate goal is to provide an integrated solution that leverages the strengths of different formal methods in order to perform more concise vulnerability analysis. Although we are initially focused on AST and Petri nets as formal models and race condition vulnerabilities as a particular case of vulnerabilities, we envision a framework that is capable of detecting other common vulnerabilities, such as out-of-bound access, integer handling issues, or heap-related vulnerabilities.

Specifically, our framework first transforms the source code of a program into an intermediate representation, which we enrich and simplify to facilitate subsequent vulnerability analysis. This intermediate representation is then converted into a formal model, which is finally analyzed with the appropriate tools for vulnerabilities. To detect these vulnerabilities, we

[1]In this paper, we use interchangeably AST as a singular and plural acronym.

previously studied the possible techniques to identify them in the model of the formalism used. The results of the analysis are then interpreted and reported to the user to allow them to fix the source code and remove any potential vulnerabilities found.

The rest of this paper is structured as follows. Section II reviews the related work. Section III presents briefly our framework. Section IV then presents an example of its application for detecting *Time-Of-Check to Time-of-Use* (TOCTOU) vulnerabilities. A TOCTOU occurs when a program checks a particular characteristic of an object (e.g., whether the file exists), and later takes some action that assumes the checked characteristic still holds [9]. Finally, Section V concludes the paper and describes future work.

II. Related Work

In this section, we review the works closely related to ours in some way. We have divided this section in different topics: works that use Petri nets as a formal model in the context of security, works that transform source code into Petri nets, and works that use AST to analyze vulnerabilities in source code.

A. Petri nets and security

The use of formal models, and Petri nets in particular, in the context of security is not a new field of research. In 2001, McDermott proposed to model attack scenarios with Petri nets, resulting in a model which the author named *attack net* [10]. These nets describe the possible attacks that program or system can suffer, rather than modeling an actual program or system behavior. Attack nets are used during the development of the tested component to discover plausible scenarios that would compromise the system or program under study.

Particular extensions of Petri nets have also been used to model different attack scenarios, such as timed Petri nets [11] or interval timed colored Petri nets [12]. For instance, in [13], Dahl and Wolthusen described and used a mechanism based on colored Petri nets with time intervals to model the TOCTOU vulnerability and other race condition attacks, both on a single computer and in a distributed environment.

In this work, we propose the use of Petri nets to analyze the occurrence of race conditions produced by external interactions in the vulnerable program itself. External elements will be modeled abstractly. Unlike these works, we are not analyzing distributed systems or environments, but the source code of a single program.

B. Source code and Petri nets

Transformation of the source code into a formal model such as a Petri net has been previously proposed in [14]. In this work, the authors translated C source code files into colored Petri nets to evaluate their behavior and improve intrusion detection systems. They achieve this by extracting information from the relationships between the source files and the Control Flow Graph [15] generated by the GNU C compiler. A Control Flow Graph is the graphical representation of all paths that might be traversed through a program during its execution. This information is later used to build sets of Petri nets that are, in turn, subsequently optimized. The verification of the Petri net model is carried out with a model verifier. Later, a tool that performs this transformation was developed and tested with several existing Unix tools [16].

These mentioned works are similar to ours. In particular, we transform the C source code into Petri nets, using the AST generated by the LLVM compiler as an intermediate model. As we discuss later in Section III, the use of the LLVM-generated AST allows our framework to be easily extended to source code for other programming languages.

C. Vulnerability analysis based on AST

AST have been already proposed as a mechanism to detect vulnerabilities in various works. For instance, Yamaguchi et al. [17] extract AST from the source code and use it to determine structural patterns and check for vulnerabilities. Recently, Feng et al. [18] transform the source code into AST and only the nodes corresponding to the user-defined functions are analyzed, transforming them into vectors that are fed to a recurrent neural network in search of vulnerabilities. Similarly, Bilgin et al. [19] transform the source code of high-level languages like Java or C++ into their corresponding AST which is later converted into features to feed Multi-layer Perceptron and Convolutional Neural Network algorithms. Another example of AST-based vulnerability scanning is [20], where the authors use AST to perform taint analysis (a type of program analysis that traces the flow of user input through a program during its execution) to analyze and detect vulnerabilities in PHP code.

In this work, we propose to transform the source code into AST and then transform it into a formal model like a Petri net. Thus, the detection of vulnerabilities is based on the analysis of the properties of the resulting Petri net. Unlike ours, the aforementioned works are based on machine learning algorithms to predict the existence of vulnerabilities.

III. Methodology

This section briefly introduces our framework for model-based analysis of vulnerabilities in source code. A sketch of how our framework works is shown in Figure 1.

The proposed framework works in two phases. As input, it needs a certain number of source code files. The first phase, *AST Extraction*, is primarily dedicated to transforming the source code files into an AST. The original AST is obtained by relying on the LLVM compiler infrastructure [21], given its modular toolchain. This AST is then analyzed to obtain a refined AST, which only contains the nodes that we identify as relevant for the vulnerability analysis. This refined AST is taken as input for the second phase of the framework, *Model-to-Model Transformation*, which is dedicated to transforming the given input (the refined AST) into a formal model. In particular, we are using Petri nets as formal models. As before, the resulting Petri net model is simplified by applying reduction techniques [22] to discard redundant and unnecessary parts of the net. Since we are exploring the state space of the Petri net model for vulnerabilities, the lower number of possible states, the better. This Petri net model is finally analyzed for vulnerabilities and the analysis results are returned to the user.

As a preliminary case study, we are working with C source files. Therefore, to produce the AST from C source files, we rely on the LLVM's native C/C++/Objective-C compiler called

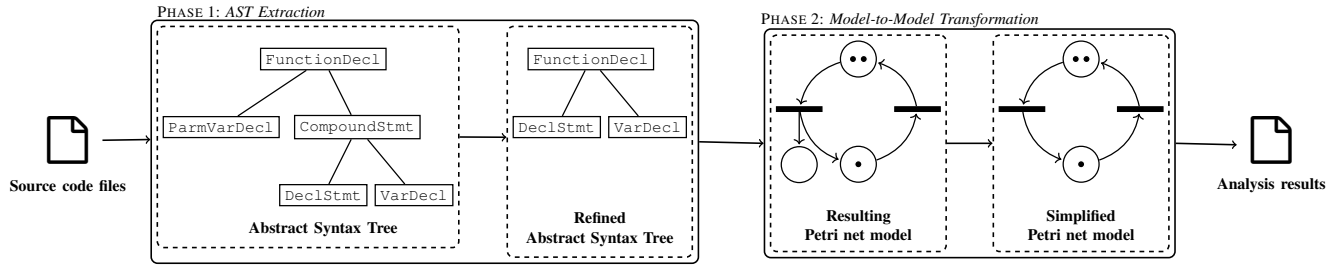


Figure 1. Our proposed framework: from source code to formal model for vulnerability analysis.

Clang [23]. Note that our framework will be able to work with source code files in other programming languages, as long as there is an LLVM-compatible compiler.

To transform the refined AST into a Petri net, we have previously defined certain C code statement pattern transformations, similar to those given in [16]. Our key idea is to transform fundamental structures like flow control and function calls into Petri nets and then combine all of them to obtain a single Petri net corresponding to the source code. As commented before, this resulting Petri net model is processed to reduce the number of places and transitions as much as possible, applying different reduction techniques [22]. This reduced Petri net model is then analyzed for vulnerabilities and the analysis results are ultimately returned to the user. Note that our framework will be able to work with other formal models, as long as there is a transformation of AST elements into the corresponding formal model.

IV. Case Study: Detecting TOCTOU Vulnerabilities

In this section, we describe the application of the proposal to detect race condition vulnerabilities, specifically file-based TOCTOU [24]. Listing 1 illustrates a simple example of a file-based TOCTOU vulnerability. On line 4 there is a check for write permission on a file (identified by a string) with the `access` system call. Once the verification is successful, the file is opened (line 6) and some data is appended to the file. If this program is run with *setuid* permission (that is, users can temporarily run it with other privileges to perform a specific task), the adversary can take advantage of the race window between the operations on lines 4 and 6 to compromise the system and ultimately elevate privileges. These types of vulnerabilities occur on filesystems with *weak* synchronization mechanisms (i.e., they do not provide methods to ensure that filesystem objects remain unchanged between consecutive interactions with them). Given the non-deterministic nature of race conditions, the success of an attack is highly dependent on the attacker's precise and timely actions during the execution of the vulnerable program.

Our goal is to automatically detect these vulnerabilities in source code files that can be several thousand lines of code. First, Clang's AST function is used to transform the input source code into its corresponding AST representation (first step of PHASE 1 in Figure 1). The result of invoking the command `clang -Xclang -ast-dump=JSON source_file.c` is dumped into a JSON file that is then parsed by a Python script to simplify the resulting AST (second step of PHASE 1). This script primarily removes AST nodes that are not of interest for the vulnerability detection analysis and simplifies the AST structure. In Listing 1, the nodes of interest are those that represent control structures, system calls, or references to the files being manipulated. Next, we apply an AST-to-Petri nets transformation based on patterns (PHASE 2). A transformation pattern has been defined for each type of AST node that generates a Petri net fragment as a result. These patters are similar to those proposed in [16]. After all AST nodes have been transformed, the resulting Petri net fragments are combined to obtain a single Petri net. To reduce the complexity of the net and facilitate its analysis, place and transition refinement techniques are applied to the resulting Petri net. Finally, the reachability graph of the reduced Petri net is calculated and a set of Linear Temporal Logic properties are evaluated to detect possible vulnerabilities. This property verification allows us to analyze all the execution histories of the source code to identify those that could suffer a race condition attack. Full automation of this evaluation process is still a work in progress.

Listing 1. Example of a file-based TOCTOU vulnerability in source code).

```
1  char *filename = argv[1];
2
3  // Check permissions
4  if(!access(filename, W_OK)){
5      // Open the file
6      file = fopen(filename, "a+");
7
8      // Write to file the user input
9      fwrite(buffer, sizeof(char),
10                 strlen(buffer), file);
11     fwrite('\n', sizeof(char), 2, file);
12     fclose(file);
13 }else
14     printf("No permission, exiting!\n");
```

V. Discussion and Future Work

The combination of AST representations and Petri net models is a promising approach to detect errors in source code and execution vulnerabilities, as the related work review concludes. The first version of the proposed framework is based on these two formal languages and aims to take advantage the mathematical foundations of both to analyse race condition vulnerabilities. Nevertheless, the framework has been designed to be easily adaptable to work with other formalisms, analysis techniques or programming languages.

We are currently focused on analyzing C source files and the use of the Clang compiler for the generation of the AST. Our next step is to consider C++ as well, which we anticipate will be smooth and nearly effortless given that LLVM provides

native support for the language. In a similar way, once both C and C++ are fully supported, we intend to move on to other languages such as Java, where previous work has already explored the field of static analysis [25], but using other analysis techniques.

Our goal is also to further extend the set of models we use to analyze source files in the first analysis phase. For instance, combining AST with other source code representations such as call graphs [26], control flow graphs [27], program dependence graphs [28], or code property graphs [29] will help refine the analysis of the source code that is performed before its transformation into Petri nets. This, in turn, would imply an improvement in the Petri nets regarding the correspondence between the model and the source code. Additionally, once new models are added and tested, we can assess which models are best suited to detecting different vulnerabilities, which we consider a key contribution of this research.

REFERENCES

[1] M. Hinchey, M. Jackson, P. Cousot, B. Cook, J. P. Bowen, and T. Margaria, "Software engineering and formal methods," *Commun. ACM*, vol. 51, no. 9, pp. 54–59, sep 2008.

[2] J. Jones, "Abstract syntax tree implementation idioms," *Pattern Languages of Program Design*, 2003, proceedings of the 10th Conference on Pattern Languages of Programs (PLoP2003).

[3] E. B. Duffy, B. A. Malloy, and S. Schaub, "Exploiting the clang ast for analysis of c++ applications," in *Proceedings of the 52nd annual ACM southeast conference*, 2014.

[4] C. E. McDowell and D. P. Helmbold, "Debugging Concurrent Programs," *ACM Comput. Surv.*, vol. 21, no. 4, pp. 593–622, dec 1989.

[5] E. Lee, "The Problem with Threads," *Computer*, vol. 39, no. 5, pp. 33–42, 2006.

[6] T. Warszawski and P. Bailis, "ACIDRain: Concurrency-Related Attacks on Database-Backed Web Applications," in *Proceedings of the 2017 ACM International Conference on Management of Data*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 5–20.

[7] J. Yang, A. Cui, S. Stolfo, and S. Sethumadhavan, "Concurrency Attacks," in *Proceedings of the 4th USENIX Conference on Hot Topics in Parallelism (HotPar'12)*. USA: USENIX Association, 2012, p. 15.

[8] T. Murata, "Petri Nets: Properties, Analysis and Applications," in *Proceedings of the IEEE*, vol. 77, no. 4, April 1989, pp. 541–580.

[9] M. Bishop and M. Dilge, "Checking for Race Conditions in File Accesses," in *Computing Systems*, vol. 9, no. 2, 1996, pp. 131–152.

[10] J. P. McDermott, "Attack Net Penetration Testing," in *Proceedings of the 2000 Workshop on New Security Paradigms*, ser. NSPW '00. New York, NY, USA: Association for Computing Machinery, 2001, pp. 15–21.

[11] W. Zuberek, "Timed petri nets definitions, properties, and applications," *Microelectronics Reliability*, vol. 31, no. 4, pp. 627–644, 1991.

[12] W. M. P. van der Aalst, "Interval timed coloured petri nets and their analysis," in *Application and Theory of Petri Nets 1993*, M. Ajmone Marsan, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 453–472.

[13] O. M. Dahl and S. D. Wolthusen, "Modeling and execution of complex attack scenarios using interval timed colored petri nets," in *Proceedings of the 4th IEEE International Workshop on Information Assurance (IWIA 2006), 13-14 April 2006, Egham, Surrey, UK*. IEEE Computer Society, 2006, pp. 157–168.

[14] J. Voron and F. Kordon, "Transforming sources to petri nets: a way to analyze execution of parallel programs," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, SimuTools 2008, Marseille, France, March 3-7, 2008*, S. Molnár, J. R. Heath, O. Dalle, and G. A. Wainer, Eds. ICST/ACM, 2008, p. 13.

[15] F. E. Allen, "Control flow analysis," in *Proceedings of a Symposium on Compiler Optimization*. New York, NY, USA: Association for Computing Machinery, 1970, pp. 1–19.

[16] J.-B. Voron and F. Kordon, "Evinrude: A Tool to Automatically Transform Program's Sources into Petri Nets," *Petri Net Newsletter*, vol. 75, pp. 19–38, Oct. 2008. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01175966

[17] F. Yamaguchi, M. Lottmann, and K. Rieck, "Generalized vulnerability extrapolation using abstract syntax trees," in *Proceedings of the 28th Annual Computer Security Applications Conference*, ser. ACSAC '12. New York, NY, USA: Association for Computing Machinery, 2012, pp. 359–368.

[18] H. Feng, X. Fu, H. Sun, H. Wang, and Y. Zhang, "Efficient vulnerability detection based on abstract syntax tree and deep learning," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, July 2020, pp. 722–727.

[19] Z. Bilgin, M. A. Ersoy, E. U. Soykan, E. Tomur, P. Çomak, and L. Karaçay, "Vulnerability prediction from source code using machine learning," *IEEE Access*, vol. 8, pp. 150 672–150 684, 2020.

[20] A. Kurniawan, B. S. Abbas, A. Trisetyarso, and S. M. Isa, "Static taint analysis traversal with object oriented component for web file injection vulnerability pattern detection," *Procedia Computer Science*, vol. 135, pp. 596–605, 2018, the 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life.

[21] C. Lattner and V. Adve, "LLVM: A compilation framework for lifelong program analysis and transformation," San Jose, CA, USA, Mar 2004, pp. 75–88.

[22] S. Haddad, "A reduction theory for coloured nets," in *Advances in Petri Nets 1989*, G. Rozenberg, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 209–235.

[23] LLVM, "Clang: a C Language Family Frontend for LLVM," https://clang.llvm.org/, (Accessed Apr 6, 2022).

[24] R. Raducu, R. J. Rodríguez, and P. Alvarez, "Defense and Attack Techniques against File-based TOCTOU Vulnerabilities: a Systematic Review," *IEEE Access*, vol. 10, pp. 21 742–21 758, 2022.

[25] V. B. Livshits and M. S. Lam, "Finding security vulnerabilities in java applications with static analysis." in *USENIX security symposium*, vol. 14, 2005, pp. 18–18.

[26] D. DaCosta, C. Dahn, S. Mancoridis, and V. Prevelakis, "Characterizing the 'security vulnerability likelihood' of software functions," in *Proceedings of the International Conference on Software Maintenance*, ser. ICSM '03. USA: IEEE Computer Society, 2003, p. 266.

[27] S. S. Anju, P. Harmya, N. Jagadeesh, and R. Darsana, "Malware detection using assembly code and control flow graph optimization," in *Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India*, ser. A2CWiC '10. New York, NY, USA: Association for Computing Machinery, 2010.

[28] A. Johnson, L. Waye, S. Moore, and S. Chong, "Exploring and enforcing security guarantees via program dependence graphs," in *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI '15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 291–302.

[29] F. Yamaguchi, N. Golde, D. Arp, and K. Rieck, "Modeling and discovering vulnerabilities with code property graphs," in *2014 IEEE Symposium on Security and Privacy*, 2014, pp. 590–604.

(Work in Progress) FATR: a Framework for Automated Analysis of Threat Reports

Juan Caballero†, Gibran Gómez†◇, Srdjan Matic†, Gustavo Sánchez†, Silvia Sebastián†◇, and Arturo Villacañas†
†IMDEA Software Institute
◇Universidad Politécnica de Madrid

***Abstract*—To adapt to a constantly evolving landscape of cyber threats, organizations actively need to collect Indicators of Compromise (IOCs), i.e., forensic artifacts that signal that a host or network might have been compromised. IOCs are used to gain visibility into the fast-evolving threat landscape, timely identify early signs of attacks, and develop adequate countermeasures. IOCs can be collected through commercial and open-source IOC feeds. But, they can also be extracted from security reports distributed using a wide array of sources such as blogs and social media. These sources need to be continuously monitored for updates and do not impose any limitations on the type and format of the shared information. To address these challenges, this paper presents FATR, a work-in-progress automated framework for collecting security reports from RSS feeds and social media such as Telegram and Twitter, and to extract IOCs from those reports. FATR can deal with multiple document types such as HTML, PDF, and text files, and can extract over 40 IOC types. We have run FATR for one year, extracting over 248k unique IOCs.**

***Index Terms*—Indicators of Compromise, IOC, Cyber Threat Intelligence, RSS, Twitter, Telegram**

I. Introduction

Cyber Threat Intelligence (CTI) is defined as "the set of knowledge, skills and experience-based information intended to help mitigate potential attacks and harmful events occurring in cyberspace" [1]. Access to CTI information is crucial for any organization that wants to gain visibility into the fast-evolving threat landscape, timely identify early signs of attacks, and develop adequate countermeasures. An essential piece of CTI is the extraction of *Indicators of Compromise* (*IOCs*), forensic artifacts representing, among others, malicious IPs, domains, and file hashes. Traditionally, IOCs have been collected and made available through commercial and open-source feeds, typically focused on one or a small set of indicator types (e.g., [2], [3]). In recent years, social media has turned into an effective medium for exchanging and spreading cybersecurity information. Social media is widely used, not only by companies in the security business, but also by cybersecurity experts that often rush to share their discoveries [4]. An ever-growing number of threat-related posts are published on social media, providing insights about new vulnerabilities, malware, and attacks. For example, Twitter users frequently report blockchain addresses related to malicious campaigns such as those appearing in ransomware notes. These addresses can be collected and used to investigate how cybercrime profits flow across different wallets and services [5]. Security vendors have increasingly been extracting IOCs from threat reports to power protection systems. Unfortunately, while commercial CTI feeds follow consistent schemes and data formats, social media does not set any burden on the format of the information that is shared. Moreover, multiple reports can focus on different aspects of the same threat, providing complementary information. In this work, we address these challenges by building an automated platform for gathering security reports from RSS feeds, Twitter, and Telegram, and automatically extracting IOCs from the collected reports.

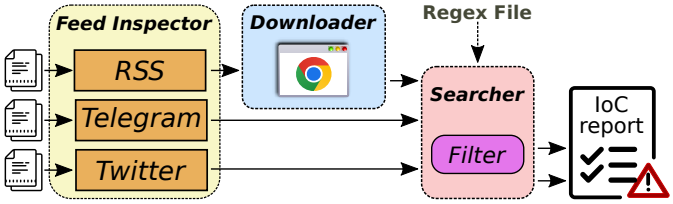


Figure 1. The architecture of FATR.

While the project is a work-in-progress, we can already report some contributions:

- We develop FATR, an automated framework to collect security reports distributed through RSS, Twitter, and Telegram.
- FATR extracts over 40 IOC types from HTML pages, PDF documents, and text files.
- We provide a quantitative assessment of FATR by extracting and analyzing 248k IOCs from documents collected during one year of monitoring.

II. Approach

Figure 1 depicts the architecture of FATR. It consists of three main modules: the *Feed Inspector*, the *Downloader*, and the *Searcher*. The Feed Inspector uses three independent components that collect security reports from RSS blogs, Telegram channels, and Twitter accounts. Each component takes as input a list of *origins* and outputs a list of tuples, that contain both the origin and the *entries* fetched from that origin. For RSS, the origin is the feed and the entries are the RSS entries the feed provides. Each RSS entry contains the URL of a report as well as report metadata such as the publication date and the title. For Telegram, the origin is the channel and the entries are the messages posted on the channel. For Twitter, the origin is the account and the entries are the account tweets. The Downloader leverages an instrumented browser to fetch the document pointed by the URL in an RSS entry. The Telegram and Twitter components do not need to use the Downloader, as their entries already contain the content of messages and tweets. Finally, the Searcher applies regular expressions to extract IOCs from the content. Before generating the final list of IOCs, the Searcher filters generic values that do not correspond to real IOCs.

Table I
LIST OF INDICATORS AND THE CORRESPONDING CLASS.

Class	Indicators
contact	• **User contact:** email, phone number.
social	• **Handle IDs:** Facebook, GitHub, Instagram, LinkedIn, Pinterest, Skype, Telegram, Twitter, WhatsApp, YouTube. • **Channel IDs:** Youtube.
file	• **Hashes:** md5, sha1, sha256.
network	• **Network information:** Fully Qualified Domain Name, Effective Second Level Domain, Internet Content Provider number, IP address, CIDR format IP range, Tor onion address, URL.
analytics	• **Advertiser IDs:** Google Adsense, Google Analytics, Google Tag Manager.
ipr	• **Text:** copyright, trademark.
blockchain	• **Digital wallets:** Bitcoin, Bitcoin Cash, Dashcoin, Dogecoin, Ethereum, Litecoin, Monero, Tezos, Zcash.
payment	• **Accounts:** IBAN, WebMoney.
vulnerability	• **IDs:** CVE.

Table I summarizes the 44 IOC types that FATR extracts. IOCs are grouped into 9 classes that cover indicators associated to users (*e.g.,* phone numbers, IBANs or LinkedIn profiles), malware (*e.g.,* file hashes and CVEs), endpoints (*e.g.,* IP addresses and domain names), advertising (*e.g.,* Google Adsense and Google Analytics identifiers), and intellectual property (*e.g.,* copyright strings and trademarks).

Feed Inspector. The RSS module receives as input a list of RSS feeds. Feeds are constantly updated, and at a given moment each feed provides only a limited number of entries (*e.g.,* the 10 most recent ones). To minimize the risk of missing entries, the RSS module queries each feed on a daily basis. Between two consecutive visits, a feed might not have been updated, therefore the Inspector uses an incremental approach to avoid storing the same entry multiple times. After fetching a feed, the module extracts the unique URLs in the entries and passes them to the Downloader module. URLs that were previously crawled can be ignored to avoid downloading the same content multiple times.

The Telegram and Twitter modules leverage dedicated APIs to query each service. Using these APIs they can fetch not just the most recent messages, but any message that was ever posted on an account or channel. The two modules operate similarly. Each of them takes as input a list of origins (*i.e.,* Telegram channels or Twitter accounts) that should be monitored, and connect to each origin on a daily basis to fetch all new entries (*i.e.,* tweets or messages). A tweet or Telegram message is only collected once through the API. However, it is possible to collect tweets with the same content from multiple accounts, e.g., if they re-tweet the same original message.

We manually created our initial origin lists by including RSS feeds, Twitter accounts, and Telegram channels for prominent companies, cybersecurity new websites, and well-known security experts.

Downloader. The Downloader takes as input a URL extracted from an RSS entry, and tries to download the content pointed to by the URL. The module retrieves content using Selenium, a popular framework used for testing Web applications [6]. We instrument Selenium to render URLs within a fully fledged instance of Google Chrome. The Downloader is able to follow redirects, supports dynamic content executed with JavaScript, and, in addition to HTML pages, it can also handle plain text documents as well as other MIME types such as PDFs. In case our instrumented browser did not succeed in retrieving the content, the Downloader makes an additional attempt with the *python requests* library [7]. For successfully retrieved content, the Download stores the document, the URL, and the origin (*i.e.,* feed) from where the entry was obtained. Downloaded documents are then filtered to remove downloads with HTTP status code errors and those where the title states the webpage was not found. Then, it extracts the readable text of the document. For webpages, it uses Mozilla's Readability.js library, also used by Firefox's Reader View [8]. For PDF files, it uses the pdfminer.six library [9].

Searcher. The Searcher takes as input the text from an entry (*i.e.,* document, tweet, Telegram message) and uses regular expressions to identify IOCs. We use a regular expression for each of the 44 supported IOCs. We choose regular expressions because they are an efficient technique for identifying, in a given string, IOCs with some intrinsic structure such as email addresses or URLs. To extract IOCs, the Searcher applies a threefold approach: matching, validation, and filtering. During the matching process, the Searcher applies the regular expression to identify candidate IOCs. The validation uses a function specific to each IOC type to validate that the candidate is indeed an IOC. Separating matching from validation helps in preventing regular expressions from becoming too complex; complex regular expressions are hard to understand and to manage over time when tweaks are required. For example, a common validation ensures that a domain name (*fqdn*) contains a valid top-level domain (TLD). It is possible to build *fqdn*, *url*, and *email* regular expressions capturing all valid TLDs; however, the IANA TLD list [10] already contains over 1,500 TLDs and may still grow. Thus, it is simpler to take as input a file with the IANA approved TLDs and to check that the candidate *fqdn* indeed has a valid TLD in that list. The advantage of this approach are shorter regular expression that are easier to maintain. Another example are phone numbers extracted using Google's *libphonenumber* library [11]. To check if a sequence of digits is a phone number in a national format (*i.e.,* without an international prefix), we need to know the country to which the number belongs to. We address this issue using the *langdetect* library [12] to identify the document language, and then we map the language to a set of possible countries. Finally, we validate if a candidate phone number is in the correct format for a given country. In case of blockchain addresses (*e.g.,* Bitcoin, Ethereum), they typically embed a checksum, which can be validated. For example, for Bitcoin addresses the validation computes the checksum of the first 21 bytes and checks that it matches the last 4 bytes in the candidate address. We use the *coinaddr* library to validate blockchain addresses.

A limitation of regular expressions is that it is hard to build them for indicators without a well-defined structure such as physical addresses and organization names. Moreover, regular expressions for different IOCs could occasionally match the same string. Some examples are network prefixes in CIDR format, which have the same structure of a URL with an IP address and without scheme (e.g., "1.2.3.4/24"); hashes that match blockchain addresses (e.g. md5 hashes that match the

Table II
DATA COLLECTION SUMMARY.

Source	Orig.	Start Date	Entries	IOCs (*pre-filtering*)
rss	278	2021/04/01	45,719	155,000 (*235,583*)
telegram	8	2021/09/09	24,542	23,062 (*30,999*)
twitter	383	2021/10/28	279,761	70,858 (*229,522*)

Bitcoin address format); and addresses that could be valid in multiple blockchains (e.g. Bitcoin and Litecoin).

We design our validation to minimize the likelihood of identifying incorrect IOCs and of producing multiple IOCs of different types for the same value. Unfortunately, the set of candidate IOCs might contain some generic string values that pass the validation. The Searcher implements filtering to remove such *generic indicators*. First, it considers generic any indicator that embeds a domain name that appears in the lists of URLs provided as input to the Feed Inspector and the Downloader. This same filtering rule is applied with the domains from the Tranco top-100K list [13]. Next, the filtering excludes IP addresses and network ranges of private or local networks (*e.g.,* "192.168.1.0/24"). Finally, to remove contact emails that often appear in security reports, the Searcher does not consider as IOCs email addresses of known providers[1] that appeared more than 10 times in the same origin.

III. EVALUATION

Table II summarizes the contribution of each source in terms of collected entries and extracted IOCs. RSS is both the oldest and the most stable source, and contributes on average with more than 3.8k new URLs every month. Telegram is the most dynamic source, and every month each one of the 8 monitored channels generate around 400 new messages. The most recently added source is Twitter, which since October 2021 contributed with around 280k unique tweets.

The last column of Table II shows the IOCs before (in brackets) and after applying the filtering. Filtering is extremely important to avoid that the extracted IOCs contain generic indicators. In the case of Twitter, more than two thirds of the candidate indicators are removed by the filtering. Each source is affected differently by the filtering: for RSS and Telegram almost 75% of the indicators are removed because they self-reference a domain name that appears in our list of sources. In the case of Twitter, 158k indicators are excluded using the rule that checks if the indicator contains a domain name from the Tranco list.

Of the three sets of sources, RSS provides the highest amount of IOCs, with an average of 3.5 IOCs per each successfully downloaded report. This ratio drops to 1 for Telegram, and around 0.25 in case of Twitter. A likely explanation for the higher contribution of RSS feeds is that webpages usually have much more content than tweets and Telegram messages. Moreover, the majority of our RSS feeds come from blogs of cybersecurity companies and magazines, which are known for curated and high-quality content.

Figure 2 reports the IOCs grouped by classes and sources, after applying the filtering. IOCs from the *network* class are the most popular across all sources. The source with the

[1]Our manually curated list includes: Gmail, Protonmail, Yahoo, AOL, Tutanota, Hotmail, QQ, Mail.ru, Outlook, Yandex

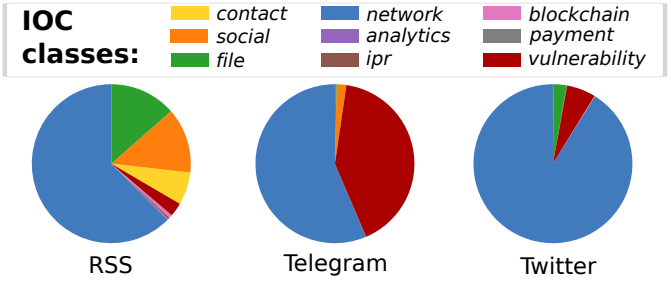


Figure 2. Distribution of IOC classes according to the different sources.

largest variety of IOCs is RSS, where 38% of the IOCs are not from the *network* class. For Telegram, our monitored channels generated IOCs that are either URLs, fully qualified domain names, IP addresses, or CVE identifiers. Interestingly, together the *contact* and *social* classes represent only 2% of the IOCs in Telegram, and they account only for 68 of the IOCs extracted from Twitter.

In Figure 3 we report how different sources contribute to each IOC class. We observe a long tail distribution, where 10% of origins generate more than 80% of the IOCs. This behavior is common across all of the sources, and IOC classes. The IOCs extracted from Telegram and Twitter are heavily dominated either by the *network* or *vulnerability* classes. In Telegram, a single channel @cybsecurity is responsible for 70% and 90% of IOCs from those two classes. In Twitter, two accounts dominate these two classes: @threatmeter that accounts for 90% of the *vulnerability* IOCs, and @ecarlesi that is responsible for 70% of the IOCs related to *network*. @threatmeter is an automated account, with no followers and focused on publicizing CVEs. On the other hand, @ecarlesi is a private account with more than 140k tweets. The IOCs originating from the RSS feeds are the most varied both with respect to the classes and to the number of origins that generate them. In this case we observe less specialization than for the other sources, but still a limited number of origins produce the majority of the indicators. For example, Dancho Danchev's blog is responsible for 30% of the IOCs including more than half of IOCs belonging to the *network* and *contact* classes. Among the top-10 IOC contributors we find: four websites on the Medium platform, two news websites of security companies, two news aggregators focused on cryptocurrencies, and a security blog.

IV. RELATED AND FUTURE WORK

Over time, several threat intelligence tools and platforms have been developed [14]–[21]. These platforms focus on the automatic extraction of IOCs from different sources including technical articles [14]–[17], social networks [22], [23], and public/private IOC feeds [24] . Regular expressions are a common choice for IOC extraction, as most IOCs follow a well defined structure that can be matched, *e.g.,* IP addresses are formatted as a set of four base-10 written hexadecimal bytes separated by dots. Commonly, such platforms focus on a limited set of IOCs (URLs, IP addresses, and file hashes). Our approach also relies on regular expressions to detect IOCs, but we cover a significantly larger number of IOC classes (*i.e.,* network and social handles, contact information, CVEs), and support a variety of document formats (HTML, PDF, plain

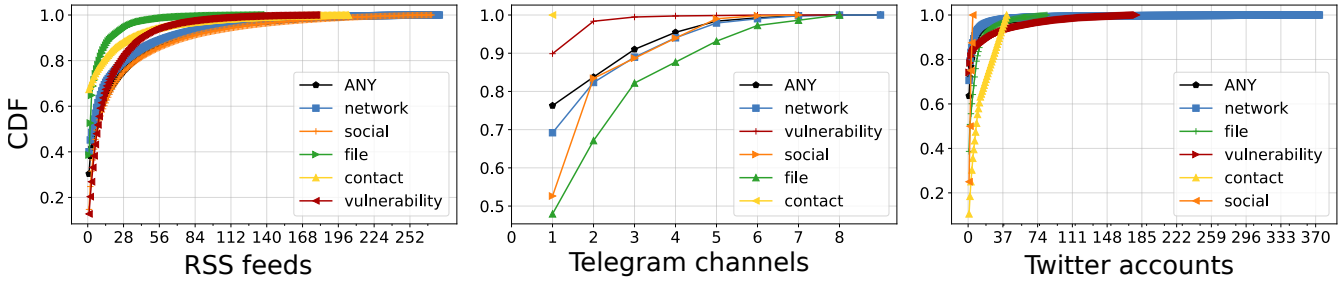


Figure 3. Contributions of each set of sources in relation to the different classes of IOCs identified.

text). Moreover, FATR uses a real web browser to download documents, allowing us to follow complex redirection chains and render dynamic content. A common challenge is to select the best sources to track. Niakanlahiji et al. [22] combine graph theory, machine learning (ML), and text mining to build reputation models to identify valuable Twitter users to follow, based on their posts. Li et al. [24] define a special set of metrics that can be used to measure the quality of an IOC feed. We start with a manually generated list of origins, and then perform the analysis in Section III to identify the best and worst contributors. This is possible because FATR can backtrack each IOC to its origin, and can aggregate them to identify the most active sources, as well as sources that specialize on particular indicator types. We can easily include new sources to our framework, and remove those that are less active, or that do not generate IOCs. In addition, the validation of the IOCs found is considered an open problem. For example, in [18] authors propose to identify web-IOCs by searching for external resources added to a web application once it gets compromised. Still, it is extremely challenging to figure out that an external resource (*e.g.,* a URL or a JavaScript snippet) is indeed an indicator of the attack, and not just a generic indicator that appears also in non-compromised web sites. Our platform leverages the insight that some IOCs can be validated, significantly lowering generic indicators, *e.g.,* Bitcoin addresses can be easily validated using their embedded checksum. In addition, FATR also includes filtering to remove generic IOCs, *e.g.,* a URL whose domain is the same as an origin we track.

V. Conclusions

In this paper we present FATR, a work-in-progress platform for the automatic extraction of IOCs in security reports obtained from RSS feeds and social media. FATR can deal with multiple document types such as HTML, PDF, and text files, extracts 44 IOC types, and performs both validation and filtering of the extracted IOCs. We run FATR for one year, collecting over 248k unique IOCs. An open challenge is how to identify and extract the context related to the IOCs, in order to enrich the IOCs and increase their threat intelligence value. Prior work tackles this challenge through natural language processing (NLP) approaches that search for grammatical connections between IOCs and surrounding terms, as well as by training ML classifiers to categorize the IOCs. In our future work, we plan to explore similar approaches based on NLP and ML, both to detect the context of IOCs, and to improve the selection of sources.

References

[1] Bank of England, https://www.bankofengland.co.uk/-/media/boe/files/financial-stability/financial-sector-continuity/understanding-cyber-threat-intelligence-operations.pdf, 2016. 1
[2] CrowdStrike, 2022, https://www.crowdstrike.com/falcon-platform/. 1
[3] PhishTank, 2022, https://phishtank.org/. 1
[4] C. Sabottke, O. Suciu, and T. Dumitras, "Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits," in *USENIX Security*, 2015. 1
[5] G. Gomez, P. Moreno-Sanchez, and J. Caballero, "Detecting Cybercriminal Bitcoin Relationships through Backwards Exploration," Tech. Rep., August 2022. [Online]. Available: https://arxiv.org/abs/2206.00375 1
[6] Software Freedom Conservancy, https://www.selenium.dev/, 2022. 2
[7] Python Software Foundation, https://github.com/psf/requests, 2022. 2
[8] Mozilla, https://github.com/mozilla/readability, 2022. 2
[9] Y. Shinyama, P. Guglielmetti, and P. Marsman, https://github.com/pdfminer/pdfminer.six, 2019. 2
[10] Internet Assigned Numbers Authority, https://data.iana.org/TLD/tlds-alpha-by-domain.txt, 2022. 2
[11] D. Drysdale, "phonenumbers python library," https://github.com/daviddrysdale/python-phonenumbers, 2022. 2
[12] M. Danilák, https://github.com/Mimino666/langdetect, 2021. 2
[13] V. L. Pochat, T. V. Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *NDSS*, 2019. 3
[14] X. Liao, K. Yuan, X. Wang, Z. Li, L. Xing, and R. Beyah, "Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence," in *CCS*, 2016. 3
[15] Z. Zhu and T. Dumitras, "Chainsmith: Automatically learning the semantics of malicious campaigns by mining threat intelligence reports," in *Euro S&P*, 2018. 3
[16] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "Ttpdrill: Automatic and accurate extraction of threat actionsfrom unstructured text of cti sources," in *ACSAC*, 2017. 3
[17] J. Zhao, Q. Yan, J. Li, M. Shao, Z. He, and B. Li, "Timiner: Automatically extracting and analyzing categorized cyber threat intelligence from social data," *Computers & Security*, 2020. 3
[18] O. Catakoglu, M. Balduzzi, and D. Balzarotti, "Automatic extraction of indicators of compromise for web applications," in *WWW*, 2016. 3, 4
[19] InQuest, https://github.com/InQuest/ThreatIngestor, 2020. 3
[20] InQuest, https://github.com/InQuest/python-iocextract, 2019. 3
[21] A. Buescher, https://github.com/armbues/ioc_parser/, 2017. 3
[22] A. Niakanlahiji, L. Safarnejad, R. Harper, and B.-T. Chu, "Iocminer: Automatic extraction of indicators of compromise from twitter," in *IEEE Big Data*, 2019. 3, 4
[23] H. Shin, W. Shim, S. Kim, S. Lee, Y. G. Kang, and Y. H. Hwang, "Twiti: Social listening for threat intelligence," in *WWW*, 2021. 3
[24] V. G. Li, M. Dunn, P. Pearce, D. McCoy, G. M. Voelker, and S. Savage, "Reading the tea leaves: A comparative analysis of threat intelligence," in *USENIX Security*, 2019. 3, 4

QuantumSolver: Librería para el desarrollo cuántico

José Daniel Escánez-Expósito
Universidad de La Laguna
Tenerife, Spain
jdanielescanez@gmail.com

Pino Caballero-Gil
Universidad de La Laguna
Tenerife, Spain
pcaballe@ull.edu.es

Francisco Martín-Fernández
IBM Research
NY, USA
paco@ibm.com

***Resumen*—En este documento se presenta una breve descripción de la propuesta en desarrollo de un *toolset* cuántico *opensource*, llamado *QuantumSolver*, con licencia MIT. La herramienta desarrollada incluye varios algoritmos con distintas funcionalidades, como la generación de números aleatorios, la resolución del problema de Bernstein-Vazirani y la distribución de claves cuánticas (protocolo BB84). Se exponen aquí los principales detalles de la implementación del *toolset*, así como algunas conclusiones obtenidas de la investigación realizada de las funcionalidades incluidas.**

***Index Terms*—Computación cuántica, Qiskit, Números aleatorios, Algoritmo de Bernstein-Vazirani, Protocolo BB84**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

El interés en las tecnologías relacionadas con la computación cuántica ha crecido notablemente en los últimos años, cobrando el tema un gran auge hoy en día. Su utilidad para vulnerar los algoritmos criptográficos actuales ha generado la ya imperiosa necesidad de la creación de nuevos protocolos seguros para las comunicaciones. Por otra parte, dadas algunas de sus curiosas propiedades, está claro que el fomento de este modelo de computación generará importantes avances tecnológicos para el conjunto de la sociedad [1].

El principal objetivo de este trabajo es fomentar el uso de tecnologías cuánticas mediante el desarrollo de un *toolset opensource* de algoritmos cuánticos, con repositorio público en GitHub y licencia MIT [2], persiguiendo la abstracción y el encapsulamiento sencillos de *software* cuántico con diferentes funcionalidades. Entre las librerías que han implementado algunos de los algoritmos y protocolos que se tratan aquí destacan las publicadas en [3] y [4].

QuantumSolver está dirigido tanto a un usuario totalmente ajeno a la informática, que, por ejemplo, desee obtener un número aleatorio gracias a la computación cuántica; como a un programador experimentado, que, por ejemplo, aspire a contribuir en la implementación de esta librería. Debido a que la propuesta pretende satisfacer a una amplia variedad de posible público de los programas ejecutables disponibles, se ha desarrollado la posibilidad de ejecución del software por medio de dos interfaces diferentes: Interfaz por Línea de Comandos (*Command Line Interface*, CLI) e Interfaz Web. Esta última está más orientada al público general, y permite que cualquier usuario pueda ejecutar algoritmos cuánticos en *hardware* cuántico real, sin tener ningún tipo de experiencia previa con la programación informática.

II. Detalles de la implementación

QuantumSolver es una librería cuántica desarrollada en Python3 gracias a *Qiskit*, que es el SDK de código abierto que IBM ofrece para trabajar con ordenadores cuánticos a nivel de pulsos, circuitos y módulos de aplicación [5]. Cuenta con dos componentes principales: *QExecute* y *QAlgorithmManager*.

II-A. QExecute

QExecute es el motor de ejecución de *QuantumSolver*. Se encarga de la autenticación contra los servicios de IBM, que ofrecen acceso a su *hardware* (tanto *hardware* cuántico real como simuladores) por medio de un *API token* de *"IBM Quantum Experience"* [6] [7]. Además cuenta con un modo invitado para no generar la inevitable necesidad al usuario de obtener el *token* teniendo que crear una cuenta en *IBM Quantum*. En este modo solo se permite ejecutar haciendo uso del simulador local *'aer_simulator'*, por lo que no se podrá utilizar el *hardware* cuántico real proporcionado por IBM. *QExecute* cuenta con métodos para la visualización del listado de los *backends* disponibles y la selección del deseado para realizar la ejecución. Además, es el componente encargado de realizar la propia ejecución de los circuitos cuánticos.

II-B. QAlgorithmManager

QAlgorithmManager es el gestor de algoritmos cuánticos de *QuantumSolver*. Se encarga de agrupar y listar todos los algoritmos disponibles, además de seleccionar el que se desee ejecutar. También permite gestionar los argumentos de los diferentes algoritmos y del intercambio de información entre ellos y el programa principal.

II-C. QAlgorithm

QAlgorithm es la entidad que se corresponde con un algoritmo cuántico cualquiera. Se trata de una clase abstracta que puede servir como plantilla para añadir de manera intuitiva un nuevo algoritmo a la librería. Cualquier entidad válida derivada de esta representa un algoritmo que *QuantumSolver* puede ejecutar. Estas entidades, siguiendo la plantilla de *QAlgorithm*, contienen información relevante sobre el algoritmo en cuestión: nombre, descripción, parámetros, maneras en que se debe analizar y tratar el resultado de la ejecución del circuito, y en que se deben interpretar y comprobar los parámetros que sean introducidos como una lista de cadenas de texto. El método principal de la entidad es la generación parametrizada del circuito cuántico correspondiente al algoritmo.

III. Programa principal

En la pantalla de inicio, el programa principal de *QuantumSolver* ofrece las alternativas de, o bien ejecutar el modo

invitado, o bien autenticarse usando un *API token* de IBM. En cualquier caso, se despliega un menú que contiene las opciones de visualización y selección de los *backends* y algoritmos disponibles. Una vez elegido un algoritmo, se solicita la introducción de los parámetros ligados a él. Cuando *backend*, algoritmo y parámetros han sido establecidos, se despliegan dos opciones. Por una parte, se permite ejecutar el algoritmo una única vez y obtener el resultado, además de una representación gráfica del circuito. Por otra parte, es posible ejecutar el algoritmo varias veces para observar su comportamiento representado en un histograma generado. A esta última opción se la ha denominado modo experimental.

IV. Interfaces

Para *QuantumSolver* se ha desarrollado una versión web que cuenta con un *backend* en Python3, utilizando el *framework* Flask, y un *frontend* usando TypeScript, React, HTML5 y CSS. De las dos interfaces ofrecidas para ejecutar *QuantumSolver*, la interfaz web (ver Fig. 1) es más intuitiva para el público general que la basada en línea de comandos (ver Fig. 2), reuniendo ambas las mismas funcionalidades.

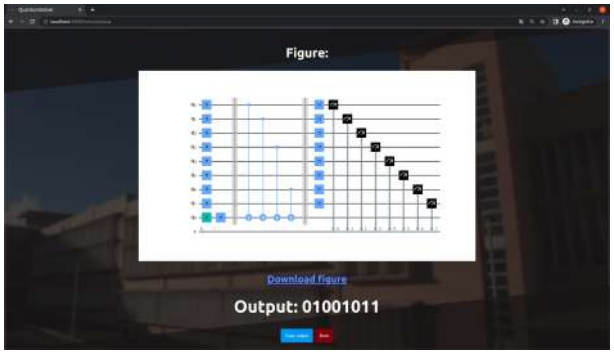


Figura 1. Interfaz web de *QuantumSolver*

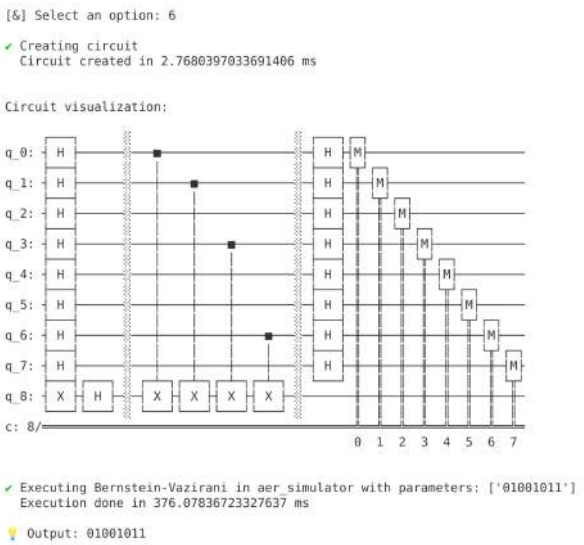


Figura 2. Interfaz *CLI* de *QuantumSolver*

V. Generación de números aleatorios

El algoritmo cuántico *QRand* implementado en *QuantumSolver* recibe como parámetro un número natural (n) y permite generar un circuito de cuya ejecución resulta un número aleatorio entre 0 y $2^n - 1$. Su funcionamiento se basa en la inicialización de n cúbits, por defecto a $|0\rangle$; la aplicación de una puerta lógica Hadamard [8] [9] a cada uno, para generar un estado de superposición en el que se tenga la misma probabilidad de medir 0 o 1 (ver Ec. (1)); y finalmente la medición del resultado, haciendo colapsar cada cúbit en un estado aleatorio e interpretándose como un número binario.

$$|00\ldots0\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \tag{1}$$

VI. Algoritmo de Bernstein-Vazirani

El algoritmo cuántico de Bernstein-Vazirani [10] incluido en *QuantumSolver* recibe como parámetro una clave formada por una cadena binaria de tamaño n, que se utiliza para codificar un oráculo que brinda información acerca de dicha clave. Concretamente, ante una cadena candidata, la información que devuelve el oráculo es la confirmación de si el número de coincidencias de 1 entre las cadenas clave y candidata es par o impar (ver Ec. (2)).

$$f_s(x) = (s * x)(mod\ 2) \tag{2}$$

La Ec. (2) refleja que el oráculo realiza el producto binario entre las parejas de bits de la clave a adivinar y la cadena candidata, y luego le aplica una puerta XOR o suma módulo 2 a los n bits resultantes. Clásicamente, se podría resolver este problema con n consultas al oráculo.

$$\begin{cases} f_s(100\ldots0) = s_0 \\ f_s(010\ldots0) = s_1 \\ f_s(001\ldots0) = s_2 \\ \ldots \\ f_s(000\ldots1) = s_{n-1} \end{cases} \tag{3}$$

En el caso cuántico se necesita una única consulta al oráculo, que devolverá correctamente la clave con un 100 % de probabilidad (sin contar con posibles errores de ruido generados por el *hardware*).

La implementación realizada del algoritmo [11] requiere $n+1$ cúbits. De ellos, n son para codificar la entrada (formada por n cúbits con valor $|0\rangle$). El restante es adicional, para la salida del oráculo cuántico (inicializado con el valor $|1\rangle$, obtenido al aplicar una puerta cuántica X [9] a un cúbit que por defecto tiene el valor $|0\rangle$). Además, se deberán aplicar puertas lógicas Hadamard a los $n+1$ cúbits antes y después del oráculo; excepto a la salida del mismo, que no lo precisará dado que no será medida. Internamente, este oráculo debe implementarse aplicando puertas *CNOT* con control en aquellos cúbits que se correspondan con los bits de la clave que estén a 1, y objetivo en la salida del oráculo. Esta puerta *CNOT* es el "equivalente" al *XOR* clásico.

La caracterización del oráculo, aplicado sobre la cadena x candidata, se muestra en la Ec. (4).

$$|x\rangle \xrightarrow{f_s} (-1)^{s\cdot x}|x\rangle \tag{4}$$

La transformación realizada de los n cúbits que codifican la entrada $|00\ldots0\rangle$ al aplicarles las puertas Hadamard de la inicialización se muestra en la Ec. (5).

$$|00\ldots0\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \tag{5}$$

La aplicación del oráculo cuántico se muestra en la Ec. (6).

$$\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \xrightarrow{f_a} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{a\cdot x}|x\rangle \tag{6}$$

El paso final, aplicando a cada cúbit anterior una puerta Hadamard, se muestra en la Ec. (7).

$$\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{a\cdot x}|x\rangle \xrightarrow{H^{\otimes n}} |a\rangle \tag{7}$$

Se observa que, al aplicar las operaciones descritas, el resultado es la clave codificada en el oráculo.

VII. Protocolo BB84

El protocolo criptográfico BB84 [12] para la distribución de claves cuánticas forma parte de la librería *QuantumSolver*.

VII-A. Entidades

Se ha implementado una entidad principal (*Participant*) y sus entidades derivadas (*Sender* y *Receiver*). Se supone que una instancia de la clase *Sender* quiere comunicarse con otra de la clase *Receiver*, de forma que la comunicación sea secreta gracias al protocolo BB84, que permite la generación de una libreta de un solo uso que solo comparten emisor y receptor. La simulación del canal cuántico se describe mediante circuitos cuánticos de Qiskit [13]. La única diferencia entre las entidades *Sender* y *Receiver* es que la primera tiene un método para enviar un mensaje (inicializando un circuito cuántico) y la segunda para recibirlo (añadiendo la fase de medición al circuito). La clase base *Participant* contiene los métodos para la generación y muestra de valores, ejes, claves y libretas de un solo uso, entre otros.

VII-B. Fundamento

La entidad emisora escoge al azar valores para codificar cada uno de los cúbits a transmitir y ejes en los que codificarlos, resultando las posibilidades indicadas en la Tabla I.

Tabla I
Posibilidades valor - eje y circuitos asociados

Valor	Eje	Circuito
$\|0\rangle$	Z	$q:$ ——
$\|0\rangle$	X	$q:$ —H—
$\|1\rangle$	Z	$q:$ —X—
$\|1\rangle$	X	$q:$ —X—H—

A continuación, la entidad receptora recibe esos circuitos (cada uno representando un cúbit) y, de manera aleatoria, elige ejes en los que medirlos. Aproximadamente el 50 % de los cúbits serán medidos correctamente, es decir, en el mismo eje que el emisor. El restante 50 % deberá ser descartado dado que al medirlos en el eje equivocado, se tendrá exactamente un 50 % de probabilidad de emitir el valor codificado correcto, lo que implica la pérdida de la correspondiente información. Para saber cuáles son los valores a descartar, ambas entidades hacen públicos los ejes en los que se midieron los cúbits, dado que no hay riesgo al realizar tal acción. Así desechan aquellos valores en los que los ejes no coinciden.

Los valores resultantes tras los descartes podrían ser considerados la clave generada, pero antes hay que verificar su seguridad. Se da una incoherencia entre la clave enviada por el emisor y la recibida por el receptor medida con el eje correcto, en aproximadamente el 50 % de los cúbits que hayan sido medidos en el eje incorrecto por un atacante [14]. En la Tabla II se ilustran los 4 casos posibles de mediciones de cúbits, todos con un 25 % de probabilidad de ocurrir.

En el último caso de la Tabla II se observa que, con una probabilidad del 50 %, se aborta el protocolo por detectar la comunicación comprometida. Teniendo en cuenta que cada uno de los cuatro casos tiene una probabilidad del 25 %, cuando se envía un único cúbit el protocolo es abortado con una probabilidad $p_{detectar}$ del 12.5 %. Cuando se envían n cúbits, dado que cuando se detecta algún ataque el protocolo se aborta, la probabilidad de abortarlo se deduce de la de no detectar ningún ataque, a partir de la Ec. (8).

$$1-(1-p_{detectar})^n = 1-\left(1-\frac{1}{4}*\frac{1}{2}\right)^n = 1-\left(\frac{7}{8}\right)^n \tag{8}$$

Este valor también se puede obtener aplicando el Teorema de Bayes con las probabilidades de no detectar ataque sobre cúbit no desechado (3/4), y sobre cúbit desechado (1).

Para verificar el resultado del protocolo, el receptor publica la mitad de la clave obtenida pues así el emisor puede compararla con la correspondiente mitad de su clave y, si ambas coinciden, la ejecución del protocolo se considera segura y se puede utilizar la mitad restante de la clave generada como clave secreta compartida. En caso contrario, se deduce que alguna entidad intermedia ha lanzado un ataque de escucha secreta o *eavesdropping*, interceptando los cúbits enviados y midiéndolos antes de remitirlos al receptor legítimo.

VII-C. Programa

La librería *QuantumSolver* permite la ejecución de una implementación realizada del protocolo BB84. Al ejecutar el programa, se despliega un menú que facilita la visualización y selección el *backend* disponible. Tras elegirlo, se dan dos opciones al usuario. Por una parte, puede correr el algoritmo una única vez y visualizar la traza entre los diferentes participantes en la comunicación, para lo que es necesario especificar una cadena de caracteres como mensaje y un valor entre 0 y 1 como densidad de intercepción (la probabilidad con la que el receptor intermedio medirá cada cúbit). Por otra parte, puede ejecutarlo varias veces y mostrar un mapa de calor que representa con colores más claros las condiciones en que ha habido más ocasiones en las que la comunicación ha sido considerada segura, y en colores más oscuros las comunicaciones en las que se detectado una intercepción. Para este modo experimental, se deben especificar diferentes condiciones: la longitud máxima del mensaje en número de bits, que definirá el eje x en el mapa de calor, este tomará

Tabla II
CASOS POSIBLES DE LAS MEDICIONES DE UN CÚBIT EN LA FASE DE VERIFICACIÓN DE BB84

Medición de emisor y receptor legítimos	Medición del atacante intermedio	Conclusión sobre ese cúbit
Distinto eje	Eje del emisor	Es desechado en la fase de descarte de los valores, aunque el receptor intermedio lo haya medido con un 100 % de probabilidad de obtener valor correcto
Distinto eje	Eje del receptor	Es desechado en la fase de descarte de los valores, aunque el receptor intermedio lo haya medido con un 50 % de probabilidad de obtener valor correcto (total incertidumbre del valor)
Mismo eje	Mismo eje que ambos	Ha sido interceptado por el atacante sin que se aborte el protocolo, con un 100 % de probabilidad de que el valor final sea correcto
Mismo eje	Eje contrario a ambos	Hay un 50 % de probabilidad de abortar el protocolo, en caso de que en el emisor colapse con el valor contrario al emitido por el emisor

valores enteros positivos hasta ese máximo; el valor del *step* de la densidad de intercepción, que definirá el eje y del mapa de calor, este tomará $\frac{1}{step}$ valores entre 0 y 1; y el número de repeticiones para cada instancia generada del problema.

Las Fig. 3 y 4 muestran dos ejemplos de mapas de calor generados, correspondientes a los parámetros de la Tabla III.

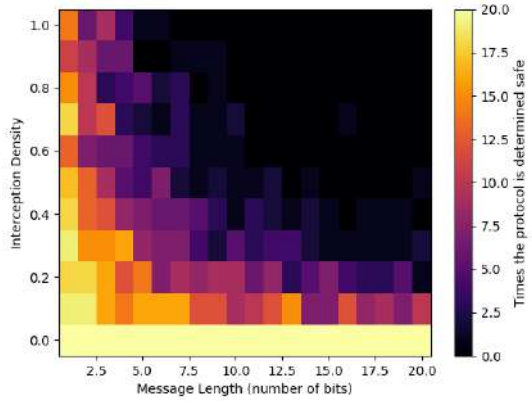


Figura 3. Primer ejemplo de mapa de calor de BB84

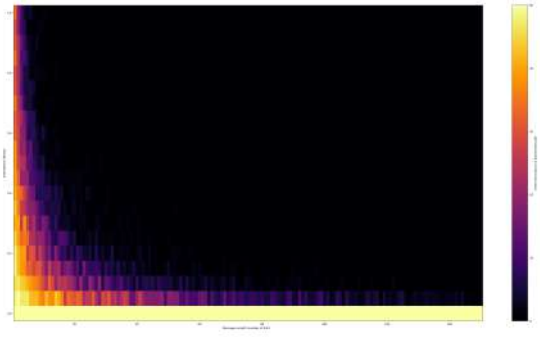


Figura 4. Segundo ejemplo de mapa de calor de BB84

Tabla III
PARÁMETROS RELEVANTES DE LOS MAPAS DE CALOR

Figura	Longitud máxima del mensaje	*Step* de densidad de intercepción	Repeticiones de cada instancia	Tiempo de ejecución (aprox.)
3	20 bits	0.1	20	5 minutos
4	150 bits	0.05	50	16 horas

VIII. CONCLUSIONES

La implementación presentada de la librería para el desarrollo cuántico *QuantumSolver* permite, de manera sumamente accesible, la ejecución de diversos algoritmos cuánticos en *hardware* cuántico real y simuladores proporcionados por IBM. También ofrece una sencilla arquitectura de entidades, respaldada por una batería de pruebas unitarias (*unitary test*), que facilita enormemente la adición de nuevos algoritmos a la librería. Precisamente, el objetivo principal en cuanto a la continuación del desarrollo de la propuesta es ampliar el número de algoritmos ahora disponibles en el *toolset*, incluyendo posiblemente los algoritmos B92, E91 y de Grover.

AGRADECIMIENTOS

Esta investigación ha sido posible gracias al proyecto RTI2018-097263-B-I00 financiado por el Ministerio de Ciencia, Innovación y Universidades, la Agencia Estatal de Investigación y el Fondo Europeo de Desarrollo Regional, y a la Cátedra de Ciberseguridad Binter-Universidad de La Laguna.

REFERENCIAS

[1] E. Gibney, “Hello quantum world! Google publishes landmark quantum supremacy claim”, Nature, vol. 574, no. 7779, pp. 461-462, 2019.
[2] J. D. Escánez, “QuantumSolver”. [Online]. Available: https://github.com/alu0101238944/quantum-solver/. [Accessed: 17-Apr-2022].
[3] Tudorache, A. G., Manta, V. I., and Caraiman, S. (2021). Implementation of the Bernstein-Vazirani Quantum Algorithm Using the Qiskit Framework. Bulletin of the Polytechnic Institute of Iaşi. Electrical Engineering, Power Engineering, Electronics Section, 67(2), 31-40.
[4] Warke, A., Behera, B. K., and Panigrahi, P. K. (2020). Experimental realization of three quantum key distribution protocols. Quantum Information Processing, 19(11), 1-15.
[5] IBM, “Qiskit”. [Online]. Available: https://qiskit.org/. [Accessed: 17-Apr-2022].
[6] IBM, “IBM Quantum”, May-2016. [Online]. Available: https://quantum-computing.ibm.com/. [Accessed: 17-Apr-2022].
[7] IBM, “User account - IBM Quantum”, May-2016. [Online]. Available: https://quantum-computing.ibm.com/composer/docs/iqx/manage/account/#account-overview. [Accessed: 17-Apr-2022].
[8] IBM, “Learn Quantum Computation using Qiskit” [Online]. Available: https://qiskit.org/textbook/preface.html. [Accessed: 17-Apr-2022].
[9] IBM, “Single qubit Gates” [Online]. Available: https://qiskit.org/textbook/ch-states/single-qubit-gates.html. [Accessed: 17-Apr-2022].
[10] E. Bernstein and U. Vazirani, “Quantum Complexity Theory”, SIAM Journal on Computing, vol. 26, no. 5, pp. 1411-1473, 1997.
[11] IBM, “Bernstein-Vazirani Algorithm” [Online]. Available: https://qiskit.org/textbook/ch-algorithms/bernstein-vazirani.html. [Accessed: 17-Apr-2022].
[12] C. H. Bennett and G. Brassard, “Quantum cryptography: Public key distribution and coin tossing,” arXiv, 1984, doi: 10.48550/ARXIV.2003.06557. [Online]. Available: https://arxiv.org/abs/2003.06557 [Accessed: 17-Apr-2022].
[13] IBM, “Quantum Key Distribution” [Online]. Available: https://qiskit.org/textbook/ch-algorithms/quantum-key-distribution.html. [Accessed: 17-Apr-2022].
[14] W. Dür, S. Heusler, “What we can learn about quantum physics from a single qubit”, 06-Dec-2013. [Online]. Available: https://arxiv.org/pdf/1312.1463.pdf. [Accessed: 17-Apr-2022].

Sesión IV: Vulnerabilidades y ciber amenazas

A Review of Kubernetes Security Vulnerabilities, Attacks and Practices

Santiago Figueroa-Lorenzo [1,2]
sfigueroa@ceit.es

Saioa Arrizabalaga [1,2]
sarrizabalaga@ceit.es

[1]CEIT-Basque Research and Technology Alliance (BRTA), Manuel Lardizabal 15, 20018 Donostia / San Sebastián, Spain.
[2]Universidad de Navarra, Tecnun, Manuel Lardizabal 13, 20018 Donostia / San Sebastián, Spain.

***Abstract*—Kubernetes is open-source software for automating deployment, scaling and management of containerized services. Organizations, such as Tesla and the US Department of Defense use Kubernetes for deploying and managing their containerized applications, delivering benefits in terms of deployment and service scalability. Despite widespread benefits of the technology, Kubernetes infrastructure is susceptible to security incidents, for example, the Tesla AWS service incident in 2018. A systematic understanding of Kubernetes security best practices can be useful for practitioners in mitigating vulnerabilities and attack vectors in Kubernetes infrastructure deployments. Therefore, the purpose of this paper is, firstly, to review both the main attack vectors targeting Kubernetes and the vulnerabilities associated with them. Second, to identify the main Kubernetes security practices including (i) implementation of authentication and authorization mechanisms, (ii) implementation of sandbox technologies, and (iii) implementation pod and network custom security policies.**

***Index Terms*—Containers, DevOps, SecDevOps, Kubernetes, security, practices**

Type of contribution: *Original research*

I. Introduction

Kubernetes is an open-source technology for automating deployment, scaling and management of containerized services [1]. Kubernetes is widely regarded the most popular container orchestration tools and is used by organizations such as Trivago, Airbnb and Adidas [2]. The benefits of using Kubernetes have been widely reported. For instance, the use of Kubernetes at the U.S. Department of Defense reduced the software deployment effort from eight months to one week [3]. Adidas, on the other hand, has significantly reduced the loading time of its e-commerce platform [2].

Despite the reported benefits, numerous security flaws have been reported in Kubernetes. For example, it was reported in 2018 that attackers gained access to Testla resources hosted in Amazon Web Services (AWS) via insecure Kubernetes shell [4]. At this moment, there are more than 140 documented vulnerabilities in the CVE[1] database related to Kubernetes. Therefore, a systematized study of attack vectors and associated vulnerabilities, as well as security best practices could support practitioners in securing their Kubernetes infrastructures.

However, it is not possible to conduct a systematized study based on scientific contributions due to the lack of contributions that consider both attack vectors and vulnerabilities, as well as the best practices for Kubernetes security. Fortunately, there are important resources such as the CNCF Financial User Group [5] and the CVE for documenting attack vectors and vulnerabilities respectively, as well as other resources such as the Kubernetes Hardening Guide [6] y the CIS Kubernetes [7], for documenting security best practices. Therefore, the objectives of this paper are:

1) Review the main attack vectors impacting Kubernetes.
2) Associate those attack vectors with vulnerabilities collected in the CVE database.
3) Review the main security practices for Kubernetes.

The remainder of this paper is organized as follows: we briefly review main concepts of this paper in Section II. In addition, we describe the main attacks vectors collected, as well as, the vulnerabilities associated to them in Section III. Next, section IV details the main security best practices for Kubernetes. Finally, conclusions and future research lines are detailed in the last section.

II. Background

First, we provide the background of Kubernetes in Section II-A. Since this paper relates attack vectors to vulnerabilities, in Section II-B we define CVEs as a source of information. Finally, we validate the methodology to be used for the security best practices review in Section II-C.

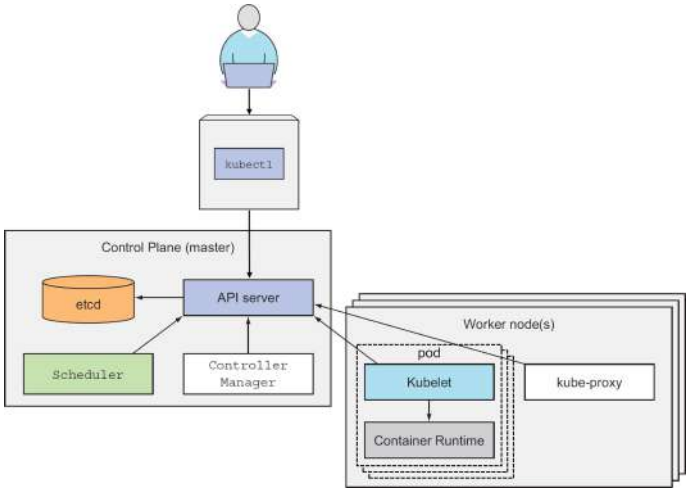


Figure 1. The components that make up Kubernetes cluster.

A. Kubernetes overview

A Kubernetes infrastructure is commonly referred to as a Kubernetes cluster [1]. A Kubernetes cluster is a collection of virtual or physical machines referred as nodes. Figure

[1]**Common Vulnerabilities and Exposures (CVE)**: https://cve.mitre.org/

1 shows that there are two types of nodes: master and workers. A master node represents the control plane and includes these components: *API server*, *scheduler*, *controller*, and *etcd* [1]. *API server* functionality is to orchestrates all the operations within the cluster. The *controller* monitors the status of the cluster through the *API server* and can change the current state to another one. The *scheduler* is responsible for monitoring recently created pods without an allocated node, so it chooses a node for them to run on. The *etcd* is a key-value based database that stores Kubernetes cluster configuration information. Users (Figure 1) use the *Kubectl* "command line tool" to interact with the *API server*.

The worker nodes host the applications that run on Kubernetes [1]. As also shown in Figure 1, worker nodes comprise next components: *kube-proxy*, *kubelet* and *pod*. *kube-proxy* maintains the network rules on nodes, enabling network communication with pods from network sessions inside or outside the cluster. *kubelet* is an agent that ensures containers are running inside a *pod*. The *pod* is the smallest Kubernetes object, which must contain at least one active container. A *container* is a software unit responsible for packaging application code so that it can run in any environment [1].

B. CVE: Common Vulnerabilities and Exposures

CVE is a source of vulnerability information to identify vulnerabilities (through a unique identifier, e.g., CVE-2022-1547) and associate to these specific versions of code bases (e.g., software and shared libraries). Vulnerabilities are defined as weaknesses in the computational logic (e.g., at the code level) of software and hardware components that, when exploited, negatively impact on the security tenants: confidentiality, integrity, or availability. Therefore, mitigation of vulnerabilities usually involves changes at the code level, but may also involve changes to the specification or protocol, going as far as deprecation (e.g., removal of affected protocols or affected functions altogether).

C. Methodology validation for Kubernetes security practices review

In the scientific literature through digital libraries such as IEEE Xplore, it is not easy to find articles that study security best practices in Kubernetes.This scenario is extensible to key technologies such as DevOps and SecDevOps. Considerable research documents how practitioners use Internet artifacts (e.g., blog posts) to make recommendations for best practices. Several articles have studied how to transfer the practices gathered from internet artifacts to the scientific domain, including systematic studies to identify challenges in microservices development, as well as to identify practices used in Continuous Deployment (CD) [8]. Other fields commonly studied in this way involve Identifying the security practices used in the organization that has adopted DevOps [9], software testing [10] and even Kubernetes security practices [11]. We adopt existing methodologies, including as an internet artifact the security reports for Kubernetes created by Kubernetes security stakeholders such as the Kubernetes Hardening Guide [6] and the CIS Kubernetes [7].

III. Security Considerations: Attack Vectors and Vulnerabilities

This section first studies the main attack vectors present in Kubernetes, associating in each case vulnerabilities associated with them (Section III-A). Then, the vulnerabilities found are summarized Section III-B.

A. Main Attack Vectors on Kubernetes

CNCF Financial User Group has released a threat modeling exercise that targeted a generic Kubernetes cluster [5]. The main objective of this study was to provide a detailed overview of potential threats and mitigations. Based on this study, the main attack vectors are summarized bellow:

Service Token: Exploiting a service token has become a frequent attack vector, since by default a service token is automatically mounted on each pod. If a container is compromised, the attacker will have an exploit mechanism using those credentials. This is a targeted attack assuming container compromise. For instance, the vulnerability *CVE-2020-8565* exploits the fact that in Kubernetes, authorization and bearer tokens are included in the log files when the log level is set to at least 9 [12].

Compromised container: This attack vector is a major point of focus within the cluster, as it provides a remote execution point for an attacker. For example, the vulnerability *CVE-2022-23648* allows that containers launched with a specially-crafted designed images to gain access to read-only copies of arbitrary files and directories on the host [13].

Network endpoints: Kubernetes endpoints defines which pods (or other servers) are exposed through a service. Kubernetes endpoints should be secured from internal malicious actors, preventing an easy attack vector. Thus, if an attacker is able to compromise a container they gain access to the endpoints if the pods network policy permits. For instance, the vulnerability *CVE-2022-24829* exploits that multiple endpoints do not require authentication, allowing an adversary to gain access to the application erroneously [14].

Denial of Service: The goal of this attack vector is to exhaust all resources. Despite event rate throttles added to Kubernetes to mitigate denial of service attacks, this mechanisms are still in its infancy. Thus, the vulnerability *CVE-2019-19922*, allows an adversary to compute the stray requests needed to decrease (due to slice expiration) performance of an entire Kubernetes cluster, also ensuring that the DDoS attack is sending the required number of stray requests [15].

RBAC Issues: The RBAC authorization plugin in Kubernetes, uses user roles as the key factor to determine whether a subject may perform the action or not. A subject, i.e., a human, a ServiceAccount, or a group of users or ServiceAccounts, is associated with one or more roles and each role is allowed to perform certain verbs on certain resources [16]. Different attacks are based on the misconfiguration of RBAC policies. For example, the vulnerability *CVE-2019-11253* exploits the fact that default RBAC policy authorizes any anonymous user to make requests, which causes an excessive consumption of CPU/memory affecting the availability of the resource [17].

B. Vulnerability analysis using CVE

Table I provides an analysis of vulnerabilities associated with the attack vectors defined in the previous section. These vulnerabilities have been collected from the CVE database. When the study was conducted, the CVE database had 140 documented vulnerabilities impacting Kubernetes. The Table I shows total vulnerabilities associated with the attack vectors defined in section III-A constitute 43.6%, with the *compromised container* being the most common attack vector with 23%. The remaining 56.4% of the documented vulnerabilities for Kubernetes can be exploited from attack vectors not summarized in Table I. Next section provides a review of security best practices for Kubernetes infrastructures.

Table I
NUMBER OF VULNERABILITIES ASSOCIATED WITH THE DEFINED ATTACK VECTORS.

Attack Vector	Vulnerabilities (Count)	Vulnerabilities (%)
Service Token	13	9.3
Compromised container	32	23
Network endpoints	10	7.1
Denial of Service	3	2.1
RBAC Issues	3	2.1
Total	61	43.6

IV. KUBERNETES SECURITY PRACTICES

The review of Internet artifacts, using the methodologies reviewed in section II-C, has derived 12 Kubernetes security practices. The Internet artifacts include security reports, blog posts, videos and presentations. Each security practices are described below:

A. Authentication (AuthN) and Authorization (AuthZ)

The best practice of enforcing AuthN and AuthZ rules prevents unauthorized users gains access and performs unauthorized activities within the Kubernetes cluster. Kubernetes uses AuthN plugins such as client certificates, bearer tokens, or an authenticating proxy to authenticate API requests [19]. AuthZ in Kubernetes involves evaluating each authenticated API request against all policies to allow or deny the request [19]. The set of tasks for securing AuthN and AuthN over Kubernetes infrastructure are listed below:

- An admission controller is a tool that intercepts requests arriving at the API Server after they are authenticated and authorized but before they are persisted in the volume. The CIS recommendation for Kubernetes promotes to enable the admission controllers [7]. There a couple of implementations that could be used as a Admission Controllers, such as, a Kubernetes native implementation based on gatekeeper[2], or plain Open Policy Agents (OPAs) to enforce a custom policy. OPA details is provided in section IV-B.
- Default configurations for both AuthN and AuthZ allows any anonymous user to perform malicious activities. For instance, if an adversary guesses the default configuration of an insecure admission, he will gain access to the admission controller and will be able to execute malicious commands. The CIS Kubernetes recommendation is to change default configurations of both AuthN and AuthZ [7].
- Anonymous access to the API server enabled by default should be restricted [7].
- Impersonation feature has benefits in terms of debugging. However, this feature can also be used by an attacker to compromise the security of the cluster. For this reason, impersonation is a feature that must be tightly controlled [6].
- The official documentation promotes AuthN based on OpenID[3], while the use of attribute based access control (ABAC) and role-based access control (RBAC) for AuthZ [19].

B. Kubernetes Security Policies

Security policies are usually applied at different levels such as network and *pod*. Additionally, there is a set of generic policies applicable at different points in the Kubernetes infrastructure. Network, *pod* and generic policies are discussed below:

- *Network-specific policies*: Network policies are applied to protect pods from unwanted network communications. All pods can communicate with each other in the default configuration because within the same node, all pods use a common Linux bridge. Recommended policies include restricting traffic between pods, restricting access to the API server, and reducing network exposure. CIS Kubernetes promotes firewalls to block all undesired network communication from network policy plugins like Calico[4] and restricting database access from the pods [7]. In addition, container compatible IDS and IPS systems are recommended to be used, e.g., Falco[5] [20].
- *Pod-specific policies*: Pod policies focus on protecting both pods and containers. They manage the execution of workloads on the cluster. In case of not defining a secure context for a pod, the container faces the risk of running with root privileges and write permissions on the root file system, which makes the cluster vulnerable. CIS Kubernetes recommends containers within a *pod* be run as a non-root user with read-only permission, i.e. run without privileges, and enabling Linux security modules [7]. It is also recommended to reduce the attack surface by installing the minimal version of the operating systems (if possible, drop Linux capabilities and use AppArmor profiles [21] or Security Enhanced Linux [22]).
- *Generic policies*: Generic security policy practices are essential to protect Kubernetes cluster components from external attackers. By default, TCP ports for *kubelet*, *API server*, *etcd*, and network plugins should not remain open and should require AuthN to have visibility. All system users should have the least privilege by default. Public access to cluster nodes via *ssh* must be restricted. Kubernetes Hardening Guide recommends to create audit policies for logging configured for each Kubernetes

[2]**Gatekeeper**: https://github.com/open-policy-agent/gatekeeper

[3]**OpenID**: https://openid.net/

[4]**Calico**: https://www.projectcalico.org/calico-networking-for-kubernetes/

[5]**Falco**: https://falco.org/

cluster at the *API server* level [6]. Beyond this basic configurations, the Open Policy Agents (OPA) is an open source, general-purpose policy engine that unifies policy enforcement across the stack and appears to be the future in the *pod* security policy space [18].

C. Vulnerability scanning

Continuous Delivery (CD) processes can be the starting point to materialize the vulnerability scanning. However, it can also be conducted at the *pod* component level. Both cases are discussed below:

- Pod components, such as containers can can contain both exploitable weaknesses and malicious malware. If persistent weaknesses in a Kubernetes cluster can be exploited, then the whole container orchestration system, and the containerized applications provisioned, become susceptible to attack. So, it is recommendeded scanning containers for vulnerabilities with tools, such as Dockscan[6], Clair[7] and Aqua-MicroScanner[8] [20].
- Inspecting images and deployment configurations within the CD process components can prevent vulnerabilities in clusters in the clusters and thus prevent attackers from gaining access later once the images have been deployed. Using a trusted private registry to pull images and test both code and images for vulnerabilities in CD process should be a common security practice.

D. Logging

Monitoring logs is recommended at (i) application level, (ii) container level, and (iii) Kubernetes clusters level. If monitoring logs is not enabled, users may have difficulty troubleshoot outcomes, such as attacks and outages. The following are some recommendations for applying the good practice of log-based monitoring:

- Logs must be constantly monitored.
- Alerts should be generated when the thresholds established for log metrics are altered.

E. Namespace separation

To avoid resource sharing it is common to use namespace separation. A *namespace* in Kubernetes implies the isolation of a logical virtual cluster which is part of a physical cluster [19]. Separating namespaces allows resources to be isolated between namespaces. The *default namespace* is created when no *namespace* is assigned to a resource. It is recommended that each team in an organization has a separate *namespace* to improve the management and operation of both development and production environments, his also prevents the exploitation of vulnerabilities in the entire resource when an attacker accesses, for example, the *default namespace*.

F. Etcd security: encryption and access

Encryption and access restrictions are a common security practice on *etcd* [19]. It is recommended that *etcd* only be available from the API server, using firewalls for protection and isolation, limiting access via the API [7]. By default, Kubernetes stores secrets in plain-text in *etcd* [16]. Therefore, accessing *etcd* involves retrieving sensitive information, such as usernames, passwords, and queries. Although Kubernetes support *etcd* encryption, the key used is kept in plain-text in a master node's configuration file. For that reason, it is recommended to use secret management tools for encryption such as *Vault*[9] [20].

G. Continuous update

Keeping the cluster up to date with the latest security patches is a mandatory practice. It is recommended to apply continuous updates for applications deployed in pods [7]. Continuous Update can prevent vulnerabilities such as *CVE-2019-16276* [23] associated with weakness *CWE-444 Inconsistent Interpretation of HTTP Requests (HTTP Request Smuggling)*. The implication of not applying the security update (on October 16, 2019 [25]) generated to avoid that vulnerability would be that the cluster will continue to be susceptible to a DoS attack.

For continuous updates, rolling update strategy is recommended to ensure availability of deployed applications [26]. *kubectl* tools allows to perform rolling updates [19].

H. CPU and memory limits

Adding both a CPU and memory consumption limit to a *pod* or *namespace* provides a way to mitigate malicious attacks. By default, Kubernetes resources start with unbounded memory request and CPU access capabilities. If an attacker initiates a DoS attack on a *pod* within the cluster a high volume of requests will result in *kube-scheduler* launching a new *pod* as well as an instance of a container. This process will continue until all CPU/memory is consumed. Therefore, we recommend configuring the resource amount according to the following guidelines:

- Defining a maximum number of instances for a *container*.
- Defining the number of CPU consumed by an application.
- Defining the maximum amount of memory for a *pod* and a *namespace*.

I. Sandbox Technologies

Even if the pods are well secured, there is no guarantee that the neighbor's pods will not be used to attack them. As a *container* shares its kernel with its host and all containers within a *pod* communicate via localhost as they share the network stack (this is enabled by a "joint container", i.e., a common Linux network namespace), the host has access to a lot of information about the containers it runs, such as its network, and the list of processes that the containers run. So if a *container* can break through the security layer that prevents it from accessing host processes (which is known as privilege escalation), there is nothing to stop it from accessing information from other containers, potentially yours. To solve this problem it would be enough to use virtual machines (VMs), although with this, we would have the slow startup and the high use of resources by the VMs. For example, if QEMU were used, more powerful machines would be required, i.e.,

[6]**Dockscan**: https://github.com/kost/dockscan

[7]**Clair**: https://github.com/quay/clair

[8]**Aqua-MicroScanner**: https://www.aquasec.com/news/microscanner-new-free-image-vulnerability-scanner-for-developers/

[9]**Vault**: https://www.vaultproject.io

we would have to pay more for almost the same service. To solve these conflicts arises Firecracker[10], which is a way to run VMs, but its main purpose is to be used as a *container* execution interface, which makes it use very few resources by design.

J. TLS support in Kubernetes

It is essential to enable Transport Layer Security (TLS) in communications between Kubernetes components (*API Server*, *etcd*, *kubelet* and *kubectl*). For example, it is widely known that the use of X509 TLS Client Certificates is the best way to authenticate to the *API Server*.

K. Separate sensitive workload

If an attacker gains access to the *kubelet* credentials of a node, then the adversary gain access to secrets and therefore, gain control of the entire system. However, by applying this practice, this adversary is unable to access both sensitive applications and the associated secrets. It is recommended to apply the utilities provided by Kubernetes, such as *taints and tolerations* [19] to control where and when a *pod* can be deployed.

L. Access to Metadata

The practice of securing sensitive cluster metadata can be very useful, to avoid exposing *kubelet* admin credentials through the gateway provided by Kubernetes metadata APIs. Enabling resources in Google Kubernetes Engine (GKE), e.g., Workload Identity [27] protects the leaking of sensitive information through metadata service.

V. Conclusions

As Kubernetes usage becomes more popular, Kubernetes security is critically important for practitioners. A systematization of best practice knowledge could be useful for securing Kubernetes infrastructure. For this reason, a study of the main attack vectors associated with Kubernetes has been conducted, determining that of the 140 vulnerabilities currently documented in the CVE database, 43.6% are the result of the impact of these attack vectors. In addition, we have performed a qualitative analysis of Internet artifacts, such as Kubernetes security reports to identify 12 security practices for Kubernetes. Our article is intended to help practitioners secure Kubernetes infrastructure. Future lines of research include the establishment of a detailed classification of documented vulnerabilities in CVE according to the level of impact on security tenants using the Common Vulnerability Scoring System (CVSS).

Acknowledgment

This research was been supported by Elkartek program of the Basque Government. Key projects have been “TRUSTIND-Creating Trust in the Industrial Digital Transformation” with grant number KK-2020/00054 and “REMEDY-REak tiME control and embeddeD security” with grant number KK-2021/00091.

[10]**Firecracker**: https://firecracker-microvm.github.io/

References

[1] S. Miles, ”Kubernetes: A Step-By-Step Guide For Beginners To Build, Manage, Develop, and Intelligently Deploy Applications By Using Kubernetes (2020 Edition)”. Independently Published, 2020. [Online]. ”Available: https://books.google.com/books?id=M4VmzQEACAAJ”.

[2] ”Kubernetes User Case Studies”, May 2020. [Online]. ”Available: https://kubernetes.io/case-studies”.

[3] The Linux Foundation, ”With Kubernetes, the U.S. Department of Defense Is Enabling DevSecOps on F-16s and Battleships”, May 2020. [Online]. Available: https://www.cncf.io/case-study/dod.

[4] Ars Technica, ”Tesla cloud resources are hacked to run cryptocurrencymining malware”, February 2018. [Online]. ”Available: https://arstechnica.com/information-technology/2018/02/teslacloud-resources-are-hacked-to-run-cryptocurrency-mining-malware”.

[5] Jonathan Meadows, ”K8s Attack Tree - Summary”, February 2019. [Online]. ”Available: https://github.com/cncf/financial-user-group/tree/main/projects/k8s-threat-model”.

[6] National Security Agency (NSA) and CISA, ”Kubernetes Hardening Guide”, March 2022. [Online]. ”Available: https://www.cisa.gov/uscert/ncas/current-activity/2022/03/15/updated-kubernetes-hardening-guide”.

[7] Center for Internet Security (CIS), ”Securing Kubernetes”, March 2022. [Online]. Available: https://www.cisecurity.org/benchmark/kubernetes.

[8] A. Rahman, et all, “Synthesizing continuous deployment practices used in software development,”in Proceedings of the 2015 Agile Conference, ser. AGILE ’15. USA: IEEE Computer Society, 2015, p. 1–10. [Online]. Available: https://doi.org/10.1109/Agile.2015.12.

[9] A.Rahman, et all, “Software security in devops: Synthesizing practitioners’ perceptions and practices,” in Proceedings of the International Workshop on Continuous Software Evolution and Delivery, ser. CSED ’16. New York, NY, USA: ACM, 2016, pp. 70–76. [Online]. Available: http://doi.acm.org/10.1145/2896941.2896946.

[10] V. Garousi, M. Felderer, and T. Hacaloglu, “Software test maturity assessment and test process improvement: A multivocal literature review,” Information and Software Technology, vol. 85, pp. 16 – 42, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584917300162.

[11] B. F. Crabtree and W. L. Miller, Doing qualitative research. sage publications, 1999.

[12] National Vulnerability Database, “CVE-2020-8565 Detail.” [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2020-8565.

[13] National Vulnerability Database, “CVE-2022-23648 Detail.” [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2022-23648.

[14] National Vulnerability Database, “CVE-2022-24829 Detail.” [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2022-24829.

[15] National Vulnerability Database, “CVE-2019-19922 Detail.” [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2019-19922.

[16] Marko Lukša, ”Kubernetes in Action.”, Manning Publications Co., 2018, ISBN: 9781617293726.

[17] National Vulnerability Database, “CVE-2019-11253 Detail.” [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2019-11253.

[18] Open Policy Agent, “Policy-based control for cloud native environments.” [Online]. Available: https://www.openpolicyagent.org/docs/latest/overview.

[19] Kubernetes, “Production-grade container orchestration.” [Online]. Available: https://kubernetes.io/docs.

[20] Rowan Baker, ”Threat Modelling: Securing Kubernetes Infrastructure & Deployments - Rowan Baker, ControlPlane.” [Online]. Available: https://kccnceu20.sched.com/event/Zeow.

[21] Kubernetes, “Restrict a Container’s Access to Resources with AppArmor.” [Online]. Available: https://kubernetes.io/docs/tutorials/security/apparmor/.

[22] Kubernetes, “Configure a Security Context for a Pod or Container.” [Online]. Available: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/.

[23] National Vulnerability Database, ”CVE-2019-16276”, September 2019, [online] Available: https://nvd.nist.gov/vuln/detail/CVE-2019-16276

[24] National Vulnerability Database, ”CVE-2019-11253”, October 2019, [online] Available: https://nvd.nist.gov/vuln/detail/CVE-2019-11253.

[25] Google Cloud, ”Security bulletins.”, [online] Available: https://cloud.google.com/kubernetes-engine/docs/security-bulletins/.

[26] VMware, ”Best Practices in Kubernetes Security.”, [online] Available: https://k8s.vmware.com/kubernetes-security-best-practices/

[27] Google Cloud, ”Protecting cluster metadata.”, [online] Available: https://cloud.google.com/kubernetes-engine/docs/how-to/protecting-cluster-metadata

Plataforma Europea para adquisición de Consciencia de Situación del Ciberespacio

Jorge Maestre Vidal
INDRA, 28108 Madrid, Spain
jmaestre@indra.es

Marco Antonio Sotelo Monge
INDRA, 28108 Madrid, Spain
masotelo@indra.es

Francisco Antonio Rodríguez López
INDRA, 28108 Madrid, Spain
farodriguez@indra.es

Mónica Mateos Calle
Mando Conjunto del Ciberespacio,
28023 Madrid, Spain
monicamateos@et.mde.es

Juan Manuel Estevez Tapiador
Universidad Carlos III de Madrid,
28911 Madrid, Spain
jestevez@inf.uc3m.es

Victor Villagrá González
Universidad Politécnica de Madrid,
28040 Madrid, Spain
victor.villagra@upm.es

Daniel Tornero Sánchez
S2 Grupo,
46010 Valencia, Spain
daniel.tornero@s2grupo.es

Rebeca Gómez Henche
Innotec Security,
28034 Madrid, Spain
rebeca.gomez@innotec.security

Mario Aragonés Lozano
Universidad Politécnica de Valencia,
46022 Valencia, Spain
maarlo9@teleco.upv.es

***Resumen*—La Plataforma Europea para la adquisición de Consciencia de Situación del Ciberespacio (ECYSAP) plantea como reto fundamental el desarrollo de principios teóricos, métodos analíticos y prototipos orientados a facilitar que el personal encargado de la toma decisiones sea capaz de percibir, comprender y proyectar lo que está sucediendo en el ciberespacio, cómo afecta a su mando, y qué nuevas oportunidades operativas revela. El proyecto aborda importantes retos y brechas tecnológicas del mercado de cara a alcanzar una autonomía estratégica europea, entre ellos la identificación de terrenos cibernéticos esenciales, la evaluación del impacto de riesgos cibernéticos sobre las líneas de esfuerzo, tareas y objetivos de la misión, o la identificación de nuevos Cursos de Acción en base a lo anterior. ECYSAP desarrolla una solución nativa militar, escalable e interoperable, coherente con el marco Ético/Regulatorio Europeo y sus valores, la proporcionalidad y el cumplimento de las Reglas de Enfrentamiento.**

***Index Terms*—ciberdefensa, ciberespacio, consciencia de situación, gestión de riesgos, soporte a decisión**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Tal y como como plantea la visión europea del ciberespacio como ámbito de operación [2], dicho ámbito comprende las siguientes, pero a su vez interrelacionadas capas: física (niveles geográficos, electromagnéticos, etc.), lógica (redes, información, servicios, etc.) y ciberpersona (dimensiones cognitivas, sociales, etc.), debiendo comprenderse como un conjunto indivisible de mutua interdependencia. Las operaciones en el ciberespacio se caracterizan por el empleo de capacidades cibernéticas con el fin de alcanzar objetivos militares dentro de estas capas, o a través de las mismas, siendo su operativa equiparable a las del resto de ámbitos de operación (marítimo, terrestre, aeroespacial y cognitivo). Debido a esto, la operativa militar en el ciberespacio plantea una complejidad intrínsecamente ligada al desafío de comprender en tiempo real el estado de los diferentes planos de procesamiento de datos en el que coexisten sus diversos actores, así como en su relación con las líneas de esfuerzo, tareas y objetivos de las misiones que habilitan; donde aliados, enemigos y elementos neutrales habrán de coexistir. Con el fin de dar soporte a dicha operativa, el personal con la responsabilidad de tomar decisiones deberá alcanzar un alto nivel de Consciencia de Situación del Ciberespacio (CSC), entendiéndose el mismo como un estado mental [4] que permita a la persona percibir, comprender y proyectar lo que está sucediendo en el ciberespacio, entender cómo afecta a su mando, e identificar las nuevas ventanas de oportunidad que revela; todo esto asumiendo un elevado nivel de incertidumbre como elemento de fricción clausewitziano donde *"el azar, la propia naturaleza humana y las conjeturas"* jugarán un papel esencial de la operativa, y por ende demandarán una gran eficacia, capacidad de respuesta y flexibilidad en la práctica del arte operacional [1]. Pero a pesar de la necesidad de operar con el adecuado nivel de CSC, y tal y como expresó en 2019 la Agencia Europea de Defensa (EDA) [5], *"el análisis del estado del arte en CSC sugirió que no existen soluciones comerciales ni de código abierto que se ajusten adecuadamente a todas las capacidades planteadas por los usuarios finales militares (europeos). Aunque algunas soluciones parciales habilitantes puedan encontrarse en nuestros mercados de manera segregada, a día de hoy no se dispone de soluciones completas."*, lo que deja entrever un importante desafío, a la vez que la incipiente necesidad de adquirir con rapidez y suficiente autonomía estratégica a nivel europeo, nuevos desarrollos, y capacidad de operación sobre ellos.

En respuesta a este reto, la Plataforma Europea para adquisición de Consciencia de Situación del Ciberespacio (ECYSAP) [3] se plantea como el mayor esfuerzo europeo hasta la fecha para alcanzar los principios teóricos, metodológicos, analíticos y primeros prototipos que faciliten que el personal militar encargado de la toma decisiones sea capaz de percibir, comprender y proyectar lo que está sucediendo en el ciberespacio, cómo afecta a su mando, y qué nuevas oportunidades operativas revela. ECYSAP constituye un esfuerzo conjunto que ha involucrado a la Comisión Europea (CE), EDA, España (ESP), Italia (IT), Francia (FR), Estonia (EE), y que abarca un

consorcio industrial liderado por Indra (ES), donde es importante destacar una gran participación española, la cual aúna importantes actores locales como S2 Grupo (ESP), Innotec (ESP), Universidad Politécnica de Madrid (ESP), Universitat Politècnica de Valencia (ESP) y Universidad Carlos III de Madrid (ES). Dicho ecosistema nacional colabora diariamente con el resto de socios estratégicos europeos entre los que se encuentran Leonardo S.P.A (IT), Airbus Cybersecurity (FR), Cybernetica (EE) o Cy4Gate (IT). ECYSAP se alinea con el desarrollo del Sistema Europeo de Mando y Control desde el plano estratégico al táctico (ESC2), que a su vez se integra en el Sistema de Mando y Control Estratégico (C2) para las misiones y operaciones en la Política Común de Seguridad y Defensa (PCSD) europea [6].

La presente publicación ha sido motivada por la voluntad de dar a conocer ECYSAP a la comunidad española de investigación en el área de la ciberseguridad, indagar en sus principales desafíos, revisar el estado actual del proyecto y explorar sinergias con otras líneas de investigación que pudieran incentivar futuras acciones conjuntas. En base a esto, y además de la presente introducción, su exposición se desglosará en las siguientes seis secciones: en la Sección 2 se presentarán los objetivos esenciales del proyecto; en la Sección 3 se presentarán las diferentes acciones que lo preceden; en la Sección 4 se esbozará la visión conjunta para alcanzar los objetivos planteados; en la Sección 5 se describirá el estado del proyecto; y en la sección 6 se comentarán algunas de las lecciones hasta ahora aprendidas.

II. Ambición

El Consorcio de ECYSAP ha adquirido el compromiso mayor con las partes involucradas, de desarrollar y aplicar *fundamentos teóricos, metodológicos, nuevos prototipados, y su integración con el fin de proporcionar una plataforma operativa europea que facilite la adquisición de la CSC en tiempo real, con capacidades defensivas de respuesta rápida y apoyo a la toma de decisiones para los usuarios finales nativos militares*. Esto dará lugar a un marco para la CSC integrador y modular, con fines de seguridad Nacional/Europea y operaciones militares expedicionarias, que se convertirá en un sistema defensivo en tiempo real capaz de orquestar respuestas cibernética bajo distintas posibles configuraciones que rijan el nivel de intervención humana en el ciclo de respuesta; todo esto mediante su interconexión con nodos inteligentes de sensado y actuación en el ciberespacio, a la vez que colaborando con sistemas de defensa existentes (ya sean legados o de nueva generación). Con un carácter marcadamente nativo orientado a la propia operación militar en el ciberespacio, ECYSAP podrá valerse del emergente ecosistema de capacidades inherentemente duales para explorar soluciones que ayuden al cumplimento de los siguientes ocho objetivos secundarios.

El primero será 1) el proporcionar capacidades avanzadas de supervisión y análisis que permitan una rápida identificación de los ataques y amenazas en el ciberespacio sobre el que ocurren las operaciones militares. Esto irá sucedido del 2) desarrollo de capacidades para la evaluación dinámica de riesgos centrada en la misión, valiéndose para ello de la correlación de las situaciones que se observan en el ciberespacio y su propagación a los objetivos, tareas, líneas de esfuerzo, etc. de las misiones planeadas o en curso. Se 3) diseñarán, implementarán e integrarán funcionalidades avanzadas de análisis, simulación y predicción capaces de apoyar la toma de decisiones y facilitar la aplicación de Cursos de Acción (CoAs) anticipatorios en base a la proyección del estado de situación adquirido. Con el fin de dar respuesta a las amenazas detectadas, ECYSAP 4) integrará subsistemas para la identificación, selección, planificación y ejecución de los CoA más adecuado tanto a nivel cibernético, como de misión. Con el fin de dar viabilidad a la usabilidad de los sistemas resultantes, se 5) aportará un entorno visual y comprensible con capacidades de configuración que faciliten que el personal militar involucrado comprenda el estado del entorno operativo, y pueda operar ECYSAP con mayor comodidad. La plataforma 6) integrará sistemas de gestión de evidencias, notificación e intercambio de información que permitan compartir la Imagen Operacional adquirida, así como el desarrollar una Imagen Operacional Común. ECYSAP constituirá una 7) plataforma para la CSC auditable, segura y sin ataduras de cara a su futura certificación y catalogación; cuyos resultados será 8) validados por usuarios finales y demostrados en escenarios de demostración relevantes apoyados por los países participantes.

III. Antecedentes

El proyecto ECYSAP tiene como punto de partida la experiencia y los hallazgos de una secuencia de proyectos y acciones previamente orquestados desde la EDA. Remontándose atrás en el tiempo, y como respuesta a la consolidación del ciberespacio como quinto ámbito de operaciones militares de la OTAN [1], el EDA Project Team Cyber Defence (PT CD) identificó la necesidad de desarrollar capacidades autónomas europeas que habrían de facilitar a los mandos militares de la Unión el comprender y gestionar el riesgo de ciberataques en todos sus niveles operativos.. En el año 2015 el grupo de trabajo CySAP Ad Hoc Working Group (AHWG) vinculado a la iniciativa *"Apoyo de la industria al proyecto Cat B - Paquete para la adquisición de la Consciencia de Situación del Ciberespacio"* [7], elaboraría bajo el liderazgo de España (Indra como líder industrial) una línea de acción común, un conjunto de requisitos y un caso de negocio que introdujera qué elementos operativos serían necesarios para que las Fuerzas Armadas europeas alcanzaran una CSC viable y sostenible; reafirmándose la necesidad de preservar la autonomía europea a lo largo de su ciclo de vida. En estos términos, en el año 2016 la EDA dio continuidad a esta línea de capacitación por medio del diseño de una referencia arquitectónica capaz de albergar dichas soluciones en concordancia con el Marco Arquitectónico de la OTAN (NAF v.3) [8], así como el desarrollo de un conjunto de requisitos de usuario comunes a nivel de sistema [9]. Este trabajo sería complementado por un nuevo proyecto de apoyo a su desarrollo denominado *"Generación de conjuntos de datos para la validación de herramientas de ciberdefensa"*[10], [11], que ambicionaría el proveer de un marco metodológico y herramientas (procesos y conjuntos de datos) para facilitar su verificación y validación. Tomando los resultados de dichos proyectos como base, a finales del año 2020 concluiría un primer prototipado rápido denominado CySAP-RRP (hasta TRL 4) que integraría un subconjunto de sus funciones esenciales (gestión dinámica de riesgos cibernéticos, valoración del impacto de dichos riesgos

a nivel de misión, y soporte a decisión) [12], cerrándosele así una primera espiral de desarrollo de capacidad orientada a probar la viabilidad de los conceptos inicialmente planteados.

En este contexto, ECYSAP plantea el inicio de una nueva espiral del desarrollo de capacidad con la motivación de alcanzar un nivel de madurez TRL 7, ampliando significativamente el nivel de ambición, integrando la visión de nuevos participantes, y concluyendo con las primeras demostraciones avanzadas en interconexión con sistemas de defensa reales.

IV. ECYSAP: Una Visión Conjunta

La visión conjunta elaborada por ECYSAP para adquirir una CSC viable para los distintos actores involucradas parte por la distinción conceptual del entorno de operaciones en dos grandes niveles: 1) Dominio Cibernético (CIS), el cual abarca las capas física, lógica y ciberpersona; y el 2) Dominio de Misión (MI), donde coexisten operaciones, tareas, objetivos, líneas de esfuerzo, etc. Desde la perspectiva del Dominio Cibernético, algunas de las cuestiones fundamentales que ECYSAP pretende responder son: *"¿cuáles son los riesgos de CIS y qué impacto podrían llegar a causar sobre los activos cibernéticos en el entorno operativo?*, *"¿cómo gestionar estos riesgos teniendo en cuenta desde la fase de planificación de la misión, hasta su ejecución?*, *"¿cómo podría el personal de toma de decisiones mitigar e incluso anticipar estos riesgos?*, dada la conciencia de situación adquirida, *"¿cómo podría mejorarse la seguridad, resiliencia y CoAs relacionados con riesgos CIS a lo largo de una misión militar?*.

Desde la perspectiva de la misión se responderán cuestiones del tipo: *"¿cómo impactarán los riesgos CIS en las tareas, objetivos y lineas de esfuerzo de una misión?*, *"¿qué terrenos cibernéticos son críticos e insustituibles para la correcta ejecución de la misión?*, *"¿qué CoAs a nivel de misión podrían ayudar a mitigar el impacto de las amenas sobre terrenos críticos cibernéticos?*, o *"¿qué combinación de CoAs (tanto en CIS como en MI) dan mayor viabilidad al éxito de la misión?*. Esta visión conceptual se ilustra en Fig. 1, donde un hipotético ciberataque impacta en el hardware y/o software que soporta una operación militar, y ECYSAP responde a través de un bucle de respuesta que abarca tres etapas fundamentales en la comprensión de la situación:

Gestión de riesgos desde el plano CIS, donde se percibe la situación de amenaza cibernética contra infraestructura, servicios y/o ciberpersonas. Se valorará su efecto en base a dimensiones de impacto CIS (confidencialidad, integridad, disponibilidad), se identificarán los riesgos derivados, y se ejecutarán los CoAs a nivel CIS, que incluirán rutinas y contramedidas tecnológicas concretas. Planteará respuesta a cuestiones del tipo, *"¿qué está sucediendo en los activos CIS que soportan la misión encomendada?*.

Correlación entre lo percibido en el plano CIS y la misión en curso, propagándose la implicación de las dimensiones de impacto CIS a efectos sobre los objetivos y tareas de la misión, como la posible interrupción del factor sorpresa, alteraciones en el ritmo de batalla, reducción de la flexibilidad operativa, disminución de la capacidad de concentrar fuerza, etc. Sobre este nivel actuarán las posibles opciones desde la perspectiva que ofrecen CoAs a nivel de misión. Este nivel de comprensión responderá a cuestiones del tipo, *"¿Cómo la situación actual y proyectada del ciberespacio afectará a la misión encomendada?*.

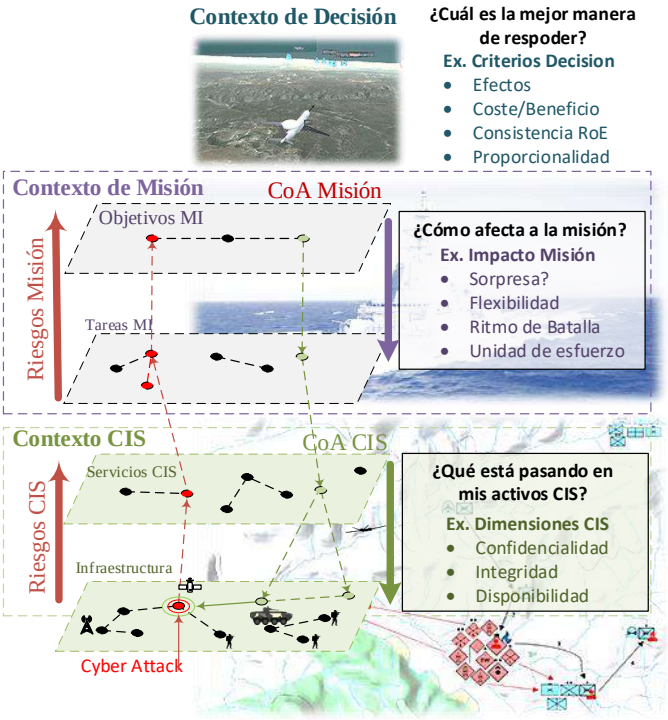


Figura 1. Niveles de comprensión de la situación del ciberespacio

Análisis de opciones/oportunidades, que abarca la interrelación de los niveles anteriores desde la perspectiva del soporte a la decisión. En este nivel se dará soporte a la identificación, valoración, priorización y planificación de CoAS. Este proceso tendrá en cuenta su proporcionalidad, coste/beneficio, nivel de consistencia con las reglas de enfrentamiento, valoraciones cognitivas frente a posibles efectos, fatiga de combate/operacional tanto en el mando como en la fuerza que ejecutará las acciones, etc. Se plantearán respuestas a cuestiones del tipo, *"¿Cuál es la mejor manera de responder a la situación percibida*?

V. Estado del Proyecto y Próximos Pasos

El proyecto ECYSAP arrancó oficialmente en diciembre del 2020, y su ejecución tendrá una duración de cuatro años. El plan de desarrollo de ECYSAP asume una metodología científica directa a lo largo de su ciclo de vida que, en paralelo a la ejecución de una estrategia de consultaría continua a usuarios finales y expertos externos, irá incorporando las lecciones aprendidas, recomendaciones y nuevas necesidades surgidas a medida que avanza el nivel de madurez de los conceptos abordados. Esta comprende cuatro fases principales de desarrollo: 1) *Análisis de Requisitos*, establecimiento de sus principios de diseño y perfilado de las bases que esbocen su concepto operativo (WP2); 2) *Diseño de la plataforma y sus componentes* (WP2-WP6); 3) *Implementación e Integración* de los diseños realizados (WP7 -WP9); y 4) consolidación a través de la *Validación y Demostración* (WP10) de los resultados alcanzados. ECYSAP actualmente se encuentra en desarrollo desde un enfoque iterativo, en el que cada fase

anterior se revisará para redefinir/ajustar los posibles cambios que se hayan identificado durante las fases posteriores.

Actualmente ECYSAP ha completado su fase inicial, habiéndose identificado y acordado su cobertura mínima por medio de requisitos funcionales, no-funcionales y de sistema. Se han concretado diferentes casos uso, realizándose los primeros esbozos de lo que dentro de varios años acabarán siendo sus escenarios de demostración nacionales, cada uno de ellos en coordinación con los Ministerios de Defensa y cibercomandos de España, Italia, Francia y Estonia; así como el apoyo de la EDA y expertos de la Comisión Europea. Esta primera fase también ha llevado al re-diseño de su arquitectura, esta vez desde la visión conjunta de ECYSAP como extensión de la alcanzada en proyectos previos (ver Sección III). Se han concretado los procedimientos para la colaboración y el apoyo con diferentes sistemas de información (Centros de Operaciones de Seguridad (SOC), Centros de Inteligencia, Sistemas de Planificación de Misiones, Herramientas de Monitorización, sistemas de actuación, etc.). En base a esto se han establecido sus procedimientos de auditoria y bastionado, concluyendo esta primera fase con la definición de una representación de conocimiento ontológica que recoge todos los niveles de información y procedimientos (CIS, Misión, CoAs, etc.) sobre los que operará la plataforma.

El foco de desarrollo de ECYSAP actual orbita en torno a la Fase 2, donde a día de hoy el Consorcio está centrando su esfuerzo en el diseño de sus capacidades orientadas a la comprensión, siempre desde la necesidad nativa del usuario final, de lo que sucede en el ciberespacio; a lo que en secciones anteriores nos referimos como Dominio Cibernético (CIS): monitorización y reconocimiento, identificación de amenazas, análisis de riesgos dinámicos, ciberinteligencia, presentación al usuario de la información, simulación, soporte a decisión de CoAs CIS, etc. También han comenzado las primeras tareas de diseño de capacidades sobre el Dominio de Misión (MI), como las de mapeo de la misión y su convergencia con el entorno CIS, gestión de riesgos a nivel de misión, colaboración/federación con otros sistemas de defensa, gestión de evidencias y cadena de custodia, etc. Se espera que, a medida que las capacidades en torno al Dominio Cibernético tomen una mayor forma, el foco del esfuerzo se desplace paulatinamente al diseño en torno al Dominio de Misión. Cabe resaltar que a medida que cada acción de diseño concluya, su fuerza de tarea se irá volcando en su correspondiente prototipo e integración, estos últimos constituyendo la tercera fase del proyecto. Si todo acontece según lo previsto, las fases de diseño y prototipado concluirán a finales del año 2023, quedando el año restante para completar su integración, consolidar los resultados y demostración; esto último dentro de la cuarta y última fase.

VI. Conclusiones y Lecciones Aprendidas

Durante la primera fase del proyecto se han evidenciado diferentes lagunas a nivel de capacidad, oportunidades y necesidades de armonización, ya sea a nivel doctrinal o de estandarización. De entras ellas, y dada la naturaleza dual de las JNIC 2022, es de resaltar la cada vez mayor incipiente diferenciación entre el portfolio de conocimientos, soluciones y productos inherentes al ámbito de la ciberseguridad, y aquellas propias de la operativa militar durante sus actuaciones en el ciberespacio. Si bien en los albores de la ciberdefensa se manifestó una tendencia a aprender e importar capacidades comerciales civiles, a medida que madura el conocimiento operativo, la propia doctrina y el pensamiento militar en este ámbito, se evidencian nuevas necesidades, así como la evolución de lo que anteriormente se consideraba dual hacia capacidades nativas. Esta problemática se traslada al personal involucrado, siendo incipiente la necesidad crítica de generar y retener talento con este conocimiento dentro del espacio europeo, y por tanto, de mejorar nuestra capacidad y autonomía estratégica en materia de ciberdefensa. Desde el punto de vista de la gestión de proyecto, esta problemática, en combinación con los riesgos inherentes al escenario geoestratégico actual (pandemia Covid'19, conflicto ruso-ucraniano, etc.) han resultado ser las mayores, aunque no únicas, dificultades en la coordinación de ECYSAP.

Desde el punto de vista técnico los principales retos también han venido de la mano de las necesidades nativas, destacándose en primer lugar la identificación de soluciones, protocolos y modelos de datos interoperables con los sistemas de defensa de nueva generación, pero también legados. Por otro lado, es importante resaltar que, si bien la respuesta a ciber incidencias tradicionalmente conlleva la actuación bajo circunstancias de alta incertidumbre, en el desarrollo de ECYSAP se prevé un mucho más complejo campo de Agramente, donde las configuraciones, procedimientos, modelos, etc. deben adaptarse a la actividad adversaria, reasignaciones de tareas dentro de la misión, cambios en las líneas de operación, etc. Otro aspecto a destacar es la gestión del automatismo inherente a la adopción de IA, el cual, a pesar de su incuestionable utilidad, no podrá adoptarse fuera de las decisiones asumidas bajo la cadena de mando, con altos niveles de fiabilidad, explicabilidad, y bajo diferentes perfiles de intervención humana durante la operación. Finalmente ha sido importante la consideración de cuestiones éticas, pero también regulatorias mayormente amparadas en doctrina y el derecho internacional, de tal manera que los cibercomandos, como actores bajo el abrigo estatal, puedan hacer uso de las capacidades que brinda ECYSAP en coherencia con sus RoEs; prestándose mucha atención a su posible percepción como actos de agresión, desproporcionalidad, etc. con las repercusiones políticas y diplomáticas que podría acarrear.

Agradecimientos

This research has received funding from the European Defence Industrial Development Programme (EDIDP) under grant agreement No EDIDP-CSAMN-SSC-2019-022-ECYSAP (European Cyber Situational Awareness Platform).

Referencias

[1] NATO AJP-5: "Allied Joint Doctrine for the Planning of Operations", https://www.coemed.org/files/stanags/01_AJP/AJP-5_EDA_V2_E_2526.pdf (visitado en 04/2022).

[2] EEAS: "European Union Military Vision and Strategy on Cyberspace as a Domain of Operations", 06 Rev 4, 2021.

[3] The European Cyber Situational Awareness Platform (ECYSAP), https://www.ecysap.eu (visitado en 04/2022).

[4] M.R. Endsley: "Situational awareness misconceptions and misunderstanding", *Journal of Cognitive Engineering and Decision Making* 9(1), 4–32, 2016.

[5] European Comisión, https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/edidp-csamn-ssc-2019 (visitado en 04/2022).
[6] The Strategic Command and Control (C2) System for CSDP missions and operations, https://www.pesco.europa.eu/project/strategic-c2-system-for-csdp-missions-and-operations/ (visitado en 04/2022).
[7] European Defence Agency, Industry Support To Cat B Project - Cyber Situational Awareness Package", 14.CAT.OP.070, http://eda.europa.eu/docs/default-source/procurement/14-cap-op-070-draft-contract-.pdf (visitado en 04/2022).
[8] NATO Architecture Framework, https://www.nato.int/cps/en/natohq/topics_157575.htm (visitado en 04/2022).
[9] European Defence Agency, "Target architecture and system requirements for an enhanced cyber situation awareness"16.CAT.OP.078, https://etendering.ted.europa.eu/cft/cft-display.html?cftId=1855 (visitado en 04/2022).
[10] D. Sandoval Rodriguez-Bermejo et al, .Evaluation methodology for mission-centric cyber situational awareness capabilities". Proc. ARES 2020, https://doi.org/10.1145/3407023.3409223 (visitado en 04/2022).
[11] R. Daton Medenou et al, ÇYSAS-S3: a novel dataset for validating cyber situational awareness related tools for supporting military operations". Proc. ARES 2020, https://doi.org/10.1145/3407023.3409222 (visitado en 04/2022).
[12] Cyber Defence Situation Awareness Package Rapid Research Prototype (CySAP-RRP), https://eda.europa.eu/news-and-events/news/2019/01/11/cyber-situation-awareness-package-(cysap)-project-launched-by-three-member-states (visitado en 04/2022).

Privacidad contextual en entornos Edge

Manuel Ruiz, Ruben Rios, Rodrigo Roman, Javier Lopez
NICS Lab, Universidad de Málaga
Campus de Teatinos s/n, 29071, Malaga
{mrr,ruben,roman,jlm}@lcc.uma.es

***Resumen*—La privacidad contextual se refiere a la protección de toda aquella información que puede desprenderse de la interacción entre usuarios y/o servicios, exceptuando los datos que el propio usuario elige transmitir. La localización, el tiempo, los patrones de uso y los diferentes parámetros necesarios para realizar la comunicación son algunos ejemplos. Este tipo de privacidad es extremadamente importante en la computación edge debido al acercamiento de los recursos de la infraestructura a los usuarios. Por ello, el objetivo de este trabajo es ofrecer un análisis y clasificación de las diferentes soluciones propuestas en la literatura respecto a la privacidad contextual en entornos edge, mostrando tanto las capacidades de los mecanismos actuales como los desafíos en este campo.**

***Index Terms*—Privacidad, Computación edge, Privacidad contextual**

Tipo de contribución: *Investigación original*

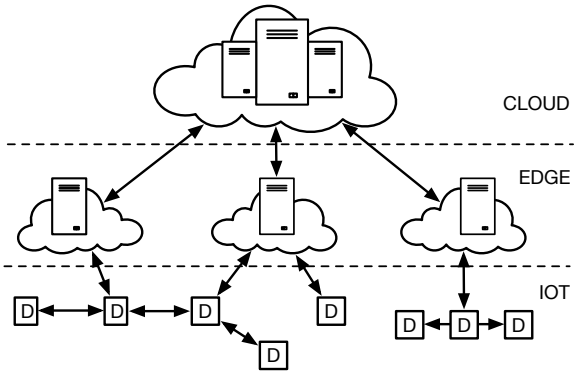


Figura 1. Infraestructura de computación edge

I. Introducción

En los últimos años se ha observado que la computación en la nube no es una solución óptima para muchos de los escenarios de aplicación previstos por la Internet de las Cosas (IoT) [1]. Aplicaciones como las redes vehiculares, la cirugía en remoto o la industria 4.0 simplemente no funcionarán si depende de un sistema remoto y centralizado como la nube. La razón es doble: (1) estas aplicaciones requieren latencias extremadamente bajas para permitir que los dispositivos reaccionen a tiempo a los cambios acontecidos en su entorno, y (2) existe un cuello de botella, en términos de ancho de banda, causado por la transmisión masiva de datos desde multitud de dispositivos en el borde de la red hasta el núcleo – donde se encuentran los servidores de la nube.

La computación en el borde o computación edge (en inglés, *Edge Computing* [2]) es un nuevo paradigma de computación que trata de dar solución a los problemas planteados anteriormente. En este paradigma, los recursos del sistema, principalmente computación y almacenamiento, se distribuyen a lo largo de un continuo que va desde el núcleo de la red hasta los dispositivos del extremo, que son los principales clientes de la infraestructura (ver Figura 1). De esta forma, al no depender directamente de servicios alojados en lugares remotos, se consigue mejorar sustancialmente los tiempos de respuesta de los dispositivos y reducir las necesidades de ancho de banda, entre de otras ventajas.

Además de las evidentes oportunidades que ofrece este nuevo paradigma, también abre la puerta a una serie de importantes retos relacionados con la gestión, coordinación y distribución de recursos. Asimismo, la naturaleza distribuida introduce una serie de importantes retos relacionados con la seguridad y privacidad [3]. Sin los mecanismos de seguridad adecuados, los potenciales beneficios que este paradigma puede aportar se verán empañados por los daños que pueden provocar los atacantes y sus desastrosas consecuencias. Por ejemplo, en un escenario edge dedicado a las redes vehiculares, un atacante puede lanzar ataques de denegación de servicio o incluso destruir físicamente parte de la infraestructura para que los vehículos sean incapaces de tomar decisiones a tiempo en caso de emergencia. Asimismo, la computación edge introduce nuevos retos de privacidad, principalmente debido al acercamiento de elementos de la infraestructura al extremo de la red. Gracias a dicho acercamiento, los operadores de la infraestructura tienen la capacidad de adquirir información que, en entornos basados en el cloud, no estaba a su alcance. De hecho, entre las novedades prometidas por el paradigma edge, se encuentra la capacidad de obtener y aprovechar información sobre el contexto en el que se despliegan los servicios virtualizados para, de esta forma, poder adaptarse a las necesidades o características del entorno y sus clientes.

Por tanto, además de los datos que un usuario puede transmitir desde sus dispositivos a la infraestructura, y que son susceptibles de ser analizados por un proveedor edge, existe también cierta información implícita al contexto que puede desprenderse de estas interacciones. Dicha información, que llamaremos información contextual, podría ser utilizada para fines maliciosos. En este sentido pueden encontrarse diferentes entidades interesadas en este tipo de información sensible. Por un lado, pueden existir proveedores edge que traten de sobrepasar los límites de la legalidad almacenando datos sobre los usuarios y su contexto, como por ejemplo su localización geográfica exacta en diversos instante de tiempo. Por otro lado, pueden existir terceras partes que se aprovechen de la infraestructura y los servicios desplegados para obtener información sensible no sólo sobre los usuarios, sino también sobre la propia infraestructura.

A fin de evitar este tipo de problemas, existen en la literatura diversas soluciones que abordan diferentes problemas de

privacidad en entornos edge. Gran parte de estas soluciones se centran en la protección de la privacidad relacionada con el contenido de los mensajes (c.f. [4], [5], [6]). Por otra parte, el número de soluciones que persiguen proporcionar mecanismos para proteger la privacidad contextual es más limitado, aun cuando la información asociada al contexto es igualmente sensible. Es por ello que en este artículo haremos una revisión y análisis de los trabajos relativos a la privacidad contextual en entornos edge, teniendo en cuenta el modelo de atacante al que tratan de hacer frente. Con esto pretendemos ofrecer una visión del estado del arte, así como promover la investigación en algunas áreas que consideramos aún requieren de nuevas soluciones o enfoques.

El resto del artículo se organiza según la estructura descrita a continuación. En primer lugar (sección II) se presentan brevemente trabajos relacionados con el estudio del estado del arte. A continuación, la sección III introduce una taxonomía de los problemas y soluciones de privacidad contextual existentes, que servirá de guía para el resto del artículo. En la sección IV se recogen los diferentes trabajos dedicados a la privacidad de la localización, mientras que en la sección V se aborda el análisis de los trabajos relacionados con la privacidad en las comunicaciones de manera general. En la sección VI, se estudia un conjunto de soluciones que surge para proteger la privacidad en el proceso de asignación de tareas, y posteriormente en la sección VII se analiza el problema de la privacidad temporal. En la sección VIII se ofrece una discusión sobre el estado del arte haciendo énfasis en las posibles líneas de investigación que consideramos más prometedores. Para finalizar, en la sección IX se muestran las conclusiones de este artículo.

II. Trabajo relacionado

Los paradigmas de computación en el borde como elemento vertebrador de un paradigma IoT completamente desarrollado ha provocado un enorme interés tanto en la academia como en la industria. En el ámbito académico se ha puesto mucho empeño en la definición del concepto y los elementos de su arquitectura [7]. Existen numerosos trabajos de investigación dedicados a estudiar las modelos de la computación en el borde (p.ej., [8], [9], [10], [2], [11]). Estos trabajos principalmente cubren aspectos generales de los paradigmas, como las tecnologías y protocolos clave, aplicaciones prometedoras además de problemas abiertos, y oportunidades. Otros autores analizan aspectos más específicos de los paradigmas de la computación edge, como el plano de la comunicación [12], el reparto de la computación [13], o el uso de redes definidas por software [14], entre otros.

También existen muchos trabajos dedicados a analizar el estado del arte de la seguridad en los paradigmas de computación en el borde. La mayoría de estos trabajos (p.ej., [15], [16], [3], [17]) suelen comenzar proporcionando una visión general del estado de los paradigmas edge y, posteriormente, realizan un análisis de las amenazas de seguridad que afectan a los diferentes componentes en estos entornos. Para finalizar, suelen presentar algunos desafíos de seguridad y oportunidades de investigación. La principal diferencia entre estos trabajos se encuentra en la clasificación de las soluciones, así como en el número y el nivel de detalle con que se analizan. Adicionalmente, estos trabajos también consideran y discuten las amenazas relacionadas con la privacidad en los paradigmas edge, pero al ser trabajos de alcance más generalista el análisis de dichas amenazas es limitado.

Existe otros trabajos de investigación que han proporcionado un análisis específico de las amenazas a la privacidad en los paradigmas de computación en el borde. Algunos de estos trabajos [18], [19], [20], [21] tienen un carácter generalista, mientras que otros se centran en aspectos muy específicos de la privacidad. Por ejemplo, Khalid et al. [22] ofrece un estudio sobre privacidad y esquemas de control de acceso, centrándose en el almacenamiento y la recuperación segura de datos. Del mismo modo, Zhang et al. [23] analiza varios métodos para la extracción, computación y la búsqueda segura de datos. En el mismo artículo, también revisan algunos mecanismos para proteger la identidad y la localización. Por último, Tian et al. [24] se centra en explorar y clasificar los retos de privacidad de localización en entornos MEC (Multi-access Edge Computing [25]).

Por lo tanto, como se desprende de los párrafos anteriores, no existen a fecha de hoy trabajos de investigación que proporcionen un estudio específico de la privacidad contextual en entornos edge. En consecuencia este es, hasta donde sabemos, el primer articulo que estudia y analiza el estado del arte de los diferentes retos y soluciones existentes de la privacidad contextual en la computación en el borde.

III. Clasificación de soluciones

En esta sección se proporciona una clasificación de amenazas y soluciones de privacidad contextual, la cual servirá de guía para la exposición de las secciones posteriores. Aunque este tipo de clasificaciones se puede realizar utilizando enfoques muy variados, en este caso nos hemos decantado por considerar en primer lugar el tipo de información a proteger, seguido por los diferentes modelos de atacante que pueden estar interesados en esa información, y, finalmente, por el tipo de soluciones desarrolladas para proteger la información frente a esos modelos de atacante. De esta forma, obtenemos una taxonomía en tres niveles, como se muestra en la Figura 2.

En el primer nivel de la clasificación, relacionado con *el tipo de información que se desea proteger*, se han considerado las siguientes categorías: localización, comunicación, reparto de tareas y registro temporal. Por lo general, la obtención de esta información podría afectar tanto a los clientes como a la propia infraestructura edge. Así pues, un atacante podía estar interesado en determinar la localización de un usuario concreto pero también la localización de servicios o aplicaciones desplegados en la infraestructura. En el segundo caso, el atacante estaría interesado, por ejemplo, en la carga de trabajo del proveedor de servicio en determinadas zonas geográficas, o en el movimiento de tareas entre dispositivos para conocer mejor su modelo de negocio, vulnerando así la privacidad del proveedor edge. Por otra parte, de las comunicaciones también se desprende gran cantidad de información sensible, por ejemplo, la dirección IP que utilizan un dispositivo de usuario es considerada un parámetro identificativo en muchos casos. Como veremos más adelante, estas dos categorías (localización y comunicación) son las que hasta la fecha han recibido una mayor atención por parte de la comunidad investigadora. Cabe mencionarse que existe cierta información contextual relacionada con las anteriores, el reparto de tareas,

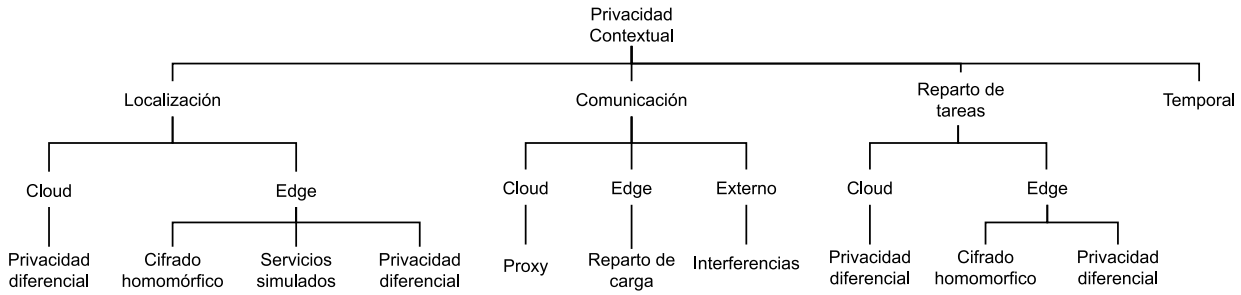


Figura 2. Clasificación de las soluciones encontradas

que hemos decidido considerar como una categoría separada debido al interés suscitado en la literatura sobre la privacidad de este procedimiento. Finalmente, la categoría temporal se refiere a los datos derivados de los patrones de uso y comportamiento. Sin embargo, pese a la importancia de esta información, no se ha encontrado literatura referente a este campo.

En el segundo nivel de la clasificación, relativo al *modelo de atacante*, es posible realizar varias separaciones atendiendo a diversas características. Por ejemplo, podemos distinguir entre atacantes internos o externos, en función de si este tiene o no acceso privilegiado a la infraestructura. Además, en relación a la naturaleza de los ataques realizados, podríamos considerar atacantes pasivos, que son aquellos que se limitan a recoger y analizar información, y atacantes activos, que además de ello se dedican a intervenir en las comunicaciones o alterar componentes del sistema. En este sentido, en la literatura se suele hacer referencia a atacantes semi-honestos (también conocidos como honestos pero curiosos), que son aquellos que no se desvían del comportamiento esperado pero tratan de obtener información a través de su participación en una comunicación o protocolo, y atacantes maliciosos, que se pueden comportar de manera arbitraria desviándose del comportamiento esperado para obtener información adicional.

Sin embargo, en este trabajo se ha decidido enfocar el modelo de atacante desde una perspectiva diferente – aunque complementaria – a las anteriormente presentadas. Esta decisión viene fundamentada por los trabajos encontrados en la literatura, que esencialmente consideran 3 atacantes posibles: (1) el servidor cloud, (2) los servidores edge y (3) atacantes externos. En esencia, esta clasificación considera los privilegios de los atacantes, si son internos o externos, y su ubicación en dentro de la infraestructura. Por lo general, todos estos atacantes se comportarán como entidades semi-honestas o pasivas.

Por último, el tercer nivel de clasificación se dedica a *los mecanismos utilizados por las soluciones propuestas*. Las principales soluciones encontradas se basan en la aplicación de mecanismos basados en privacidad diferencial y en cifrado homomórfico. En esencia, la privacidad diferencial [26] es una técnica de adición de ruido de manera que un atacante no pueda obtener información sensibles a partir del análisis estadístico del conjunto de datos. Por otra parte, el cifrado homomórfico [27] tiene como objetivo permitir la computación sobre datos cifrados, de forma que pueda seguir siendo utilizada por el resto de la infraestructura sin poner en riesgo los datos en sí mismos. También se han encontrado soluciones basadas en otros mecanismos, como la creación de servicios simulados, utilización de servidores proxy o la incorporación de señales de interferencias, que serán explicadas con detenimiento en sus respectivos apartados.

A continuación, se presenta y analiza manera detallada la investigación más relevante desarrollada hasta la fecha en cada una de estas categorías: la privacidad de la localización en la sección IV, la privacidad de la comunicación en la sección V, la privacidad en el reparto de tareas en la sección VI, y la privacidad temporal en la sección VII.

IV. Privacidad de localización

El lugar donde se encuentra un individuo o entidad en un momento determinado es información extremadamente sensible. Por norma general, los individuos son el foco de atención de atacantes aunque la localización de determinados dispositivos o recursos también puede ser de gran interés [28]. La información de localización puede servir a un atacante para identificar a una determinada persona, para crear un perfil sobre esta con sus hábitos, aficiones o gustos, e incluso para hacer un seguimiento o predecir donde estará en el futuro y atentar contra su integridad física o moral. Además, debido al acercamiento de los servicios y la infraestructura edge, esta información puede ser obtenida con más facilidad o precisión, incluso cuando el usuario no ha decidido revelarla libremente. Los mecanismos de privacidad de localización, por tanto, tratan de evitar que esta información se desprenda de las interacciones de los usuarios con el edge.

Si clasificamos las soluciones propuestas desde el punto de vista del atacante, en la literatura encontramos básicamente dos tipos de soluciones – orientadas a un cloud semi-honesto y orientadas a un cloud/edge semi-honesto.

IV-A. Cloud semi-honesto

Un proveedor de servicios cloud semi-honesto es aquel que proporcionan un servicio adecuado pero intentan extraer información de su interacción con el usuario y con los nodos edge. Las soluciones propuestas dentro de este ámbito se aplican en los nodos edge, que se considera confiable. En ambos casos propuestos, se utiliza la privacidad diferencial. En general, el nodo edge recibirá la información exacta de la ubicación del usuario, y ofuscará su contenido antes de enviarla al servidor cloud. De esta forma el proveedor de servicios nunca conocerá la localización exacta.

En [29] se propone un nuevo entorno de trabajo de privacidad diferencial llamado Pri-ENV, que permite proteger la ubicación exacta del usuario sin limitar la calidad del servicio prestado por los proveedores de servicios. Este entorno de

trabajo está compuesto por dos elementos: (1) El mecanismo de privacidad diferencial Pri-LBS y (2) un módulo PLA diseñado para permitir a los vehículos solicitar información útil basada en la localización enviada sin revelar su privacidad. Este módulo será el responsable de identificar el equilibrio entre la privacidad y la calidad del servicio mediante un nivel de privacidad ajustable. Ambos elementos se encuentran en los nodos edge de la red, que serán los encargados de aplicar las medidas de privacidad diferencial a la información. Los resultados obtenidos muestran que al aumentar el nivel de privacidad la calidad del servicio no baja drásticamente, lo que permite a los usuarios encontrar un equilibrio personalizado.

Miao et al. [30] proponen también un sistema basado en privacidad diferencial. En este caso se presenta un marco de trabajo denominado MEPA. Dentro de este marco de trabajo se muestra un algoritmo de privacidad diferencial y transmisión de peticiones denominado "Quadtree Differential Privacy" basado en "Hilbert curve division" (QTDP-H). Gracias a esta división de curvas se puede transformar un espacio de dos dimensiones en un espacio de una dimensión manteniendo poca pérdida de información, lo que permite disminuir el coste computacional que conlleva este tipo de técnicas. Comparado con los métodos tradicionales, se reducen tanto el tiempo medio de ejecución como el error medio relativo. Sin embargo, es necesario mencionar que el algoritmo propuesto no maneja bien la inconsistencia de los datos, lo que se propone en el artículo como línea de trabajo futuro.

IV-B. Cloud y edge semi-honesto

En ocasiones, el usuario no confiará en ningún elemento de la infraestructura. Así pues, en esta categoría encontramos las soluciones que también consideran a los nodos edge como atacantes semi-honestos. En este caso, es el dispositivo del propio usuario quien se encargará de proteger la privacidad de los datos de localización. Dentro de esta categoría encontramos varios enfoques. El primero estaría centrado en la utilización de cifrado homomórfico, y el segundo basado en la creación de servicios simulados que distraigan al atacante. Para finalizar, también se describe una solución basada en privacidad diferencial, similar a las anteriores.

En el caso del cifrado homomórfico, Jiang et al. [31] proponen dos protocolos de localización de la ubicación de sensores, los cuales permiten mantener su privacidad haciendo uso del cifrado homomórfico Paillier. La localización de los sensores se consigue a través del envío de la distancia del usuario con respecto a 3 estaciones base. Así, cuando el sensor quiere conocer su posición, solicita el cálculo de la distancia a las estaciones base. Éstas envían la información cifrada a los usuarios a través de los nodos edge – lo cuales no pueden extraer dicha información.

De esta forma, la información cifrada de las coordenadas del sensor pueden ser calculadas a partir de la información cifrada de la distancia obtenida con las 3 estaciones base. En todas estas comunicaciones la información se transmite cifrada, por lo que la privacidad está basada en la seguridad del esquema de cifrado. No obstante, cabe mencionarse que en ambos protocolos propuestos la clave pública del sensor y la localización de las estaciones base son públicas, por lo que un atacante externo puede elegir una localización y simular una interacción legítima.

Otro enfoque es el utilizado por He et al. [32], quienes consideran el uso de servicios simulados dentro de la red para dificultar las escuchas externas por parte de un atacante. El atacante intentará observar la trayectoria de los servicios mientras migran por los distintos nodos edge. Los servicios creados serían instancias independientes del mismo servicio que el usuario está utilizando, indistinguible del servicio original. Adicionalmente, respecto al patrón de movimiento de estos servicios, se estudian distintas estrategias basadas tanto en la imitación del comportamiento de los usuarios en la red como en la utilización de movimientos optimizados para minimizar la detección o el seguimiento del usuario real.

Así, una de las estrategias de optimización propuestas consigue llevar la precisión del seguimiento del atacante a cero cuando la movilidad del usuario real es lo suficientemente aleatoria. Asimismo, si el usuario real siempre permanece conectado al mismo nodo edge, es más apropiado utilizar la estrategia de la imitación de usuarios reales. No obstante, estos enfoques basado en el uso de servicios simulados presentan varios problemas. El inconveniente principal es el aumento del uso de recursos de la red. Además, si el atacante conoce las estrategias utilizadas por dichos servicios simulados, la utilidad de los mismos puede reducirse al mínimo.

Finalmente, Kaur et al. [33] plantean otra solución basada en privacidad diferencial para el caso de los datos de localización. El enfoque es similar a las soluciones vistas en la sección anterior, excepto por la incorporación de un nuevo elemento: el Secure Service Offloader (SSO). El SSO consiste en una nueva capa de nodos entre los nodos edge y el dispositivo, que sería la encargada de aplicar la privacidad diferencial a los datos que recoge de los dispositivos, evitando así la información sin ofuscar sea transmitida a los nodos Edge.

V. Privacidad en la comunicación

Del análisis de las comunicaciones, aunque estas estén protegidas mediante técnicas criptográficas seguras, se desprende también gran cantidad de información de información sensible, como las entidades que se comunican, la frecuencia con que lo hacen, el volumen de estas comunicaciones, etcétera. Precisamente por ello, se ha dedicado un gran esfuerzo de investigación a proporcionar soluciones capaces de proteger frente a atacantes con diversas capacidades de análisis de tráfico. Aunque la mayor parte de soluciones está enfocada a las comunicaciones en Internet, también se han estudiado estos problemas y desarrollado soluciones en otros entornos especializados, como las redes de sensores o entornos edge, que mostraremos a continuación.

Al igual que en el apartado anterior, volvemos a clasificar los trabajos de investigación según su consideración respecto a los atacantes.

V-A. Cloud semi-honesto

Suponiendo únicamente un servidor cloud honesto pero curioso, tenemos el trabajo de Zhang et al. [34], [35]. En él se presenta un sistema escalable basado en MEC, llamado Mobility Support System (MSS), que permite ocultar el tráfico y la localización de red del usuario móvil a los nodos de la red. El sistema se basa en crear un proxy de red dinámico y distribuido por cada usuario para conseguir minimizar la sobrecarga del tráfico y el coste computacional. El proxy

manejará el tráfico entrante y saliente del usuario. Los nodos objetivo serán los nodos al que se encuentra dirigido el tráfico, que pueden ser desde un servidor web a otro nodo móvil con un agente MSS. Además, dentro de este sistema se añade un nuevo elemento: el proveedor de servicio de movilidad (MSP). Este elemento manejará una flota de servidores, llamados routers virtuales (VR), que serán distribuidos dinámicamente desde los servidores centrales. Estos VR serán capaces de almacenar varios proxys.

Cuando un usuario quiere conectarse a otro nodo, el agente MSS en el host solicitará un proxy al MSP. Este proxy se asignará a una ubicación de red lo más cercana posible al nodo objetivo, y se conectará directamente a él. El tráfico entre el usuario y el nodo objetivo se envía a través del proxy utilizando una conexión entre el usuario y el proxy basada en su identidad. A continuación, la dirección del proxy será la expuesta a la red y no cambiará sin importar la ubicación del usuario. Por lo tanto, la dirección de red real del usuario y su movimiento se encuentran completamente ocultos del nodo objetivo. Cuando el nodo objetivo es un servidor estándar de internet y la conexión está vinculada a una dirección IP, MSS otorga un soporte adicional a la movilidad que permite a los protocolos de red tradicionales funcionar sin interrupción incluso si el usuario se encuentra desconectado temporalmente.

V-B. Edge semi-honesto

El trabajo de He et al. [36] trata únicamente la relación del dispositivo del usuario con el nodo edge, por lo que es este último el que se supone semi honesto. En esta investigación también se menciona la localización del usuario como elemento clave, pero además añade el patrón de uso de la red en la comunicación con los usuarios. El servidor edge puede ser capaz de extraer información estadística e incluso patrones del uso de la red de cada dispositivo basado en su historial de repartición de tareas y utilizar dicho patrón como huella para identificar la presencia de cierto usuario. Además, también podría determinar el servicio que esta ejecutándose en el lado del usuario, según el patrón de las tareas generadas por el servicio.

Para solucionar estos problemas, se propone un algoritmo de reparto de tareas basado en un proceso de decisión de Markov (CMDP) que tiene en cuenta la privacidad del usuario. Desde el punto de vista de envío de comunicación con la red, este algoritmo optimiza el retraso y rendimiento de consumo mientras que se mantiene un nivel de privacidad establecido con anterioridad.

Con el uso de este algoritmo, el dispositivo transmitirá algunas tareas – probablemente falsas – cuando las condiciones del canal sean inestables. Esto servirá para proteger su ubicación y patrón de uso. Sin embargo, como efecto secundario, este nivel de privacidad más elevado también conllevaría un mayor retraso y consumo de energía.

V-C. Atacante externo

Dentro de esta sección cabe destacar los conceptos mostrados en [37], donde se explora la seguridad a nivel físico. Se cree que este tipo de métodos basados en la teoría de la información proporcionan una mayor noción de privacidad que la criptografía y conllevan una menor carga computacional. Por lo tanto, pueden ser más apropiados para defenderse de atacantes externos en los entornos edge. Aprovechando la naturaleza inalámbrica del paradigma, se propone que el servidor edge envíe señales falsas para crear interferencia e impedir la escucha de atacantes externos, actuando sobre la privacidad general así como en la contextual. Estas señales de interferencia se generarán a la hora de la comunicación con los dispositivos finales, por lo que también se diseña un algoritmo de distribución de carga capaz de optimizar la combinación de las interferencias con las señales reales. Adicionalmente, se presenta un algoritmo para calcular la potencia óptima de las señales de interferencia. Finalmente, se presentan dos modos de operación basados en dos problemas de optimización, uno referente a la energía consumida y otro al retraso de ejecución.

Sin embargo, cabe mencionar que el trabajo habla únicamente de la comunicación de un nodo edge con un dispositivo. La inclusión de más antenas se menciona como futuras líneas de investigación, así como el estudio de nuevas técnicas de privacidad basadas en la capa física.

VI. Privacidad en el reparto de tareas

El reparto de tareas es una interesante aplicación que surge en entornos MEC con sensores móviles. En este tipo de aplicación, cobra mucha importancia la localización tanto de las tareas como del usuario que las emite y el que las recibe. Es por ello que muchos trabajos de investigación se centran únicamente en la privacidad dentro de este ámbito, en lugar de proporcionar un enfoque más genérico.

VI-A. Cloud semi-honesto

La solución propuesta por Shen et al [38], [39] se basa en el uso de técnicas de ofuscación para proteger el reparto de tareas de un servidor central semi-honesto ubicado en el cloud. El entorno propuesto se compone únicamente del servidor central, los servidores edge, y los usuarios móviles. Así, la protección de la privacidad recae sobre el servidor edge, basado principalmente en la ofuscación de información a través de un algoritmo genético.

El método de trabajo es el siguiente. Primero el servidor central publica la localización de las tareas a los servidores edge pertinentes. Después, los usuarios dentro del área designada envían su localización real a los nodos edge. Los servidores edge, tras recibir la información, ofuscan la localización de los usuarios y reparten las tareas en función de la localización ofuscada. A continuación, los usuarios que quieran participar en las tareas informarán al servidor edge, y realizarán la tarea. Tras recibir los resultados, el servidor edge enviará únicamente al servidor central los resultados y la localización ofuscada de los usuarios que han participado en completar las tareas.

VI-B. Cloud y edge semi-honesto

Dentro de la privacidad en el reparto de tareas, existen trabajos que consideran semi-honestos tanto al cloud como a los servidores edge. Uno de ellos, Ding et al. [40], propone un sistema de distribución de tareas para entornos edge basados en sensores móviles que tiene en cuenta la privacidad, y cuya característica principal es el uso del cifrado homomórfico para la localización del usuario, junto con la colaboración de varios

servidores edge para el reparto de la tarea cifrada. El esquema de comportamiento es similar a [38], [39], pero dando más peso a los solicitantes de tareas.

Primero, los solicitantes envían sus tareas al servidor central y se genera un par de claves para cada tarea. Después, el servidor central entrega las claves públicas a un servidor edge que se encuentre en la región solicitada, y entrega las claves privadas al servidor edge más cercano al primero. El primer servidor edge publica las tareas junto con sus claves públicas a los usuarios. A continuación, los usuarios solicitarán las tareas en las que se encuentren interesados mediante el envío al servidor edge de la distancia a las tareas, cifrada con la clave pública correspondiente. El servidor edge seleccionará los ganadores y los usuarios se desplazarán a la localización de la tarea, donde la completarán y subirán los datos cifrados al primer servidor edge. Después, el servidor edge cargará los datos recibidos en el servidor central junto con la distancia cifrada ofuscada. El servidor central pagará al servidor del edge y a los participantes, y por último, el servidor central es cifra y agrega los datos solicitados, y los devuelve al solicitante. Con este sistema, gracias al cifrado homomórfico, ni el servidor central ni el servidor edge pueden obtener la localización real de los usuarios durante el proceso.

En otro enfoque, Wu et al. [41] proponen añadir un elemento más a la mezcla: el centro de autorización (CA). Éste será el responsable de registrar todas las entidades del sistema y distribuir las claves necesarias. Todos los elementos del sistema son considerados semi-honestos excepto el CA, que es considerado totalmente honorable durante el desarrollo del protocolo.

El procedimiento sería el siguiente. Primero, el CA registra todas las entidades asignando los pares de claves correspondientes. Cada solicitador de tareas (TO) envía de forma anónima la tarea al servidor central. El servidor central reparte las tareas entre los servidores edge dependiendo de la localización. Estos últimos publicitan las tareas a los usuarios. Si un usuario quiere participar en alguna tarea, interactúa con el servidor edge para obtener los secretos correspondientes que además sirven como credenciales para la autorización de la tarea. Mientras tanto, el servidor central no puede saber que secretos ha solicitado el usuario. Finalmente los datos recogidos se ofuscan con un número aleatorio y se cifran con su clave pública antes de ser enviados. El servidor edge comprueba la integridad de todos los datos recogidos y calculan en colaboración con el servidor central la agregación de los mismos.

Respecto a la seguridad de los TOs, el sistema es capaz de mantener la privacidad de la identidad, de las tareas y de los resultados. Respecto a la privacidad de los participantes, el servidor central no conoce la relación entre las tareas y los participantes. Además, los nodos edge u otros usuarios no pueden identificar la información enviada de un usuario.

VII. Privacidad temporal

La privacidad temporal es otro de los aspectos a tener en cuenta dentro del paradigma de computación edge. La información temporal de conexión a la red puede ayudar a la predicción de comportamiento de individuo. Además, combinado con la localización, puede fomentar la creación de perfiles individuales.

No obstante, no se han encontrado estudios de investigación que traten en detalle el problema específico de la privacidad temporal dentro de los entornos de computación edge. Cabe mencionarse que si existen dichos estudios aplicados a otros paradigmas similares. Por ejemplo, en el entorno de las redes de sensores, se encuentran trabajos como el de Chakraborty et al. [42], que proponen mantener la privacidad temporal retrasando los envíos de algunos paquetes en algunos puntos de la ruta entre el sensor que detecta el evento y la estación base para que el atacante no pueda deducir el tiempo en el que tiene lugar dicho evento.

VIII. Desafíos Futuros

Tras una exhaustiva revisión de la literatura relativa a la privacidad contextual en entornos edge, se han detectado y analizado múltiples soluciones, que resumimos en la Tabla I.

A tenor de los resultados de este artículo, podemos afirmar que la investigación sobre privacidad contextual en entornos edge está aún en una fase de desarrollo muy temprana. Si bien existen áreas concretas donde hay ya un corpus de soluciones relativamente amplio, existen otras aún inexploradas. Como se puede observar en la Tabla I, hasta la fecha la mayor parte de soluciones se ha centrado en desarrollar soluciones relativas a la protección de información de localización y al reparto de tareas, siendo este último problema una versión especializada del primero. Otras áreas, en cambio, ha recibido poca o ninguna atención. En el ámbito de la privacidad en las comunicaciones existen pocas soluciones, aunque novedosas y variadas, pero presentan inconvenientes y/o son incapaces de dar una solución completa a los desafíos planteados. Por último, cabe destacar la ausencia absoluta de soluciones dedicadas a la protección de la privacidad temporal. En este sentido, consideramos que puede resultar de enorme interés analizar soluciones que haya surgido en otras áreas de investigación afines y estudiar si sería posible adaptar las soluciones propuestas en ellas a los entornos edge.

En lo relativo al modelo de atacante ocurre algo similar. En general, la mayoría de artículos se centran en atacantes alojados bien en el cloud o que comprenden toda la infraestructura cloud-edge. Sólo uno de los trabajos encontrados considera un modelo de atacante diferente, en concreto un atacante externo. Además, todos estos trabajos consideran un modelo de atacante semi-honesto, que trata de obtener información sensible sin excederse de sus funciones. Por tanto, el estudio de diferentes modelos de atacantes, especialmente aquellos activos o maliciosos, es un problema abierto que necesita de soluciones.

Por último, en el plano de las técnicas utilizadas para la protección de diferentes tipos de información, observamos que gran parte de ellas se sustentan en el uso de privacidad diferencial y cifrado homomórfico. Así pues, es necesario investigar otros mecanismos y técnicas que puedan ser aplicadas en entornos de computación edge y que se ajusten a su propia naturaleza. Sin duda, hay espacio para nuevas soluciones con enfoques innovadores.

IX. Conclusión

Las características de la computación edge, como la distribución y la limitación de recursos, provocan tanto la aparición de nuevos problemas de privacidad como la agravación de

Tabla I
MEDIDAS DE PRIVACIDAD SEGÚN LOS TRABAJOS DE INVESTIGACIÓN ESTUDIADOS

Referencia	Información contextual	Atacante		Técnica de protección
[29]	Localización	Cloud	Semi-honesto	Privacidad diferencial
[30]	Localización	Cloud	Semi-honesto	Privacidad diferencial
[31]	Localización	Cloud y Edge	Semi-honesto	Cifrado homomórfico
[32]	Localización	Cloud y Edge	Semi-honesto	Servicios Simulados
[33]	Localización	Cloud y Edge	Semi-honesto	Privacidad diferencial
[34], [35]	Comunicación	Cloud	Semi-honesto	Proxy
[36]	Comunicación	Edge	Semi-honesto	Algoritmo de reparto de carga
[37]	Comunicación	Externo	Pasivo	Señales de interferencia
[38], [39]	Reparto de tareas	Cloud	Semi-honesto	Privacidad diferencial
[40]	Reparto de tareas	Cloud y Edge	Semi-honesto	Cifrado homomórfico
[41]	Reparto de tareas	Cloud y Edge	Semi-honesto	Privacidad diferencial

otros existentes, en comparación con otros paradigmas afines como el paradigma cloud. Algunas de las medidas de privacidad efectivas en entornos cloud no pueden ser aplicadas directamente en la computación edge debido a dichas características. Por tanto, sería necesario adaptarlas al nuevo entorno o innovar para mitigar los problemas emergentes.

Este artículo ha revisado y analizado la literatura relativa a los problemas de privacidad contextual en entornos edge. Los aspectos más cubiertos por la literatura existente son la privacidad de la localización y la privacidad durante el reparto de tareas en la computación edge. El estado de esta investigación es bastante significativa, considerando la novedad de este paradigma y su desarrollo concurrente. Sin embargo, existen varios aspectos como la privacidad en el contexto de las comunicaciones y en el aspecto temporal, que carecen de soluciones suficientes, sobre todo si lo comparamos con otros paradigmas similares. Esto es, sin lugar a dudas, una oportunidad para investigadores interesados en dicho ámbito que pueden aportar soluciones tempranas y novedosas dentro del paradigma.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación a través del proyecto SecureEDGE (PID2019-110565RB-I00), la Consejería de Economía, Conocimiento, Empresas y Universidad de la Junta de Andalucía a través del proyecto SAVE (P18-TP-3724) y el proyecto BIG$^{\text{Priv}}$DATA (UMA20-FEDERJA-082) del Programa Operativo FEDER Andalucía 2014-2020.

REFERENCIAS

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 13–16.

[2] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, aug 2019.

[3] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, Part 2, pp. 680–698, jan 2018.

[4] S. Gupta, R. Garg, N. Gupta, W. S. Alnumay, U. Ghosh, and P. K. Sharma, "Energy-efficient dynamic homomorphic security scheme for fog computing in IoT networks," *Journal of Information Security and Applications*, vol. 58, p. 102768, 5 2021. [Online]. Available: https://abdn.pure.elsevier.com/en/publications/energy-efficient-dynamic-homomorphic-security-scheme-for-fog-comp

[5] J. N. Liu, J. Weng, A. Yang, Y. Chen, and X. Lin, "Enabling efficient and privacy-preserving aggregation communication and function query for fog computing-based smart grid," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 247–257, 1 2020.

[6] F. Yildirim Okay, S. Ozdemir, and Y. Xiao, "Fog computing-based privacy preserving data aggregation protocols," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 4, 4 2020.

[7] Z. Mahmood, Ed., *Fog Computing: Concepts, Frameworks and Technologies*. Springer International Publishing, 2018.

[8] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2018.

[9] M. Mukherjee, L. Shu, and D. Wang, "Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

[10] P. Bellavista, J. Berrocal, A. Corradi, S. K. Das, L. Foschini, and A. Zanni, "A survey on fog computing for the Internet of Things," *Pervasive and Mobile Computing*, vol. 52, pp. 71–99, jan 2019.

[11] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, "A Survey on Edge Computing Systems and Tools," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1537–1562, aug 2019.

[12] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[13] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[14] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How Can Edge Computing Benefit From Software-Defined Networking: A Survey, Use Cases, and Future Directions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2359–2391, 2017.

[15] P. Zhang, M. Zhou, and G. Fortino, "Security and trust issues in Fog computing: A survey," *Future Generation Computer Systems*, vol. 88, pp. 16–27, nov 2018.

[16] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing Fog Computing for Internet of Things Applications: Challenges and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2018.

[17] M. Yahuza, M. Y. I. B. Idris, A. W. B. A. Wahab, A. T. S. Ho, S. Khan, S. N. B. Musa, and A. Z. B. Taha, "Systematic Review on Security and Privacy Requirements in Edge Computing: State of the Art and Future Research Opportunities," *IEEE Access*, vol. 8, pp. 76 541–76 567, 2020.

[18] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Survey on Multi-Access Edge Computing Security and Privacy," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 2, pp. 1078–1124, 4 2021.

[19] B. Ali, M. A. Gregory, and S. Li, "Multi-access edge computing architecture, data security and privacy: A review," *IEEE Access*, vol. 9, pp. 18 706–18 721, 2021.

[20] T. Khalid, M. A. K. Abbasi, M. Zuraiz, A. N. Khan, M. Ali, R. W. Ahmad, J. J. Rodrigues, and M. Aslam, "A survey on privacy and access control schemes in fog computing," *International Journal of Communication Systems*, vol. 34, no. 2, 1 2021.

[21] Y. I. Alzoubi, V. H. Osmanaj, A. Jaradat, and A. Al-Ahmad, "Fog computing security and privacy for the Internet of Thing applications: State-of-the-art," *Security and Privacy*, vol. 4, no. 2, p. e145, 3 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/spy2.145https://onlinelibrary.wiley.com/doi/abs/10.1002/spy2.145https://onlinelibrary.wiley.com/doi/10.1002/spy2.145

[22] T. Khalid, M. A. K. Abbasi, M. Zuraiz, A. N. Khan, M. Ali, R. W. Ahmad, J. J. Rodrigues, and M. Aslam, "A survey on privacy and access control schemes in fog computing," *International Journal of Communication Systems*, p. e4181, oct 2019.

[23] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," *IEEE Access*, vol. 6, pp. 18 209–18 237, 2018.
[24] Z. Tian, Y. Wang, Y. Sun, and J. Qiu, "Location privacy challenges in mobile edge computing: Classification and exploration," *IEEE Network*, vol. 34, no. 2, pp. 52–56, mar 2020.
[25] IBM News Release. Ibm and nokia siemens networks announce world's first mobile edge computing platform. [Online]. Available: https://www-03.ibm.com/press/us/en/pressrelease/40490.wss
[26] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.
[27] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1–35, 2018.
[28] R. Rios, J. Lopez, and J. Cuellar, *Location Privacy in Wireless Sensor Networks*, ser. CRC Series in Security, Privacy and Trust. Taylor & Francis, 2016. [Online]. Available: https://www.crcpress.com/Location-Privacy-in-Wireless-Sensor-Networks/Rios-Lopez-Cuellar/p/book/9781498776332
[29] L. Zhou, L. Yu, S. Du, H. Zhu, and C. Chen, "Achieving differentially private location privacy in edge-assistant connected vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4472–4481, 6 2019.
[30] Q. Miao, W. Jing, and H. Song, "Differential privacy–based location privacy enhancing in edge computing," in *Concurrency and Computation: Practice and Experience*, vol. 31, no. 8. John Wiley and Sons Ltd, 4 2019.
[31] H. Jiang, H. Wang, Z. Zheng, and Q. Xu, "Privacy preserved wireless sensor location protocols based on mobile edge computing," *Computers and Security*, vol. 84, pp. 393–401, 7 2019.
[32] T. He, E. N. Ciftcioglu, S. Wang, and K. S. Chan, "Location privacy in mobile edge clouds: A chaff-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2625–2636, 11 2017.
[33] J. Kaur, A. Agrawal, and R. A. Khan, "Encryfuscation: A model for preserving data and location privacy in fog based IoT scenario," *Journal of King Saud University - Computer and Information Sciences*, 3 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S131915782200074X
[34] P. Zhang, M. Durresi, and A. Durresi, "Network Location Privacy Protection with Multi-access Edge Computing," in *Advances in Intelligent Systems and Computing*, vol. 926. Springer Verlag, 2020, pp. 1342–1352.
[35] ——, "Multi-access edge computing aided mobility for privacy protection in Internet of Things," *Computing*, vol. 101, no. 7, pp. 729–742, 7 2019.
[36] X. He, J. Liu, R. Jin, and H. Dai, "Privacy-Aware Offloading in Mobile-Edge Computing," *2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings*, vol. 2018-January, pp. 1–6, 7 2017.
[37] X. He, R. Jin, and H. Dai, "Physical-Layer Assisted Privacy-Preserving Offloading in Mobile-Edge Computing," *IEEE International Conference on Communications*, vol. 2019-May, 5 2019.
[38] H. Shen, G. Bai, Y. Hu, and T. Wang, "P2TA: Privacy-preserving task allocation for edge computing enhanced mobile crowdsensing," *Journal of Systems Architecture*, vol. 97, pp. 130–141, 8 2019.
[39] Y. Hu, H. Shen, G. Bai, and T. Wang, "Privacy-preserving task allocation for edge computing enhanced mobile crowdsensing," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11337 LNCS. Springer Verlag, 2018, pp. 431–446.
[40] X. Ding, R. Lv, X. Pang, J. Hu, Z. Wang, X. Yang, and X. Li, "Privacy-preserving task allocation for edge computing-based mobile crowdsensing," *Computers & Electrical Engineering*, vol. 97, p. 107528, 1 2022.
[41] H. Wu, L. Wang, and G. Xue, "Privacy-Aware Task Allocation and Data Aggregation in Fog-Assisted Spatial Crowdsourcing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 589–602, 1 2020.
[42] B. Chakraborty, S. Verma, and K. P. Singh, "Temporal Differential Privacy in Wireless Sensor Networks," *Journal of Network and Computer Applications*, vol. 155, p. 102548, 4 2020.

ChaosXploit: A Security Chaos Engineering framework based on Attack Trees

Sara Palacios Chavarro[1], Daniel Díaz-López[1], Pantaleone Nespoli[2]
[1]School of Engineering, Science and Technology, Universidad del Rosario, Bogotá, Colombia
{sara.palaciosc, danielo.diaz}@urosario.edu.co
[2]Department of Information and Communications Engineering, University of Murcia, 30100, Murcia, Spain
pantaleone.nespoli@um.es

***Abstract*—Security incidents may have several origins. However, many times they are caused due to components that are supposed to be correctly configured or deployed. Traditional methods may not detect those security assumptions, and new alternatives need to be tried. Security Chaos Engineering (SCE) represents a new way to detect such failing components to protect assets under cyber risk scenarios. This paper proposes ChaosXploit, a security chaos engineering framework based on attack trees, which leverages the chaos engineering methodology along with a knowledge database composed of attack trees to detect and exploit vulnerabilities in different targets as part of an offensive security exercise. Once the proposal is explained, a set of experiments are conducted to validate the feasibility of ChaosXploit to validate the security of cloud managed services, i.e. Amazon buckets, which may be prone to misconfigurations.**

***Index Terms*—Security Chaos Engineering, attack trees, cloud managed services, vulnerabilities**

Contribution type: *Original research*

I. Introduction

Site Reliability Engineering (SRE) is defined as a discipline focused on improving systems' design and operation to make them more scalable, reliable, and efficient [1]. Although SRE has been approached with different methodologies, over the last ten years, a new approach for testing the resiliency of distributed systems has emerged [2], which is known as Chaos Engineering (CE). CE is used to identify the system's immunities when damage is injected, so vulnerabilities can be found and subsequently mitigated. CE tests are designed to "build confidence in the system's capability to withstand turbulent conditions in production" [3].

Designing CE experiments implies defining a prepared and controlled environment to analyze a target system [4] and applying a scientific method that allows one to observe the environment, define a set of hypotheses, and validate them. CE has proven to be extremely useful in validating attributes of reliability and availability in a production environment. Nevertheless, checking these elements may not be enough if the ultimate goal is a holistic validation of the system's security level. It might be the case in different distributed systems, such as secure IoT services [5] or personal data managers with high-security requirements [6].

Considering what was previously said, some efforts have come up towards applying CE to cybersecurity in the last five years, known as Security Chaos Engineering (SCE). In particular, SCE aims to use the CE principles to evaluate the three most important attributes of a system from a holistic cybersecurity perspective, i.e., confidentiality, integrity, and availability [7].

Noting that this new methodology can have a great impact on new developments by reducing vulnerabilities through experimentation, we have decided to follow this innovative line to provide a new security CE framework based on attack trees, known as ChaosXploit.

The main contributions of this paper are summarized as follows:

- The proposal of a SCE framework named ChaosXploit, which uses attack trees as the main flowchart for the execution of attacks, and contains three main components: an observer, an experiment runner, and a knowledge database.
- The design of an attack tree that pursues an attack goal of extraction or modification of information of AWS S3 buckets that enriches the knowledge database of ChaosXploit.
- The execution of a set of experiments that validates the feasibility of ChaosXploit to execute an attack tree over a specific target, i.e., AWS S3 bucket, exposing multiple misconfigurations.

This paper is structured as follows: Section II collects the most recent works proposed related to SCE, exploring their pros and cons. Then, Section III describes ChaosXploit, our proposed framework to conduct SCE experiments. Next, in Section IV some experiments over ChaosXploit are proposed are executed. Finally, Section V concludes the work, showing some future work that can improve our proposal.

II. State of the art

Several research works have been proposed in the literature so far that leverage the robust capabilities of CE. Nevertheless, the application of the methodology, together with its definition, has been ambiguous.

Since the release of *Chaos Monkey* in 2011 by Netflix [8], CE has been chiefly applied to test the resilience of cloud and virtualized infrastructures, arguing on the potential benefits that the chaotic methodology could bring.

In this sense, Camacho *et al.* [9] proposed *Pystol*, a fault injection platform to test the resiliency of hybrid-cloud systems in adverse circumstances. Available as an open-source framework, Pystol exploits CE's abilities in the form of a Software Product Line (SPL) that can be mounted on top of cloud ecosystems. The platform is then tested in a production-ready environment, executed using standard Kubernetes objects and APIs and Amazon Web Services to deploy the cluster with three use cases.

Furthermore, the work in [10] presented *ChaosOrca*, an open-source CE-based fault injector for system calls in containerized applications. That is, the system can estimate the self-protection capability of any Docker-based microservice concerning system call errors. In particular, ChaosOrca formalizes the steady-state of the container by automatically recording several system metrics (e.g., CPU and RAM consumption, network I/O). Then, perturbations are injected into the system calls invoked by the dockerized application in an isolated fashion, without impacting the normal operations of possible other containers. The prototype is tested in three case studies of Docker microservices, namely Torrent, Nginx, and Bookinfo, showing promising results in detecting resilience weaknesses.

Moreover, Zhang *et al.* [11] proposed *ChaosMachine*, an open-source and extensible CE system in Java aiming to analyze the exception-handling capabilities in production environments. So, ChaosMachine can reveal potential resilience problems of try-catch blocks with an architecture composed of three components: i) a monitoring sidecar, ii) a perturbation injector, and iii) the chaos controller. The framework is then tested with three large-scale open-source Java applications summing 630k code lines with realistic workloads, demonstrating its capacities in production environments.

Recently, the main target of CE has been slightly moved from resilience to including security concerns surrounding a system. Assuming that security failures are going to happen doubtless, SCE aims at testing the security controls of a system through proactive experiments and, thus, building confidence in the system's capabilities to defend against malicious conditions. Unfortunately, since this paradigm change has happened lately, the amount of academic work and tools are still scarce. To this extent, *ChaoSlingr* is the first open-source software tool to demonstrate the possibility of applying the principles of CE to information security[1]. The system was designed to operate on AWS by a team at UnitedHealth Group led by Aaron Rinehart to exhibit a simplified manner for writing security chaos experiments [12]. From the main project, several organizations have started to utilize ChaosSlingr to design their chaotic experiments.

Additionally, the work in [7] presented *CloudStrike*, a software tool that applies Risk-Driven Fault Injection (RDFI) to cloud infrastructures. For the sake of the reader, the tool was firstly proposed in [13]. Specifically, RDFI extends the application of CE to include cloud security without losing the resilience viewpoint, i.e., by injecting security faults using attack graphs. The SCE-based proposal is then tested in some cloud services of leading platforms, namely, AWS and Google Cloud Platform. Interestingly, the authors claim they compute the risk to which the assets are exposed using the CVSS. Later on, the same authors leveraged the SCE strategies to test another proposal, CSBAuditor, a cloud security framework that can constantly monitor a specific cloud infrastructure to detect possible malicious activities [14].

Also, the application of SCE to enhance API security is defined in [15]. Particularly, the authors propose utilizing this methodology to test the configuration of the API's security controls, exposing early vulnerabilities.

III. Proposal of ChaosXploit

This section describes ChaosXploit, a SCE-powered framework composed of different modules that support the application of CE methodology to test security in different kinds of information systems. The architecture of the proposal is depicted in Figure 1, and each internal module is described in the following sections.

A. Knowledge database

The knowledge database is responsible for providing the steps required to conduct an offensive SCE experiment executed by a team (blue team) interested in maturing a defensive strategy. Thus, this module is composed of a set of attack trees and a hypothesis generator.

1) Attack trees: This module is in charge of delivering the intelligence for executing the SCE experiments. Such intelligence is represented by different attack trees, where each tree clusters different branches focused on achieving a specific attack goal, e.g., gaining access to data stored in a cloud storage solution. So, different attack goals may be pursued as attack trees are contained in the knowledge database. Each branch of an attack tree gathers different offensive actions that may be conducted to achieve the final attack goal, where an action may be a python script, an HTTP request, or some process to be run on the operating system. It is worth mentioning that attack trees for different types of targets may be defined, such as trees for user applications, managed cloud services, Kubernetes, and network devices, among others.

2) Hypothesis Generator: The intelligence contained in the attack trees needs to be converted to a hypothesis so it can be consumed by the other modules of ChaosXploit. So, the Hypothesis Generator is responsible for translating the branch actions contained in an attack tree into a form readable for the module that executes the SCE experiments, i.e. the exploiter. Each hypothesis generated by this module is a statement about the system being tested that must be refuted or confirmed by the SCE experiments, e.g. an organization will not expose private data when the recognition tool Foca[2] is pointed out to the main domain.

[1]https://github.com/Optum/ChaoSlingr

[2]https://github.com/ElevenPaths/FOCA

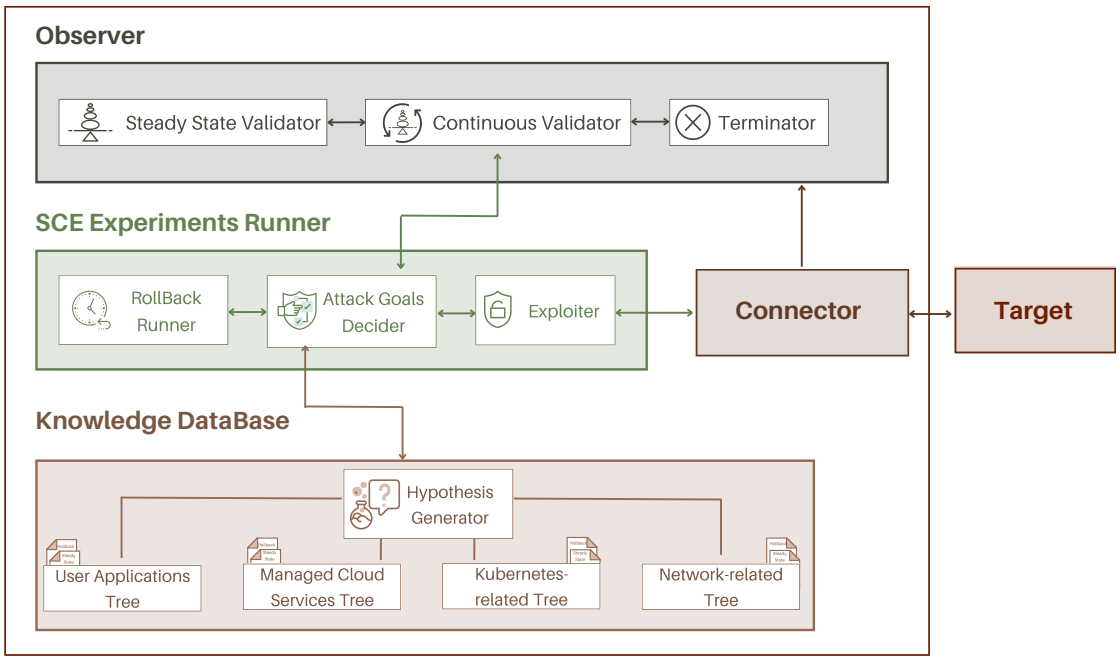


Figure 1: Proposed architecture of ChaosXploit

B. Observer

The observer groups all the activities related to the observation of both the target and the SCE experiment. This module is important because it allows for monitoring of specific conditions of the target before, along, and after the execution of the SCE experiments. This module is composed of a steady-state validator, a continuous validator, and a terminator.

1) Steady state Validator: The steady-state validator is in charge of verifying the steady-state hypothesis in the target that represents estable conditions. These conditions will depend on the attack goal and the specific hypothesis being tested. For example, a normal condition may be a well-formed response from a web server.

2) Continuous validator: The continuous validator permits verifying specific signals detected from the target, which allows determining the results of an interaction between the exploiter and the target. These signals are particularly important because they may indicate if a current action included in a branch of an attack tree was successful, so the following action in the branch should be triggered, or they simply may indicate that the target is not vulnerable and the following actions of the branch should not be executed.

3) Terminator: The terminator observes the failure states of the SCE experiment to define the actions to follow consequently. For example, if the target gets unresponsive due to the execution of a SCE experiment, a failure state will be launched and the terminator will be able to inform the Rollback Runner so it can restore the target.

C. SCE Experiments Runner

The SCE Experiments Runner is in charge of the SCE experiment's execution over a target to validate or refute a hypothesis. This component is fundamental because it not only leads the interaction with the target but also centralizes the communication with the observer and knowledge database. It consists of three main elements: attack goal decider, exploiter, and rollback runner.

1) Attack goal decider: The attack goal decider receives a defined goal attack as input to be tested over a target. Such attack goal may be contributed by the user of ChaosXploit who is interested in probing if a particular system is susceptible to a specific attack. Then, the attack goal decider requests the knowledge database for the proper attack tree that matches such a defined goal.

2) Exploiter: The exploiter executes the SCE experiment over a target to validate or refute a hypothesis. With such purpose, the exploiter performs the offensive actions defined previously by the attack tree obtained from the knowledge database. Besides, it is also able to collect information about specific responses coming from the target to define the next step in an attack.

3) Rollback runner: An experiment may contain a sequence of actions that reverse what was undone during the experiment. These actions will be called by the Rollback Runner after the Continuous Validator finishes its execution regardless of whether an error occurred in the process or not.

D. Connector

The connector is responsible for searching for the most suitable extension to connect to the target on which the user

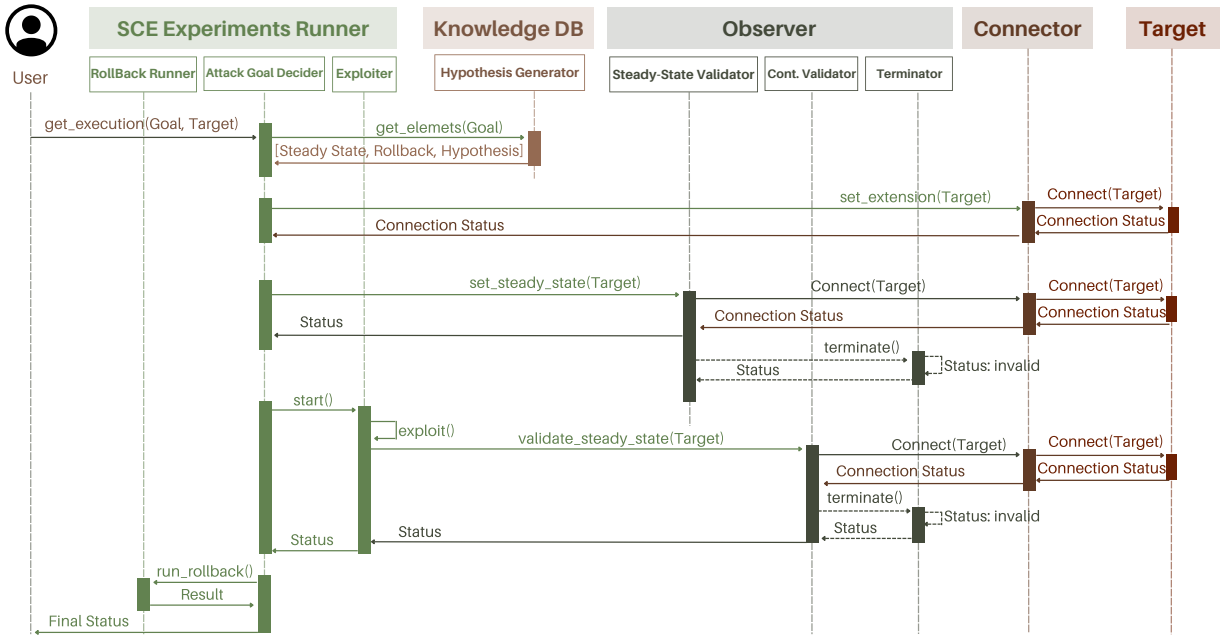


Figure 2: Flow diagram of the execution of a SCE experiment in ChaosXploit

wants to run the experiment. Once an extension has been defined, the connector establishes the link with the target and tests that the scenario is adequate to run the SCE experiment.

The interactions between the components of ChaosXploit are shown in Figure 2. First, the user of ChaosXploit requests the Attack Goal Decider the execution of a SCE experiment, informing: the attack goal to be considered and the target where the SCE experiment should be addressed. Then, the Attack Goal Decider gets from the knowledge database the steady-state of the experiment, the rollback procedure, and the most proper hypothesis (attack tree) that matches the attack goal desired by the user. The Attack Goal Decider also requests to the Connector the preparation of the extension for the target informed by the user. When a connection to the target is established and a hypothesis is defined, then the Attack Goal Decider does the following actions: i) sets the steady-state of the experiment in the Observer, ii) starts the execution of the steps defined in the first branch of the attack tree with the help of the Exploiter, and iii) keeps continuous communication with the Continuous Validator to monitor the execution of the exploitation in progress and in that way be aware of the attack goal was achieved. If the Continuous Validation fails, then the termination process is activated by the Terminator. The experiment ends with the execution of the Rollback Runner to restore everything.

IV. Experiments

Multiple experiments have been conducted using the ChaosXploit proposal mentioned in Section III, which are also available in the repository of this project[3]. Based on the fact that AWS S3 buckets and Elasticsearch databases account for nearly 45% of the cloud misconfigured and compromised technologies [16], ChaosXploit focuses on evaluating the security of the AWS S3 service on this experiment. It considers the possible configurations and whether they permit establishing a connection, whether they are public or private buckets or whether they permit getting the configured Access Control Lists (ACLs) which allow managing the access to the buckets and their objects. These lists define which AWS accounts or groups have access and what kind of permissions they have.

This section of experiments is composed of the following subsections: Settings IV-A, where the hardware and software requirements to develop the experiment, are specified. Definition of the knowledge database IV-B, where the attack tree is presented together with the specification of the branch chosen for the experiment. SCE experiment IV-C in which the steady-state and the hypothesis of the experiment are defined, as well as the input parameters and the monitored variables. Finally, Results Analysis IV-D presents the results obtained.

A. Settings

The following setup was used to make use of ChaosXploit:

- Hardware: the experiment was executed on a Fedora OS with AMD Ryzen 5 3500U CPU, 8GB RAM, and 512GB SSD.
- Internal Components: Some of the components of ChaosXploit have been built over existing modules of ChaosToolkit, as it is an open-source framework that

[3]https://github.com/SaraPalaciosCh/ChaosXploit

allows its extension and improvement to make it oriented to security purposes. ChaosToolkit was chosen since this tool simply allows automation of the experiments using *json* files. The connection to the different targets (buckets) was done using boto3 (SDK for python).

- Environment: The first version of ChaosXploit should be installed on a virtual environment with *python3.7* and *Chaostoolkit* installed.

B. Definition of the Knowledge Database

In Figure 3 it is possible to observe the attack tree implemented for this experiment. It starts with the attacker finding public buckets by either enumerating the names or searching sites such as the Wayback Machine. Then, the next action seeks to confirm if the attacker succeeds in establishing a connection to the bucket. Once the connection is established, the attacker can follow one of the 4 different branches to reach the attack goal identified in the tree as the last box: extract or modify information. These paths are described as:

- **Branch 1**: where the attacker has gained access to the bucket without any permission or authentication process. Once inside, he can inspect the objects contained in the storage system, and read the Access Control Lists (ACL). If these ACLs have permissions open to the entire public, then the attacker will be able to reach the attack goal.
- **Branch 2**: it is a path taken by the attacker in case the bucket has the access permissions properly configured. At this point, the attacker could make use of possible vulnerabilities in the AWS access control system, also known as IAM, to then elevate his privileges and gain access to the bucket's information, thus achieving the attack goal.
- **Branch 3**: in which the attacker can use brute-forcing techniques to compromise admin credentials and thereby gain access.
- **Branch 4**: where the attacker can use social engineering techniques such as phishing to compromise credentials and gain access.

It is important to note that the execution of the first branch was included in the scope of this project, as the actions included in such branch were automatable completely. Other branches could also be implemented through a combination of manual and automatic actions.

C. SCE experiment

The goal of this experiment stems from the fact that Amazon S3 allows data to be stored and protected from unauthorized access with encryption features and access management tools. However, the shared responsibility model of cloud services has led the creators of this type of storage to commit flaws during security configuration. Leaving the information open to the public, putting its confidentiality, integrity, and availability at risk.

Based on the goal of the attack tree (Extract or modify Information), it is possible to define the experiment following the scientific method as follows:

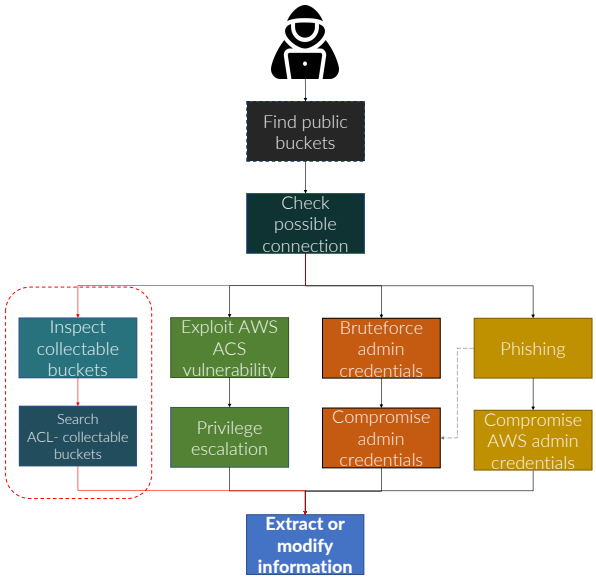


Figure 3: Attack Tree for the experimental scenario, highlighting the implemented path

- Observability: Public AWS S3 Buckets.
- Steady State: The buckets to be analyzed suggest having the access controls properly configured.
- Hypothesis: if you try to access the objects stored in the buckets, then you will not be able to see their contents or the associated access controls since they are properly configured to prevent information leaks.

Implementation of the first branch of the attack tree defined for this scenario is described below. First, the finding of public buckets was done using enumeration techniques by considering regular expressions. Since Amazon S3 has defined a series of requirements for the bucket names, this makes it very easy for the attacker to enumerate them. Then, the connection check was performed using boto3, the AWS SDK for python. With this step, we were able to clean up the buckets leaving out those that no longer exist or had invalid names. Afterward, ChaosXploit inspects the buckets to identify if their objects can be read and finally searches if there are buckets that allow access to the ACLs.

As shown in Table I, different parameters were considered as input values for ChaosXploit. First, the *domain* is an optional input that should contain the name of the organization to be analyzed. We have considered this option since ChaosXploit can be used as an internal audit tool. Therefore, with this argument, the enumeration of the buckets will be limited to all those that are related to the given domain. In case this input is not provided, ChaosXploit will generate a list of names using brute-force, wordlists, and bucket naming rules defined by AWS. Second, the number of *threads* is considered as an input, so that the process of connecting and reading buckets

may be performed in parallel on the different cores, according to the defined thread's value. Third, the *mode* indicates the type of analysis to be performed, whether it aims to find *Object-Collectable* or *ACL-Collectable* buckets. The last input, *output*, is a file name used to store the results and feed the ChaosXploit continuous validator.

Regarding the monitored variables, three were considered: i) Buckets that have public objects that can be accessed by anyone, denoted by **Object-Collectable** in Table I, ii) Buckets that have public ACLs, and can be accessed by anyone denoted by **ACL-Collectable** and iii) the **Permissions** obtained from the ACLs.

Monitored Variables	
Name	**Description**
Object-Collectable	No. of buckets that have public objects and are accessible by anyone
ACL-Collectable	No. of buckets that have public ACLs and are accessible by anyone
Permissions	No. of permissions obtained from the ACL.
Input Parameters	
Name	**Description**
Domain(Optional)	Domain name to which you want to identify the buckets
Threads	Execution Threads
Mode	Object-Collectable or ACL-Collectable
Output	Output File

Table I: Monitored variables and input parameters for experiments.

D. Results Analysis

ChaosXploit's functionality was tested using a list of 3k buckets obtained through a bucket name enumeration process which can be performed using tools such as s3enum[4], bucketkicker[5] or Sublist3r[6].

As seen in the upper left part of Figure 4, all possible actions of the attack tree were executed by ChaosXploit. It is possible to identify that for the second one (Check possible connection), out of the 3k buckets listed, 271 did not allow a connection. This is because the bucket no longer existed or had an invalid name, e.g it did not follow the common bucket naming characteristics proposed by AWS. This leaves us with 2729 buckets remaining to test.

In the case of the third act of the attack tree (Inspect collectible buckets), 2454 buckets were well configured and passed the steady-state defined in our experiment, since they did not allow reading files or permissions listed in the ACLs. However, 275 did not pass validation.

The lower left part of Figure 4 shows the file extensions that were extracted from 252 buckets that were Object Collectable. From each bucket, only the first 50 objects were collected, since some buckets had more than 100000 files stored, for a total of 7465 collected files. Of all these files it was possible to identify that more than 2000 were images (jpg and png) and approximately 1250 were categorized as others because they could be log files, folders, or had no extension.

To analyze the users and user groups associated with each bucket we first need to know that Amazon S3 has a set of predefined groups:

- **AuthenticatedUsers group** representing all AWS accounts.
- **AllUsers group** allowing anyone in the world to access the resource.
- **LogDelivery group** allowing access logs to be written to the bucket.

Additionally, AWS defines also the following types of permissions:

- **READ** Allows grantee to list the objects in the bucket.
- **WRITE** Allows grantee to create new objects in the bucket. For the bucket and object owners of existing objects, also allows deletions and overwrites of those objects.
- **READ_ACP** Allows grantee to read the bucket ACL
- **WRITE_ACP** Allows grantee to write the ACL for the applicable bucket.
- **FULL_CONTROL** Allows grantee the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket

In the upper right part of Figure 4 is possible to identify that 92 of the 257 buckets allowed the extraction of the ACLs. Up to 13 permissions per bucket were identified. These showed information about the user who owned the bucket, known as **CanonicalUser** by AWS, or about the user groups that had access to it. Then, it is worth noting that for canonical users the FULL_CONTROL permission was enabled for 84 buckets (91.3%), and in the case of the user groups, 64 (69.5%) of them allow the reading of the stored objects (READ permission) and 89 (96.7%) allow the reading of the ACLs (READ_ACP permission).

Finally, we analyze the results of those buckets that allowed the extraction of both objects and ACLs. As seen in the lower right part of Figure 4, 69 buckets (25%) allowed both tasks to be performed. These were filtered by the *AllUsers* and *AuthenticatedUsers* user groups and it was identified that 41(38.3%) from the *AllUsers* group and 17 (29.8%) from the *AuthenticatedUsers* group were allowed to read the ACLs and the objects. Nevertheless, it was identified that 11 buckets (10.3%) from the *AllUsers* group and 11 buckets (19.3%) from the *AuthenticadedUsers* group allowed the modification of their content (WRITE permission) and the alteration of the ACLs (WRITE_ACP permission), indicating a big flaw that could compromise severally the confidentiality, integrity, and availability of the stored data.

With these results, we have noticed the importance of not only providing a tool for the detection of flaws or vulnerabilities but also seeing it as an aid to infer possible mitigations to prevent the exploitation of such vulnerabilities.

[4]https://github.com/koenrh/s3enum
[5]https://github.com/craighays/bucketkicker
[6]https://github.com/aboul3la/Sublist3r

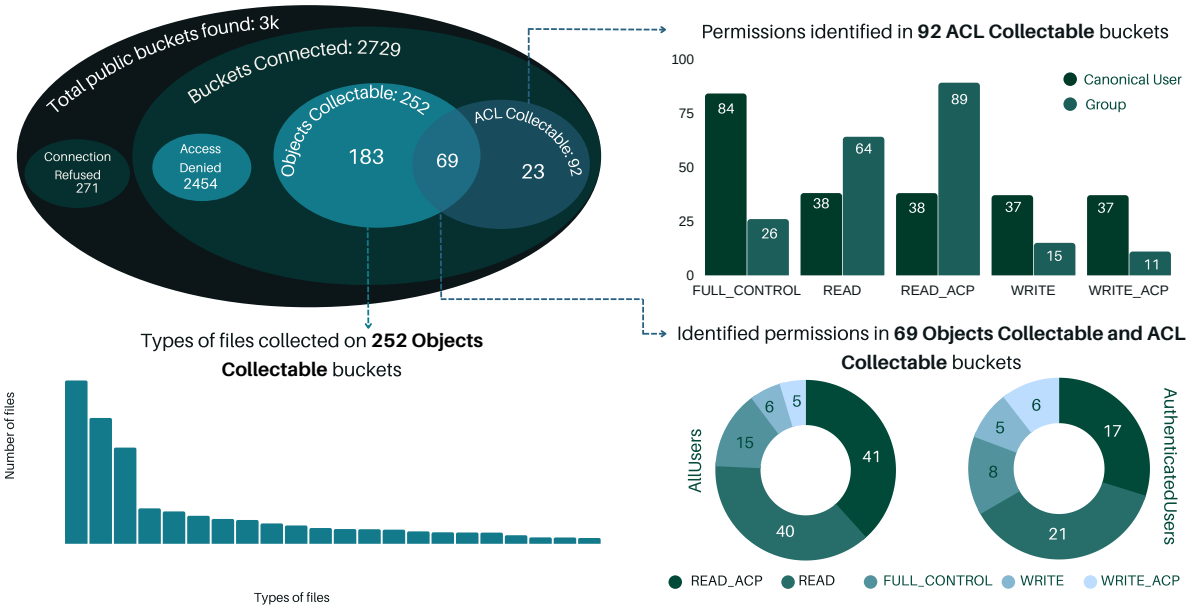


Figure 4: Results of the execution of each action included in the first branch of the attack tree

V. Conclusions and Future Work

No one could expect the impactful digital revolution we live in, changing substantially how we live our lives with great benefit. On the downside, such a change also implies the existence of ill-motivated entities that constantly try to attack connected systems to damage the confidentiality, integrity, or availability of the provided services. Such threat entities use increasingly advanced techniques, for example, based on malware campaigns [17] or threats addressed to a specific technology [18]. Over the last ten years, a novel paradigm has emerged, the so-called Chaos Engineering, whose main objective consists of testing the resiliency of distributed and complex systems. More recently, the paradigm has evolved to embrace the entire cybersecurity ecosystem, i.e., the Security Chaos Engineering, to defend the system assets against cyber-attacks through continuous and rigorous experimentations on possible security holes and consequent mitigations.

In this paper, we proposed ChaosXploit, a SCE-powered framework that can conduct Security Chaos Engineering experiments on different target architectures. Based on the hypothesis generated by the knowledge database and the attack representations, ChaosXploit executes SCE experiments over a target to find a potential security problem as an ultimate goal. Also, ChaosXploit features an observer which is in charge of verifying the change between the steady state of a certain hypothesis and the current state of the system. To prove the capabilities of ChaosXploit, a set of experiments was conducted on several AWS S3 buckets, evaluating their security characteristics with SCE. Results demonstrated that our approach can be successful, highlighting several unprotected buckets for a specific attack path. ChaosXploit was made publicly available for the cybersecurity community through the repository of the project[7].

Future work will explore the possibility of widening the ChaosXploit framework target architectures to include other use cases, systems, or providers. Besides, integrating a recommendation module to suggest countermeasures once a security flaw is discovered is worth investigating. Moreover, the performance of ChaosXploit should be further evaluated to prove its usefulness in performance-demanding scenarios.

Acknowledgment

This work has been supported by Universidad del Rosario (Bogotá) through the project “IV-TFA043 - Developing Cyber Intelligence Capacities for the Prevention of Crime”.

References

[1] B. Beyer, C. Jones, J. Petoff, and N. R. Murphy, *Site Reliability Engineering: How Google Runs Production Systems*, 1st ed. O’Reilly Media, Inc., 2016.

[2] A. Basiri, L. Hochstein, N. Jones, and H. Tucker, “Automating chaos experiments in production,” *CoRR*, vol. abs/1905.04648, 2019. [Online]. Available: http://arxiv.org/abs/1905.04648

[3] “Principles of chaos engineering,” https://principlesofchaos.org/, last time accessed: 2021-11-16.

[4] M. Pawlikowski, *Chaos Engineering: Site reliability through controlled disruption*. Manning, 2021.

[5] D. Díaz-López, M. Blanco Uribe, C. Santiago Cely, D. Tarquino Murgueitio, E. Garcia Garcia, P. Nespoli, and F. Gómez Mármol, “Developing secure iot services: A security-oriented review of iot platforms,” *Symmetry*, vol. 10, no. 12, 2018. [Online]. Available: https://www.mdpi.com/2073-8994/10/12/669

[7] https://github.com/SaraPalaciosCh/ChaosXploit

[6] D. Díaz-López, G. Dólera Tormo, F. Gómez Mármol, J. M. Alcaraz Calero, and G. Martínez Pérez, "Live digital, remember digital: State of the art and research challenges," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 109–120, 2014, 40th-year commemorative issue. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790613002905
[7] K. A. Torkura, M. I. Sukmana, F. Cheng, and C. Meinel, "CloudStrike: Chaos Engineering for Security and Resiliency in Cloud Infrastructure," *IEEE Access*, vol. 8, pp. 123 044–123 060, 2020.
[8] A. Basiri, N. Behnam, R. de Rooij, L. Hochstein, L. Kosewski, J. Reynolds, and C. Rosenthal, "Chaos engineering," *IEEE Software*, vol. 33, no. 3, pp. 35–41, 2016.
[9] C. Camacho, P. C. Cañizares, L. Llana, and A. Núñez, "Chaos as a Software Product Line—A platform for improving open hybrid-cloud systems resiliency," *Software - Practice and Experience*, no. April 2021, pp. 1–34, 2022.
[10] J. Simonsson, L. Zhang, B. Morin, B. Baudry, and M. Monperrus, "Observability and chaos engineering on system calls for containerized applications in Docker," *Future Generation Computer Systems*, vol. 122, pp. 117–129, 2021. [Online]. Available: https://doi.org/10.1016/j.future.2021.04.001
[11] L. Zhang, B. Morin, P. Haller, B. Baudry, and M. Monperrus, "A Chaos Engineering System for Live Analysis and Falsification of Exception-Handling in the JVM," *IEEE Transactions on Software Engineering*, vol. 47, no. 11, pp. 2534–2548, 2021.
[12] A. Rinehart and K. Shortridge, "Security Chaos Engineering Gaining Confidence in Resilience and Safety at Speed and Scale," Tech. Rep., 2021.
[13] K. A. Torkura, M. I. Sukmana, F. Cheng, and C. Meinel, "Security chaos engineering for cloud services: Work in progress," in *2019 IEEE 18th International Symposium on Network Computing and Applications, NCA 2019*. Institute of Electrical and Electronics Engineers Inc., sep 2019.
[14] K. A. Torkura, M. Sukmana, F. Cheng, and C. Meinel, "Continuous auditing and threat detection in multi-cloud infrastructure," *Computers and Security*, vol. 102, p. 102124, 2021. [Online]. Available: https://doi.org/10.1016/j.cose.2020.102124
[15] S. Sharieh and A. Ferworn, "Securing apis and chaos engineering," in *2021 IEEE Conference on Communications and Network Security (CNS)*, 2021, pp. 290–294.
[16] Rapid7, "2021 cloud misconfiguration report," 2021.
[17] I. Martínez Martínez, A. Florián Quitián, D. Díaz-López, P. Nespoli, and F. Gómez Mármol, "Malseirs: Forecasting malware spread based on compartmental models in epidemiology," *Complexity*, vol. 2021, 2021.
[18] P. Nespoli, D. Díaz-López, and F. Gómez Mármol, "Cyberprotection in iot environments: A dynamic rule-based solution to defend smart devices," *Journal of Information Security and Applications*, vol. 60, p. 102878, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212621001058

Simulador de APTs realistas basado en el marco de MITRE ATT&CK

Xavier Larriva-Novo, Víctor A. Villagra, Oscar Jover, Mario Sanz Rodrigo, Carmen Sánchez-Zas, Manuel Álvarez-Campana
Dpto. Ingeniería Sistemas Telemáticos, ETSI Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, España
Av. Complutense 30
xavier.larriva.novo@upm.es, victor.villagra@upm.es, oscar.jwalsh@alumnos.upm.es, mario.sanz@upm.es, carmen.szas@upm.es, manuel.alvarez-campana@upm.es

Resumen- **Las Amenazas Persistentes Avanzadas (APTs) son hoy en día el tipo de amenaza más sofisticadas y difíciles de abordar para los sistemas y las redes actuales. Mediante el uso de tácticas, técnicas y procedimientos (TTP) altamente sofisticados divididos en múltiples etapas, los atacantes consiguen controlar de forma remota las máquinas infectadas y extraer información confidencial de organizaciones y gobiernos.**

Es por ello que los entornos de formación y entrenamiento en ciberseguridad deben contar con herramientas de apoyo que permitan conocer y practicar con las distintas amenazas que existen actualmente y que están continuamente avanzado. Este articulo propone el modelado formal de APTs aleatorios tomando en cuenta la matriz de MITRE ATT&CK y el marco de referencia STIX2.1 como lenguaje de compartición de información de amenazas. Finalmente, este articulo propone la simulación realista de APTs aleatorios para su desarrollo en ciber ejercicios y la formación de expertos en ciberseguridad.

***Index Terms*- Amenaza Persistente Avanzada (APT), STIX, ciberseguridad, MITRE ATT&CK**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Los ataques informáticos son una amenaza generalizada para cualquier sistema informático. Los ataques evolucionan al mismo tiempo que los sistemas. Es por ello por lo que los entornos de formación y entrenamiento en ciberseguridad deben contar con sistemas que permitan conocer y practicar sobre las distintas tipologías de amenazas que existen actualmente y que están continuamente avanzando. Las amenazas más sofisticadas se integran en lo que se denominan "Amenazas Avanzadas Persistentes", del inglés Advanced Persistent Threats (APTs) [1].

Aunque cada APT se adapta a su objetivo y, por tanto, es potencialmente único, su evolución suele seguir un determinado patrón. Un APT normalmente comienza con un paso de reconocimiento inicial, seguido de un compromiso inicial; una vez que se establece un punto de apoyo, el atacante intentará elevar sus privilegios y también avanzar hacia su objetivo, lo que implica más reconocimiento interno y compromisos; el paso final, la finalización de la misión, suele ser la exfiltración de datos.

El área denominada "Inteligencia de Amenazas", en inglés Threat Intelligence, consiste en modelar cualquier información que permita identificar, evaluar, monitorizar y responder frente a una posible amenaza.

Existen diversos formatos de detección y representación de amenazas a partir de diversas fuentes de información. Entre los diversos modelos y formatos de representación de amenazas se encuentran diversos modelos como MITRE Cyber Analytics Repository [2], Cyber Kill Chain (CKC)[3], Unified Cyber Kill Chain[4] y formatos y lenguajes como Structured Threat Information Expression (STIX) [5] o Kusto Query Language (KQL) [6].

El modelo de MITRE se ha utilizado para dividir un ataque complejo en etapas consecutivas para ayudar a los analistas a estudiar, concentrarse y resolver los ataques etapa por etapa, permitiendo desarrollar estrategias de mitigación para cada una de las etapas.

En base a esto, el objetivo del sistema propuesto consiste en el desarrollo de un simulador de Amenazas Persistente Avanzadas (APTs), orientada a la ejecución de cibermaniobras adaptativas y personalizables basados en modelo de MITRE ATT&CK.

II. Modelo de amenazas

En el panorama de las amenazas los eventos ocurren a una mayor velocidad todos los días; las grandes cantidades de datos involucrados en la inteligencia de amenazas cibernéticas y el intercambio de información de amenazas hacen necesaria la automatización para ayudar al análisis humano a ejecutar acciones defensivas a velocidad de una máquina. La combinación de todos estos factores requiere representaciones de información de amenazas estructuradas y estandarizadas.

A. STIX

STIX (*Structured Threat Information Expression*) es un lenguaje y un formato de serialización. STIX es un formato XML que tiene como objetivo ampliar el intercambio de indicadores para permitir la gestión y el intercambio generalizado de amenazas cibernéticas tomando en cuenta el espectro completo de su comportamiento.

B. Unified Cyber Kill Chain

El *Unified Kill Chain* (UKC) proporciona información sobre la disposición ordenada de las fases de los ciberataques de un extremo a otro y cubre diversos vectores de ataque, uniendo y ampliando los modelos existentes.

C. Repositorio de análisis cibernético de MITRE

El CAR de MITRE es una base de conocimientos de análisis desarrollada por MITRE basada en el modelo de adversario MITRE ATT&CK. CAR también contiene un modelo de datos para los datos observables que se utilizan para ejecutar las analíticas y los sensores que se utilizan para recopilar esos datos. CAR contiene análisis asignados a técnicas específicas de ATT&CK y describe la hipótesis a alto nivel, la implementación de pseudocódigo, las pruebas unitarias y el modelo de datos utilizado para desarrollarlos, de modo que los análisis se puedan transcribir a varias plataformas. CAR está destinado a ser utilizado por defensores cibernéticos en toda la comunidad y sirve como un mecanismo para compartir análisis basados en el comportamiento que se pueden utilizar para la detección de adversarios. Los análisis de CAR se desarrollaron para detectar los comportamientos del adversario en ATT&CK.

D. Matriz Mitre ATT&CK

La matriz MITRE ATT&CK cuenta con una variedad de técnicas utilizadas en diversas etapas por los adversarios para lograr un objetivo específico. Esos objetivos se clasifican como tácticas en la Matriz ATT&CK. Los objetivos se presentan linealmente desde la etapa inicial de reconocimiento hasta el objetivo final de exfiltración e/o impacto. Dentro de la matriz ATT&CK, se clasifican las siguientes etapas de ataque:

- Reconocimiento: recopilación de información sobre la organización objetivo.
- Desarrollo de recursos: establecimiento de una infraestructura de comando y control.
- Acceso inicial: intento de ingreso en la red de la víctima.
- Ejecución: intento de ejecución de código malicioso.
- Persistencia: cambio en las configuraciones del sistema de la víctima para tratar de mantener su punto de apoyo.
- Escalada de privilegios: aprovechamiento de vulnerabilidades para obtener permisos de nivel superior.
- Evasión: uso de procesos confiables para ocultar malware.
- Acceso a credenciales: robo de nombres y contraseñas de cuentas mediante técnicas como el *Keylogger*.
- Descubrimiento: intento de descubrir el entorno explorando los elementos que pueden ser controlados.
- Movimiento lateral: uso de credenciales legítimas para pivotar a través de los múltiples sistemas de la organización víctima.
- Recopilación: recogida de datos de interés para el objetivo del adversario.
- Comando y control: comunicación con los sistemas comprometidos para su control.
- Exfiltración: robo de datos.
- Impacto: manipulación, interrupción o destrucción de sistemas y datos.

III. Diseño del Sistema de Simulación

En el siguiente apartado se expondrá los diversos componentes del modelado de APTs basado en la matriz de MITRE ATT&CK presentado en este artículo. Los componentes se dividen en: fuentes de información, modelo formal basado en STIX y los elementos de salida del modelo como se presentan en la Fig 1.

A. Fuentes de Información de APT: Marco MITRE ATT&CK

El marco de MITRE ATT&CK (Adversarial Tactics, Techniques and Common Knowledge) es una base de conocimientos a nivel global, pública y de constante evolución de tácticas, técnicas y procedimientos utilizados por los atacantes. Refleja las distintas fases del ciclo de vida de los ataques llevados a cabo por un atacante y las plataformas objetivo. Se utiliza como base para el desarrollo de modelos y metodologías de amenaza específicas en el sector privado, en el gobierno y en la comunidad de productos y servicios de ciberseguridad

Acceso a la fuente de información

Para poder acceder a esta fuente de información, se cuenta con un modelo de datos implementado en el lenguaje de programación SQLite. La base de datos cuenta con un registro de los objetos de los que MITRE ATT&CK tiene conocimiento incluidos: patrones de ataque, malware, herramientas, indicadores, campañas, etc. Será desde aquí en el cual, mediante consultas a la base de datos, se permitirá la extracción de información para generar los componentes de nuestro modelo de APT.

B. Modelado de APT

Definición del modelo formal basado en STIX

Para el modelo formal se toma en consideración el modelo Structured Threat Information Expression. STIX es un lenguaje y formato de serialización adoptado como estándar internacional para el intercambio de inteligencia sobre amenazas cibernéticas (CTI)

C. Modelo de APT

El programa de simulación de APTs se encarga de describir el modelo formal de APTs indicando los parámetros y etapas que caracterizan el APT a utilizar en el ejercicio. Estos parámetros permiten definir el número de patrones de ataque que se extraen de la base de datos MITRE ATT&CK. Adicionalmente, el generador de simulación lee el modelo

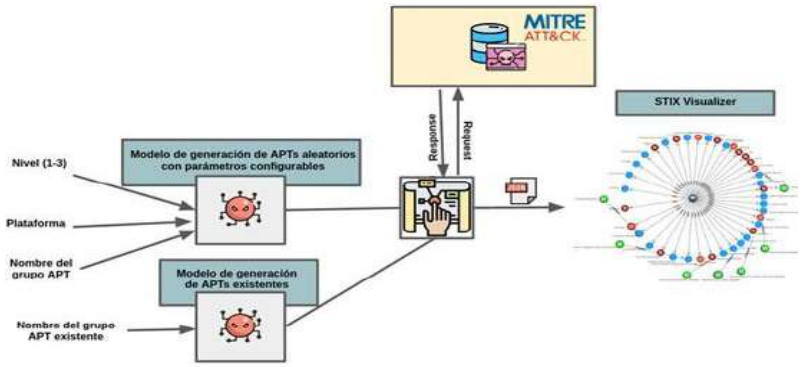


Fig. 1. Diseño del sistema de simulación

previamente creado para describir el despliegue de la red que representa el escenario real en el cual se basa el ejercicio. Esto genera la secuencia de APT indicando los diferentes pasos que componen la ejecución de la APT sobre el escenario concreto.

Generación de un APT Existente

El modelo de un APT existente, consiste en obtener todos los objetos STIX vinculados a dicho grupo APT. En este caso, el modelado es diferente al modelado de los otros dos casos anteriores, puesto que no se genera ningún valor aleatorio, sino que se extraen de la BBDD ATT&CK todos los objetos STIX relacionados y se insertan en un archivo JSON para ser posteriormente visualizados. A continuación, se describen los diferentes pasos que conforman la definición del modelo y que generan, como resultado, una amenaza persistente avanzada real:

- *Configuración de los parámetros iniciales.*

Elección del grupo APT existente: Este parámetro permite seleccionar el grupo APT del que quiere generar el modelo de todas las opciones del grupo ATT&CK.

Generación del grupo APT existente.

Los grupos APT están dentro de un objeto STIX llamado conjunto de intrusiones o intrusion set. Una vez se ha encontrado el conjunto de intrusiones que corresponde a ese grupo APT, se buscan todos los objetos STIX que tengan como referencia externa el ID de grupo asignado a ese grupo APT.

El resultado de la generación de un APT para cada uno de estos modelos es un archivo JSON con todos los identificadores SDOs y SROs modelados y añadidos a un 'bundle', que permite agrupar los objetos resultantes en un mismo conjunto. Un 'bundle' se utiliza como contenedor para una colección de objetos STIX. Tiene un identificador único asociado (UUID) y en la parte de objetos, vienen incluidos todos los objetos STIX generados.

Generación de un APT Aleatorio personalizable

La parametrización de APT aleatorios viene dado por diversos parámetros configurables a nivel de usuario. Esta opción permite la posibilidad de configurar y parametrizar el APT generado con el objetivo de adaptarlo a diferentes necesidades. A continuación, se describen los diferentes parámetros que conforman la definición del modelo y que generan, como resultado, una amenaza persistente avanzada única y aleatoria pero adaptada a los parámetros introducidos por el usuario:

- *Configuración de la plataforma objetivo:*

Este parámetro permite la opción de configurar la plataforma objetivo a la que está dirigida el APT aleatorio generado. Se define de un listado de las plataformas disponibles actualmente. En caso de no configurar este parámetro, se elegirá una plataforma aleatoria de entre todas las plataformas disponibles.

- *Elección del nombre del grupo APT aleatorio.*

Este parámetro permite elegir un nombre para el grupo APT aleatorio que va a ser generado. Con este nombre, se modela el objeto STIX de actor de amenaza. En caso de no configurar este parámetro, se elegirá aleatoriamente mediante la combinación de un adjetivo y el nombre de un animal en inglés.

- *Configuración de patrones de ataque específicos.*

Este parámetro permite configurar los patrones de ataque que se incluirán en el APT aleatorio. Los patrones de ataque elegidos serán los únicos que aparecerán en el APT aleatorio final generado. En el caso de elegir una plataforma objetivo, sólo se dispondrá de patrones de ataque que pertenecen a esa plataforma. En caso de no disponer de este parámetro, se elegirán los patrones de ataque aleatorios de entre todos los patrones de ataque existentes.

- *Configuración del nivel.*

Este parámetro permite identificar el nivel de dificultad del APT aleatorio que va a ser generado. Como en el caso anterior, afecta al número de patrones de ataque que van a ser incluidos por cada fase dentro del APT y por lo tanto al malware, herramientas, acciones recomendadas y vulnerabilidades. El número de patrones de ataque aleatorios que se van a elegir por cada fase se puede calcular como un número aleatorio entre el máximo número de ataques que tiene esa fase del número de ataques totales que hay en la base de datos de ATT&CK y el nivel. Actualmente, esta fórmula ha sido modificada por otra más sencilla, que genera el número aleatorio de ataques por fase basándose exclusivamente en el nivel elegido por el usuario. Por lo tanto, si el usuario ha elegido el nivel difícil (nivel 3), se elegirá por cada fase un número aleatorio entre 1 y 3 que corresponderá al número de ataques aleatorios a incluir en esa fase.

IV. Prototipo del sistema simulación

El sistema de generación de APTs permite la ejecución secuencial de un conjunto de habilidades asociadas al APT generado. Este sistema de generación de APTs se detalla a continuación den la Fig. 2.

El sistema de generación de APTs contiene dos interfaces de comunicación. La interfaz de entrada es la encarga de leer los parámetros predefinidos para la generación de modelo de simulación APT. Por otro lado, la interfaz de salida permite la intercomunicación con las maquinas o sistemas víctimas.

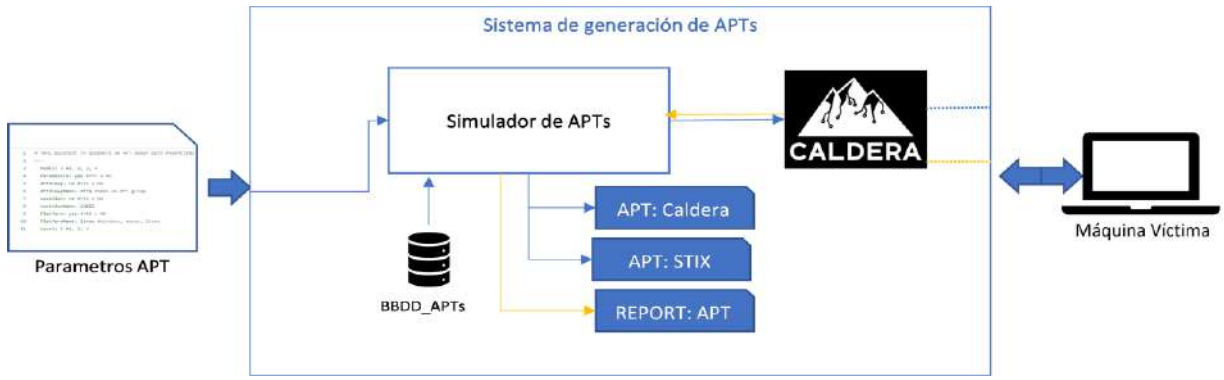


Fig. 2. Prototipo del sistema de simulación de APTs.

El sistema de generación de APTs cuenta con diversos componentes detallados a continuación:

Parámetros APT : Son aquellos parámetros que se detallan en la Sección III el cual permite permiten configurar diversas amenazas (CTI de la BBDD APTs). Estas amenazas se definen el contexto y el modelo propuesto para la generación de un APT aleatorio. En este contexto el APT aleatorio se define por un conjunto de TTP, con actores de amenazas, ubicaciones y cursos de acción recomendadas según la información de MITRE ATT&CK.
BBDD_APTs: Esta base de datos se encuentra implementada en SQLite. En el cual se encuentra el registro de datos recopilados de los objetos que MITRE ATT&CK tiene publicados como: patrones de ataque, malware, herramientas, indicadores de compromiso, cursos de acción, entre otros. En relación al contenido de las herramientas asociadas a los diversos APTs en esta base de datos se encuentra una distribución de técnicas y tácticas para la ejecución de diversos payloads relacionados a las fases de un APT.

Simulador de APTs: Es parte del sistema de generación de APTs que permite desarrollar los diversos APTs parametrizados. Este módulo permite generar el modelo secuencial con para ser instalado en una maquina víctima. Los diversos modelos generados son basados en las distintas fases de la matriz de MITRE ATT&CK. Este modelo genera los modelos específicos parametrizados **(APT: Caldera)**, con el objetivo que el sistema de orquestación Caldera pueda ejecutarlos en una maquina víctima.

Caldera: es el sistema de orquestación que permite la ejecución de diversas técnicas adversarias de manera automática sobre un determinado sistema. El objetivo de este elemento consiste en evaluar la ejecución de diversas herramientas asociadas a un APT ejecutado sobre una maquina victima [7]. Para que la maquina victima permita la ejecución del APT sobre esta, previamente deberá suscribirse de manera remota a la plataforma de simulación en el cual, posteriormente se ejecutarán de manera automática el conjunto de habilidades asociadas a dicho APT.

APT: STIX: Este es archivo de formato JSON con los objetos del dominio STIX (SDO) y relaciones STIX (SRO) que conforman el modelo. Este modelo este compuesto por un identificador único asociado (UUID) y en la parte de objetos, vienen incluidos todos los objetos STIX generados.

REPORT: APT: Este es un fichero de tipo JSON el cual indica los resultados obtenidos por la ejecución del APT. Aquí se indica la fase en la que se encuentra la herramienta que se ha ejecutado, el payload asociado, así como la respuesta del sistema ante la ejecución de dicha herramienta. Entre otros parámetros se encuentran los tiempos de ejecución que ha tardado las herramientas en obtener la información.

PCx: Las maquinas victimas permitirán la ejecución de los APTs generados. Estas máquinas ejecutarán las operaciones predeterminadas de manera secuencial. La ejecución de las diversas herramientas se ejecuta de manera consecutiva y coherente. Este resultado podrá ser satisfactorio o no en caso de que el sistema pueda obtener una respuesta ante la herramienta del APT propuesto.

V. Conclusiones

En este trabajo se define un modelo genérico que permite caracterizar APTs aleatorias y se proponen las bases de una plataforma de simulación de APTs tanto aleatorias como de grupos APTs existentes, con el objetivo de servir como plataforma de ciber ejercicio para expertos en ciberseguridad. Para lograr estos objetivos, se utiliza la matriz de MITRE ATT&CK como referencia, el lenguaje de compartición de información de amenazas STIX 2.1 para modelar las amenazas. Estos patrones tienen unas fases asociadas únicas de la cadena que utiliza MITRE ATT&CK como referencia, y con la que asigna consecuentemente el malware, las herramientas y las vulnerabilidades de esa fase.

Por otro lado, el lenguaje STIX no posee un objeto fase, sino que la fase viene como propiedad dentro de los objetos de dominio STIX (SDOs). Esto ha sido una limitación de cara a la representación visual con el visualizador de STIX, puesto que cuando se muestra el modelo del APT aleatorio, se muestran todos los objetos SDOs y sus relaciones, pero no organizados de manera secuencial en base a la fase en la que se encuentra cada objeto STIX creado. Este problema puede ser parcialmente solventado para cada modelo APT aleatorio generado, puesto que el visualizador de STIX permite personalizar la vista y mover los objetos STIX a donde se considere. Se propone también como trabajo futuro, el diseño de un visualizador que permita mostrar directamente los objetos STIX secuencialmente ordenados por las fases a las que pertenecen.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto COBRA (10032/20/0035/00) del programa COINCIDENTE 2019 trabajando en colaboración con el MCCE, DGAM/SDG. PLATIN. Este proyecto ha sido concedido por el Ministerio de Defensa, así como por las ayudas FJCI-2017-34926 y RYC2015-18210, concedidas por el Gobierno de España y con-financiadas por el Fondo Social Europeo.

Referencias

[1] P. Chen, L. Desmet, and C. Huygens, “A Study on Advanced Persistent Threats,” in *Communications and Multimedia Security*, Berlin, Heidelberg, 2014, pp. 63–72. doi: 10.1007/978-3-662-44885-4_5.

[2] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, “Mitre att&ck: Design and philosophy,” *Technical report*, 2018.

[3] “Cyber Kill Chain Analysis Section II: Studies and Analysis of Cybercrime Phenomenon 3 International Journal of Information Security and Cybercrime 2014.” https://heinonline.org/HOL/LandingPage?handle=hein.journals/ijisc3&div=21&id=&page= (accessed Apr. 19, 2022).

[4] P. Pols and J. van den Berg, “The unified kill chain,” *CSA Thesis, Hague*, pp. 1–104, 2017.

[5] S. Barnum, “Standardizing cyber threat intelligence information with the structured threat information expression (stix),” *Mitre Corporation*, vol. 11, pp. 1–22, 2012.

[6] M. Copeland, “Kusto Query Language and Threat Hunting,” in *Cloud Defense Strategies with Azure Sentinel*, Springer, 2021, pp. 185–211.

[7] P. Zilberman, R. Puzis, S. Bruskin, S. Shwarz, and Y. Elovici, “SoK: A Survey of Open-Source Threat Emulators,” *arXiv:2003.01518 [cs]*, Oct. 2020, Accessed: Apr. 19, 2022. [Online]. Available: http://arxiv.org/abs/2003.01518

Estudio de modelado de periféricos para habilitar emulaciones de firmware embebido

Xabier Gandiaga, Urko Zurutuza, e Iñaki Garitano

Departamento de Electrónica e Informática
Escuela Politécnica Superior
Mondragon Unibertsitatea
Goiru 2, E-20500 Arrasate-Mondragón
Email: {xgandiaga,uzurutuza,igaritano}@mondragon.edu

***Resumen*—Los sistemas embebidos aumentan cada vez más en número y con ello también lo hacen los ataques dirigidos a estos. Uno de los factores clave para reducir la superficie de ataque es descubrir y corregir vulnerabilidades en el firmware embebido. El análisis dinámico es uno de los métodos más empleados para estos fines. Escalar el análisis dinámico es necesario para acelerar este proceso, lo que conlleva crear emulaciones del firmware que permitan prescindir del coste de compra de hardware. Identificamos el modelado de periféricos como problema central para habilitar dichas emulaciones. Listamos las características deseables de un proceso de modelado de periféricos, los retos a tener en cuenta y los diferentes procesos que se han utilizado para resolverlos en diferentes escenarios.**

***Index Terms*—Sistemas embebidos, emulación, análisis de firmware**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

El número de sistemas embebidos presentes en la sociedad ha aumentado debido a diversos factores, entre los que se encuentran (1) el abaratamiento de los microcontroladores, (2) la implementación de las tecnologías de conectividad y (3) el que las tecnologías pertinentes se hayan vuelto más eficientes [1]. Este aumento en dispositivos ha hecho que la cantidad de ataques dirigidos a los sistemas embebidos aumente [2]. Para hacer frente a estos ataques es necesario reducir la superficie de ataque de los sistemas embebidos, siendo clave el descubrir y arreglar vulnerabilidades en el firmware. Debido a que gran parte del firmware embebido es privado [3] el análisis dinámico, analizar el firmware durante su funcionamiento, es el método principal utilizado para la búsqueda de vulnerabilidades.

El análisis dinámico permite interactuar con el firmware en ejecución, lo cual permite controlar y establecer el estado de ejecución. El análisis dinámico requiere controlar la plataforma de ejecución para lo cual se puede utilizar el hardware originario del firmware como plataforma y adaptarlo a las herramientas de análisis a utilizar. Sin embargo, para poder escalar el análisis y acelerar la búsqueda de vulnerabilidades es necesario configurar una plataforma de emulación donde ejecutar los firmware y así prescindir del coste de adquirir hardware.

Al configurar una plataforma de emulación de un firmware, el mayor de los problemas es modelar los periféricos con los que el firmware interactúa. Para poder realizar este modelado se han creado varios procesos que hacen uso de análisis estático (analizar el firmware sin ejecutarlo), análisis dinámico, instrumentación (generar trazas de la ejecución a partir de código agregado) y hardware. No obstante, los procesos desarrollados hasta ahora no consiguen hacer frente a todos los problemas que el mercado de sistemas embebidos presenta.

Entre dichos problemas destacamos los siguientes: (1) los sistemas embebidos continúan en aumento, por lo que los procesos de modelado de periféricos tienen que acelerarse mediante escalabilidad. Para ello es necesario prescindir de hardware durante el proceso de modelado y automatizar el esfuerzo manual requerido, reduciendo el nivel técnico para aplicarlos. (2) Debido a la heterogeneidad de arquitecturas y periféricos en los sistemas embebidos, es necesaria una aplicabilidad genérica. Es decir, el que un proceso de modelado de periféricos funcione entre diferentes firmware. Esto implica funcionar en diferentes arquitecturas de procesadores y diferente acceso a información sobre el firmware (código abierto o privado, p.e.). Además, (3) los modelos creados deberían de ser transferibles a emulaciones de otros firmware para así reducir el esfuerzo en emular otros dispositivos.

Este documento resume los trabajos en la literatura sobre el modelado de periféricos con el fin de habilitar emulaciones para el análisis dinámico de firmware embebido.

II. Periféricos de sistemas embebidos

Los sistemas embebidos utilizan principalmente dos métodos para incorporar periféricos; (1) integrarlos en el microcontrolador y (2) conectarlos a través de entradas/salidas digitales o algún bus de comunicación. Esto separa los periféricos en periféricos *on-chip* (módulos de memoria, temporizadores, controladores de interrupciones, interfaces/buses de comunicación, etc.) y periféricos *off-chip* (cámaras, sensores, botones, etc.).

Al emular firmware de sistemas embebidos, algunos periféricos *on-chip* ya están implementados en plataformas de emulación. Sin embargo, otros periféricos *on-chip* y casi todos los periféricos *off-chip* carecen de estas implementaciones.

III. Modelado de periféricos

Modelar un periférico consiste en interceptar las interacciones del firmware con el periférico y proporcionar respuestas válidas al firmware para que la ejecución continúe. Un modelo completo es capaz de proporcionar el firmware con

valores correctos para cualquier llamada al periférico original y puede dirigir el firmware por todas sus líneas de ejecución (una cobertura completa). También es posible sustituir la implementación del periférico por el de otro similar, aunque esto puede causar que la ejecución difiera del original. Los procesos que realizan el modelado se pueden diferenciar en dos grupos [4]:

- **Modelado a alto nivel de abstracción:** El modelado de periféricos ocurre a nivel de código del firmware. Las partes del código del firmware que interactúan con los periféricos son reemplazados por código que evita la interacción y proporciona respuestas al firmware.
- **Modelado a bajo nivel de abstracción:** El firmware interactúa con periféricos a través de escrituras y lecturas a memoria. En este tipo de modelado, se intercepta el firmware mediante código agregado a la plataforma de emulación cuando este intenta leer una respuesta de un periférico en la memoria. Después ese código escribe las respuestas correspondientes en los registros de la memoria que el firmware va a leer.

IV. Técnicas para modelar periféricos

El modelado de periféricos se realiza aplicando diferentes técnicas de análisis estático, análisis dinámico e instrumentación sobre el firmware y las plataformas de ejecución en uso para el firmware (emulaciones, hardware o ambos a la vez).

Las técnicas principales aplicadas al firmware para modelar periféricos son las siguientes:

- **Instrumentación:** Instrumentar significa agregar código o programas que generen trazas sobre la ejecución. Estas trazas se pueden utilizar para crear modelos o como base para los *inputs* de otras técnicas. Esta instrumentación puede ocurrir a nivel de binario (instrumentación de binario mediante angr y similares), a nivel de plataforma de emulación (PANDA [5], p.e.) o a nivel de hardware.
- **Fuzzing:** El fuzzing de firmware es un método de fuerza bruta que crea *inputs* a insertar al firmware. Los fuzzer generan valores en base a normas que delimitan cómo mutar unos *input* base. Estos valores base se formulan a través de la documentación del firmware o *logs* de ejecución del firmware. El fuzzer más utilizado para modelar periféricos es el American Fuzzy Loop (AFL) [6].
- **Ejecución simbólica:** La ejecución simbólica es un método de análisis dinámico de caja blanca o *whitebox*. Un ejecutor simbólico analiza rutas de ejecución de un código seleccionadas en base a un algoritmo exploratorio. Las rutas de ejecución se representan como un árbol binario que crea dos nuevas ramas, *true* y *false*, por cada condición encontrada en el código. Los valores que guían la ejecución por las ramas se guardan como valores simbólicos. Estos valores simbólicos se pueden resolver para obtener un valor concreto que guíe la ejecución hasta un estado específico del código. Las herramientas de ejecución simbólica más utilizadas para modelar periféricos son angr [7] y S2E [8].
- **Ejecución concólica:** La ejecución concólica es una mezcla de una ejecución simbólica y una ejecución de valores concretos. Se utiliza para evitar problemas que la ejecución simbólica puede llegar a tener en estados complejos del código (explosión de rutas, tiempos largos para resolver valores simbólicos, etc.). Para modelar periféricos, la ejecución concólica se basa en una plataforma de ejecución combinada con una herramienta de ejecución simbólica (QEMU [9] + angr o S2E).

V. Retos del modelado de periféricos

Los retos del modelado de periféricos listados los extraemos de los problemas a los que los procesos de modelado, presentados más adelante en el documento, hacen frente:

- **Uso de hardware:** En caso de ser necesario el uso de hardware real durante el proceso de generación del modelo, el escalado horizontal del proceso estará limitado por la cantidad de dispositivos reales disponibles.
- **Necesidad de trabajo manual:** A cuanto más trabajo manual necesite el proceso de modelado menos escalable es el proceso porque no se podrán automatizar esas partes. Una menor escalabilidad resulta en procesos más lentos. Además, el trabajo manual requiere de un conocimiento técnico mayor para aplicarse, lo cual reduce la usabilidad de la proceso en la industria.
- **Conexiones externas y control del hardware:** En caso de utilizar hardware durante el proceso para modelar los periféricos, los procesos aplicables están limitados a las conexiones y control disponibles del hardware (JTAG, serial, conexiones TCP, *stubs* de depuración insertables en el hardware, etc.).
- **Disponibilidad de herramientas de análisis:** Las herramientas de análisis de binarios más populares no soportan todas las arquitecturas disponibles, lo cual limita los procesos que los utilicen. Como ejemplo, S2E tuvo que ser re-adaptado a ARM en [10].
- **Firmware de código abierto o privativo:** En un proceso de modelado de periféricos a bajo nivel, si los firmware con los que trabajar son de código cerrado, es imposible utilizar técnicas de caja blanca de análisis dinámico sobre ellos (ejecución simbólica y ejecución concólica) para modelar periféricos. Trabajando en modelado de alto nivel, localizar el código que interactúa con los periféricos en el binario puede ser inviable por el mismo motivo.
- **Características concretas en el código:** Los procesos de modelado de periféricos a alto nivel requieren que el código utilice capas de abstracción para el control de periféricos. Estas capas de abstracción son *Hardware Abstraction Layers* (HAL) en firmware *baremetal* [11] y *Board Support Packages* (BSP) y *drivers* en firmware con sistemas operativos [12].
- **Acceso directo a memoria (DMA):** Mediante un controlador de DMA un periférico puede escribir en memoria sin interactuar con el procesador. Esto significa que durante la emulación, en aquellos casos en los cuales el periférico a modelar haga uso de DMA, el firmware puede quedarse a la espera de esta interacción. Si el modelo no lo contempla esto no ocurre y la ejecución podría verse afectada. Si se trabaja a bajo nivel, es necesario identificar patrones de acceso por DMA en el firmware para saber cuando una interacción por DMA

puede ocurrir. A alto nivel, es necesario localizar las funciones que interactúan con el controlador DMA y reemplazarlos.

- **Controladores de interrupciones:** Las interrupciones son señales lanzadas por el hardware que avisan de eventos asíncronos al firmware. Los controladores de interrupciones mantienen un listado de qué hardware puede lanzar qué interrupciones y la prioridad de estos. Una interacción con un periférico puede quedarse estancada si las interrupciones correctas no son lanzadas. Es necesario modelar qué interrupciones habilitar en cada momento e ir lanzándolos ordenadamente en modelos de bajo nivel.
- **Transferibilidad de los modelos:** Muchos procesos de modelado de periféricos modelan la interacción de periféricos *off-chip* con el procesador mediante periféricos *on-chip* que actúan como interfaces para el procesador [13]. Esto hace que los modelos creados sean modelos que combinan dos periféricos (interfaz *on-chip* y periférico *off-chip*), reduciendo su transferibilidad a firmware que utilice esa misma combinación de periféricos.
- **Cobertura del firmware:** Al utilizar un modelo para evitar una interacción con un periférico es posible que ese modelo no sea completo. Dicho modelo evitaría que un análisis sobre la emulación final obtuviese un cobertura total del firmware, pudiendo ocultar vulnerabilidades. Esto ocurre tanto trabajando en modelado a alto nivel como a bajo nivel [14]. A cuanto mayor la cobertura mayor es la fidelidad del modelo al periférico original.

Agrupándolos, destacamos los siguientes retos: (1) la escalabilidad del proceso de generación del modelo, afectado por el uso de hardware y trabajo manual, (2) la versatilidad de los procesos de modelado, afectada por la heterogeneidad del hardware, características del firmware y herramientas utilizables. Por último, (3) la calidad del modelo resultante en términos de transferibilidad del modelo y cobertura del firmware.

VI. Clasificación de procesos que habilitan emulaciones

El nivel de las emulaciones de firmware se puede clasificar en base a la cobertura del firmware que ofrecen los modelos de periféricos [15]. A su vez, los procesos para crear estos modelos se pueden clasificar en base al nivel de fidelidad que buscan para el modelo de un periférico y los retos del modelado de periféricos a los que responden. En base a esos factores los procesos de modelado se pueden separar en los siguientes grupos:

VI-A. Emulación hardware-in-the-loop

La emulación *hardware-in-the-loop* utiliza hardware para habilitar la ejecución de las emulaciones. Estos procesos no intentan modelar periféricos, sino que redirigen las llamadas a periféricos que realiza el firmware en la plataforma de emulación al periférico real. Sin embargo, están limitados en escalabilidad y aplicabilidad por estar atados a hardware. Los trabajos más relevantes son Avatar, de Zaddach, Bruno, et al. [16] el cual se utiliza como base de múltiples otros estudios ([17],[18]), Prospect, de Kammerstetter et al. [19] y Surrogates, de Koscher et al. [20]

VI-B. Modelado parcial

El modelado parcial no se preocupa por la fidelidad de la emulación final. Estos procesos (1) sustituyen periféricos no emulados por otros ya emulados y (2) responden a llamadas a periféricos con valores concretos que permitan continuar la ejecución. Estos valores se infieren de archivos de configuración en los sistemas de archivos del firmware a emular o se utilizan algunos definidos por defecto. Firmadyne, de Chen et al. [21] y FirmAE, de Kim et al. [22] sustituyen el kernel original del firmware por otro propio (reduciendo aún más la fidelidad de la emulación final) y automatizan los pasos (1) y (2), disminuyendo el nivel técnico requerido. En Ecmo, de Jiang et al. [23], se sustituyen los periféricos de forma manual con código insertado a la plataforma de emulación.

VI-C. Modelado mediante hardware

El modelado mediante hardware intenta crear modelos de periféricos de alta fidelidad que prescinden del hardware para su uso en base a información obtenida instrumentando hardware o una emulación *hardware-in-the-loop*. Los modelos creados mediante estos procesos son modelos a bajo nivel. Pretender, de Gustafson et al. [17] utiliza instrumentación sobre Avatar para obtener datos sobre la interacción de los periféricos y QEMU. Después utiliza esta información para crear modelos de periféricos basados en inteligencia artificial. Conware, de Spensky et al. [13] instrumenta dispositivos físicos que corren firmware de código abierto para obtener *logs* en los cuales basar sus modelos. Después examina el código del firmware para dividir estos modelos en dos. Un modelo para la interfaz *on-chip* y otro para el periférico *off-chip*. Estos modelos individuales son más transferibles.

VI-D. Modelado mediante trabajo manual

Los procesos en esta categoría trabajan en el modelado de periféricos de alto nivel. El trabajo que realizan se puede dividir en dos pasos principales: (1) el localizar las capas de abstracción de uso de hardware, HAL, en el código y (2) generar el código del modelo para reemplazarlos. Este código de reemplazo intenta ser lo más fiel posible al periférico original. El localizar la capa de abstracción puede ser automatizado, pero, por ahora, el código para el modelado ha sido creado manualmente en todos los trabajos. HALucinator, de Clements et al/ [11], trabaja con HALs en firmware *baremetal*. Para-rehosting, de Li et al. [24], combina el modelado a alto nivel con transferir la lógica del firmware a un programa de *userspace* de x86. Firmporter, de Xin et al. [12] trabaja con BSPs y *drivers* en firmware para RTOS de código abierto. Clements et al. [15] trabajan con firmware de RTOS de código cerrado.

VI-E. Modelado automático

En el modelado automático se crean modelos de alta fidelidad a bajo nivel mediante procesos automatizados y escalables. Las técnicas utilizadas en esta categoría son el *fuzzing*, ejecuciones simbólicas y ejecuciones concólicas. P2IM, de Feng et al. [25] crea modelos de periféricos en firmware de código cerrado mediante la clasificación de registros de memoria y un fuzzer. Mediante el fuzzer se prueban diferentes respuestas para el firmware que se escriben en direcciones de memoria de destino clasificadas manualmente. Aunque el

clasificar las direcciones de memoria es un trabajo manual, el modelado (encontrar las respuestas correctas para el firmware) es automático. DICE, también por Feng et al. [26] agrega la posibilidad de modelar periféricos que utilizan DMA en P2IM. Jetset de Johnson et al. [27], Laelaps de Cao et al. [18] y µEmu de Zhou et al. [10] utilizan ejecuciones simbólicas (Jetset) y ejecuciones concólicas (Laelaps, µEmu) para modelar los periféricos. Sin embargo, al utilizar técnicas de caja blanca, están limitados a firmware abierto. Fuzzware de Scharnowski et al. [14] utiliza una combinación de un fuzzer y ejecuciones simbólicas. Las ejecuciones simbólicas limitan los valores y las mutaciones que el fuzzer debe probar. Esto permite aumentar la cobertura del firmware en menos tiempo de ejecución.

VII. Conclusiones y líneas futuras

La emulación de firmware de sistemas embebidos presenta múltiples retos debido principalmente a la heterogeneidad en cuanto a arquitecturas y periféricos existente. Uno de los mayores retos se centra en la emulación de los periféricos, lo cual requiere la creación de un modelo o gemelo digital lo más realista posible. En este trabajo se han recogido los distintos procesos existentes para la creación de modelos de periféricos, clasificándolos en dos grupos en base a la capa de abstracción sobre la cual trabajan, a nivel de código del firmware y a nivel de plataforma de emulación. Así mismo, se han descrito los principales retos para el modelado de periféricos y los trabajos más relevantes clasificados en base a los objetivos, retos y la fidelidad de la emulación resultante.

Las líneas futuras de este ámbito incluyen: (1) automatizar las partes manuales de los procesos citados para reducir el nivel técnico que requieren en su uso, (2) extender los procesos a arquitecturas no soportadas, adaptando las herramientas de análisis usadas y (3) trabajar la transferibilidad de modelos.

Agradecimientos

Este trabajo ha sido desarrollado por el grupo de sistemas inteligentes para sistemas industriales apoyado por el Departamento de Educación, Política Lingüística y Cultura del Gobierno Vasco (IT1676-22). Ha sido parcialmente financiado por el proyecto REMEDY del Departamento de Desarrollo Económico e Infraestructuras bajo el acuerdo de subvención KK-2021/00091. Así mismo, ha sido parcialmente financiado por el proyecto VARIOT, TENtec n. 28263632 del programa Connecting Europe Facility de la Unión Europea.

Referencias

[1] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of things," *International journal of communication systems*, vol. 25, no. 9, p. 1101, 2012.

[2] P.-A. Vervier and Y. Shen, "Before toasters rise up: A view into the emerging iot threat landscape," in *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 2018, pp. 556–576.

[3] C. Maxfield. (2019) Embedded markets study - integrating iot and advanced technology designs, application development processing environments.

[4] A. Fasano, T. Ballo, M. Muench, T. Leek, A. Bulekov, B. Dolan-Gavitt, M. Egele, A. Francillon, L. Lu, N. Gregory, D. Balzarotti, and W. Robertson, "SoK: Enabling Security Analyses of Embedded Systems via Rehosting," *ASIA CCS 2021 - Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, no. Ii, pp. 687–701, 2021.

[5] B. Dolan-Gavitt, J. Hodosh, P. Hulin, T. Leek, and R. Whelan, "Repeatable reverse engineering with panda," in *Proceedings of the 5th Program Protection and Reverse Engineering Workshop*, 2015, pp. 1–11.

[6] M. Zalewski. (2021). [Online]. Available: https://github.com/google/AFL

[7] S. a. A. S. U. . S. Computer Security Lab at UC Santa Barbara. (2021). [Online]. Available: https://github.com/angr/angr

[8] V. Chipounov, V. Kuznetsov, and G. Candea, "S2e: A platform for in-vivo multi-path analysis of software systems," *Acm Sigplan Notices*, vol. 46, no. 3, pp. 265–278, 2011.

[9] F. Bellard, "Qemu, a fast and portable dynamic translator." in *USENIX annual technical conference, FREENIX Track*, vol. 41, no. 46. California, USA, 2005, pp. 10–5555.

[10] W. Zhou, L. Guan, P. Liu, and Y. Zhang, "Automatic firmware emulation through invalidity-guided knowledge inference," in *USENIX Security Symposium*, 2021, pp. 2007–2024.

[11] A. A. Clements, E. Gustafson, T. Scharnowski, P. Grosen, D. Fritz, C. Kruegel, G. Vigna, S. Bagchi, and M. Payer, "{HALucinator}: Firmware re-hosting through abstraction layer emulation," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 1201–1218.

[12] M. Xin, H. Wen, L. Deng, H. Li, Q. Li, and L. Sun, "Firmware re-hosting through static binary-level porting," *arXiv preprint arXiv:2107.09856*, 2021.

[13] C. Spensky, A. Machiry, N. Redini, C. Unger, G. Foster, E. Blasband, H. Okhravi, C. Kruegel, and G. Vigna, "Conware: Automated modeling of hardware peripherals," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2021, pp. 95–109.

[14] "Fuzzware: Using precise MMIO modeling for effective firmware fuzzing," in *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, 2022.

[15] A. A. Clements, L. Carpenter, W. A. Moeglein, and C. Wright, "Is your firmware real or re-hosted?" in *Workshop on Binary Analysis Research (BAR)*, vol. 2021, 2021, p. 21.

[16] J. Zaddach, L. Bruno, A. Francillon, D. Balzarotti *et al.*, "Avatar: A framework to support dynamic security analysis of embedded systems' firmwares." in *NDSS*, vol. 14, 2014, pp. 1–16.

[17] E. Gustafson, M. Muench, C. Spensky, N. Redini, A. Machiry, Y. Fratantonio, D. Balzarotti, A. Francillon, Y. R. Choe, C. Kruegel *et al.*, "Toward the analysis of embedded firmware through automated re-hosting," in *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*, 2019, pp. 135–150.

[18] C. Cao, L. Guan, J. Ming, and P. Liu, "Device-agnostic firmware execution is possible: A concolic execution approach for peripheral emulation," in *Annual Computer Security Applications Conference*, 2020, pp. 746–759.

[19] M. Kammerstetter, C. Platzer, and W. Kastner, "Prospect: peripheral proxying supported embedded code testing," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, 2014, pp. 329–340.

[20] K. Koscher, T. Kohno, and D. Molnar, "{SURROGATES}: Enabling {Near-Real-Time} dynamic analyses of embedded systems," in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, 2015.

[21] D. D. Chen, M. Woo, D. Brumley, and M. Egele, "Towards automated dynamic analysis for linux-based embedded firmware." in *NDSS*, vol. 1, 2016, pp. 1–1.

[22] M. Kim, D. Kim, E. Kim, S. Kim, Y. Jang, and Y. Kim, "Firmae: Towards large-scale emulation of iot firmware for dynamic analysis." New York, NY, USA: Association for Computing Machinery, 2020.

[23] M. Jiang, L. Ma, Y. Zhou, Q. Liu, C. Zhang, Z. Wang, X. Luo, L. Wu, and K. Ren, "Ecmo: Peripheral transplantation to rehost embedded linux kernels," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 734–748.

[24] W. Li, L. Guan, J. Lin, J. Shi, and F. Li, "From library portability to para-rehosting: Natively executing microcontroller software on commodity hardware," *arXiv preprint arXiv:2107.12867*, 2021.

[25] B. Feng, A. Mera, and L. Lu, "P2IM: Scalable and hardware-independent firmware testing via automatic peripheral interface modeling," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 1237–1254.

[26] A. Mera, B. Feng, L. Lu, and E. Kirda, "Dice: Automatic emulation of dma input channels for dynamic firmware analysis," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 1938–1954.

[27] E. Johnson, M. Bland, Y. Zhu, J. Mason, S. Checkoway, S. Savage, and K. Levchenko, "Jetset: Targeted firmware rehosting for embedded systems," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 321–338.

Sesión V: Seguridad IoT, IIoT, ICS

Hacia la creación de reglas CEP no supervisadas para la detección en tiempo real de ataques en entornos IoT

José Roldán-Gómez
Universidad de Castilla-La Mancha
Campus Universitario s/n, Albacete, 02006, Spain
jose.roldan@uclm.es

Jesús Martínez del Rincón
Queen's University Belfast
Belfast, BT3 9DT, UK
j.martinez-del-rincon@qub.ac.uk

Juan Boubeta-Puig
University of Cadiz
Avda. de la Universidad de Cadiz 10,
Puerto Real, 11519, Spain
juan.boubeta@uca.es

José Luis Martínez
Universidad de Castilla-La Mancha
Campus Universitario s/n, Albacete, 02006, Spain
JoseLuis.Martinez@uclm.es

***Resumen*—En los últimos años, el Internet de las cosas (IoT) ha crecido rápidamente. Sin embargo, los ataques contra el mismo también lo han hecho. Ciertas limitaciones del paradigma provocan que sea necesario estudiar nuevos métodos para detectar ataques en tiempo real debido a la dificultad de adaptar técnicas empleadas en otros paradigmas.**

En este trabajo, proponemos una arquitectura capaz de generar reglas de procesamiento de eventos complejos (CEP) para la detección de ataques en tiempo real de forma automática y completamente no supervisada. Se usa CEP porque permite procesar una gran cantidad de datos en tiempo real y porque puede desplegarse en un entorno IoT. Junto con CEP emplearemos Análisis de componentes principales (PCA), Modelos de mezcla gaussiana (GMM) y la distancia de Mahalanobis. Con esto se consiguen generar reglas CEP de forma no supervisada.

Los resultados demuestran que las reglas funcionan muy bien detectando ataques en tiempo real.

***Index Terms*—Rules generation, Cybersecurity, Internet of Things, CEP**

Tipo de contribución: *Investigación original*

I. Introduction

El internet de las cosas (IoT) ha crecido rápidamente en la última década y no parece que este crecimiento vaya a detenerse o decelerar próximamente, debido al potencial evidente que ofrece este nuevo paradigma. Prueba de esto es que ya interactuamos con ciertas aplicaciones orientadas a este paradigma a través de dispositivos como *smartphones* o *wearables*. IoT puede resultar útil en una infinidad de contextos y aplicaciones, como por ejemplo aplicaciones sanitarias, domótica, gestión de recursos y muchas otras más [1], [2], [3].

El rápido crecimiento del IoT es positivo para el desarrollo de muchas aplicaciones, sin embargo, este crecimiento también conlleva afrontar una serie de retos en diferentes ámbitos [4]. En este trabajo nos centramos en la ciberseguridad de los sistemas IoT, concretamente en la detección de ataques en red en entornos IoT.

Es imprescindible entender que las soluciones propias de otros paradigmas no siempre pueden aplicarse directamente en entornos IoT, esto se debe fundamentalmente a las limitaciones de los dispositivos de este paradigma. Dentro de estas limitaciones destacan: escasa capacidad computacional, ancho de banda limitado, sensores de bajo coste, escasa memoria y uso de baterías. Si a ello le sumamos el incremento en el uso de este tipo de dispositivos [5], [6], esto ha provocado un aumento de los cibercriminales que deciden centrarse en este paradigma.

Para detectar ataques de red en tiempo real en entornos IoT necesitamos cumplir dos requisitos innegociables. El primero es que el sistema pueda desplegarse en entornos IoT con las limitaciones mencionadas anteriormente, el segundo es que el sistema sea capaz de procesar una gran cantidad de datos, esto permite que el sistema sea escalable y funcione en redes de diferentes tamaños. El Procesamiento de Eventos Complehos (CEP) es una tecnología que cumple perfectamente estos requisitos. CEP permite recolectar una gran cantidad de datos en forma de eventos simples. Mediante reglas definidas por un experto se pueden extraer situaciones de interés de estos eventos simples, formando así eventos complejos. Esta funcionalidad es ideal, por ejemplo, para detectar ataques en red en tiempo real. En este caso usando los paquetes de red como eventos simples. El correcto despliegue de motores CEP en entornos IoT está ampliamente demostrado [7], [8], [9], [10]. Aunque CEP es muy ventajoso para la detección de ataques en tiempo real presenta una limitación, esta es la necesidad de un experto de dominio que sea capaz de definir reglas que deben cumplirse para llevar a cabo dicha detección.

Este trabajo se centra en diseñar e implementar una arquitectura capaz de generar reglas CEP de manera automática y no supervisada para detectar y clasificar ataques en red sin necesidad de un experto de dominio. Para ello aplicaremos técnicas de reducción de la dimensionalidad no supervisadas y clustering para modelar la normalidad mediante reglas, y después aplicar conceptos de detección de datos anómalos. De esta forma se podrán generar reglas efectivas y eficientes sin necesidad de datos etiquetados en entrenamiento.

El resto del artículo se compone de las siguientes sec-

ciones: La sección II describe los conceptos necesarios para comprender el artículo en su totalidad. La sección III es un breve estudio de otras propuestas para generar patrones CEP de manera automática. Posteriormente la sección IV describe el diseño y la implementación de la propuesta. Los resultados se describen y discuten en la sección V. En la sección VI se describen futuros trabajos posible. Por último se enumeran las conclusiones en la sección VII.

II. Fundamentos

Esta sección introduce los conceptos claves de este artículo: MQTT y CEP

II-A. Protocolo MQTT

MQTT (*Message Queue Telemetry Transport*) se trata de un protocolo que funciona en la capa de aplicación y se apoya en TCP/IP. Está orientado a comunicación en redes mediante un esquema publicador/subscriptor mediante topics. De esta forma los dispositivos (clientes) que requieran una información se subscriben al topic correspondiente. Los clientes que generan esta información publican en ese topic. Existe un nodo central llamado broker que se encarga de orquestar el comportamiento de la red, recibir los paquetes y reenviarlos a los nodos correspondientes. Este protocolo es especialmente útil en redes IoT porque es especialmente ligero, por estas razones es muy popular dentro del paradigma [11].

II-B. Procesamiento de eventos complejos (CEP)

CEP es una tecnología cuyo objetivo es detectar situaciones de interés, mediante la recolección y correlación de eventos. Para lograr esto, por norma general, un experto de dominio define reglas CEP que permiten comprobar situaciones específicas en los flujos de eventos. De este modo, cuando se cumple una regla, se genera un evento complejo que identifica una situación de interés. Los conceptos clave, a modo de resumen son los siguientes:

- Motor CEP: Se trata del software concreto que se emplea para realizar el procesamiento de eventos complejos. En nuestro caso se emplea Siddhi CEP.
- EPL: Se denomina EPL (*Event Processing Language*) al lenguaje empleado para definir las reglas en un motor CEP. En el caso de Siddhi CEP, el EPL se denomina SiddhiQL[12].
- Evento simple: Se conocen como eventos simples los datos en crudo que recibe el motor CEP, en el caso de la detección de ataques en red en tiempo real, estos eventos simples serán los paquetes de red. Sin embargo esto puede cambiar en función del contexto y del planteamiento del problema.
- Regla CEP: Llamamos reglas CEP a los patrones que describe e implementa un experto del dominio, estas reglas CEP describen situaciones de interés que quieren identificarse, estas reglas CEP están escritas en un lenguaje de procesamiento de eventos (Event Processing Language,EPL) que puede variar en función del motor CEP que se utilice. En este trabajos e emplea Siddhi, además en nuestro caso cada regla CEP puede identificar una familia de ataques.
- Evento complejo: Los eventos complejos identifican una situación de interés y son generados por las reglas CEP, cada vez que una de estas reglas se cumple se genera un evento complejo. En nuestro caso un evento complejo identifica que se ha detectado un ataque de una familia particular.

III. Estado del arte

Existen algunos trabajos relevantesque abordan el problema de la generación de reglas CEP desde diferentes perspectivas. Un estudio detallado de los diferentes enfoques es necesario para comprender las novedades intrínsecas de este enfoque.

Para una mejor comprensión de las diferentes propuestas, es conveniente clasificarlas. En este trabajo las clasificaremos según dos criterios. El primer criterio es la necesidad de disponer de reglas previas para la generación de nuevas reglas CEP. El segundo criterio consiste en la necesidad de etiquetar los diferentes eventos para que la propuesta funcione, es decir, supervisada o no supervisada.

III-A. Supervisado con reglas previas

En este grupo encontramos a las propuestas que necesitan datasets de entrenamiento etiquetados y reglas previas. Un enfoque diferente consiste en actualizar los patrones ya existentes. Esto permite generar nuevas reglas que ofrecen mejores resultados que las originales. Un trabajo que que encaja en esta categoría es el propuesto por Yunhao Sun et al. [13] En este trabajo, se utiliza un conjunto histórico de datos de entrenamiento y reglas CEP. En primer lugar, se utiliza una función de pérdida, que se obtiene a partir del error de las mediciones de las reglas anteriores con respecto a los resultados reales, y una función de activación basada en la curva S unipolar y la función de pérdida que determina si una regla es lo suficientemente buena para ser incluida en el nuevo conjunto de reglas. A partir de este nuevo subconjunto, se crean los clusters necesarios. A continuación, se obtiene el hiperrectángulo que contiene todos los elementos de cada familia y se genera una regla para representarla. Este trabajo es interesante aunque trata el problema de la actualización y no de la generación de nuevas reglas. además, el prefiltrado de las reglas antes del aprendizaje puede mejorar el rendimiento, lo cual es útil en un entorno de IoT.

Aunque la mayoría de los trabajos estudiados se basan en la generación de nuevas reglas para detectar eventos de Internet, existen otros enfoques. En el trabajo propuesto por Nathan Tri Luong et al. [14] CEP se utiliza para preprocesar los datos y un componente adicional, Tensor Flow, se utiliza en la implementación para realizar el entrenamiento y las clasificaciones de los diferentes eventos. En este tipo de enfoque, las reglas CEP realizan los procesos previos al entrenamiento y la clasificación. La limitación de esta arquitectura es que el cuello de botella puede trasladarse al componente encargado de realizar las clasificaciones. Esto hace que no se aproveche plenamente la capacidad de los motores CEP para procesar una gran cantidad de datos.

III-B. Supervisado sin reglas previas

En este grupo encontramos propuestas que no requieren reglas previas, sino que etiquetan eventos complejos basándose en datos históricos. Un artículo de esta categoría es el de Ralf Bruns y Jürgen Dunkel [15]. Este trabajo consigue adaptar el algoritmo bat a la búsqueda de reglas CEP estructurando

los diferentes operadores CEP, los valores de los atributos y las ventanas temporales en forma de árbol. De esta manera el algoritmo determina estos valores en la regla que representan. Los resultados obtenidos son muy buenos, y se consiguen utilizando un algoritmo poco habitual en el contexto del CEP. La única limitación de la propuesta es que necesita un contexto en el que los eventos complejos se mantengan en función de los eventos simples. Esto no siempre es fácil sin reglas previas.

Otro trabajo que consigue extraer reglas automáticamente sin reglas previas es el propuesto por José Roldán-Gómez et al. [16]. En este caso, las reglas se construyen a partir de la predicción del valor de la característica más importante para una categoría. Si la diferencia entre el valor real y la predicción supera un umbral, este simple evento no corresponde a una categoría. Los resultados obtenidos son buenos, aunque la principal limitación de este trabajo es la dificultad que puede existir para generar ciertas reglas basadas únicamente en una variable clave y un valor esperado.

Una evolución natural del trabajo anterior es la que vemos en el trabajo de José Roldán-Gómez et al. [17]. En este trabajo se reducen las dimensiones de los eventos individuales usando PCA, con esto se consiguen dos objetivos. El primero es caracterizar simplemente los eventos individuales, el segundo es mejorar drásticamente el rendimiento del motor CEP y la red del sistema reduciendo la dimensión de los eventos individuales. A partir de las etiquetas de los sucesos individuales, se calculan las medias de los sucesos reducidos. La regla consiste en una distancia euclidiana ponderada por los pesos de cada componente del suceso reducido. Esta diferencia se compara con la suma de errores de cada componente ponderada de nuevo con los pesos de cada componente y con la desviación estándar de cada componente. Los resultados de los experimentos son muy buenos, además de la reducción del tamaño del evento y la consiguiente mejora del rendimiento de la red y del motor CEP. Una pequeña limitación de este trabajo es que se trata de una forma supervisada para calcular la regla para cada categoría.

III-C. No supervisado con reglas previas

Este grupo es el menos común ya que requiere un entrenamiento no supervisado y la existencia de reglas capaces de detectar elementos de interés. Sin embargo, podemos encontrar trabajos como el trabajo de Haoyu Ren et al. [9]. Este se centra mucho en optimizar el rendimiento en entornos IoT, esta característica no es muy común en este tipo de propuestas. Para conseguir este objetivo, se utiliza un motor micro CEP y un modelo basado en Tensorflow Lite Micro con redes neuronales preentrenadas. Estas redes neuronales pueden ser actualizadas para adaptarse al comportamiento cambiante de un sistema real. El principal cambio de esta propuesta con respecto a las otras analizadas es que la salida de estas redes neuronales alimenta el motor CEP, que tiene reglas definidas manualmente. Puede parecer que esta propuesta no entra en el ámbito de la generación automática de reglas CEP. Sin embargo, es posible generar reglas sencillas que detecten la salida de las redes neuronales. De este modo, la clasificación es realizada por estas redes neuronales. La principal limitación de esta propuesta es que las reglas CEP son definidas manualmente, a diferencia de nuestra propuesta en la que se generan automáticamente.

III-D. No supervisado sin reglas previas

En este grupo encontramos propuestas que no requieren reglas o etiquetas previas en los datos. Algunos trabajos se centran principalmente en el etiquetado de eventos simples y luego utilizan algoritmos de extracción de reglas conocidos. El trabajo de Mehmet Ulvi Simsek et al. [18] realiza un estudio utilizando diferentes clasificadores para etiquetar eventos simples, luego utiliza los algoritmos más comunes para la extracción de reglas. Sus conclusiones muestran que GRU junto con el algoritmo FURIA obtienen los mejores resultados en sus experimentos. Este trabajo tiene un alto valor por el trabajo comparativo que realiza, el único pequeño inconveniente que puede tener en algunos casos. Es que los algoritmos basados en el aprendizaje profundo requieren una gran cantidad de datos.

Nuestra propuesta entraría en esta categoría. La novedad es que logramos una propuesta no supervisada y sin necesidad de reglas previas mientras realizamos una reducción de dimensión de los eventos, esto mejora el rendimiento computacional. Además, nuestra propuesta es capaz de funcionar correctamente entrenando con pocas muestras, esto es una ventaja respecto a las propuestas que se basan en deep learning. Por último, la implementación que se realiza facilita la creación y actualización de nuevas reglas en un sistema cambiante.

IV. Arquitectura propuesta

Esta sección describe la arquitectura, podemos observar un esquema gráfico en la Figura 1. Nuestra propuesta se centra en el generador de reglas CEP, pero como podemos observar los datos de entrenamiento se obtienen de la red IoT. Como se comentó anteriormente estos paquetes no están etiquetados y alimentan el generador de reglas CEP. Este generador de reglas hace uso de forma secuencial de las fases que se describen a continuación.

IV-A. Fase PCA

Esta fase se encarga de generar (o actualizar si no es la primera generación) el modelo PCA utilizando el tráfico de entrada. PCA es un método estadístico cuyo objetivo es la reducción de la complejidad de un espacio muestral mediante la reducción de las dimensiones de ese espacio. De este modo, si tenemos un elemento x representado por n variables, el objetivo es encontrar una representación con m variables donde m sea significativamente más pequeño que n. Estas nuevas variables se obtienen mediante combinaciones lineales de las originales, cada variable nueva se conoce como componente. Y cada componente es linealmente independiente de las otras componentes. El objetivo de PCA es maximizar la cantidad de información que representa cada componente. De esta forma si un elemento x está compuesto por el vector de variables $n = \{n_1, n_2, ..., n_n\}$, las nuevas variables del vector $m = \{m_1, m_2, ..., m_m\}$ tendrán la representación que podemos observar en la Ecuación 1. En este caso podemos observar que cada variable está ponderada con un peso α. Una variable con un peso mayor tiene más importancia en esa componente. Una ventaja de este modelo es la facilidad

para convertir un elemento del espacio original al reducido cuando tenemos el modelo PCA entrenado.

$$m_i = n_1 * \alpha_1 + n_2 * \alpha_2 + n_3 * \alpha_3 + ... + n_n * \alpha_n \quad (1)$$

Cada componente recoge una cantidad de información, esta cantidad se denomina ratio de varianza explicada rv. Las primera componentes siempre tiene un rv mayor que las últimas. En un escenario ideal y perfectamente lineal podría lograrse que la suma de los ratios de varianza explicada de todas las componentes fuese 1. En la practica buscamos aproximarnos lo máximo posible manteniendo la reducción de dimensiones lo más elevada que podamos.

Para implementar nuestra propuesta empleamos PCA incremental [19], esta versión permite recalcular el modelo si se añaden nuevos datos, para ello se emplea la media y las covarianzas del modelo actual. De esta forma no hay que generar un nuevo modelo desde cero si llegan nuevos datos de entrenamiento.

Una vez que se obtiene el modelo PCA entrenado se envía al broker de la red IoT, además se emplea este modelo para reducir el tráfico de entrada, también extraemos los ratios de varianza explicada en cada componente, y obtenemos la matriz diagonal de los mismos. Esta reducción es necesaria para las siguientes etapas. En este trabajo el modelo PCA reduce a 4 componentes, aunque esto es un detalle específico de la implementación y puede variar en otros escenarios.

IV-B. Fase GMM

Una vez que hemos reducido la dimensionalidad del tráfico, se emplean Modelos de mezcla gaussiana (GMM) para clusterizar el tráfico en diferentes familias.

GMM se trata de un modelo probabilístico que asume que para un conjunto de datos X existen k distribuciones normales que representan todas las categorías C presentes en los datos, dentro de las cuales se encuentran todos los elementos de X. El objetivo de GMM es encontrar la mejor combinación de los parámetros para las K distribuciones normales. De esta forma podemos agrupar los elementos en k familias diferentes.

$$p(x_i) = \Sigma_{k=1}^{K} p(x_i|c_k)p(c_k) \quad (2)$$

La Ecuación 2 describe la probabilidad del elemento $x_i \in X$ como la suma de probabilidades compuestas que tiene de pertenecer a cada familia, de forma que $p(x_i) = 1$. Esto quiere decir que GMM asume que todos los elementos se encuentran dentro de esas distribuciones, como se comentó anteriormente.

$$p(x_i) = \Sigma_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \quad (3)$$

La ecuación 3 representa el modelo GMM como una combinación lineal las K distribuciones normales. De forma que π_k es el coeficiente de mezcla para cada distribución y ofrece un estimado para cada una de las distribuciones normales. Por su parte $\mathcal{N}(x|\mu_k, \Sigma_k)$ es llamada componente del modelo de mezcla, modela y describe cada una de las distribuciones normales, μ_k es la media y Σ_k es la covarianza.

La enorme ventaja de GMM es que permite cierta flexibilidad en cada categoría, de tal forma que 2 normales pueden ser muy distintas, además no tiene un sesgo para grupos circulares funcionando bien incluso en cierta distribuciones no lineales [20].

En este caso se emplea una versión variacional del algoritmo [21], esta permite inferir una cantidad óptima de distribuciones normales. El objetivo de usar esta versión es no tener que indicar el número de familias K a priori, esto permite que el proceso sea completamente no supervisado.

GMM nos permite generar familias anónimas sin necesidad de etiquetar los datos de entrenamiento previamente, esto permite agrupar los diferentes datos de entada en las diferentes familias. Con esa agrupación se extrae la media de cada familia, también se aprovecha esta etapa para calcular la matriz de covarianzas, aunque esta es común a todos los datos de entrenamiento reducidos con PCA.

IV-C. Fase Umbral

En esta etapa se calcula el umbral para cada familia empleando la distancia de Mahalanobis. La distancia de Mahalanobis se trata de una función de distancia que tiene en cuenta la matriz de covarianza para ponderar la misma [22]. La ventaja fundamental de la distancia de Mahalanobis es que tiene en cuenta las diferencias de escala que pueden existir entre las diferentes variables así como la correlación que pueda existir entre las mismas (aunque esta última propiedad no la necesitamos en nuestro caso).

En esta propuesta empleamos la distancia de Mahalanbois para ver la diferencia de cada elemento reducido mediante PCA con respecto a las categorías que obtuvimos anteriormente con GMM.

$$d(x-\mu) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \quad (4)$$

La ecuación 4 describe como se calcula la diferencia entre el elemento x y la media de una categoría μ. Σ^-1 representa la matriz de covarianzas inversa, como vemos lo que se realiza es una transformación para que estas covarianzas tengan peso en la función distancia.

En nuestro caso particular, como vamos a aplicar la distancia de Mahalanobis a elementos reducidos con PCA podemos mejorar la función distancia si tenemos en cuenta el ratio de varianza explicada de las diferentes componentes y ponderamos con las mismo. De esta forma, nuestra función distancia dará mayor peso a las componentes con un rv mayor. El primer paso obtener la matriz VE como la matriz diagonal con los ratios de varainza explicada de cada componente.

$$VE = diag(rv_1, rv_2, ..., rv_m) \quad (5)$$

$$d(x-mu) = \sqrt{(x-\mu)^T (\Sigma^{-1} \times VE) (x-\mu))} \quad (6)$$

La Ecuación 5 muestra cómo se obtiene la matriz que utilizamos para ponderar los ratios de varianza explicados. La Ecuación 6 muestra cómo queda la distancia de Mahalanobis ponderando estos ratios de varianza explicada.

Empleando la ecuación 6, cada elemento se compara con la media de cada familia. Una vez que tenemos todas las distancias podemos calcular el umbral para esa familia, para ello se usa el elemento más lejano de la familia respecto a la media y el más cercano no perteneciente a la familia respecto a la media de la familia. Con estas distancias se calcula el punto medio, que define el umbral para esa categoría.

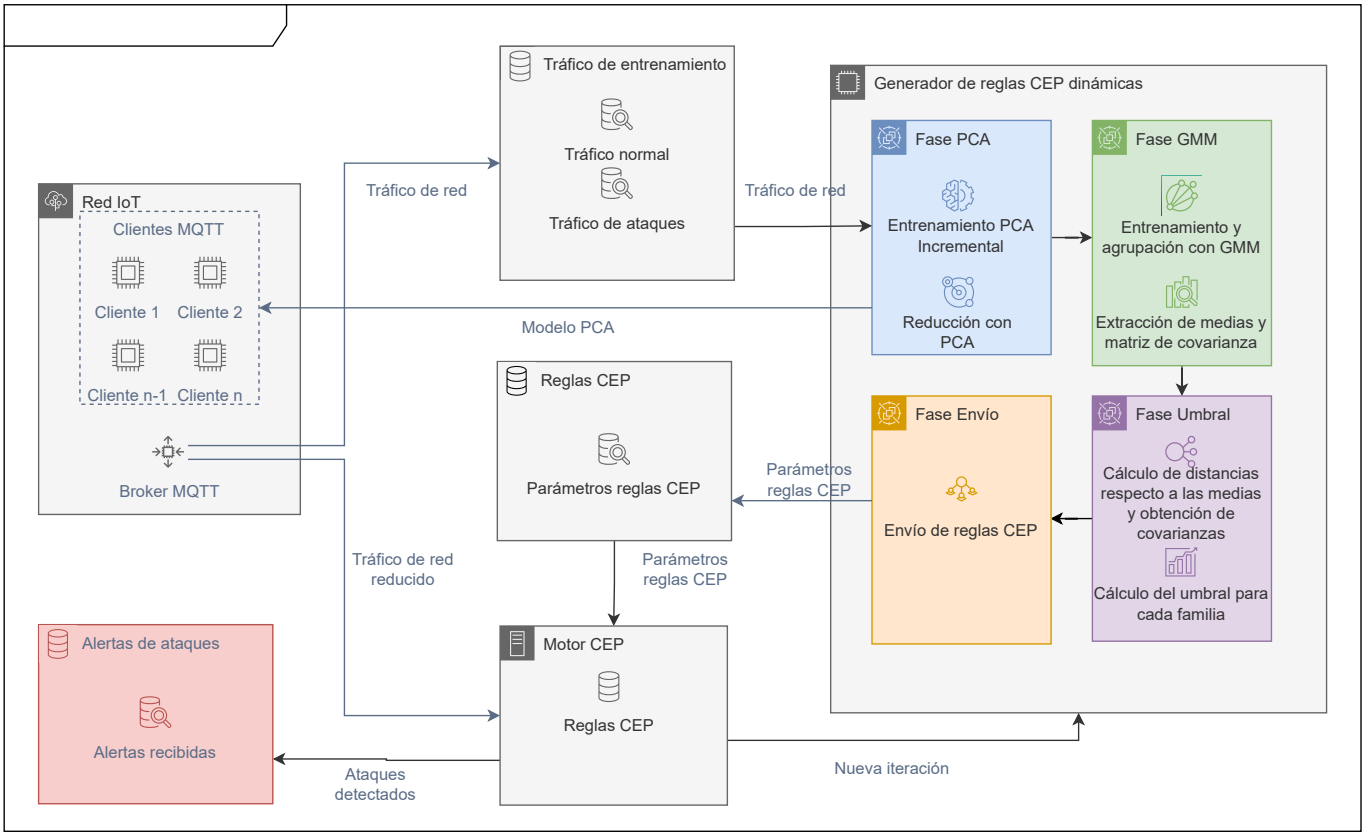


Figura 1. Diagrama de la arquitectura propuesta para detectar ataques IoT en tiempo real.

IV-D. Fase Envío

En esta etapa se envían los parámetros de las reglas al motor CEP. El código Siddhi se envía la primera vez, pero no es necesario enviarlo nuevamente en las siguientes iteraciones. Esto nos permite generar reglas CEP dinámicas, esto es una novedad muy interesante de nuestra propuesta.

En cuanto al funcionamiento de reglas CEP dinámicas, cuando el generador de reglas CEP genera nuevas reglas, no es necesario generar un nuevo fichero Siddhi, que es utilizado para generar una aplicación en el motor Siddhi, en su lugar se hace uso de tablas dinámicas que contienen los parámetros de las reglas actuales. Esta ventaja de implementación reduce el trasiego de datos en red al actualizar o generar nuevas reglas, además facilita enormemente la implementación, la creación y la actualización.

Una vez en funcionamiento el broker reduce los paquetes con PCA y los envían al motor CEP. Con estos paquetes reducidos se calcula la distancia de los mismos paquete respecto a la media de cada familia con la Ecuación 6, si esta distancia es inferior al umbral de esa familia se considera que el paquete pertenece a esa familia. En caso de que un paquete no entre dentro del umbral de ninguna familia se considera que ese paquete es una anomalía.

Las partes más importantes de la aplicación Siddhi pueden verse en el listado 1. Como vemos las partes más importantes del código pueden dividirse en 4 bloques separados por espacios.

El primer bloque de código define una tabla donde se almacenan los parámetros necesarios para cada regla. En este caso necesitamos almacenar un *id*, el cual es unívoco dentro de una iteración, la iteración actual, la media de esa familia, el umbral y la matriz de covarianzas inversa. El siguiente bloque se encarga de obtener la diferencia de cada evento que llega respecto a las medias de las familias almacenadas en la tabla. Los últimos 2 bloques implementan la ecuación 6, esta se realiza en dos partes para mejorar la legibilidad.

La gran ventaja de esta implementación es que crear o actualizar reglas, consiste simplemente en actualizar la tabla porque la estructura se mantiene. Estas tablas, pueden almacenarse en una base de datos o almacenarse de forma volátil en memoria. Esto sumado al funcionamiento no supervisado de la propuesta ofrece una solución apta para desplegarse sin necesidad de un experto de dominio.

Podemos observar también que el motor CEP puede solicitar nuevas iteraciones al generador de reglas. En nuestros experimentos estas nuevas iteraciones vienen definidas por los datasets de entrenamiento, esto permite generar experimentos reproducibles. En un despliegue real podrían iniciarse nuevas iteraciones cuando se obtienen un número determinado de anomalías, o cuando transcurre un tiempo específico. Esto dependerá del tipo de red y aplicaciones que se tengan.

V. Experimentos y resultados

En esta sección se describen los experimentos realizados, además se analizan y discuten los resultados de los mismos.

El escenario que planteamos es una red MQTT con tres clientes legítimos y un broker. Los clientes generan datos numéricos y los envían al broker, esto permite simular un escenario de envío de temperaturas. Para demostrar que se clasifican correctamente ataques desconocidos se han im-

Listing 1. Aplicación Siddhi para detectar ataques en tiempo real

```
@primaryKey('idRule')
@index('idRule')
define table ParametersTable(idRule int,iteration int,m1 double,m2 double,m3 double,m4 double,
threshold double, x00 double, x01 double, x02 double, x03 double, x10 double, x11 double,
x12 double, x13 double, x20 double, x21 double, x22 double, x23 double, x30 double, x31 double,
x32 double, x33 double);


from ReducedEvent as re left outer join ParametersTable as pt
select pt.idRule as idFam , re.c1-pt.m1 as d1,re.c2-pt.m2 as d2,re.c3-pt.m3 as d3,re.c4-pt.m4
as d4, re.c1 as c1, re.c2 as c2, re.c3 as c3, re.c4 as c4
insert into MeanDiffEvent;


from MeanDiffEvent as md join ParametersTable as pt
 on md.idFam==pt.idRule
select md.idFam,md.d1,md.d2,md.d3,md.d4,
((md.d1*pt.x00)+(md.d2*pt.x10)+(md.d3*pt.x20)+(md.d4*pt.x30)) as cd1,
((md.d1*pt.x01)+(md.d2*pt.x11)+(md.d3*pt.x21)+(md.d4*pt.x31)) as cd2,
((md.d1*pt.x02)+(md.d2*pt.x12)+(md.d3*pt.x22)+(md.d4*pt.x32)) as cd3,
((md.d1*pt.x03)+(md.d2*pt.x13)+(md.d3*pt.x23)+(md.d4*pt.x33)) as cd4,
md.c1 as c1, md.c2 as c2, md.c3 as c3, md.c4 as c4
insert into ComputedMeanDiffEvent;


from ComputedMeanDiffEvent as cm join ParametersTable as pt
 on cm.idFam==pt.idRule
 select pt.iteration,cm.idFam, cm.c1, cm.c2, cm.c3, cm.c4
 having math:sqrt((cm.d1*cm.cd1)+(cm.d2*cm.cd2)+(cm.d3*cm.cd3)+(cm.d4*cm.cd4))<pt.threshold
 insert into DetectedEvent;
```

plementado diferentes ataques, estos ataques son diferentes entre sí para demostrar el correcto funcionamiento de nuestra propuesta. Los ataques son los siguientes:

- *Subscription fuzzing*: Este ataque consiste en intentar suscribirse a diferentes topics, se puede usar cuando tenemos acceso a un sistema MQTT.
- *Disconnection wave*: Consiste en spoofear el *id* del protocolo MQTT y lanzar el comando de desconexión, si no se configura correctamente es posible robar el *id* del dispositivo legítimo y expulsarlo del sistema. El objetivo de este ataque es desconectar todos los dispositivos del sistema.
- *TCP syn scan*: Se trata del escáner clásico empleado para comprobar qué puertos TCP están abiertos. El atacante inicia con un un paquete SYN. Si recibe un SYN/ACK supone que el puerto está abierto, si recibe un RST supone que está cerrado.
- *UDP scan*: Se trata de enviar paquetees UDP a cada puerto a escanear, si se recibe una respuesta UDP se considera el puerto abierto, si no se recibe ninguna respuesta el puesto está abierto o filtrado, un paquete del tipo ICMP *port unreachable error* quiere decir que el puerto está cerrado y cualquier otro tipo de error ICMP quiere decir que el puerto está filtrado.
- *Xmas scan*: Se trata de un escáner bastante poco usual en la actualidad, sin embargo lo empleamos en el escenario porque es diferente del escáner UDP y del TCP SYN. Se trata de enviar a cada puerto TCP un paquete con las banderas FIN, PSH y URG a 1. Si no se recibe respuesta se considera el puerto abierto o filtrado, si se recibe un RST se considera un puerto cerrado, si se recibe cualquier paquete ICMP *unreachable error* se considera un puerto filtrado.
- *Telnet connection*: Se trata de paquetes que tratan de conectar por Telnet con diferentes usuarios y contraseñas, para simular la primera etapa de Mirai. La idea es comprobar la propuesta ante un escenario muy usual [23].

Utilizando los paquetes normales y los ataques se generan datasets de entrenamiento y de testing. Los experimentos se realizan en varias iteraciones. En la primera iteración se entrena solo con paquetes normales, esto sería lo normal si se despliega la arquitectura por primera vez, aunque es posible hacer el primer entrenamiento con paquetes de ataques sin problema. A partir de la primera iteración se introducen nuevos ataques en cada iteración y se entrena el modelo nuevamente con los paquetes que no hayan sido clasificados por ninguna regla previa. Este proceso busca demostrar que la arquitectura propuesta es capaz de detectar nuevos ataques de forma no supervisada e incremental. Cabe destacar que la primera vez que se manda un ataque al motor CEP se envía el dataset de entrenamiento, esto es para evitar que el motor entrene con el dataset de testing, posteriormente se usa el dataset de testing de ese ataque para demostrar que la regla creada funciona correctamente. El dataset es accesible desde el siguiente repositorio https://data.mendeley.com/datasets/cpnh332y5t/draft?a=e93dc79e-8f25-4122-a9c8-862e619d06b3 [24].

Un detalle importante a tener en cuenta, es que la primera vez que se detecte un ataque este no entrará en ninguna regla CEP porque es una anomalía. Con el entreno posterior del modelo con los nuevos datos se generará la nueva regla CEP. Para evaluar los resultados de estos experimentos se usan las métricas clásicas siguientes:

- $Precision = \frac{TP}{TP+FP}$

Resultados de primera iteración	TP	FP	TN	FN	Precision score	Recall score	F1 score
Tráfico normal	7936	0	3820	0	1	1	1
Tráfico *Subscription fuzzing*	819	0	10936	1	1	0,99878049	0,99938987
Tráfico *Disconnection wave (training dataset)*	3000	1	8755	0	0,99966678	1	0,99983336

Tabla I
RESULTADOS DE LA ITERACIÓN DONDE ENTRA *Disconnection wave*

Resultados de la ultima iteración	TP	FP	TN	FN	Precision score	Recall score	F1 score
Tráfico normal	7780	0	18131	156	1	0,98034274	0,99007381
Tráfico *Subscription fuzzing*	819	0	25247	1	1	0,99878049	0,99938987
Tráfico *Disconnection wave*	16999	0	9067	1	1	0,99994118	0,99997059
Tráfico *TCP syn scan + UDP scan*	160	1	25906	0	0,99378882	1	0,99688474
Tráfico *Telnet conection*	51	0	26016	0	1	1	1
Tráfico *XMAS scan*	100	0	25967	0	1	1	1

Tabla II
RESULTADO DE LOS EXPERIMENTOS REALIZADOS

- $Recall = \frac{TP}{TP+FN}$
- $F1Score = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$

Donde TP son los positivos reales, FP son los falsos positivos y FN son los falsos negativos.

De esta forma una puntuación de *Recall* alta significa que una regla CEP detecta muchos eventos que pertenecen realmente a esa familia, una puntuación alta de *Precision* quiere decir que esa regla CEP no detecta muchos falsos positivos. Por último *F1 Score* hace uso de las dos puntuaciones para obtener una métrica equilibrada entre ambas.

En la Tabla I vemos los resultados de una iteración intermedia, la segunda en este caso, como vemos los resultados son muy buenos. Es necesario comprender que en esta iteración solo existen las reglas capaces de detectar tráfico normal y tráfico del ataque *subscription fuzzing*. Esto quiere decir que el tráfico de *disconenction wave* se detecta como una anomalía, ya que no entra dentro de ninguna regla CEP, esto se debe a que no se han generado las familias para detectar este ataque porque el modelo aún no ha entrenado con estos datos. Esta tabla ilustra lo bien que funcionan las reglas para detectar tráfico del que se tienen muestras en entrenamiento y cómo es capaz de detectar situaciones anómalas sin problemas. De los ataques de tipo *Subscription fuzzing* solo se falla en la detección de un paquete. Por su parte, del tráfico *Disconnection wave* también se detectan todos los paquetes excepto uno.

La tabla II muestra los resultados con reglas creadas para todas las familias de ataques. Esta tabla demuestra lo bien que funcionan las reglas CEP generadas de forma completamente no supervisada por nuestra propuesta. Un detalle importante es que en nuestro escenario GMM, se juntó a los escáner de tipo UDP y TCP en las mismas familias. Por eso se clasifican con las mismas reglas. Este comportamiento se debe a que la propuesta está modelando los comportamientos de los ataques, por eso dos ataques muy parecidos se clasifican juntos. En este caso ocurre porque existen muy pocos elementos de escáner tanto TCP como UDP por eso no tienen un gran peso en el modelo. Incluso con este comportamiento, que puede presentarse en un entorno real, el sistema obtiene unos resultados muy buenos. El peor resultado, en cuanto a *F1 score* se refiere, es de 0,98 para el tráfico normal, esto es un indicativo de lo bien que funcionan las reglas CEP generadas por nuestra propuesta.

VI. TRABAJOS FUTUROS

En esta sección se describen futuros trabajos que pueden suponer una mejora de la propuesta presentada.

La primera ampliación posible consiste en entrenar el modelo variando los ataques en cada etapa y entremezclarlos para comprobar si existen diferencias significativas en el funcionamiento. En algunos casos esto puede asimilarse más a un escenario real.

Esta propuesta emplea el motor Siddhi CEP, pero sería muy interesante adaptar la propuesta a otros motores CEP y realizar una comparativa del rendimiento a nivel computacional. Esto nos permitiría entender que motor CEP es mejor en cada situación.

Por último, sería interesante implementar un etiquetado de los ataques anómalos y la generación de descriptores que permitan a un experto o al propio sistema conocer los nuevos ataques que llegan.

VII. CONCLUSIONES

En este trabajo se propone una arquitectura centrada en el paradigma IoT capaz de generar y actualizar reglas CEP de manera no supervisada para detectar y clasificar ataques en red en tiempo real sin la necesidad de un experto de dominio. La integración de CEP y de PCA para reducir la dimensión de los paquetes hace que la arquitectura sea óptima entornos IoT.

Las reglas generadas por nuestra arquitectura funcionan muy bien. Los resultados obtenidos son muy buenos generados de una manera no supervisada y de forma incremental, de tal modo que pueda desplegarse en todo tipo de entornos y aprender constantemente.

La arquitectura permite detectar anomalías de forma exitosa, posteriormente estas anomalías entrenan al modelo y se generan nuevas familias. Además la propuesta es exitosa al entrenar de forma incremental y clasificar nuevas familias de ataques gracias al entrenamiento con iteraciones.

Todas estas conclusiones obtenidas demuestran que nuestra propuesta puede desplegarse de forma exitosa en entornos IoT con limitaciones reducidas y generar reglas CEP dinámicas sin supervisión que son capaces de detectar ataques en red en tiempo real.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Ciencia, Innovación y Universidades y los Fondos FEDER de la Unión Europea [FPU 17/02007 y FPU 17/03105, RTI2018-093608-B-C33 y RTI2018-098156-B-C52], por la Universidad de Castilla La Mancha [número de subvención DO20184364]. Este trabajo también ha sido financiado por la JCCM [número de beca SB-PLY/17/180501/ 000353], y el Plan de Investigación de la Universidad de Cádiz y el Grupo Energético de Puerto Real S.A. bajo el proyecto GANGES [IRTP03_UCA].

REFERENCIAS

[1] A. A. AlZubi, M. Al-Maitah, A. Alarifi, Cyber-attack detection in healthcare using cyber-physical system and machine learning techniques, Soft Computing 25 (18) (2021) 12319–12332. doi:10.1007/s00500-021-05926-8.

[2] P. Asghari, A. M. Rahmani, H. H. S. Javadi, Internet of Things applications: A systematic review, Computer Networks 148 (2019) 241–261. doi:10.1016/j.comnet.2018.12.008.

[3] I. Calvo, M. G. Merayo, M. Núñez, A methodology to analyze heart data using fuzzy automata, Journal of Intelligent & Fuzzy Systems 37 (6) (2019) 7389–7399. doi:10.3233/JIFS-179348.

[4] M. M. Sadeeq, N. M. Abdulkareem, S. R. Zeebaree, D. M. Ahmed, A. S. Sami, R. R. Zebari, IoT and cloud computing issues, challenges and opportunities: A review, Qubahan Academic Journal 1 (2) (2021) 1–7.

[5] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, E. K. Markakis, A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues, IEEE Communications Surveys Tutorials 22 (2) (2020) 1191–1221. doi:10.1109/COMST.2019.2962586.

[6] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, B. Sikdar, A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures, IEEE Access 7 (2019) 82721–82743. doi:10.1109/ACCESS.2019.2924045.

[7] G. Ortiz, J. Boubeta-Puig, J. Criado, D. Corral-Plaza, A. Garcia-de Prado, I. Medina-Bulo, L. Iribarne, A microservice architecture for real-time IoT data processing: A reusable Web of things approach for smart ports, Computer Standards & Interfaces 81 (2022) 103604. doi:10.1016/j.csi.2021.103604.

[8] D. Corral-Plaza, I. Medina-Bulo, G. Ortiz, J. Boubeta-Puig, A stream processing architecture for heterogeneous data sources in the Internet of Things, Computer Standards & Interfaces 70 (2020) 103426. doi:10.1016/j.csi.2020.103426.

[9] H. Ren, D. Anicic, T. A. Runkler, The synergy of complex event processing and tiny machine learning in industrial IoT, in: Proceedings of the 15th ACM International Conference on Distributed and Event-based Systems, DEBS '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 126–135. doi:10.1145/3465480.3466928.
URL https://doi.org/10.1145/3465480.3466928

[10] J. Roldán-Gómez, J. Boubeta-Puig, G. Pachacama-Castillo, G. Ortiz, J. L. Martínez, Detecting security attacks in cyber-physical systems: a comparison of Mule and WSO2 intelligent IoT architectures, PeerJ Computer Science 7 (2021) e787, publisher: PeerJ Inc. doi:10.7717/peerj-cs.787.
URL https://peerj.com/articles/cs-787

[11] D. Soni, A. Makwana, A survey on MQTT: A protocol of internet of things(IoT), 2017.

[12] Query Guide - Siddhi, accessed 8/01/2022.
URL https://siddhi.io/en/v5.1/docs/query-guide/

[13] Y. Sun, G. Li, B. Ning, Automatic Rule Updating based on Machine Learning in Complex Event Processing, in: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), 2020, pp. 1338–1343, iSSN: 2575-8411. doi:10.1109/ICDCS47774.2020.00176.

[14] N. N. T. Luong, Z. Milosevic, A. Berry, F. Rabhi, An open architecture for complex event processing with machine learning, in: 2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC), 2020, pp. 51–56, iSSN: 2325-6362. doi:10.1109/EDOC49727.2020.00016.

[15] R. Bruns, J. Dunkel, Bat4CEP: a bat algorithm for mining of complex event processing rules, Applied Intelligence (Mar. 2022). doi:10.1007/s10489-022-03256-2.
URL doi.org/10.1007/s10489-022-03256-2

[16] J. Roldán, J. Boubeta-Puig, J. Luis Martínez, G. Ortiz, Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks, Expert Systems with Applications 149 (2020) 113251. doi:10.1016/j.eswa.2020.113251.
URL https://www.sciencedirect.com/science/article/pii/S0957417420300762

[17] J. Roldán-Gómez, J. Boubeta-Puig, J. M. Castelo Gómez, J. Carrillo-Mondéjar, J. L. Martínez Martínez, Attack Pattern Recognition in the Internet of Things using Complex Event Processing and Machine Learning, in: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2021, pp. 1919–1926, iSSN: 2577-1655. doi:10.1109/SMC52423.2021.9658711.

[18] M. U. Simsek, F. Yildirim Okay, S. Ozdemir, A deep learning-based CEP rule extraction framework for IoT data, The Journal of Supercomputing 77 (8) (2021) 8563–8592. doi:10.1007/s11227-020-03603-5.
URL https://doi.org/10.1007/s11227-020-03603-5

[19] D. A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental Learning for Robust Visual Tracking, International Journal of Computer Vision 77 (1) (2008) 125–141. doi:10.1007/s11263-007-0075-7.
URL https://doi.org/10.1007/s11263-007-0075-7

[20] E. Patel, D. S. Kushwaha, Clustering Cloud Workloads: K-Means vs Gaussian Mixture Model, Procedia Computer Science 171 (2020) 158–167. doi:10.1016/j.procs.2020.04.017.
URL https://www.sciencedirect.com/science/article/pii/S1877050920309820

[21] D. M. Blei, M. I. Jordan, Variational inference for Dirichlet process mixtures, Bayesian Analysis 1 (1) (2006) 121–143, publisher: International Society for Bayesian Analysis. doi:10.1214/06-BA104.
URL https://projecteuclid.org/journals/bayesian-analysis/volume-1/issue-1/Variational-inference-for-Dirichlet-process-mixtures/10.1214/06-BA104.full

[22] R. De Maesschalck, D. Jouan-Rimbaud, D. L. Massart, The Mahalanobis distance, Chemometrics and Intelligent Laboratory Systems 50 (1) (2000) 1–18. doi:10.1016/S0169-7439(99)00047-7.
URL https://www.sciencedirect.com/science/article/pii/S0169743999000477

[23] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, et al., Understanding the mirai botnet, in: 26th USENIX security symposium (USENIX Security 17), 2017, pp. 1093–1110.

[24] J. Roldán-Gómez, Hacia la creación de reglas CEP no supervisadas para la detección en tiempo real de ataques en entorno IoT (dataset), https://data.mendeley.com/datasets/cpnh332y5t/draft?a=e93dc79e-8f25-4122-a9c8-862e619d06b3 (2022).

MECInOT: Emulador de escenarios de Industrial Internet of Things y Multi-Access Edge Computing para su análisis de seguridad

Sergio Ruiz Villafranca
Instituto de Investigación en Informática de Albacete (i3a)
Universidad de Castilla-La Mancha
sergio.rvillafranca@uclm.es

José Roldán Gómez
Instituto de Investigación en Informática de Albacete (i3a)
Universidad de Castilla-La Mancha
Jose.Roldan@uclm.es

José Luis Martínez
Instituto de Investigación en Informática de Albacete (i3a)
Universidad de Castilla-La Mancha
joseluis.martinez@uclm.es

José Miguel Villalón Millán
Instituto de Investigación en Informática de Albacete (i3a)
Universidad de Castilla-La Mancha
josemiguel.villalon@uclm.es

***Resumen*—En los últimos años se está produciendo una gran revolución en el modo de gestionar los entornos industriales. El uso de nuevos dispositivos de la Internet de las Cosas (IoT) en estos entornos está produciendo lo que se conoce con el nombre de industria 4.0. Estos dispositivos IoT se caracterizan por su simplicidad y por su poca capacidad computacional. En estos escenarios, es de especial importancia solventar los nuevos problemas de seguridad que pueden aparecer, especialmente en aquellas infraestructuras consideradas críticas. Para facilitar esta tarea se presenta MECInOT, un emulador que permite a los investigadores generar diferentes escenarios de red sin la necesidad del gasto asociado al equipo industrial. Este emulador es completamente flexible, gracias a la virtualización de dispositivos, ajustándose a las necesidades que puedan surgir en este tipo de escenarios. A partir del emulador propuesto, se pueden desplegar y analizar medidas de seguridad para medir y evaluar su impacto.**

***Index Terms*—Industrial Internet of Things, Industria 4.0, Ciberseguridad, openLEON, Docker, MEC**

Tipo de contribución: *Investigación en ciberseguridad*

I. Introducción

Con el paso del tiempo se han ido produciendo sucesivas revoluciones industriales, con el objetivo de mejorar la productividad y viabilidad de las fábricas. Últimamente, el principal foco reside en la optimización de los recursos materiales y humanos gracias a los avances tecnológicos que están apareciendo. En este entorno, en la actualidad nos encontramos en la cuarta revolución industrial, también llamada industria 4.0 o IIoT (*Industrial Internet of Things*), que contempla ya los avances y las implementaciones de tecnologías IT, (*Information Technologies*). En esta cuarta revolución se considera la inclusión de las nuevas tecnologías como inteligencia artificial, *big data*, *cloud computing*, *edge computing*, virtualización, sistemas ciberfísicos, sensores IoT (*Internet of Things*), junto con los protocolos OT (*Operational Technologies)* que tradicionalmente se han establecido en este ámbito [1]. Incluso, ya existen autores [2] que están proponiendo soluciones a los problemas derivados de la implementación de la industria 4.0, abocando por la denominación de la industria 5.0. En esta futura industria se busca la personalización y el enfoque de las ventajas y funcionalidades de estas tecnologías en función de ámbito de producción de cada compañía.

El término de IIoT [3] nace al añadir dispositivos M2M (*machine to machine*) a Internet, incluyendo el uso de las tecnologías IoT en entornos industriales. Esto provoca que, gracias a la información que aportan estos dispositivos a las nuevas industrias, se genere una gran cantidad de datos que serán transportados por la red. Tal y como se menciona en [4], se espera que el 50 % de los dispositivos conectados a Internet en 2023 serán de este tipo. Este hecho hará que más de 14.7 mil millones de estos dispositivos estén conectados a Internet.

Por otro lado, una de las tecnologías que han revolucionado el mundo de la informática en los últimos años ha sido el *Cloud Computing* [5]. Su uso ha servido a muchas empresas como una opción económica de obtener una capacidad de cómputo moldeable en función de las necesidades que puedan surgir en momentos concretos. Sin embargo, con el incremento del número de dispositivos conectados, se está comenzando a apreciar una reducción en el rendimiento de la red de los principales proveedores de Cloud (Amazon Web Services, Azure Cloud, Google Cloud). Esto esta provocando que aparezcan retardos en las comunicaciones, haciendo que ciertas funcionalidades con restricciones temporales severas tengan problemas a la hora de cumplir con sus prestaciones, produciéndose latencias superiores a las deseadas [6]. La solución propuesta para esta situación ha sido la definición e implementación de *Edge & Fog Computing* [5], lo que nos permite acercar el procesamiento de la nube a las redes empresariales. Esto a su vez, abre la oportunidad de la aparición de nuevas aplicaciones que eran inviables en *Cloud Computing*, como todas aquellas asociadas a aplicaciones en tiempo real y con restricciones temporales muy exigentes, como la conducción autónoma o a la gestión y manejo de la información recibida por los drones no tripulados. Por otra parte, *Edge Computing* también ofrece la posibilidad de mejorar la gestión de un tráfico heterogéneo y distribuido, como el que se produce con el IIoT [7].

A pesar de las grandes aportaciones y ventajas que ofrecen

estas tecnologías, esta nueva situación provoca la aparición de nuevas vías de entrada y vulnerabilidades que pueden ser aprovechadas por los atacantes. Antes de la aparición del IIoT, las industrias se encontraban más aisladas de Internet, por lo que el acceso malintencionado a sus infraestructuras críticas era más difícil. Por otro lado, si la adopción e implementación de estas nuevas tecnologías no se realiza siguiendo una serie de buenas prácticas y con especial enfoque en evitar la aparición de riesgos en los sistemas, se pueden dar incidencias de seguridad sin que el personal de la empresa este preparado para poder afrontar esta situación [3]. En este punto, existe una fuerte demanda de herramientas que permitan emular en entornos controlados los procesos industriales, en una linea a lo que introduce el gemelo digital para entornos industriales. A pesar de las limitaciones que existen en la virtualizacion de entornos industriales, estas herramientas pueden servir para la mejora de los protocolos de red usados en dichos entornos.

Con esta premisa se presenta MECInOT, un emulador que está basado en openLEON [8], y que se enfoca en el despliegue de diferentes escenarios industriales caracterizados por el uso simulado de dispositivos IIoT. Con este trabajo se busca proporcionar una herramienta de virtualizacion de la capa de red IIoT. También, se quiere otorgar la capacidad de modificar e introducir nuevos dispositivos y servicios al emulador de los ya existentes en él. Incluso, la incorporación de la comunicación con *hardware* real, con el fin de dar soporte al paradigma de gemelo digital [9]. Además, se incluye un conjunto de herramientas en MECInOT, para realizar pruebas de seguridad en los entornos diseñados. Con el fin, de dar soporte a las investigaciones realizadas en el ámbito de la ciberseguridad en los entornos IIoT y MEC, y comprobar el impacto de los diferentes modelos de amenaza presentes en ellos.

II. Fundamentos Técnicos

En este apartado profundizaremos en algunas partes teóricas de las arquitecturas que desplegará el emulador, y en posibles riesgos y problemas que encontramos en los entornos IIoT.

II-A. Multi-Access Edge Computing (MEC)

MEC es una evolución del concepto de *Edge Computing*, que fue estandarizado por el *European Telecommunications Standards Institute* (ETSI), y que busca mejorar el rendimiento de las comunicaciones entre el entorno de *Cloud Computing* y las tecnologías relacionadas con IoT e IIoT. Uno de los puntos más destacados de MEC es el uso de la virtualización para permitir la computación de diferentes aplicaciones y la gestión del tráfico heterogéneo que tiene que soportar [10]. Como ya hacía *Edge Computing*, al acercar la computación que tradicionalmente se realizaba en los entorno *Cloud*, podemos reducir la latencia de las comunicaciones y evitar posibles saturaciones en la red. Esto se traduce en una mejor experiencia para los usuarios finales de las aplicaciones, permitiendo el desarrollo de aplicaciones que en base a las limitaciones de las tecnologías que había hasta ahora era inviable. Además, el uso de las nuevas redes móviles 5G y las futuras 6G [6], permitirá reducir aún más las latencias entre los dispositivos, lo cual posibilitará que las aplicaciones en tiempo real puedan desplegarse en este tipo de arquitecturas.

Para poder implementar MEC, debemos de hacer uso de varias tecnologías de virtualización sobre las que se asienta este nuevo paradigma [11]. Estas tecnologías son:

- ***Network Function Virtualization* (NFV) [12].** Como se ha mencionado previamente, el uso del IIoT va a producir que una gran cantidad de dispositivos IoT estén conectados en entornos industriales. Muchos de estos dispositivos van a hacer uso de protocolos de comunicación propietarios, ya que son diseñados para entornos muy concretos. Esto obligaría a utilizar dispositivos de red específicos para poder gestionar este tráfico. Gracias a NFV, se puede gestionar todo este tráfico heterogéneo de manera automatizada, reduciendo los costes que conlleva el uso de dispositivos especializados, respecto a dispositivos generales. En esencia, NFV se trata de un software que puede ser instalado en cualquier dispositivo físico que permita la comunicación con él. ETSI especifica algunas aplicaciones que se podrían abordar con NFV, como las funciones de conectividad *Dynamic Host Configuration Protocol* (DHCP) o *Network Address Translation* (NAT), funciones de seguridad con *firewalls*, y gestión del tráfico procedente de redes móviles.
- ***Software Defined Networking* (SDN).** Tradicionalmente, en las redes de comunicaciones la capa de datos se correspondía con la parte de la red que se ocupaba de gestionar el tráfico generado por los usuarios. Por otro lado, la capa de control se ocupaba de la gestión de aquellas operaciones de enrutamiento. Ambas capas trabajaban conjuntamente en los propios dispositivos de red, provocando que estos dispositivos se ralentizaran. Con el incremento del número de dispositivos conectados en cada red, se comprobó que esto era una opción no escalable y poco flexible. En este punto se comenzó a utilizar *scripts* para la reconfiguración de las redes, produciéndose muchos errores de configuración[13]. Con SDN se busca solventar estos problemas separando ambas capas. El objetivo principal es no predefinir de manera tan clara la red, y poder monitorizar y gestionar la red sin la necesidad de regirse bajo unos dispositivos de red concretos o propietarios. Para ello, se usa una arquitectura estructurada en base a la definición de la red, y el reparto de los recursos vía software gracias a la virtualización de redes [14].
- ***Network Slicing* [10].** Esta técnica consiste en definir diferentes redes virtuales o capas en función del criterio que se considere, y a cada una de ellas se destinará una cantidad de recursos diferentes según sus necesidades. Esto permite, generar múltiples redes personalizables partiendo de la red física desplegada, aportando flexibilidad a la hora de repartir los recursos de red según la demanda de cada capa.
- ***Service Function Chaining* (SFC).** El cambio de las redes tradicionales a las redes definidas por software y virtualizadas supone un esfuerzo para los equipos IT de las empresas [11]. Por ello, el objetivo de SFC es facilitar esta transición de manera dinámica. Realmente, SFC es un concepto muy parecido a NFV. La principal diferencia entre ambos radica en que SFC soluciona los problemas que pueden surgir de proveer un servicio que

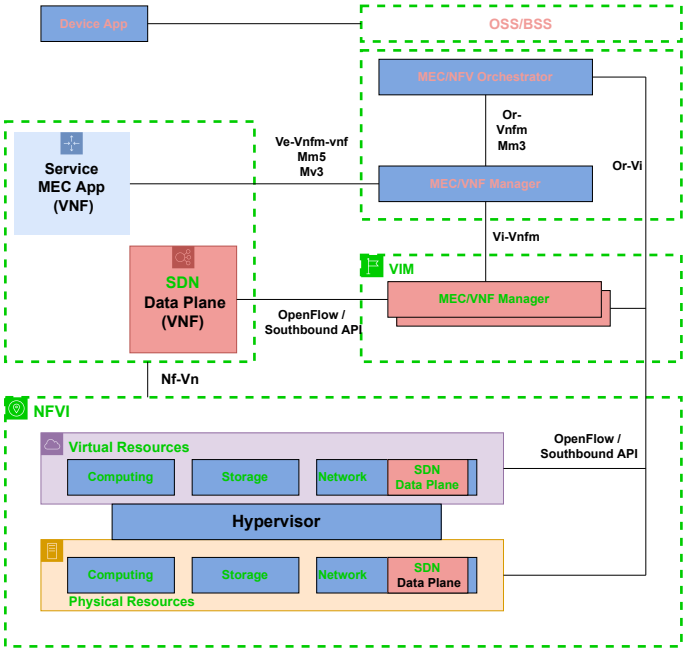


Figura 1. Arquitectura de referencia de MEC [10]

se encuentra dentro de una cadena de servicios en un dispositivo. Por tanto, SFC se ocupa de gestionar que servicios se están ejecutando en el dispositivo de red y cuáles pueden ser usados [14].

En la Figura 1 se puede observar un ejemplo de implementación de una arquitectura MEC con los servicios básicos de NFV y SDN que hemos visto. En ella se puede observar la comunicación entre el controlador de SDN y el orquestador NFV.

III. Estado del Arte

Existen algunos trabajos interesantes orientados a la emulación en entornos IIoT. En esta sección se realiza un breve estudio de los más destacados.

El emulador IIoT Testbed [15] permite la emulación de aplicaciones IIoT basadas en el *middleware Data Distribution Service* (DDS), permitiendo a los usuarios la gestión de procesos, en los cuales se pueden modificar múltiples parámetros, como por ejemplo *Quality of Service* (QoS). La gestión y creación de nuevos procesos IIoT se realiza a través de una interfaz gráfica web, facilitando su uso. Sin embargo, se trata de un emulador muy limitado en cuanto a funcionalidad y flexibilidad, ya que se encuentra centrado en la gestión de los datos de los procesos. Por esto, no es posible usarlo en nuestro entorno al no permitir introducir nuevos procesos y nuevas aplicaciones con diferentes protocolos.

Fogify [16] establece un entorno de desarrollo y despliegue de aplicaciones basadas en *Fog Computing* o MEC. Este emulador permite la creación y definición de redes, el despliegue de múltiples nodos *Edge* y la implementación de aplicaciones IoT. Además, también permite el análisis de prestaciones de los nodos desplegados y de la propia red, para comprobar diferentes fallos o rendimientos de las aplicaciones. Por contra, para poder usar Fogify se exige un alto dominio tanto de la herramienta, como de la base sobre la que se fundamenta su despliegue: *Docker Swarm*. Por otro lado, esta herramienta pueda quedar sin soporte en el corto plazo. Por último, se encuentran dificultades a la hora de desarrollar aplicaciones basadas en protocolos OT, imposibilitando la creación de un entorno IIoT heterogéneo.

Fogbed [17] utiliza conjuntamente las herramientas Containernet [18] y Maxinet [19], que derivan de Mininet, para permitir a los usuarios diseñar y desplegar las topologías que deseen. Al basarse en el lenguaje de programación Python, con un simple *script* es posible desplegar una topología cuyos nodos finales serán contenedores, lo que permite una gran flexibilidad a la hora de desplegar diferentes aplicaciones y servicios. El principal problema que encontramos con Fogbed, es que no se incluye la conectividad móvil que es tan característica en los entornos MEC. Además, al usar la versión por defecto de Containernet no se permite la creación de redundancia en la topología MEC. Esto puede generar posibles fallos de conectividad a la hora de incrementar la complejidad de los despliegues y aplicaciones.

El último emulador de este tipo de redes encontrado es openLEON [8]. Este emulador solventa los problemas que se mencionan de Fogbed, al implementar un controlador que permite el uso de *spanning-tree* en la topología, permitiendo la redundancia en la topología. Además, implementa en la topología desplegada con Containernet el emulador srsLTE [20] que junto con el *hardware* apropiado permite la conectividad LTE con toda la topología. Por esto, se considera openLEON como uno de los emuladores más apropiados para ser usado en el despliegue de entornos IIoT. Sin embargo, openLEON está centrado en el despliegue de la parte MEC, y no cuenta con servicios ni aplicaciones IIoT. Esto hace que no se puedan realizar experimentaciones y pruebas de seguridad en entornos de estas características. Por tanto, en este trabajo se considera usar openLEON como base de nuestro emulador para el despliegue de la topología MEC. El emulador propuesto, MECInOT, integra esta parte, junto con el despliegue de las topologías industriales con sus correspondientes dispositivos y servicios. Esto supone un aumento de la funcionalidad que no podemos encontrar en el resto de trabajos mencionados.

IV. Descripción de MECInOT

A lo largo de esta sección se detallan los aspectos de diseño y de metodología que se han tenido en cuenta a la hora de elaborar este emulador.

IV-A. Diseño

A la hora de plantear el diseño sobre el que se desarrolla MECInOT se han considerado varios puntos que deben ser abordados:

- **Emulación realista de entornos físicos.** Permitiendo realizar pruebas y obtener datos de una manera análoga a realizarlas sobre una topología real, con la ventaja de no necesitar los dispositivos reales para llevarlas a cabo.
- **Flexibilidad a la hora de elaborar diferentes escenarios en función de las necesidades de las investigaciones.** Gracias al uso de la virtualización de dispositivos mediante contenedores, somos capaces de modificar el número de dispositivos, su funcionamiento, o incluso la realización de múltiples subredes destinadas a una función específica.

- **Facilidad en la inclusión de dispositivos reales en los escenarios de prueba.** En función de la naturaleza de la experimentación, es posible que sea interesante realizar una evaluación de prestaciones en las que se incluyan dispositivos reales. En esta evaluación se pueden obtener métricas como colisiones, retrasos en los paquetes, rendimientos, etc. Es por ello, que gracias a la virtualización sobre la que se basa el emulador, es posible incluir dentro de las redes que despliega dispositivos físicos de red, como *IoT gateway*.
- **Enfoque para pruebas en ciberseguridad.** El diseño de este emulador se ha centrado en posibilitar la realización de pruebas de ciberseguridad en entornos IIoT. Para ello, se proporciona una serie de *scripts* y un contenedor atacante que permite a los usuarios, de forma sencilla, lanzar diferentes tipos de ataques, y comprobar su impacto en el escenario a estudio.

IV-B. Metodología de despliegue del emulador

En un escenario real de industria 4.0, es necesario el despliegue de cada dispositivo real en la red industrial o empresarial, y proporcionar una configuración de red que cubre las necesidades especificas del escenario. En el caso del despliegue de MECInOT, el primer paso es la virtualización de la máquina encargada de la creación de la red interna industrial, y de la máquina que se ocupará del despliegue de la arquitectura MEC. A continuación, se debe definir la comunicación entre la red interna industrial, y la topología MEC. Para que ésta sea posible, tanto la máquina real, como las máquinas virtuales que gestionan la red IIoT y la arquitectura MEC deben tener una dirección IP perteneciente a la misma subred, configurando el adaptador de red de las máquinas virtuales en modo puente. Además, es necesario la realización de una configuración de *routing* dentro de la máquina virtual que gestiona la red industrial, que nos permita comunicar ambas subredes.

A continuación, se deben establecer las direcciones de subred utilizadas para la red IIoT y la que contiene la arquitectura MEC. En la configuración del emulador se ha establecido la dirección 172.19.0.0/16 para la red industrial. En ella se encontrarán los dispositivos IIoT virtualizados en contenedores con la herramienta Docker-Compose que nos facilitará el rápido despliegue y modificación de los escenarios. En el caso de la red usada para el despliegue de la arquitectura MEC, se ha dejado la que ofrece openLEON en su configuración base, que es la dirección privada 10.0.0.0/12. Se comprueba que existe conexión entre los *hosts* de la red IIoT, y los de la red MEC.

El último paso consistirá en el despliegue de una arquitectura IIoT sobre la que se desea realizar las pruebas. Esta arquitectura, estará compuesta por un conjunto de dispositivos IIoT, ya implementados en los escenarios de prueba que proporciona el emulador, y por las topologías de redes industriales que serán descritas en el siguiente apartado.

IV-C. Topología industrial

Con la finalidad de poder emular una red de una industria 4.0, en MECInOT se ha implementado una red empresarial única haciendo uso de la interfaz que crea Docker-Compose para levantar los diferentes contenedores. Por otro lado, también se permite definir diferentes subredes para cada uno de los protocolos de comunicación que podemos encontrar en un escenario industrial, y que se van a definir a continuación:

Internet of Things. En el emulador se han incluido aplicaciones con los protocolos más utilizados en el ámbito del IoT: MQTT (*MQ Telemetry Transport*), CoAP (*Constrained Application Protocol*) y AMQP (*Advanced Message Queuing Protocol*) [21], [22].

Operational Technologies. Los protocolos que se han agregado al emulador son aquellas versiones industriales que emplean el protocolo de transporte TCP: Modbus/TCP, que es el estándar industrial más utilizado, S7 que es un protocolo de comunicación propietario usado por los PLC (*Programmable Logic Controller*) de Siemens, y OPC UA (*Open Platform Communications United Architecture*), el cual está considerado como uno de los protocolos que permiten una convergencia entre protocolos IT y OT [22].

Information Technologies. En este grupo de protocolos ubicamos aquellos que son los más utilizados por los usuarios de la red de Internet, como pueden ser el el HTTP (*Hypertext Transfer Protocol*) [23].

Como se ha comento previamente, nuestro emulador está diseñado para una evaluación de seguridad de redes IIoT. Es por ello, que debemos incluir un conjunto de herramientas que permitan de forma sencilla ejecutar ataques en entornos industriales, y evaluar su impacto. En este punto se ha incluido un nodo que utiliza como base la distribución Kali Linux para usarla en el análisis de seguridad.

De cara a evaluar el impacto de un posible ataque a la infraestructura o a los datos de ésta, se ha considera incluir los siguientes tipos de ataques:

- **Ataque de manipulación de paquetes.** Para este caso, el nodo atacante debe realizar un ataque *Man in The Middle* para capturar los paquetes a modificar. Se modificarán los datos incluidos en los paquetes transmitidos en la red IIoT. El resto de campos que componen los paquetes no son modificados.
- **Ataque de fuerza bruta.** Se ha implementado un *script* que lleva a cabo un ataque de diccionario sobre un login de un servidor web.
- ***Payload* en trama HTTP.** Con el objetivo de tratar de aprovecharnos de una vulnerabilidad tipo *Shellshock* [24] de un servidor web, se ha automatizado con un *script* el lanzamiento de una trama HTTP modificada con un *payload* en el campo *User-Agent*.
- **Escaneo de red.** Se ha preparado un *script* con el que se puede realizar de manera automatizada y sucesiva un escaneo sobre los dispositivos de la red. Este *script* de ataque también permite el lanzamiento de diferentes tipos de escaneos en una misma ejecución.
- **Ataques de denegación de servicio.** Se han implementado métodos de denegación de servicio tradicionales como los *ping of the dead* [25], y los basados en inundación o saturación de los puertos con el lanzamiento masivo de tramas TCP. Además, se han adaptados estos ataques en función del protocolo objetivo. En concreto, para el protocolo AMQP, se realiza la inclusión de dispositivos falsos con el fin de saturar la cola de mensajes, y que

un dispositivo legitimo no llegue a establecer comunicación. Por otro lado, aprovechando que CoAP se trata de un protocolo que utiliza UDP como protocolo de capa de transporte, se ha implementado un ataque de amplificación de UDP, utilizando uno de los métodos del servidor CoAP para denegar el servicio a algunos de los dispositivos de la red. Este último ataque consiste en utilizar un paquete UDP muy ligero, y usurpar la IP del dispositivos que se quiera atacar. De este modo podemos aprovecharnos del funcionamiento del protocolo UDP para que envíe un paquete mayor del enviado por el atacante.

- **Ataques de escaneo de dispositivos.** Este ataque trata de aprovecharse de una configuración básica de un *broker* de MQTT para obtener información adicional del mismo. Esta tarea se encuentra automatizada con un *script*, con lo que es posible obtener información de los dispositivos y los *topics* de comunicación.

IV-D. Topologia MEC

Como ya se ha comentado anteriormente, la topología MEC es desplegada gracias al emulador openLEON. En él, se puede diferenciar una red de centro de datos, y otra con conectividad móvil:

- **Centro de datos.** openLEON implementa esta parte de la topología con el emulador Containernet [18], que es una expansión del emulador Mininet [26], permitiendo la creación de topologías cuyos *hosts* están implementados mediante contenedores. Por otro lado, el esquema de la topología está basado en un centro de datos de *3-Tier*, lo cual hace que nos encontremos una topología de red jerárquica de 3 niveles. En dicha topología aparecen 2 *core switches*, y 2 *switches* de agregación por cada uno de ellos. Además, la arquitectura se compone de un total de 64 hosts conectados entre 8 switches que se encuentran en el nivel *Top of Rack*. Esta estructura ofrece redundancia de conexiones entre los distintos elementos con el fin de evitar puntos de fallo único en la red. Por ello, es de interés la implementación del protocolo *spanning-tree* en los dispositivos de red. Debido a que el controlador SDN por defecto de Mininet no lo soporta, los desarrolladores de openLEON decidieron utilizar el controlador RYU, que es un módulo del lenguaje de programación Python que soporta el protocolo OpenFlow para su implementación en la topología.
- **Comunicación móvil.** La parte que distingue a openLEON de otros emuladores estudiados es el soporte en las comunicaciones móviles como LTE. OpenLEON implementa esta parte a través del emulador srsLTE [20], y necesita para su funcionamiento de una serie de dispositivos hardware reales, como antenas y estaciones LTE. Con este módulo se permite la comunicación entre los dispositivos conectado vía LTE con los nodos del *Edge*.

En la Figura 2 se encuentran representadas todas las partes que conforman el emulador. Se puede observar las diferentes funcionalidades que aporta cada una de ellas al conjunto general de MECInOT.

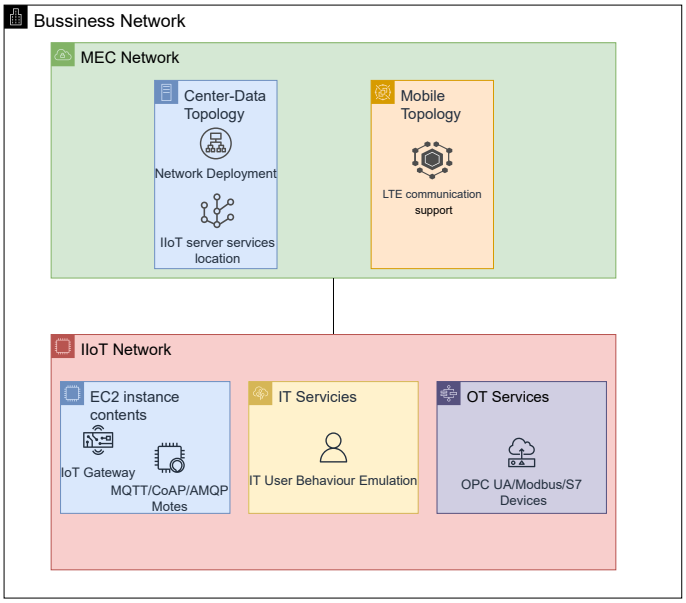


Figura 2. Diagrama lógico del emulador MECInOT.

V. Evaluación del escenario

En esta sección se muestran algunos ejemplos de escenarios que pueden ser desplegados en MECInOT, explicando su funcionamiento. Por otro lado, se evalúa el consumo de recursos de la máquina donde corre el emulador sobre los escenarios estudiados. Por último, se comprueba la viabilidad del emulador para la realización de pruebas orientadas a la ciberseguridad en entorno IIoT desplegado.

V-A. Hardware utilizado

Para la ejecución del emulador propuesto se ha utilizado un portátil con una CPU Intel i7-10875H a 2.30GHz con 32GB de RAM, teniendo instalado Windows 10 20H2. Por otro lado se ha seleccionado el hipervisor VirtualBox 6.1.16r. Además, se utilizará como hardware adicional una Raspberry Pi Zero 2 W que hará la función de *IoT gateway* en aquellos escenarios que sea necesarios.

V-B. Escenario industrial OT

En este escenario encontraremos exclusivamente los protocolos industriales que ya han sido mencionados anteriormente. A continuación se describe el reparto de los nodos entre las topologías y se describirá su funcionamiento.

El escenario de estudio del **protocolo OPC UA** está compuesto por un nodo cliente que se encuentra en la red industrial, y un nodo servidor alojado en la topología MEC. El funcionamiento básico de la comunicación entre ambos nodos será la lectura de una cadena de caracteres aleatoria que genera el servidor en un tiempo comprendido entre 1 a 9 segundos.

Para el caso del **protocolo S7** se despliegan dos nodos clientes, uno de lectura y otro de escritura, que se localizarán en la red industrial, y el nodo servidor alojado en la topología MEC. El funcionamiento es muy similar al del protocolo OPC UA. El cliente escritor modifica el valor de una cadena alojada en el servidor, y esta es leída por el segundo cliente.

Para el caso del **protocolo ModBus/TCP** se despliegan dos clientes localizados en la red industrial, con las mismas funciones que ya se han mencionado para el protocolo S7, y un servidor alojado en la topología MEC. En este escenario uno de los clientes escribe una lista de valores con valor verdadero o falso de manera alterna en el servidor, y cada cierto tiempo el segundo cliente lee estos valores almacenados.

Además, se despliega un nodo atacante en la red junto con el resto de dispositivos. Este nodo, descrito en la sección previa, cuenta con herramientas de escaneo y de aprovechamiento de vulnerabilidades. Además posee una serie de *scripts* enfocados para la realización de ataques.

V-C. Escenario fábrica 4.0

Se trata de una expansión del escenario anterior en el que se introduce la comunicación entre dispositivos IoT y de usuarios IT. Esto permite emular el compartimiento que se puede encontrar en una fábrica 4.0 y que se encuentra perfectamente conectada a diferentes servicios de red. En el caso de los protocolos IoT se diferenciará su comportamiento, pues su tráfico va dirigido a un *IoT gateway* localizado en la red empresarial. Al igual que en el escenario anterior, también se ha introducido un nodo que emula a un posible atacante en la red.

Con el **protocolo MQTT**, se ha tratado emular que la compañía tiene una serie de sensores de temperatura en diferentes puntos de la red industrial y que emitirá la información de temperatura en grados Celsius cada 3 segundos. La función del *IoT gateway* es simplemente implementar el *MQTT Broker* para establecer la comunicación con un subscriptor que se encuentra en uno de los nodos *Edge* de la topología MEC.

En el caso del **protocolo CoAP**, disponemos de un cliente localizado en la red industrial que escribe mensajes en el *IoT gateway* y que serán leídos por otro cliente alojado en la topología MEC.

Para el **protocolo AMQP**, se despliega un cliente que publica en la cola RabbitMQP alojada en el *IoT gateway* un número aleatorio entre el 33 y el 126 y que será recibido por un cliente alojado en la topología MEC.

Por parte de los **protocolos IT**, se han incluido una serie de servidores HTTP y HTTPS, siendo uno de ellos un login habitual de usuarios y un servidor *streaming* de contenido multimedia, tratando de simular la obtención de imágenes de un vídeo. Por parte de los clientes, que se encuentran exclusivamente en la red industrial, se han creado unos *scripts* que periódicamente consultarán y realizarán peticiones a los distintos servidores.

Un esquema lógico de la comunicación entre las partes de la topología descrita se puede ver en la Figura 3. Donde se puede diferenciar como se destina cada tipo de tráfico por los diferentes elementos que se han incluido en este escenario.

V-D. Estudio de rendimiento

Una vez descritos algunos de los escenarios que se pueden desplegar con el emulador, se va a realizar un estudio del consumo de recursos que conlleva el despliegue del escenario más complejo, que corresponde al escenario de una fábrica 4.0, debido a la mayor cantidad de contenedores que se deben desplegar y la cantidad de servicios y aplicaciones que se utilizan. La métrica que vamos a utilizar es el porcentaje de

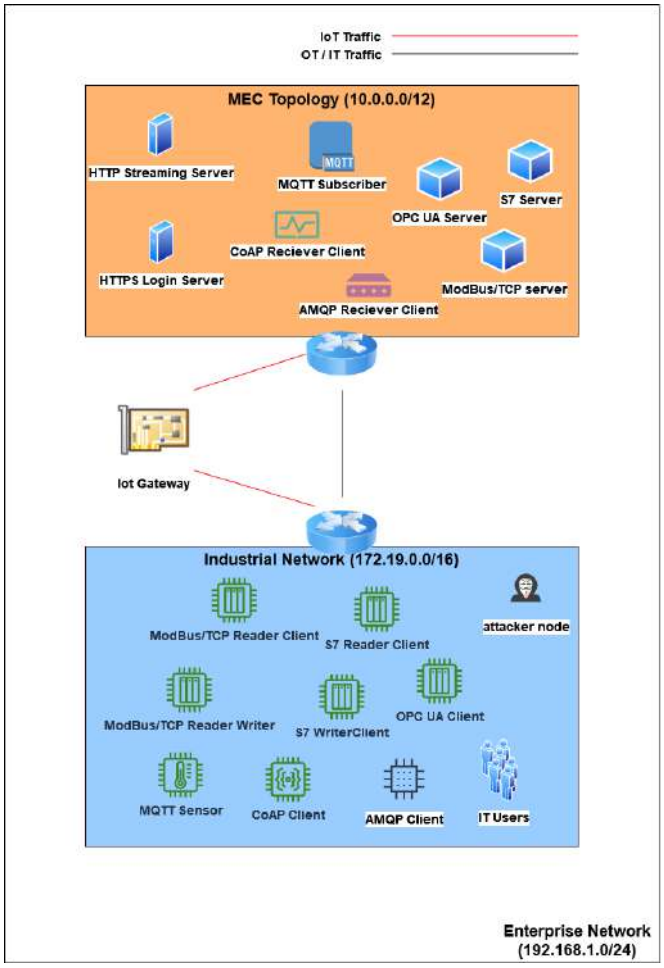


Figura 3. Esquema de red del escenario de una fábrica 4.0.

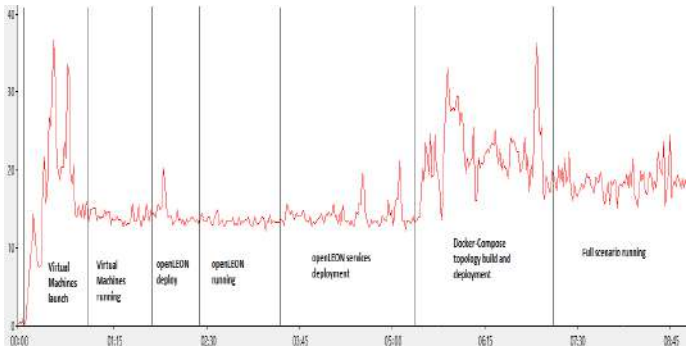


Figura 4. Evolución del consumo de CPU con el despliegue del escenario completo.

uso total del procesador de la máquina anfitriona con Windows 10 y utilizaremos la herramienta de Monitor de Rendimiento que proporciona el mismo sistema operativo para la obtención de los datos.

En la figura 4, se puede observar la evolución del consumo de CPU a lo largo de los 10 minutos que dura la ejecución del escenario. Podemos observar desde el coste que supone el levantamiento de las máquinas virtuales que contienen las dos partes del emulador hasta el impacto que tiene la ejecución en el sistema. También se aprecian los aumentos de carga según se despliegan las diferentes partes de los emuladores, destacando el pico de consumo cuando se hace la construcción

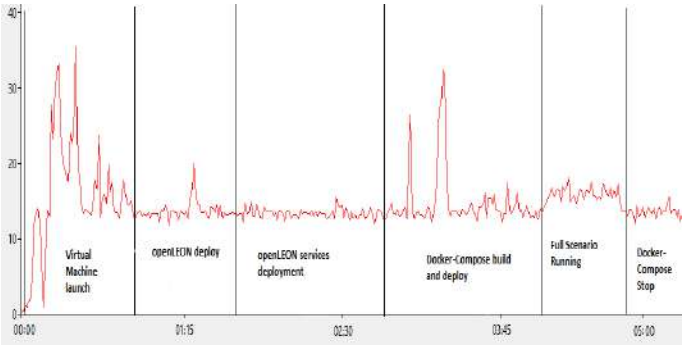


Figura 5. Evolución del consumo de CPU con el despliegue del escenario solo con protocolos OT.

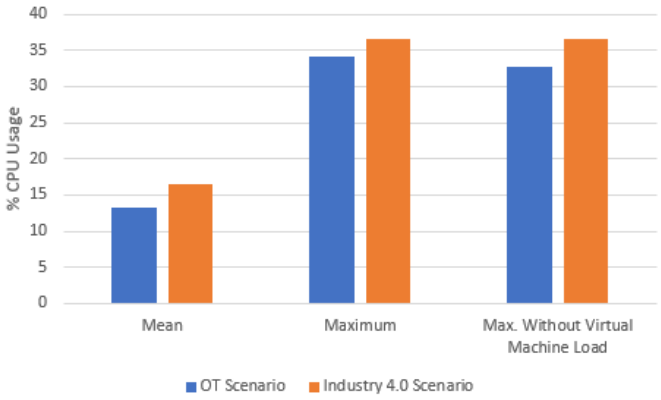


Figura 6. Gráfica comparativa del uso de CPU entre ambos escenarios.

de las imágenes de los contenedores y su despliegue.

Para comprobar la flexibilidad y como escalaría el consumo de recursos en función de un escenario con mayor complejidad, se realiza una comparativa con el despliegue del escenario que únicamente tiene en cuenta las aplicaciones relacionadas con los protocolos OT. Lo que implica que se despliega un escenario más simple que el que se utilizó para el primer experimento. La Figura 5 muestra la evolución de la ejecución de este único escenario durante una ejecución de 5 minutos, en la que también se muestra el impacto del lanzamiento de ataques a través del nodo atacante y que se ha reflejado en dicha figura.

Tal y como se muestra en la Figura 6, podemos comprobar cómo según la complejidad del escenario los recursos necesarios son mayores. Sin embargo, teniendo en cuenta el uso medio de CPU en ambos casos, podemos ver que la diferencia es únicamente del alrededor del 2 %, por lo que sería posible seguir incrementando el número de dispositivos virtualizados y el número de aplicaciones, hasta llegar a alcanzar altas tasas de utilización del procesador de la máquina. Otro apunte que podemos observar es que el pico de mayor utilización del procesador se produce durante el despliegue de las máquinas virtuales. Por la parte, en el escenario de industria 4.0 se observa que el mayor pico de consumo no viene dado por el despliegue de las maquinas virtuales, sino que viene asociado por la construcción y despliegue de las imágenes y sus correspondientes contenedores. Este experimento demuestra la viabilidad de desplegar los diferentes escenarios en un sistema con unos recursos limitados.

```
s7_client_writer_1      | [*] Message send: pxCbpppfKvyNaoTqbIsD
s7_client_writer_1      | [*] Message send: rGAIJVhgSZbnikwoLTlJ
s7_client_writer_1      | [*] Message send: VrDqAiCARQCdxAJrHray
s7_client_writer_1      | [*] Message send: aQjhdJixzSRZeKBHDSsL
s7_client_writer_1      | [*] Message send: wMeCoFEtJfMaVupHZxVv
```

Figura 7. Información enviada por el cliente escritor.

```
s7_client_reader_1      | 53435566844169441025
s7_client_reader_1      | 77410953794301182351
s7_client_reader_1      | 55736648159120147726
s7_client_reader_1      | 20972574760450636806
s7_client_reader_1      | 23851989352500968793
```

Figura 8. Datos recibidos por el cliente lector.

V-E. Análisis de Seguridad

Tras analizar el rendimiento del emulador, se va a realizar una prueba de concepto enfocada a la realización de un ciberataques, en concreto de un intento de modificación de paquetes del escenario de OT.

Primero se debe de acceder al nodo atacante del escenario y utilizar la herramienta *arpspoof* para poder capturar el tráfico que envíe o reciba el nodo que se desea, en esta prueba nuestro objetivo va a ser el cliente lector del protocolo S7.

Posteriormente, debemos de lanzar el *script* de manipulación de paquetes, que se ocupará de modificar los datos de los paquetes que lee el cliente del servidor. En la Figura 7 y Figura 8 se muestra como el cliente escritores envía cadenas de caracteres, mientras que el lector recibe del servidor números enteros, esto implica que nuestro ataque de manipulación está surgiendo efecto y estamos obteniendo el comportamiento esperado.

VI. Conclusiones

Este artículo se presenta el emulador MECInOT para de redes IIoT con soporte MEC; se presenta su funcionalidad, estructura y su potencial uso en la emulación de entornos industriales y, sobre todo, para el análisis de seguridad de este tipo de infraestructuras. Se han presentado las partes y potenciales usos de la herramienta propuesta y, se ha evaluado desde un punto de vista de aplicación y como herramienta para realizar auditorías de seguridad. Se ha demostrado que el emulador es escalable y permite emular topologías realmente complejas en ordenadores personales, gracias al uso de la tecnología de contenedores y virtualización. Finalmente, se ha presentado como se pueden desplegar ataques o intrusiones en el emulador y, se abre la puerta a la inclusión e investigación de técnicas de detección de intrusiones o amenazas y de nuevas aplicaciones.

VII. Trabajo Futuro

En este apartado se comentará algunos aspectos a mejorar del emulador y algún campo donde se podría aprovechar su uso.

- **Optimización en la imagen de los contenedores.** Reducción del tamaño de los contenedores para reducir el tiempo y consumo empleado en el despliegue de la topología.
- **Unificación del emulador en una única máquina virtual.** Actualmente el emulador utiliza dos máquinas virtuales interconectadas a través de la red doméstica,

conllevando un uso de recursos que se podrían ahorrar si se llegase a utilizar únicamente una máquina virtual.

- **Mejorar la administración y gestión de los contenedores.** A medida que los escenarios se vayan ampliando y aumentando en complejidad, la estabilidad de los mismos puede ser comprometida si no se usa una plataforma de gestión de los dispositivos virtualizados. Por ello, se puede realizar una migración del uso de Docker Compose a un *cluster* de Kubernetes a la hora del despliegue del escenario. Esto a su vez, nos permite automatizar algunos pasos en los que se ve implicado el usuario con Docker-Compose, como puede ser el reinicio y levantamiento de un contenedor en caso de fallo en la red o el despliegue de nuevos contenedores en determinadas situaciones.
- **Implementación de nuevas funcionalidades en la red.** Al tratarse de un emulador de despliegues de redes orientado a la ciberseguridad, puede ser de utilidad aplicarlo para la implementación de *firewalls* o IDS (*Intrusion Detection System*) para probar diferentes formas de reducir, anular o detectar los posibles daños que puede provocar un atacante en cada situación, aprovechándonos de los ataques ya implementados en el emulador.
- **Introducción de nuevas aplicaciones.** Gracias a la topología desplegada con openLEON, se podrían incluir aplicaciones basadas en las tecnologías emergentes orientadas a mejorar la seguridad y privacidad en MEC. Por ejemplo, los contratos inteligentes basados en *blockchain*.
- **Inclusión de ataques específicos.** Los ataques incluidos en MECInOT están diseñados para suplir los ataques genéricos, que se encuentran en los entornos IT con un enfoque a dispositivos OT. Sin embargo, es necesario incluir las herramientas necesarias para poder explotar las vulnerabilidades, que se pueden encontrar en los protocolos de red OT y en dichos dispositivos. Además, también se debe de enfocar dichos ataques al aprovechamiento de las vulnerabilidades encontradas en los entornos vitualizados y en el paradigma MEC [27].

Agradecimientos

Este trabajo ha sido realizado bajo la financiación europea del Fondo Social Europeo Plus (FSE+) con el contrato predoctoral en la Universidad de Castilla La-Mancha en el proyecto con identificador PI001482, por el Ministerio de Ciencia, Innovación y Universidades y los Fondos FEDER de la Unión Europea, referencia RTI2018-098156-B-C52 y la beca FPU 17/02007. Finalmente, también ha sido financiado por la JCCM [proyecto SB-PLY/17/180501/ 000353] y [proyecto SBPLY/21/180501/000195].

Referencias

[1] D. Ivanov, C. Tang, A. Dolgui, D. Battini, and A. Das, "Researchers' perspectives on industry 4.0: multi-disciplinary analysis and opportunities for operations management," *International Journal of Production Research*, pp. 1–24, 08 2020.

[2] P. K. R. Maddikunta, Q.-V. Pham, P. B, N. Deepa, K. Dev, T. R. Gadekallu, R. Ruby, and M. Liyanage, "Industry 5.0: A survey on enabling technologies and potential applications," *Journal of Industrial Information Integration*, vol. 26, p. 100257, 2022.

[3] L. L. Dhirani, E. Armstrong, and T. Newe, "Industrial iot, cyber threats, and standards landscape: Evaluation and roadmap," *Sensors*, vol. 21, no. 11, p. 3901, 2021.

[4] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2018–2023," *Cisco White Paper*, pp. 1–36, 2020.

[5] A. T. Atieh, "The next generation cloud technologies: A review on distributed cloud, fog and edge computing and their opportunities and challenges," *ResearchBerg Review of Science and Technology*, vol. 1, no. 1, p. 1–15, Oct. 2021.

[6] T. K. Rodrigues, J. Liu, and N. Kato, "Application of cybertwin for offloading in mobile multiaccess edge computing for 6g networks," *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16 231–16 242, 2021.

[7] D. Borsatti, G. Davoli, W. Cerroni, and C. Raffaelli, "Enabling industrial iot as a service with multi-access edge computing," *IEEE Communications Magazine*, vol. 59, no. 8, pp. 21–27, 2021.

[8] C. Fiandrino, A. Pizarro, P. Mateo, C. Andrés Ramiro, N. Ludant, and J. Widmer, "openleon: An end-to-end emulation platform from the edge data center to the mobile user," *Computer Communications*, vol. 148, pp. 17–26, 12 2019.

[9] C. Alcaraz and J. Lopez, "Digital twin: A comprehensive survey of security threats," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2022.

[10] A. Filali, A. Abouaomar, S. Cherkaoui, A. Kobbane, and M. Guizani, "Multi-access edge computing: A survey," *IEEE Access*, vol. 8, pp. 197 017–197 046, 2020.

[11] M. Liyanage, P. Porambage, and A. Y. Ding, "Five driving forces of multi-access edge computing," *arXiv preprint arXiv:1810.00827*, 2018.

[12] J. Liu, Q. Li, R. Cao, W. Tang, and G. Qiu, "Mininet: An extremely lightweight convolutional neural network for real-time unsupervised monocular depth estimation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, pp. 255–267, 2020.

[13] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[14] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116 974–117 017, 2020.

[15] R. S. Auliva, R.-K. Sheu, D. Liang, and W.-J. Wang, "Iiot testbed: A dds-based emulation tool for industrial iot applications," in *2018 International Conference on System Science and Engineering (ICSSE)*, 2018, pp. 1–4.

[16] S. Moysis, G. Zacharias, T. Demetris, P. George, and D. Marios D., "Fogify: A fog computing emulation framework," in *Proceedings of the 5th ACM/IEEE Symposium on Edge Computing*, ser. SEC '20. New York, NY, USA: Association for Computing Machinery, 2020.

[17] A. Coutinho, F. Greve, C. Prazeres, and J. Cardoso, "Fogbed: A rapid-prototyping emulation environment for fog computing," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.

[18] M. Peuster, H. Karl, and S. van Rossem, "Medicine: Rapid prototyping of production-ready network services in multi-pop environments," in *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2016, pp. 148–153.

[19] P. Wette, M. Dräxler, and A. Schwabe, "Maxinet: Distributed emulation of software-defined networks," *2014 IFIP Networking Conference*, pp. 1–9, 2014.

[20] I. Gomez-Miguelez, A. Garcia-Saavedra, P. Sutton, P. Serrano, C. Cano, and D. Leith, "srslte: an open-source platform for lte evolution and experimentation," 10 2016, pp. 25–32.

[21] B. Mishra and A. Kertesz, "The use of mqtt in m2m and iot systems: A survey," *IEEE Access*, vol. 8, pp. 201 071–201 086, 2020.

[22] D. Silva, L. I. Carvalho, J. Soares, and R. C. Sofia, "A performance analysis of internet of things networking protocols: Evaluating mqtt, coap, opc ua," *Applied Sciences*, vol. 11, no. 11, p. 4879, 2021.

[23] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over http dataset," in *Proceedings of the 3rd multimedia systems conference*, 2012, pp. 89–94.

[24] C. Mary, "Shellshock attack on linux systems-bash," *International Research Journal of Engineering and Technology*, vol. 2, no. 8, pp. 1322–1325, 2015.

[25] R. Ahmad and I. Alsmadi, "Machine learning approaches to iot security: A systematic literature review," *Internet of Things*, vol. 14, p. 100365, 2021.

[26] K. Kaur, J. Singh, and N. S. Ghumman, "Mininet as software defined networking testing platform," in *International Conference on Communication, Computing & Systems (ICCCS)*, 2014, pp. 139–42.

[27] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

An Industrial Control System physical-digital framework applied to connected Wastewater Treatment Process for identifying cyber threats

Álvaro García-García
Área de TIC - Industria 4.0
Fundación Cidaut
alvgar@cidaut.es

Cristian Velasco-Merino
Área de TIC - Industria 4.0
Fundación Cidaut
crivel@cidaut.es

Enrique Rodríguez-Núñez
Área de TIC - Industria 4.0
Fundación Cidaut
enrrod@cidaut.es

***Abstract*—The current growth of cyber threats in Industrial Control Systems (ICS) and critical infrastructures, particularly for legacy environments, has boosted security concerns regarding cyber-physical systems management and critical services exposure. Legacy systems, rather undemanding in terms of cybersecurity, are now exposed to new digital contexts of operation and execution of connected processes, integrating long-run equipment, convergent networks and communication protocols. In order to test and study cyber attacks without risk of real system damaging, for developing cyber defense solutions, diverse testbed methodologies to reproduce real ICSs used in critical infrastructures are discussed in this paper. In this way, a hybrid testbed based on a physical-digital cybersecurity framework, reproducing a legacy Wastewater Treatment Process (WWTP), is presented as a case study. Physical and digital components are combined at real level to study convergent cyber threats, with the focus on the evaluation of attacker activity and characterisation of the impact on specific critical processes, powered by passive monitoring and virtual reality.**

***Index Terms*—Industrial Control System, Cyber security, Cyber-Physical system, Testbed, Cyber threat**

Contribution: *Research in progress*

I. Introduction

Emerging cyber threats pose a challenge to Industrial Control Systems (ICS) security, particularly for legacy systems [1]. These infrastructures, now exposed to new contexts of operation and execution of connected processes, integrate equipment, networks and protocols designed to accomplish both high-performance and critical functionalities. In this way, the term ICS includes a diverse range of vulnerable operational technologies (OT) and components which allow full or partially automated control of industrial facilities [2]. Moreover, legacy systems are rather undemanding in terms of the availability, integrity and confidentiality requirements that security solutions must face against cyber attacks. Thus, the progressive introduction of new Industry 4.0 enabling technologies in the productive ecosystem, such as cyber-physical systems, has therefore meant a change in the strategies with which industrial sectors must be protected in a connected and distributed paradigm [3].

Advanced Persistent Threats (APTs) [4] for ICSs and critical infrastructures, have become one of the main concerns due to the serious impact caused both by intrusions to industrial processes and to the continuity of operations and critical services for the population (energy production, transportation, drinking water production, wastewater treatment processes, health, manufacturing, etc.) [5].

The variability and intensity of the attacks which compromise industrial networks and systems is evolving along with the constant development and expansion of digital technologies. This progress presents new challenges to managers and infrastructures, that are limited by the security practices and restrictions inherited in OT. As a result, new security strategies are needed for new threats. On one hand, evaluating alternate solutions to address industrial attack vectors and, on the other hand, increasing the speed and efficiency in the subsequent development of prevention, defense and response methodologies for critical infrastructures and services. Research into industrial testbeds represents an applied approach for developing cyber defense methods. In addition, new technologies such as digital twins of instrumentation and control systems allow testing the system's resistance to cyber threats carrying out cyber attack tests without risk of real system damaging [6]. Thus, by the physical-digital convergence, the digital twin concept is able to represent an abstraction of industrial systems' reality allowing for multiple interaction levels between processes, systems and humans within the virtual space [7]. It is with this perspective that the development and availability of digital twin frameworks as cyber threat knowledge instruments based on deception, intrusion detection and situational awareness, might usefully be encouraged to foster learning about the risks that the connection between OT and IT spaces entail [8].

As such, this digitised approach presents a low-cost and flexible ICS testbed solution to research in the field of cybersecurity [1], but real devices are more suitable for studying cyber attacks in high fidelity physical environments. Nevertheless, when physical and digital approaches are combined, it is offered an alternative option to implement cyber-physical systems of high reliability, provided that the existing synchronization problem between the real system and its digital twin counterpart can be assured.

In this work, we discuss different ICS testbed solutions to generate knowledge in cyber threats and present a case study of a connected cybersecurity environment to achieve a real physical-digital ICS replication of a Wastewater Treatment Process (WWTP). This research presents the proposed methodology, focused on identifying attacker activity and characterise the impact assessment on specific critical pro-

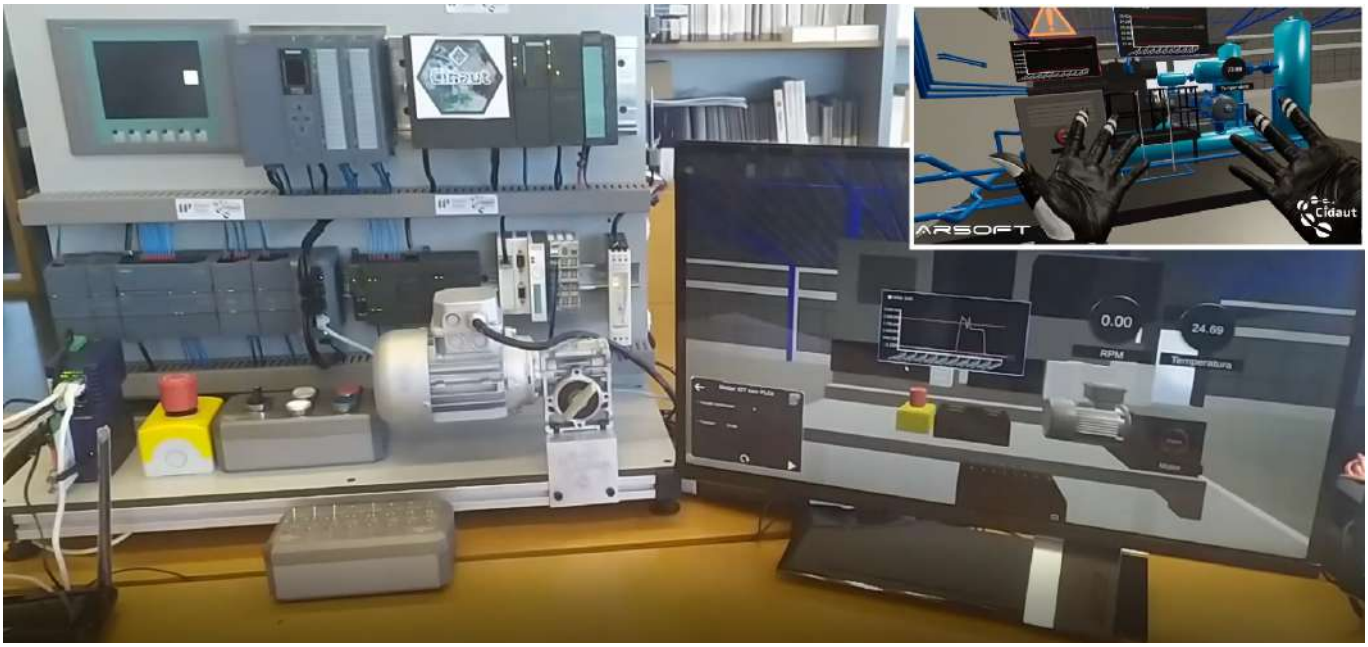


Fig. 1. Connected cybersecurity environment to achieve a real physical-digital ICS replication of a WWTP

cesses. Also, the system architecture is described.

The remaining of the paper is organized as follows. Section 2 considers different testbed methodological approaches. Next, Section 3 describes the case study, system architecture and shows preliminary findings. Finally, Section 4 presents the conclusions derived from the applied research.

II. Methodology

In recent years, many of the industrial cybersecurity efforts have increasingly focused on the characterisation and instruction of testing ICS environments [9], called testbeds, to reproduce real-world systems and critical infrastructures [10]. This section outlines some ICS testbed solutions build on existing literature and presents the proposed physical-digital ICS replication approach.

A. Methodologies used to replicate Industrial Control Systems

Scaled-down replicas of a real ICS, help researchers to acquire knowledge by studying the tactics, techniques, and procedures (TTP) used by a cyberattacker. Nevertheless, this learning is conditioned by the accurate reproduction of the real system, how it works under operational conditions, and the diversity of components of which it is composed. [1] classifies a testbed as Physical, Virtual and Hybrid, whose common OT functional ICS components include: (i) hardware devices such as Programmable Logical Controllers (PLCs), Remote Terminal Units (RTUs), and Intelligent Electronic Devices (IEDs), (ii) electrical and mechanical field devices, such as sensors and actuators, (iii) software applications such as Human Machine Interface (HMI), and (iv) communication devices, gateways and protocols.

With regard to *physical testbeds*, highly realistic ICS environments are characterised using the same real hardware components such as PLCs, RTUs, SCADA, etc., HMI, and network layers. This approach provides researchers with physical measurement instruments and actuators in order to study the response to different cyber threat and specific vulnerabilities. Unlike the latter, *virtual testbeds* are based on software simulations and emulations designed to reproduce virtual components regarding ICS environments. They allow a low-cost solution where virtual machines are common tools used to leverage laboratory studies in order to observe the behavior of different systems exposed to cyber threats. However, *hybrid testbeds* combine both physical parts and software simulations. Since the physical layer of a real ICS will be maintained together with the virtual layer, a good trade-off between accuracy and flexibility allow makes it possible to characterise new cyber-phisical systems and environments.

B. Proposed physical-digital ICS replication approach

This paper presents an hybrid physical-digital methodology in order to characterise an ICS testbed which replicates, as much as possible, a real industrial controlled scenario. Through a connected approach is aimed at providing the convergence between physical processes -related to the target environment- and digital tools to encourage the system characterisation for both deception and detection. In particular, a threefold objective is pursued to generate cyber security knowledge: (i) To provide physical OT approaches which combine PLCs, industrial protocols, actuators and other real process control systems such as SCADA; (ii) To provide network exposed services managed by a passive monitoring digital infrastructure, to collect data which is stored in a Security Information and Event Management (SIEM) solution managing cyber threats, and (iii) To provide a virtual reality system as an interactive visualisation of the digital twin approach.

Some examples of this physical-digital approach are also presented in the literature [1]. SANS Institute developed CyberCity Testbed, a physical representation of an entire city. This testbed mixes emulated components and physical PLCs, routers and controllers in order to test security measures

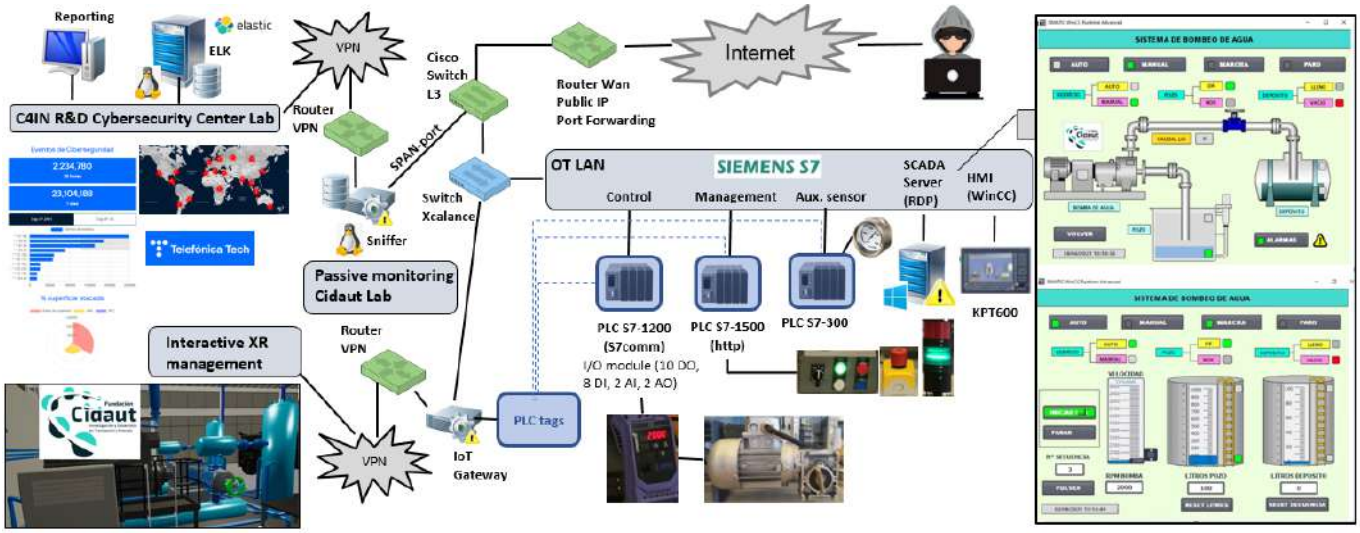


Fig. 2. WWTP testbed modular system architecture

on the ICS field. A different testbed example for ICS was developed by NIST (National Institute of Standards and Technology), to emulate three real-world industrial systems without replicating the entire plant or assembling a complete system. The first of them, the Tennessee-Eastman chemical process [11], has been used as a reference to implement hybrid testbeds through developing virtual simulations [12]. This approach, allows simulating real-world interfaces between control devices, sensors and actuators avoiding to construct an entire plant operation [9].

III. Case Study

In this work we present a connected physical-digital cybersecurity framework (Figure 1), as example of a WWTP infrastructure in critical domains that depend on ICSs.

A. Motivation

There exist previous operational testbeds in the water domain, demonstrating their utility to conduct ICS cybersecurity research [13]. The framework presented in this work is designed to monitor, register and identify the attacker activity from the Internet in real time. In addition, the system allows the digital characterisation of the impact assessment on specific critical water treatment processes. A cyber-physical layer is provided with the digital twin concept as well as augmented reality/virtual reality technologies. The cybersecurity framework proposed for this case study also provide researchers with a private laboratory network, mapping the augmented interfaces of real processes. Through digital and virtual models [6], learning opportunities applied to Industry 4.0 using representation and visualisation layouts [14], are offered.

The design and implementation of the hybrid WWTP testbed has been coordinated sharing information between process experts and automation control engineers to implement the process physical layer, with the aim of making the cyber threats research scenario the most realistic possible. The recognition, detection, analysis and management of events associated with vulnerabilities of the hybrid testbed are supported by passive network monitoring. This study (https://aristeo.elevenlabs.tech/) will be conducted in cooperation with security engineers of C4IN R&D Cybersecurity Center (Telefónica TECH Cyber & Cloud, León. Spain). In addition, all events are registered in a SIEM based on a ELK Stack infrastructure (Elasticsearch, Logstash and Kibana).

B. System architecture

The WWTP testbed system architecture (see Figure 2) is composed of four operational modules: (i) A physical OT module that integrates the ICS components and system's operation, including three legacy Siemens S7 PLCs (S7-1200 for control, S7-1500 for management, and S7-300 for auxiliary sensors), analogical and digital I/O, a water pump, a frequency converter, a temperature sensor, safety panel buttons, a light beacon, an HMI Siemens KPT600 running the process control application, a SCADA server running the Siemens Simatic WinCC software control application on a Windows 10 host, and a dedicated OT network infrastructure exposed to Internet; (ii) A passive monitoring module that integrates a layer 3 Cisco switch, enabling port mirroring to capture and process all exposed network traffic; (iii) An ELK Stack infrastructure, hosted by C4IN Cybersecurity Lab. This layer analyse, classify and map all information regarding registered cyber threats in the WWTP testbed; and (iv) An interactive Virtual Reality layer of the WWTP, connecting both the virtual app and the OT hardware components through an IIoT Gateway inside lab private network. A bidirectional synchronisation to update PLC tags is used.

Different system actuators work in a fully automated way, being controlled by the HMI and Siemens WinCC application. The legacy physical OT infrastructure and networks (TCP/IP and PROFINET) are designed to be exposed for remote management in a vulnerable way. Siemens PLCs are not updated with latest firmware version, and the WAN router forwards remote management protocols such as S7comm and http. In the same way, HMI KPT600 management protocol and the SCADA server RDP protocol are accessible for remote connections.

Fig. 3. Reputation list report view provided by Aristeo (Telefónica TECH Cyber & Cloud)

C. Preliminary results

The cybersecurity framework has proved its value under real attack situations. The most common threat tipologies are active scanning based, making use of the Siemens s7comm+ characteristics (S7 Communication proprietary protocol) to execute a remote exploit; or Microsoft Windows remote desktop protocol based, aimed at compromising the SCADA workstation within the OT network. The testbed captures all the adversary moves, including a considerable amount of industrial-based protocols to find a compatibility via to run exploits or gain remote access. Moreover, Aristeo system (see Figure 3) classifies and scores all the interactions related to security events and threats. All IP address numbers involving threat events targeted on the WWTP testbed, regarding hardware components and devices exposed, are registered: hacking events derived from remote code execution, port scanning, etc. In addition, the IP address number first seen date and its associated dynamic changes, proposes a predictive approach to identify campaign-based attacks and provides the date on which the adversary infrastructure appears.

IV. Conclusions and Future Work

Cyber-physical systems, have boosted legacy security concerns in a new industrial network context regarding the remote management and monitoring of ICSs and critical services exposure. The development of ICS-based cybersecurity research environments, such as testbeds, is intended to expose industrial infrastructures under safe conditions by providing a high realistic scale model. In this way, a hybrid environment is, probably, the most balanced option for developing a real system-wide test in order to research the behavior of cyber threats. Furthermore, the hybrid approach provide researchers with adaptive scenarios to replicate the impact of new attack vectors. In this paper we have presented a hybrid characterisation of a cybersecurity framework, reproducing an exposed WWTP critical infrastructure by legacy physical-digital components combined at real level. The convergent system architecture also encourages the study of cyber threats, powered by passive monitoring and virtual reality. In the future, machine learning algorithms applied to proactive threat detection, are expected to improve cyber defense trends.

Acknowledgements

We thank ARSoft (Salamanca, Spain) for providing EyeFlow resources to develop the virtual and augmented reality systems, and C4IN R&D Cibersecurity Center of Telefónica TECH Cyber & Cloud (León, Spain) for providing reporting resources and the cloud management infrastructure.

References

[1] M. Conti, D. Donadel, and F. Turrin, "A Survey on Industrial Control System Testbeds and Datasets for Security Research," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 4, pp. 2248–2294, 2021.

[2] D. Sullivan, E. Luiijf, and E. J. M. Colbert, *Components of Industrial Control Systems*. Cham: Springer International Publishing, 2016, pp. 15–28. [Online]. Available: https://doi.org/10.1007/978-3-319-32125-7_2

[3] J. E. Rubio, C. Alcaraz, R. Roman, and J. Lopez, "Current cyber-defense trends in industrial control systems," *Computers and Security*, vol. 87, p. 101561, 2019. [Online]. Available: https://doi.org/10.1016/j.cose.2019.06.015

[4] S. Quintero-Bonilla and A. M. del Rey, "A new proposal on the advanced persistent threat: A survey," *Applied Sciences (Switzerland)*, vol. 10, no. 11, 2020.

[5] L. Cazorla, C. Alcaraz, and J. Lopez, "Cyber Stealth Attacks in Critical Information Infrastructures," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1778–1792, 2018.

[6] K. Semenkov, V. Promyslov, A. Poletykin, and N. Mengazetdinov, "Validation of complex control systems with heterogeneous digital models in industry 4.0 framework†," *Machines*, vol. 9, no. 3, 2021.

[7] C. Semeraro, M. Lezoche, H. Panetto, and M. Dassisti, "Digital twin paradigm: A systematic literature review," *Computers in Industry*, vol. 130, 2021.

[8] C. Alcaraz and J. Lopez, "Digital twin: A comprehensive survey of security threats," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2022.

[9] R. Candell, D. M. Anand, and K. Stouffer, "A cybersecurity testbed for industrial control systems," *ISA Process Control and Safety Symposium 2014, PCS 2014*, pp. 873–888, 2014.

[10] B. Green, R. Derbyshire, W. Knowles, J. Boorman, P. Ciholas, D. Prince, and D. Hutchison, "ICS Testbed Tetris: Practical Building Blocks towards a Cyber Security Resource," *CSET 2020 - 13th USENIX Workshop on Cyber Security Experimentation and Test, co-located with USENIX Security 2020*, 2020.

[11] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers and Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993.

[12] D. Chen, Y. Peng, and H. Wang, "Development of a Testbed for Process Control System Cybersecurity Research," *Proceedings of the 3rd International Conference on Electric and Electronics*, vol. 69, no. Eeic, pp. 158–161, 2013.

[13] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," *2016 International Workshop on Cyber-physical Systems for Smart Water Networks, CySWater 2016*, no. Figure 1, pp. 31–36, 2016.

[14] N. Tvenge, O. Ogorodnyk, N. P. Østbø, and K. Martinsen, "Added value of a virtual approach to simulation-based learning in a manufacturing learning factory," *Procedia CIRP*, vol. 88, pp. 36–41, 2020. [Online]. Available: https://doi.org/10.1016/j.procir.2020.05.007

Metodología y herramientas para análisis y evaluación de seguridad frente a ataques de radiofrecuencia en vehículos

Roberto Gesteira Miñarro
Instituto de Investigación Tecnológica - ICAI
C/ Alberto Aguilera 25, Madrid
rocky@alu.icai.comillas.edu

Juan Blázquez Sánchez
Instituto de Investigación Tecnológica - ICAI
C/ Alberto Aguilera 25, Madrid
201709458@alu.comillas.edu

Gregorio López López
Instituto de Investigación Tecnológica - ICAI
C/ Alberto Aguilera 25, Madrid
gllopez@comillas.edu

Rafael Palacios Hielscher
Instituto de Investigación Tecnológica - ICAI
C/ Alberto Aguilera 25, Madrid
Rafael.Palacios@iit.comillas.edu

Miguel Ángel Blázquez Puras
CESVIMAP
C/ Jorge de Santayana 18, Ávila
BPMIGU1@mapfre.com

Rubén García Fernández
CESVIMAP
C/ Jorge de Santayana 18, Ávila

Enrique Zapico Alonso
CESVIMAP
C/ Jorge de Santayana 18, Ávila
ezapico@cesvimap.com

José María Cancer Aboitiz
CESVIMAP
C/ Jorge de Santayana 18, Ávila
cancera@cesvimap.com

***Resumen*—Los automóviles son sistemas complejos cuyo funcionamiento se basa en millones de líneas de código e incorporan una amplia gama de tecnologías de comunicación, así como de dispositivos y aplicaciones de terceros. Por lo tanto, la ciberseguridad es un tema especialmente relevante hoy en día en el sector del automóvil y lo será aún más en los próximos años, en parte debido a la regulación que se planea entre en vigor al respecto. En este contexto es necesario disponer de metodologías y herramientas que permitan evaluar y clasificar los vehículos dependiendo de su nivel de exposición y protección frente a ataques de ciberseguridad. En este artículo presentamos una metodología y un conjunto de herramientas que permiten obtener los mensajes intercambiados entre mandos de radiofrecuencia y vehículos y analizar, evaluar y comparar los protocolos utilizados por diferentes modelos de vehículos desde el punto de vista de la ciberseguridad.**

***Index Terms*—Radiofrecuencia, ciberseguridad, vehículos**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

La ciberseguridad representa actualmente un tema candente en la industria del automóvil que ha comenzado a regularse recientemente (especialmente, desde 2019). Así, el Reglamento UN n.º 155 hará obligatorio que los fabricantes de equipos originales (OEM) mantengan un sistema de gestión de ciberseguridad (CSMS) y el Reglamento UN n.º 156 hará obligatorio que los OEM tengan un sistema certificado de gestión de actualizaciones de *software* (SUMS). Aunque los vehículos no deben cumplir con ninguna homologación de ciberseguridad en este momento, el Reglamento UN No. 155 establece que dicho CSMS debe estar certificado a partir de 2022 para nuevas homologaciones y a partir de 2024 para nuevos registros.

Teniendo en cuenta que una regulación de ciberseguridad similar en otros sectores como la banca ha llevado más de una década, esto representa un desafío sin precedentes para la industria del automóvil que se espera impulse una ola de investigación e innovación en este área.

Dado que en todos los países desarrollados es obligatorio tener un seguro asociado a los vehículos, las compañías de seguros aparecen como un actor clave en este contexto, ya que necesitan estimar los riesgos de ciberseguridad de los vehículos para reflejarlo adecuadamente en las pólizas.

Concretamente, es necesario disponer de metodologías rigurosas que permitan evaluar y clasificar los vehículos dependiendo de su nivel de exposición y protección frente a ataques que permitan: (i) robar el vehículo (ya sea mediante ataques de radiofrecuencia o mediante ataques a aplicaciones que controlen el acceso al vehículo); (ii) robar datos contenidos en el vehículo (siendo especialmente relevantes los de carácter personal); (iii) secuestrar el vehículo; o (iv) conseguir controlarlo de manera remota.

Este artículo se centra en ataques de radiofrecuencia (RF). La mayoría de trabajos que se han llevado a cabo anteriormente en este ámbito se han centrado en intentar abrir o robar el vehículo, trabajando con señales capturadas en formato analógico. El objetivo de este trabajo, en cambio,

es definir una metodología que permita conocer y evaluar el funcionamiento de los mecanismos de apertura y arranque por RF de diferentes marcas de vehículos desde el punto de vista de la ciberseguridad. Para conseguir dicho objetivo es fundamental trabajar con las señales en formato digital, por lo que es necesario desarrollar una serie de herramientas soporte que permitan obtener, analizar y reproducir dichas señales digitales.

El resto del artículo se organiza de la siguiente manera. La sección II revisa brevemente algunos trabajos previos relacionados con ataques a mandos a distancia y a llaves manos libres. La sección III describe los recursos *hardware* y *software* que se han utilizado en este trabajo. En la sección IV se explica la metodología de trabajo propuesta. La sección V presenta los resultados obtenidos hasta la fecha aplicando dicha metodología al análisis del funcionamiento de mandos a distancia RF de diferentes marcas de vehículos. Finalmente, la sección VI resume las principales conclusiones del artículo y las futuras líneas de trabajo.

II. Estado del Arte

Actualmente, la mayoría de los automóviles en circulación utilizan canales de RF para comunicarse con otros dispositivos. Estos canales pueden presentar problemas de seguridad, ya que la comunicación viaja en forma de ondas que se propagan por el aire, de manera que un atacante puede interceptar dichas ondas.

En el caso de los vehículos, la gran mayoría ofrecen la posibilidad de abrir las puertas mediante un mando a distancia. Para ello, el mando emite una señal y el vehículo la recibe. Si esta señal es válida, entonces el vehículo se abre.

En automóviles más modernos, existen otro tipo de llaves que funcionan por proximidad (tienen un menor rango de alcance). El protocolo de comunicación con este tipo de llaves (conocidas como *keyless*) es más complejo en el sentido de que puede ser bidireccional y puede depender de otros canales de comunicación.

II-A. Ataques al mando a distancia

Antiguamente, el mando a distancia enviaba una señal con un código. El vehículo recibía ese código y, si era correcto, entonces se abría. Este código siempre era el mismo, lo cual es un problema.

Existe un ataque conocido como ataque de *replay*, que consiste en capturar la señal del mando y replicarla. En un caso como éste, con que un atacante capture una vez la señal del mando ya le basta para poder abrir el vehículo todas las veces que quiera. Esto es un problema grave ya que permite a un atacante entrar al vehículo sin forzar una cerradura o romper una ventana, lo cual puede desembocar en robo de objetos, robo de información o incluso robo de vehículo si lo consigue arrancar, entre otras consecuencias.

En este caso habría que analizar si el código es totalmente fijo o tiene algún tipo de caducidad con el paso del tiempo.

Para corregir este problema del código único, se implementaron los códigos evolutivos (*rolling codes* en inglés). En este caso, el mando y el vehículo contienen un algoritmo generador de números pseudo-aleatorios (PRNG) inicializado con la misma semilla. De esta manera, el vehículo genera una lista de códigos válidos (por ejemplo, 100 números) y el mando va enviando uno a uno el siguiente número generado por el algoritmo. Así, si el vehículo recibe un código de la lista de códigos válidos, se abrirá. Además, en principio los códigos anteriores al código recibido deberían descartarse y se deberían generar más códigos para tener una lista con la misma cantidad de códigos [1].

En primer lugar, existe una vulnerabilidad intrínseca de denegación de servicio. Si el mando se pulsa un número elevado de veces hasta salirse de la ventana de códigos válidos del vehículo (más de 100 veces, por ejemplo), entonces el mando deja de estar sincronizado con el vehículo y será necesario abrirlo de forma manual o volver a sincronizar el mando.

El ataque de *replay* ya no funciona de manera tan sencilla. Sin embargo, se puede realizar de una manera algo más sofisticada. El código capturado, en este caso, será de un solo uso, pero suficiente para poder abrir el vehículo.

A la hora de capturar un código válido del mando, es necesario que el vehículo no reciba la señal. De lo contrario, el vehículo se abrirá normalmente y el código será descartado. Entonces, es necesario introducir interferencias (mediante una antena que haga de *jammer*) en la banda de trabajo del mando para que el vehículo no reciba el código correctamente.

Paralelamente, con otra antena se deben capturar al menos dos códigos del mando (suponiendo que la víctima pulsa repetidas veces el mando para abrir su vehículo). Y la parte importante es que el vehículo debe abrirse replicando el primer código capturado y no con un código del mando de la víctima (de lo contrario, todos los códigos capturados quedarían invalidados). Así, se obtendría al menos un código válido para poder abrir el vehículo una vez que la víctima ha terminado de utilizarlo. La otra opción es esperar a que la víctima abra y cierre el coche de forma manual [2].

Otra alternativa es capturar una serie de señales, analizarlas para extraer los códigos e intentar hacer ingeniería inversa para obtener el algoritmo del PRNG y la semilla utilizada [1].

II-B. Ataques a llaves sin contacto

Primeramente, hay que tener en cuenta que el protocolo de llaves *keyless* es petición-respuesta. Cuando se inicia el protocolo, el vehículo envía una petición, la llave responde y entonces el coche se abre. Como funcionalidad adicional, las llaves *keyless* permiten arrancar el motor sin introducir la llave en el contacto (normalmente se pulsa un botón dentro del vehículo).

Los métodos empleados para iniciar el protocolo de llaves *keyless* varían mucho en función del fabricante. En algunos casos, es necesario poner la mano en la maneta del coche, mientras que en otros basta con acercarse al automóvil. Otros fabricantes, optan por usar canales de baja frecuencia (LF) o de alta frecuencia, como Bluetooth Low Energy (BLE) [3].

En este caso, para comprometer el vehículo se precisa de un ataque de *relay* (o *proxy*). El ataque consiste en capturar la señal que emite la llave y transferirla al vehículo mediante un repetidor, para lo cual se necesita un grupo de al menos dos atacantes [4].

El escenario de ataque se da cuando la víctima está alejada de su vehículo y uno de los atacantes consigue aproximarse a su llave para capturar la señal. Al momento de capturar la

señal, esta se envía al segundo atacante, que está próximo al vehículo de la víctima y consigue abrir el vehículo. En caso de que el vehículo se pueda arrancar con la llave *keyless*, entonces por el mismo procedimiento el segundo atacante consigue encender el motor.

En estos casos no siempre funciona el ataque de *replay* clásico, ya que la señal de la llave puede tener un tiempo de expiración, por eso es necesario transmitirla justo al recibirla. En caso de que funcione, deberá realizarse habiendo instalado previamente un *jammer* en el vehículo de la víctima para obligarle a que tenga que abrir y cerrar el coche de forma manual, a la par que se capturan los códigos emitidos por la llave [2].

Para prevenir este tipo de ataques, algunas llaves *keyless* incorporan una protección que consiste en que se desactivan mientras no detecten movimiento (con el uso de un acelerómetro). De esta manera, ya no están emitiendo señales continuamente, sino solo durante un tiempo determinado. Aun así, mediante técnicas de ingeniería social, se podría conseguir activar la llave *keyless* y realizar el ataque de *relay*. Otra forma de prevención consiste en guardar la llave en una funda con protección contra ataques de RF (una jaula de Faraday) [5].

De nuevo, otra alternativa es analizar la señal que emite la llave *keyless* y realizar ingeniería inversa para poder obtener un algoritmo que genere señales válidas y así poder abrir (y arrancar) el vehículo de forma correcta. Este proceso puede ser más complicado que en el caso de los mandos a distancia porque existen llaves *keyless* que incorporan algoritmos criptográficos, tanto estándares como propietarios [3].

III. Recursos y herramientas

III-A. Hardware

Como dispositivos *hardware*, se emplearán HackRF One y YARD Stick One, ambos de la empresa Great Scott Gadgets.

HackRF One servirá para realizar un reconocimiento básico de radiofrecuencia. Este dispositivo es un transceptor *half-duplex* que permite operar en frecuencias desde 1 MHz hasta 6 GHz y que puede llegar a una razón binaria de 20 millones de muestras por segundo. Además, se trata de un dispositivo *open-source*.

Con HackRF One y el *software* adecuado, se puede visualizar el espectro de frecuencia en tiempo real, se pueden capturar señales y también replicar dichas capturas.

YARD Stick One, por su parte, también es un transceptor *half-duplex* que opera en frecuencias 300-348 MHz, 391-464 MHz y 782-928 MHz. Este dispositivo acepta distintos tipos de modulaciones digitales (ASK OOK, GFSK, 2-FSK, 4-FSK, MSK) y razones binarias de hasta 500 kbps.

Con YARD Stick One y `rfcat` se pueden capturar señales sabiendo sus parámetros de RF y también generar señales sintéticas a partir de los bits que se quieran transmitir.

III-B. Software

Como *software*, se utilizará GNU Radio Companion, GQRX, `inspectrum` y `rfcat`. Estos programas se pueden instalar fácilmente en un sistema operativo Linux (como Ubuntu o Kali Linux).

GNU Radio Companion es un proyecto *open-source* que proporciona un entorno de programación gráfico basado en bloques de procesamiento de señales para interactuar con dispositivos de *Software-Defined Radio* (SDR). Se usará junto con HackRF One para capturar señales y reproducir señales capturadas.

GQRX es un programa basado en GNU Radio Companion que muestra el espectro en frecuencia en formato *waterfall* y es capaz de procesar la señal de radio recibida. El uso que se dará a este programa será para identificar la frecuencia de trabajo con HackRF One.

`inspectrum` es un programa que muestra la potencia de una señal en tiempo y frecuencia. Se utiliza para analizar una captura de señal y poder extraer sus características e incluso los símbolos codificados como bits.

`rfcat` es una librería de Python que se utiliza exclusivamente para interactuar con YARD Stick One mediante una consola de comandos interactiva o bien mediante un *script*.

IV. Metodología propuesta

En primer lugar, habrá que identificar la banda de trabajo del vehículo y el mando o la llave *keyless* (normalmente será la banda de 433 MHz o 868 MHz, que son bandas no licenciadas). Para el caso de llaves *keyless*, es posible que su banda de trabajo sea en baja frecuencia (canales de 22 kHz, 125 kHz y 134.2 kHz) o en alta frecuencia (BLE, en 2.4 GHz) [3].

Una vez identificada la frecuencia, se procede a realizar una captura de la señal con HackRF One. Esta captura puede ser reproducida para efectuar un ataque de *replay*. No obstante, si el ataque se queda en este punto, sería un estilo *script-kiddie*. Además, la captura no es perfecta, contiene ruido, y es probable que el ataque no funcione como debería.

Como se ha comentado anteriormente, esta metodología se amplía realizando un análisis de la señal capturada digitalmente. Para entender la señal, será necesario tomar varias muestras y representarlas en `inspectrum`, para poder extraer los símbolos y los bits codificados.

Una vez que se conoce el formato binario de la señal capturada, se puede usar YARD Stick One para sintetizar una señal que porte la información extraída. Para que esta señal sintética funcione correctamente, será necesario realizar las siguientes tareas de manera iterativa:

- Configurar los parámetros de YARD Stick One.
- Transmitir la señal sintética con YARD Stick One.
- Capturar la señal sintética con HackRF One.
- Analizar la captura de la señal sintética con `inspectrum` y comparar con la señal que se quiere suplantar.

En el momento en que la señal original sea equivalente a la señal sintética, el dispositivo YARD Stick One estará correctamente configurado tanto para transmitir como para recibir. Además, YARD Stick One será capaz de demodular una señal y extraer los bits codificados de forma automática. Con esto, se consigue que el análisis de la información enviada por un cierto dispositivo (una llave de vehículo) sea mucho más eficiente.

V. Resultados preliminares

Para los resultados obtenidos hasta la fecha, no se indicarán los fabricantes de vehículos analizados por motivos de privacidad.

V-A. Análisis de señales

La Tabla I compara las señales usadas por modelos de vehículos de distintos fabricantes.

Tabla I
CARACTERÍSTICAS DE SEÑALES CAPTURADAS.

Fabricante	Modulación	Longitud
Marca A	2-FSK	Una trama de 168 bits
Marca B	2-FSK	3 tramas de 120 bits
Marca C	ASK OOK	Una trama de 312 bits
Marca D	ASK OOK	6 tramas de 312 bits y 27 tramas de 32 bits

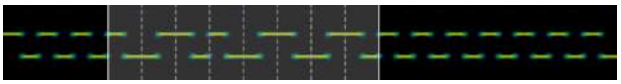


Figura 1. Muestra de señal capturada para un vehículo de la Marca A.

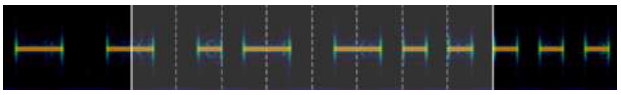


Figura 2. Muestra de señal capturada para un vehículo de la Marca D.

Como se observa en la Tabla I, los fabricantes analizados utilizan para transmitir la información modulaciones digitales sencillas como 2-FSK (Fig. 1) o ASK OOK (Fig. 2). Además, en todos los casos analizados se utiliza codificación Manchester, en la que la información se codifica por flanco (p.ej., flanco de subida: 0 lógico y flanco de bajada: 1 lógico, o viceversa según el convenio), proporcionando así sincronismo a nivel de símbolo.

Todas las señales cuentan con una secuencia de sincronismo al comienzo de la trama (Fig. 3) que consiste en una sucesión de 1 lógicos hasta llegar a un 0 lógico que indica el comienzo de los datos transmitidos. El uso de la codificación Manchester permite precisamente que esta secuencia de sincronismo tenga efecto en el receptor y pueda establecer el periodo de símbolo para demodular la señal.

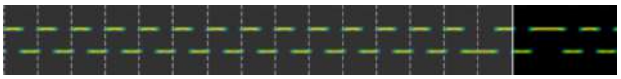


Figura 3. Secuencia de sincronismo de señal capturada para un vehículo de la Marca A.

V-B. Resultados individuales para el modelo de la Marca A analizado

Tras realizar el procedimiento descrito en la sección IV con la señal capturada del modelo de Marca A, se consiguió configurar YARD Stick One para poder transmitir y recibir usando `rfcat`.

Estos son algunos ejemplos de la información transmitida por el mando del vehículo cuando se pulsa el botón de apertura (en hexadecimal):

- `ffffffffffffe7b5fee66491d0592c722769755eb`.
- `ffffffffffffe7b9fee66491db9af74d04ba1ae4e`.
- `ffffffffffffe7b1fee66491d34a91610d4848854`.
- `ffffffffffffe7befee66491deb94957806d52b0c`.

Como se puede observar, hay varios dígitos que coinciden en las cuatro muestras, lo cual indica que la información está dividida en diferentes secciones binarias. Tomando la primera muestra, se pueden identificar las siguientes secciones:

- Secuencia de sincronismo: `ffffffffffffe`.
- Acción: `7b`.
- Valor cambiante: `1f`.
- Valor fijo: `ee66491d`.
- Código evolutivo: `0592c722769755eb`.

Hasta el momento, se sabe que la acción será `7b` para abrir el vehículo, `3b` para cerrarlo y `5d` para abrir el maletero. Por otro lado, por el momento se desconoce la función del valor cambiante, aunque se cree que puede tratarse de algún método de detección de errores. El valor fijo se corresponde con un identificador de la llave. De esta manera, solamente las llaves configuradas para un vehículo funcionarán (típicamente, dos llaves: la de uso habitual y la de repuesto). Por último, el código evolutivo es un número pseudo-aleatorio de 64 bits, lo cual es suficientemente robusto ante un ataque de fuerza bruta.

Finalmente, cabe destacar que se logró transmitir una de estas secuencias de bits extraídas mediante YARD Stick One y abrir el vehículo, disponiéndose de un vídeo que lo demuestra que podrá mostrarse en la conferencia.

VI. CONCLUSIONES

La ciberseguridad es un tema de especial relevancia actualmente en el sector del automóvil, existiendo la necesidad de metodologías y herramientas que permitan evaluar y comparar la ciberseguridad de diferentes marcas y modelos de vehículos.

En este artículo se presenta una metodología y un conjunto de herramientas soporte que permiten obtener trazas de los mensajes intercambiados entre mandos de radiofrecuencia y vehículos, lo que a su vez permite analizar, evaluar y comparar los protocolos utilizados por diferentes modelos de vehículos desde el punto de vista de la ciberseguridad. Hasta el momento se han analizado trazas de varios modelos y se ha conseguido capturar un mensaje de apertura de un vehículo y reproducirlo para abrirlo, simulando un posible ataque. Sin embargo, trabajar con la representación binaria de los mensajes intercambiados entre mando y vehículo permite ir más allá de realizar meros ataques. Por poner un ejemplo, se podría comprobar si una vez recibido un código pseudoaleatorio realmente se descartan los códigos anteriores no utilizados; de no ser así, esto representaría una vulnerabilidad desde el punto de vista de ciberseguridad.

Actualmente estamos trabajando en la captura y análisis de más señales de mandos de diferentes marcas de vehículos, para poder comparar los protocolos utilizados desde el punto de vista de la ciberseguridad, así como en el análisis del funcionamiento de las llaves *keyless* , que son un caso especialmente relevante ya que permiten no sólo abrir el vehículo, sino también ponerlo en marcha.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por MAPFRE a través del proyecto TIBET (inTegración de cIberseguridad y Biomecánica en vEhículos y Tráfico).

Referencias

[1] A. Mohawk, 5 de feb. de 2016. [En línea]. Disponible en: https://www.andrewmohawk.com/2016/02/05/bypassing-rolling-code-systems/ (Accedido: 05-04-2022) (citado en p. 2).

[2] O. Ibrahim, A. Hussain, G. Oligeri y R. Pietro, «Key is in the Air: Hacking Remote Keyless Entry Systems,» en 10 de ene. de 2019, págs. 125-132, ISBN: 978-3-030-16874-2. DOI: 10.1007/978-3-030-16874-2_9 (citado en pp. 2, 3).

[3] L. Wouters, B. Gierlichs y B. Preneel, «My other car is your car: compromising the Tesla Model X keyless entry system,» *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2021, n.º 4, págs. 149-172, ago. de 2021. DOI: 10.46586/tches.v2021.i4.149-172. [En línea]. Disponible en: https://tches.iacr.org/index.php/TCHES/article/view/9063 (citado en pp. 2, 3).

[4] J. Edkins, *Keyless car theft: what is it and how to prevent it*, 11 de feb. de 2022. [En línea]. Disponible en: https://www.carwow.co.uk/blog/keyless-car-theft-prevention (Accedido: 05-04-2022) (citado en p. 2).

[5] *Keyless car theft: What is a relay attack, how can you prevent it, and will your car insurance cover it?* Leasing.com. [En línea]. Disponible en: https://leasing.com/guides/relay-car-theft-what-is-it-and-how-can-you-avoid-it/ (Accedido: 05-04-2022) (citado en p. 3).

Implicaciones de seguridad en MAS Desplegados en Infraestructuras de Carga basadas en OCPP

Cristina Alcaraz, Alberto Garcia y Javier Lopez
Computer Science Department, University of Malaga,
Campus de Teatinos s/n, 29071, Malaga, Spain
{alcaraz, albertogr, jlm}@lcc.uma.es

***Resumen*—El interés actual por desplegar infraestructuras de carga de vehículos eléctricos para el ahorro energético y la sostenibilidad es cada vez más palpable, lo que llama la atención a muchas comunidades, especialmente a la científica, para explorar, entre otras cosas, la influencia de las nuevas tecnologías de información en los procesos operacionales. Teniendo en cuenta este escenario, este artículo, por tanto, analiza cómo el uso de los sistemas de multi-agente pueden beneficiar las tareas de monitorización, mantenimiento y de seguridad, y propone una arquitectura específica en base a los actores especificados en el protocolo OCPP (Open Charge Point Protocol). Esta arquitectura constituye la base para analizar los diversos tipos de amenazas que agentes software pueden sufrir, clasificándolas de acuerdo a las características funcionales e interacciones con los diversos elementos de la infraestructura. Esta agrupación y el conjunto de ataques abordados están basados en el SP-800-19 definido por el National Institute of Standards and Technology, y formalizados siguiendo la metodología de árboles de ataque. El estudio revela la importancia que tiene analizar los riesgos que esta tecnología puede traer a este escenario, proporcionando, además, un conjunto de recomendaciones que sirvan de guía para aplicaciones futuras.**

***Index Terms*—Ciberseguridad, infraestructuras de carga de vehículos eléctricos, sistemas multi-agente, árboles de ataques**

Tipo de contribución: *Investigación original*

I. Introducción

La evolución que está viviendo el sector de la movilidad está motivada entre otras cosas por la concienciación ecológica que cala cada vez más en la sociedad debido a la necesidad de sustituir las tradicionales fuentes de energía por otras con un menor impacto en el medio ambiente. Esto se refleja en el crecimiento del mercado de vehículos eléctricos, en el que se espera un volumen de crecimiento anual de 21,7 % para alcanzar 233,9 millones de unidades y los 2.495,4 millones de dólares para el año 2027 [1]. Esta demanda obliga a desplegar infraestructuras de carga tanto públicas como privadas, que sostengan este nuevo modelo de transporte. Uno de los protocolos más extendido actualmente para dar soporte a este tipo de infraestructura es el protocolo OCPP (Open Charge Point Protocol) [2], el cual define un conjunto de actores y una arquitectura base en la que estaciones de carga se conectan a sistemas centralizados a cargo de la gestión y el control del suministro de energía, ofreciendo un conjunto de servicios específicos para la autorización de transacciones, gestión y configuración dinámica de las estaciones, mecanismos para reservar energía y diagnóstico. Aprovechando estos recursos fundamentales para diseñar y desplegar infraestructuras de carga, este artículo propone la aplicación de un sistema multi-agente (Multi-Agent System (MAS)) para tratar y optimizar aún más algunas de las implicaciones en materia de operación, mantenimiento y seguridad de infraestructuras de carga. Con esto, además, nos acercamos a la concepción idónea de interconectar Tecnologías de la Información (TI) con las Tecnologías Operacionales (TO) para beneficiar las operaciones en tiempo real y el modelo de negocio.

Para hacer una revisión del estado del arte, en el trabajo [3] se aborda un conjunto de problemas presentes actualmente en las estaciones de carga y propone el uso de la IA (Inteligencia Artificial) y tecnología blockchain para solventarlos, pero no estudia la aplicación y las implicaciones que tiene las nuevas TI tal como se aborda en este artículo. Por otro lado, algunos autores ya han llevado a cabo estudios previos sobre la utilización de agentes en redes de carga de vehículos eléctricos. Por ejemplo, en [4] se desarrolla un simulador basado en un MAS para analizar el comportamiento de los usuarios dentro de una red de cargadores. Su objetivo es determinar de forma óptima dónde desplegar las estaciones de carga para satisfacer la demanda. En [5], se diseña un MAS que tiene como función gestionar el consumo de los usuarios para minimizar el impacto en la red eléctrica. También simula el sistema planteando cuatro escenarios distintos para verificar su eficacia. El mismo objetivo se persigue en [6], en la que, además, se define una estructura jerárquica que mejora la planificación, y en [7] y [8] se realizan predicciones del comportamiento de los usuarios teniendo en cuenta factores sociales. Sin embargo, en todos estos trabajos, los agentes software (SW) se usan principalmente para la simulación y la planificación, no para desplegarlos junto a la red de carga como si fuese una red de monitorización secundaria, además de que no exploran al detalle las implicaciones que tiene el despliegue de dichos agentes desde el plano de la seguridad.

El presente artículo da un salto y propone, por un lado, una infraestructura de carga basada en un MAS para garantizar una monitorización constante de las estaciones de carga, midiendo estados de salud para contribuir con el mantenimiento predictivo y a la detección de amenazas, además de cumplir con el esquema estandarizado de interconexión del protocolo OCPP. Por otro lado, y en base a esta arquitectura, (i) se identifica y clasifica tipos de amenazas al sistema propuesto y de acuerdo al SP 800-19 definido por el National Institute of Standards and Technology (NIST) en [9], (ii) se formaliza y prioriza algunos ataques siguiendo la metodología de árboles de ataques, y (iii) se proporciona recomendaciones de seguridad como guía para implementar soluciones TI-TO seguras en el escenario propuesto.

El resto del artículo queda estructurado como sigue: en la sección II se presenta la arquitectura específica basada en OCPP y se introducen los agentes. En la sección III se

clasifican las amenazas siguiendo la SP-800-19. Esta clasificación se complementa con la aplicación de la metodología de modelado de amenazas con árboles de ataque en la sección IV. Más tarde, en la sección V, se exponen algunas recomendaciones, con el fin de reforzar la seguridad en este escenario. Por último, las conclusiones de este estudio se recogen en la sección VI, junto con algunos detalles del trabajo futuro.

II. Despliegue de un MAS en redes de carga

Como se indica en la introducción, el escenario considerado en este trabajo consiste en una infraestructura basada en estaciones de carga (comúnmente conocidos como cargadores eléctricos) para vehículos eléctricos, y por el que hace uso del protocolo OCPP detallado en [2] y definido por el Open Charge Alliance. El protocolo considera los fundamentos tradicionales de cliente-servidor y de los principios de control entre las estaciones de carga y el sistema central, e incluye instrucciones de comando y control (C&C) necesarias para permitir que sistemas de control puedan gestionar y autorizar las recargas deseadas y diagnosticar en tiempo real las estaciones. Esta forma de conectar estaciones con sistemas de control, y de una manera estandarizada, ha llamado la atención a muchas manufactureras (ej. ABB, Schneider Electric, GARO, hypercharger, Legrand y efacec, entre otros), convirtiéndolo en un estándar de facto fuertemente apoyado por la industria [10]. Es tanta su influencia, que actualmente se encuentra disponible la versión OCPP 2.0.1 [2] que aborda los siguientes actores de interés también para nuestro estudio:

- CS (*Charging Station*): es el sistema ciber-físico, a través del cual, los usuarios de la infraestructura de carga pueden recargar sus respectivos vehículos eléctricos. Por tanto, una CS es la interfaz principal entre el usuario y la red de carga, cuyos elementos lo conforman: un controlador encargado de gestionar los sensores y actuadores integrados en su propia estación, además de incluir un contador inteligente para contabilizar el consumo eléctrico.
- CSMS (*Charging Station Management System*): está encargado de gestionar, supervisar y controlar las CS existentes en la red de carga. Procesa las peticiones de los usuarios que quieren hacer uso de los servicios de carga y se comunica con las CS involucradas para controlar la transacción de energía.
- EVSE (*Electric Vehicle Supply Equipment*): es el componente de la CS que aporta energía al vehículo. Cada CS puede contener uno o más EVSE, y cada uno es gestionado de forma independiente por la CS. En este trabajo se usa el concepto de redes de CS y redes EVSE de forma intercambiable.
- CSO (*Charging Station Operator*): entidad encargada de la administración y el mantenimiento de la red EVSE. Suele hacer referencia a una persona empleada para dichas tareas de gestión y control local.
- EMS (*Energy Management System*): dispositivo que controla la producción y el consumo real de energía de forma local o por áreas, según las políticas y normativas pre-establecidas. Para este control, se requiere, además, que el EMS guarde estrecha relación y comunicación con el CSMS para producir y distribuir la energía de acuerdo a la demanda real.
- EV (*Electric Vehicle*): corresponde con el destinatario principal de la energía, cuando el usuario, dueño del EV, hace uso de los servicios de carga. En este punto, merece la pena destacar que un EV puede también actuar como fuente móvil de energía hacia la red eléctrica (en base a una comunicación y flujo de energía bidireccionales). Estos casos son definidos como V2G (Vehicle-to-grid).
- Infraestructura de tarificación: es la entidad encargada de establecer los precios, registrar la deuda y realizar el cobro por los servicios de carga ofrecidos a los usuarios.

El protocolo OCPP está diseñado para integrar prácticamente cualquier técnica comúnmente usada en el sector, soportando diversos métodos de autorización de usuarios: mediante tarjetas RFID, usando tarjetas de crédito, autorización remota a través del CSMS, etc. También contempla la posibilidad de que la CS pierda, de forma temporal, la conexión con el CSMS, pasando a operar en un estado "ofline" en el que la CS es capaz de gestionar de forma local una lista de autorización obtenida previamente del CSMS. Si no dispone de tal lista, la CS aceptará a cualquier usuario, por motivos de continuidad del negocio, para realizar la autenticación una vez que se haya recuperado la conexión con el CSMS y así poder tarificar y cobrar el servicio. Además, OCPP plantea tanto situaciones en las que el inicio de la transacción de energía es solicitado por la CS, como situaciones en las el CSMS inicia el proceso de carga ("remote control"). Otra de las características que implementa OCPP 2.0.1 es la posibilidad de reservar una CS (o bien un EVSE de la CS) con un período de antelación, de forma que ningún otro usuario pueda usarla durante ese tiempo.

II-A. El rol de los agentes software en CS

Dentro de la arquitectura propuesta, un conjunto de agentes SW se distribuyen en cada componente principal del sistema de carga (ya sea en la CS y el CSMS) con objeto de recolectar localmente y de manera dinámica información relevante que puede ayudar a mejorar las funciones primarias de cada estación y que puede ser vital para analizar el estado de salud de una red, favoreciendo, a su vez, las decisiones a realizar en los respectivos sistemas de CSMS y EMS. Además, estos agentes pueden tener la capacidad para compartir información que puede ser útil para intensificar aún más la "consciencia situacional", y explicar de primera mano: (i) qué ocurre dentro de un punto de carga y en qué momento determinado, además de (ii) identificar qué elementos hardware, software y red son afectados dentro de una CS, y (iii) qué vecinos pueden también estar implicados. Esta característica esta representada en la figura 1, donde podemos ver cómo las CS se pueden comunicar con el CSMS a través del protocolo OCPP, mientras una red paralela se despliega con respecto a la red de OCPP para contribuir en mejorar la calidad de los recursos de control frente a posibles fallos imprevistos o ataques inesperados, muchos de los cuales pueden provenir de insiders (ej. CSO con intenciones maliciosas) u outsiders (ej. el público general con acceso a las CS, o terceros/proveedores de servicios HW o SW) con intereses para manipular valores relevantes de componentes CS [11], [12], e impactar consecuentemente en el suministro de energía y/o dañar el modelo de negocio y su cadena de valor.

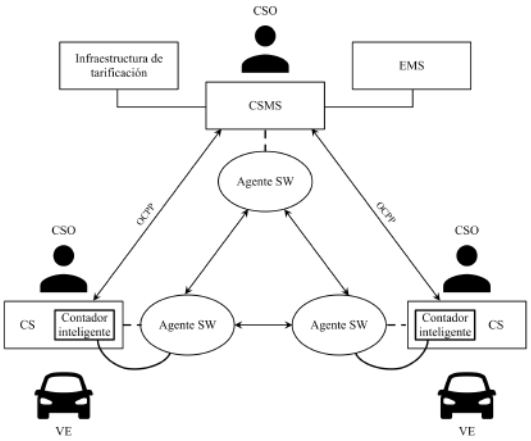


Figura 1. Inclusión del MAS a la red de carga

En nuestro escenario, se consideran fundamentalmente dos roles distintos en función de (i) las tareas de las que se encarga, de (ii) la información que gestiona y de (iii) dónde se sitúe en la red de carga. Cada CS es la plataforma en la que se hospeda un agente SW que aquí llamamos *agente de monitorización*, ya que su función principal será la de obtener información local relativa al consumo de energía de la estación y su funcionamiento interno (recursos HW y SW disponibles, estado de los canales de comunicación con otros nodos de la red, tráfico de red, etc.), y en definitiva el estado de salud de la CS. Para conocer información referente al consumo (metering), este agente tiene acceso al contador local de la CS, con esto se consigue una doble verificación del consumo eléctrico (a través de los reportes del protocolo OCPP y del sistema MAS), mientras que se consigue un diagnóstico más completo. Por otro lado, el CSMS implementará su propio agente SW con un rol distinto al de los anteriores, al que denominamos *agente colector*. La inclusión de este agente permite recolectar la información obtenida por los agentes de monitorización presentes en todas las CS de la red, por lo que tiene un rol fundamental en la monitorización, diagnóstico y detección de amenazas en toda la infraestructura de carga. Aparte de esto, los recolectores pueden tener la capacidad de liderar funciones específicas de control (ej. interrumpir la toma de medición por los agentes de monitorización, solicitar datos de diagnóstico o cambiar de manera remota valores de configuración) y respuesta frente a incidentes.

Aplicando el enfoque MAS, mejoramos la capacidad de conocer con mayor detalle el estado de toda la red, reforzando la monitorización ofrecida por los tradicionales IDS/IPS (Intrustion Detection/Prevention Systems), SIEM (Security Information and Event Management), SOC (Security Operation Center) y firewalls, permitiendo tener una prevención y/o detección más temprana frente ataques específicamente diseñados para estos tipos sistemas ciber-físicos. Por otro lado, el uso de los agentes para comandar acciones de respuesta frente a amenazas potenciales del estilo APT (Advanced Persistent Attacks) contra las CS, permite al CSO tener un mayor control sobre la red y flexibilidad a la hora de elaborar planes de mitigación y recuperación.

Pese a esto, la incorporación del MAS debe realizarse con sumo cuidado, mirando especialmente en ciertos aspectos relacionados con la seguridad, de lo contrario se estarían introduciendo nuevas amenazas con nuevos patrones o vectores de ataque que pueden ser aprovechados por adversarios. Esto se debe a que los agentes no dejan de ser componentes SW que, si no son protegidos correctamente frente accesos no autorizados, son fácilmente manipulables por actores maliciosos. Esta connotación software añade además, la necesidad de precisar en los procesos de validación de códigos fuentes desde que éstos pueden presentar serios bugs que pueden ser fácilmente explotables por atacantes. Estos errores software puede incluso provenir de frameworks, librerías y herramientas proporcionadas por terceros, lo que conlleva a numerosos agujeros de seguridad. Por tanto, las siguientes secciones, exploran clases de amenazas a estos tipos de agentes SW (monitorización y recolectores), agrupándolos de acuerdo a sus interacciones.

III. Clasificación de amenazas de seguridad

Aunque existen metodologías efectivas para clasificar tipos de amenazas en sistemas basados en agentes software, como puede ser el tradicional modelo CIA (Confidentiality, Integrity and Availability), nosotros seguimos la metodología propuesta por el SP (Special Publication) 800-19, titulada "*Mobile agent security*" y definida por el NIST en [9]. Según este SP es posible clasificar las amenazas contra agentes software teniendo presente sus principales componentes así como sus relaciones, para luego categorizar las amenazas e identificar los posibles vectores de ataque. Por tanto, en las siguientes subsecciones vamos a analizar las interacciones que tienen los agentes con cada elemento de la infraestructura de carga, para posteriormente extraer posibles tipos de ataques en la siguiente sección.

III-A. Clase 1: agentes SW y las estaciones de carga (CS)

La estación de carga es el principal actor y componente ciber-físico en donde agentes software pueden desplegarse para la monitorización constante de los recursos de la CS, y, por lo tanto, el componente por el cual se espera que agentes puedan funcionar durante todo su ciclo de vida. Esto también significa que tanto una CS como un agente SW guardan una estrecha relación entre ellos, y esto, a su vez, es lo que conduce a establecer una categoría de amenazas en el que se incluyen todas aquellas que pueden poner en riesgo el bienestar y la seguridad de la CS. Por ejemplo, agentes SW maliciosos o "rogue" (supuestos agentes legítimos del sistema) podrían ser integrados para intencionadamente impactar en el comportamiento de la CS, alterar datos de telemetría o de medición, violar operaciones críticas o corromper servicios esenciales.

Dicho de otro modo, el perfil de un agente malicioso dentro esta categoría consiste en una pieza de código con autonomía suficiente y privilegios para: (i) realizar tareas de gestión de archivos de registros, (ii) medir variables relativas al funcionamiento y al rendimiento de la CS, y (iii) ejecutar comandos C&C para la dar respuesta a incidentes de seguridad. Esta forma de abusar o escalar privilegios, les permiten, a su vez, a tomar acciones indebidas que podría cambiar o detener otros servicios necesarios y esenciales para el correcto funcionamiento de la CS (p. ej. denegación de los heartbeats al CSMS), del proceso de carga de un EV (p. ej. interrumpir

la carga de un conector del EVSE) o provocar serios apagones por impactar en el grid (p. ej. retornar energía en escenarios V2G). Aparte de esto, agentes maliciosos pueden acceder a datos sensibles y registros almacenados en la CS con la posibilidad de filtrarlos a un actor malicioso (p. ej. un gateway externo al sistema para exfiltración de los datos), o incluso, alterarlos para engañar al propio MAS o al sistema de pago de la red de carga. Es más, agentes legítimos pueden ser diseñados para efectuar tareas de monitorización con la capacidad para analizar el tráfico de red y las comunicaciones entre la CS y sus componentes relacionadas (ya sea otra CS o el CSMS). Si estos agentes son manipulados, entonces pueden tener también la capacidad para liderar acciones de escuchas indebidas.

III-B. Clase 2: agentes SW y el sistema de control (CSMS)

En la mayoría de escenarios, las CS pueden no ser los únicos nodos en los que se ejecutan agentes SW, sino que muchas veces es necesario situar la centralización de datos y comandos en otros agentes desplegados fuera de las estaciones. Un lugar idóneo para ello, es el CSMS puesto que este gobierna todas las CS de la red y tiene una visión global de todo el sistema. Los agentes presentes en el CSMS están diseñados para obtener datos de todos los demás agentes SW desplegados por la red, con el fin de elaborar estadísticos y mantener informado al CSMS. Por lo tanto, estos tipos de colectores necesitan mantener enlaces de comunicación con el sistema central.

Al tratarse el CSMS de un nodo de gestión centralizada, es una pieza clave dentro de la infraestructura de carga, por lo que los agentes integrados en él pueden ser manipulados por insiders maliciosos (ej. un CSO) quienes pueden tomar el control de toda la red de carga y lanzar numerosos tipos de ataques. Por ejemplo, la manipulación de datos falsos correspondientes a estados de salud hacia el CSMS, y la interrupción de operaciones de control hacia el CSMS/CS para violar la ejecución natural de otros servicios esenciales como la gestión de transacciones de energía o su control (p. ej. abusar de los canales de comunicación y aislar al CSMS o una CS, en esta última se podría sobrecargar al controlador y obligar a la estación a gestionar operaciones en modo offline como es indicado en [2] y [11]). Además, en el CSMS se almacenan datos y estadísticos relativos al consumo y al funcionamiento de toda la red EVSE, por lo un agente "rogue" integrado en el CSMS tendría acceso a tal información, con lo cual podría obtener una imagen detallada de toda la topología e información sensible de la red (a nivel de información y operación). Insiders también podrían inyectar malware o bombas lógicas (ej. backdoors) en los códigos de los agentes, otorgándoles la capacidad para engañar a los CSO sobre el estado real de la red, consiguiendo que amenazas de tipo APT puedan surgir en estos contexto, en el que insiders/outsiders podrían navegar de manera sigilosa de un nodo a otro de la red de carga y de control (p. ej. de un agente SW malicioso a otro). Por otro lado, insiders y outsiders podrían suplantar la identidad del CSMS para gestionar instrucciones de C&C hacia los agentes SW de las estaciones a fin de causar DoS y, consecuentemente, fraude o robo de energía. Es decir, adversarios con el control total de un agente SW en una estación podría causar exhaustación para interrumpir servicios de comunicación y aislar la estación como es señalado arriba. De este modo, la estación podría entrar en modo offline por requisitos del protocolo OCPP [2] (ver sección II), obligándola a tratar las autorizaciones (probablemente ilícitas) en local a fin de garantizar el servicio y la continuidad de negocio. Pero también, entidades supuestamente lícitas falseando la identidad de colectores o el CSMS podrían recibir información del sistema para liderar ataques contra la confidencialidad. Esta última amenaza puede ocurrir si medidas de seguridad en canales de comunicación (p. ej., usando TLS) no son consideradas. Incluso, siendo consideradas y dependiendo de las medidas de seguridad, es posible conllevar ataques MiTM, corrompiendo los canales de comunicación tal como se detalla en [13] para TLS 1.3 y en [11] para OCPP sobre TLS.

III-C. Clase 3: agentes SW y otros agentes SW

La propia definición de los MAS menciona que un agente tiene la capacidad de interaccionar con otros agentes en la búsqueda de realizar las tareas para las que fue diseñado. En el caso de las infraestructuras de carga, estas interacciones se refieren a las comunicaciones que se dan entre los agentes de distintas estaciones para intercambiar información sobre el estado de la red y el consumo de energía que los usuarios están demandando. También en esta categoría se distinguen las comunicaciones entre los agentes de las CS y los agentes (los colectores) que se encuentran en el CSMS, tanto para el proceso de recolección de información por áreas, como para el envío de comandos por parte de CSO a las estaciones. Si alguno de los agentes involucrados en estas interacciones ha sido comprometido por un insider/outsider, podría usarse para engañar al sistema MAS en su totalidad y extender la zona de influencia del atacante y su superficie de ataque.

Un agente comprometido podría hacerse pasar por otro agente SW con el fin de engañar y entorpecer el funcionamiento del sistema de monitorización MAS. Además, también podría mentir sobre su rol, suplantando a un agente recolector, haciendo que todos los agentes de la red le envíen su información al agente equivocado. Un agente malicioso también puede negarse a responder a las peticiones de otros agentes que intenten comunicarse con él, o incluso, realizar otros tipos de ataques de DoS, como, por ejemplo: flooding (provocar el colapso de una red por el envío masivo de paquetes), selective forwarding (retransmitir de manera selectiva) o crear agujeros negros (black holes) propiamente dicho, muchos de los cuales definidos en [11] y [14]. Otro problema que puede surgir de la interacción entre dos agentes es el caso en el que el agente malicioso niegue haber realizado una acción, como puede ser el de enviar cierta información a otro agente. Esto puede provocar dificultades a la hora de realizar auditorías del sistema que intenten descubrir cómo se ha producido un incidente de seguridad.

Asociando todos estos ataques al modelo tradicional CIA y priorizando los riesgos dentro del contexto de energía y control, observamos que el hecho de que agentes maliciosos integrados cerca de los componentes principales de telemetría y control, pueden conllevar a serios ataques contra la disponibilidad e integridad de operaciones y servicios esenciales. Sin embargo, esto no quita la relevancia que tienen los ataques contra la confidencialidad.

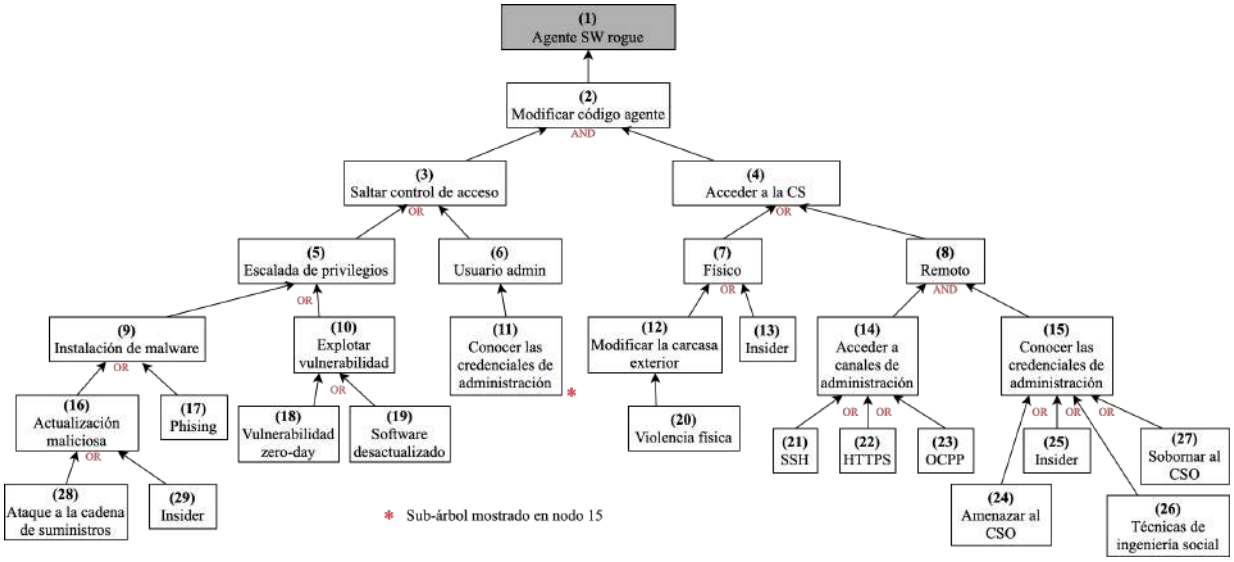


Figura 2. AT para obtener el control de un agente SW

IV. Modelado de ataques contra agentes SW

Para estudiar la viabilidad de la arquitectura propuesta en la sección II, basada en un MAS para infraestructuras de carga, se explora en esta sección algunas de las amenazas ya mencionadas en la sección anterior, a fin de identificar posibles vulnerabilidades dentro de nuestro sistema. En la literatura encontramos una gran cantidad de metodologías usadas para el modelado de ataques, como el modelo STRIDE [15], basado en el análisis de amenazas por componentes, la metodología STPA-sec [16], que estudia acciones que pueden llevar al sistema a un estado no seguro, o el enfoque OCTAVE, basado en el riesgo operacional y prácticas seguras [17]. Sin embargo, una de las más populares debido a su versatilidad son los árboles de ataque (Attack Tree, AT) [18] [19] [20] [21].

Los AT son un tipo de grafos lógicos que modelan secuencias de múltiples posibles acciones llevadas a cabo por actores maliciosos para evadir un sistema de defensa y realizar un ataque. Estos árboles se confeccionan en base a tres componentes gráficos principales [20]: nodos, aristas y puertas lógicas. Entre los nodos, destacan: el nodo raíz, que representa el objetivo del ataque; los nodos internos, por su parte, se corresponden con objetivos parciales; mientras que los nodos hoja, son los ataques que consideramos atómicos (es decir, que no pueden descomponerse más, o simplemente aquellos que no interesa especificar más detalles). Las puertas lógicas pueden ser conjuntivas, lo que implica que todos los requisitos deben cumplirse, o disyuntivas, o que supone que cualquiera de los requisitos es suficiente para llegar al nivel inmediatamente superior. Entre las fortalezas de los AT encontramos el hecho de que se pueden representar el formato texto, lo que posibilita la traducción a otros lenguajes y formas de representación de los datos como XML [19] o JSON [22]. Por tanto, son fáciles de automatizar y de integrar con otros sistemas de información [23]. Además, los árboles pueden almacenarse en un repositorio y reusarse para generar nuevos árboles que extiendan los ya existentes. Como contrapartida, nos encontramos con que su construcción se basa en el conocimiento y la experiencia del analista que los confecciona, por lo que es fácil omitir posibles ataques y rutas. Otra debilidad es que no existe un estándar para su creación, simulación y análisis [21].

Una vez explicado la técnica, vamos a analizar mediante AT, algunos de los objetivos que pueden tener insiders u outsiders en el contexto de infraestructuras de cargas controladas por un MAS. Como no es viable presentar un estudio exhaustivo de todos los posibles objetivos de un atacante por restricciones de espacio, en este análisis sólo se tendrán en cuenta algunos de los ataques que afecten a la integridad y a la disponibilidad, que, por su parte, son los tipos de ataques que más nos interesan, ya que pueden: ocasionar daños en la infraestructura y su monitorización, impedir a los usuarios que puedan cargar sus VE o proceder con fraudes relevantes en el consumo abusivo de energía.

Concretamente, se ha considerado una **ataque contra la integridad**, donde el objetivo es ganar el control de un agente SW integrado dentro de una CS correspondiente a la primera categoría definida en la sección anterior (ver subsección III-A), modificando su comportamiento (convirtiéndolo en un agente "rogue") para comprometer la CS. Esto lo capacita para llevar a cabo ataques más complejos, que le pueden suponer mayores beneficios y ocasionar un mayor impacto en la infraestructura. En la figura 2 se presenta el árbol de ataque que desglosa las rutas a seguir por un adversario para comprometer un agente SW en el MAS (1 − ilustra el nodo en la figura) propuesto en este artículo. Todos los vectores de ataque considerados en este punto pasan por alterar el código fuente que implementa el agente SW (2), para lo cual, es necesario: (i) acceder de alguna forma a la CS (4) y (ii) escalar privilegios para alcanzar dicho código (3). Es importante resaltar que el acceso a la CS puede ser tanto físico (7), como remoto (8). Si tenemos en cuenta los casos en los que un atacante (insider u outsider) se presenta en los dominios físicos de la estación de carga, podemos diferenciar dos casos. El primero de ellos es que el atacante acceda a los puertos físicos de la CS, pensados para las tareas de administración, y mediante la manipulación física de la carcasa externa del

cargador (12). Esto lo conseguiría directamente ejerciendo violencia física contra el sistema (20). Sin embargo, también hay que contemplar la posibilidad de que el atacante sea un operario o técnico de mantenimiento (13).

No es necesario que el atacante se encuentre en el mismo lugar que la CS para perpetrar un ataque. El controlador de la CS está conectado a la red y expone servicios a través de diferentes protocolos para acceder a él de forma telemática y llevar a cabo tareas de administración y mantenimiento. En este caso, un atacante puede usar estos canales para alcanzar la CS, pero para ello antes deberá ser capaz de infiltrarse en la red en la que se encuentra el cargador y localizar la dirección y los puertos en los que se están ejecutando dichos servicios (14). Los protocolos más comunes para este cometido son SSH, HTTPS y OCPP sobre TLS (como indicado en la figura como 21, 22 y 23, respectivamente) que por defecto están protegidos por mecanismos de seguridad (p. ej., la autenticación se puede establecer con usuario y contraseña, criptografía de clave pública o certificados). Por tanto, su uso está vinculado a personal autorizado, por lo que otra tarea que deberá liderar un atacante será la de robar las credenciales de acceso (15), ya sea mediante técnicas de ingeniería social (26), amenazando (24) o sobornando (27) a algún operario del sistema, es decir, a un CSO. También está la posibilidad de que el atacante sea el propio operario o el administrador del sistema (p. ej. un trabajador descontento) (25).

Una vez el atacante ha alcanzado la CS objetivo, deberá eludir el control de acceso para llegar al código fuente del agente SW (3), con el fin de comprometerlo. Esto puede llegar a ser sencillo si el atacante conoce las credenciales (7) de una cuenta con los privilegios de administración (6), que ha podido conseguir de múltiples formas (24, 25, 26, 27), como se menciona en el párrafo anterior. En cambio, si el atacante usa una cuenta sin los privilegios para modificar el código del agente, deberá valerse de otras técnicas para realizar una escalada de privilegios (5). Una de ellas podría ser la de aprovecharse de alguna vulnerabilidad del sistema (10) que aún no haya sido parcheada, ya sea porque el personal de administración no ha realizado bien su trabajo (19), o porque se trate de una vulnerabilidad desconocida, zero-days (cero día), (18). Otra forma de escalar privilegios sería mediante la instalación de un malware (9), la cual puede ejecutar una puerta trasera para permitir el control desde una localización remota. Para infectar la CS, el atacante cuenta con múltiples opciones, como las campañas de phishing o spear phishing (17). También, es importante contemplar la opción en la que el atacante comprometa el sistema de actualizaciones (28), de forma que la CS lleve a cabo una actualización maliciosa (16) e instale el malware sin que el administrador se percate. Una vez más, aparece la opción del insider, ya que puede autorizar actualizaciones no oficiales (29).

Una vez que el atacante se ha hecho con el control del agente, éste es capaz de plantearse ataques más sofisticados que pueden incluso ser de tipo APT para persistir aún más en el tiempo. Esto supone que el árbol descrito en la figura 2 se convierte en un componente de muchos otros árboles de ataque que representan diversos objetivos, como se puede apreciar en la figura 3. Aquí se muestra una de las características que se han comentado anteriormente y que hacen

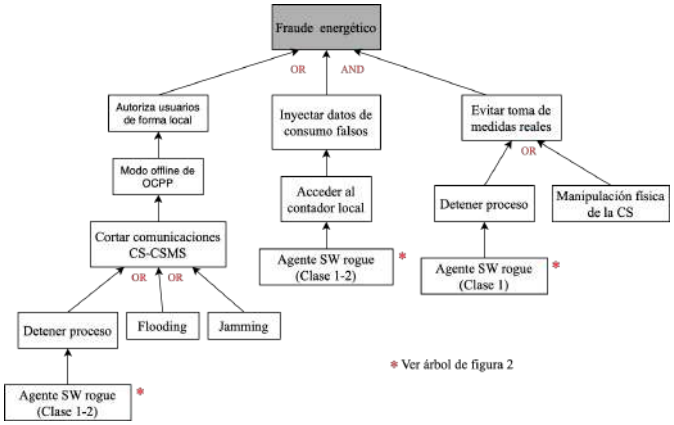


Figura 3. AT para cometer fraude energético

a los AT ser una herramienta muy versátil en el análisis de amenazas: *su capacidad para reutilizarse y componer árboles más complejos*. En la figura 3 se observa un árbol para un **ataque de fraude de energía** que sirve de ejemplo para ilustrar este hecho. En él, el nodo raíz del árbol de la figura 2 se ha convertido en un nodo hoja, esto significa que el objetivo final que considerábamos anteriormente se convierte en el punto de partida para un nuevo objetivo del atacante.

En la figura 4 se presenta un ejemplo de **ataque de denegación de servicio** en el que el objetivo del atacante es impedir las comunicaciones entre dos agentes SW. Por motivos de extensión del documento no va a explicarse detalladamente todo el árbol, aunque sí es interesante pararse en algunas partes del mismo. Este árbol está estructurado de forma que diferencia entre vectores de ataque provenientes de la capa física (4), los del nivel de red (2) y los del nivel de aplicación (5). Si atendemos a los ataques de red (2), vemos que la mayoría de ellos (black-hole attack (5), flooding (6) y selective forwarding (7)) requieren antes de un ataque de suplantación de identidad (13). De esta forma un atacante necesita llevar a cabo un ataque de ARP spoofing (19) y, a través de un agente rogue modificado (20), afirmar ser otro agente legítimo de la red para engaña a los agentes de otras estaciones (y también al agente recolector del CSMS). La excepción que no necesitaría una suplantación de identidad previa, es el ataque de replay (8), en el que el atacante, únicamente captura un paquete de una conversación antigua y lo envía múltiples veces al objetivo que desea dejar fuera de servicio en un corto período de tiempo.

A nivel de aplicación (3) se pueden detener las comunicaciones entre dos agentes, lanzando una señal de terminación (9) al proceso encargado de escuchar en el puerto correspondiente. Esto lo puede realizar de forma automática un malware previamente instalado en el nodo (14) o, nuevamente, un agente SW cuyo comportamiento ha sido modificado (16). Un atacante también tendría la opción de infiltrarse en la CS y obtener una terminal remota (15), conectándose a través de protocolos como SSH (26) o Telnet (27). Además, de ello también necesitará tener los permisos suficientes como para detener procesos del sistema (22), que pueden obtenerse mediante una escalada de privilegios (24), o usando las credenciales de un CSO (25).

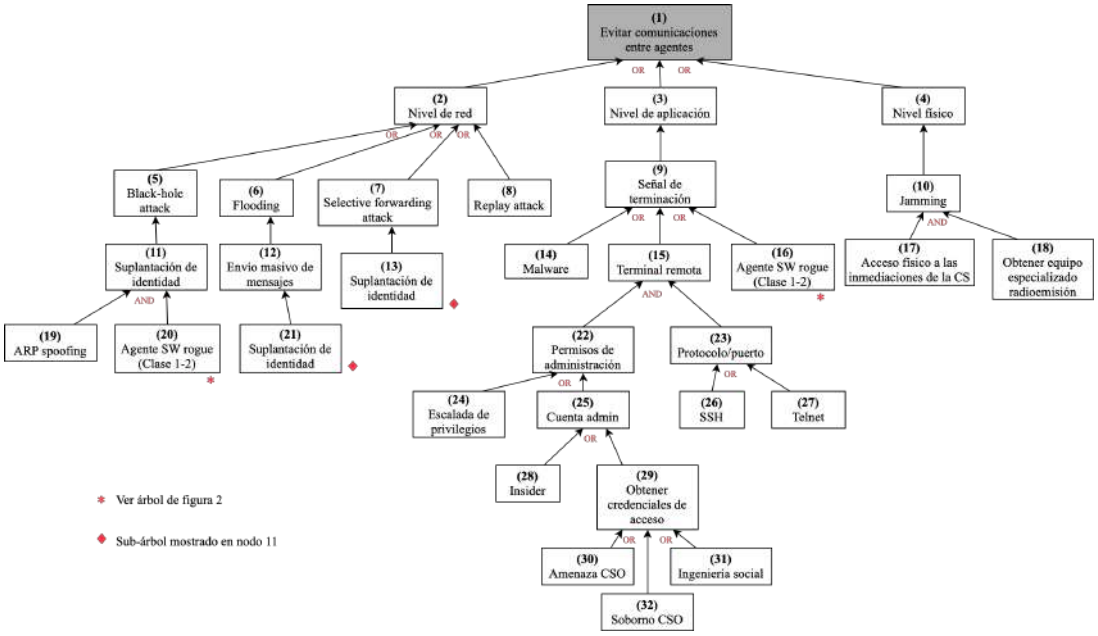


Figura 4. AT para impedir que dos agentes se comuniquen

V. Recomendaciones de seguridad

Para garantizar la seguridad de un MAS desplegado en una infraestructura de carga de VE, deben cumplirse una serie de requisitos de seguridad para reducir cualquier incremento de riesgo que vaya en contra del buen funcionamiento y rendimiento de los posibles agentes SW desplegados en las CS, en el CSMS y entre CS. Uno de los requisitos de seguridad a abordar durante el diseño de un MAS y en relación con el CIA, es la importancia que tiene el intercambio fluido de datos (ya sea con otros agentes o con la plataforma en la que se hospedan), vital para garantizar una monitorización fiable de la infraestructura de carga. Esto implica configurar y usar protocolos y mecanismos de seguridad bajo algoritmos de cifrado robustos, uso de certificados digitales como medio de autenticación, firmas digitales para autenticación y prueba de las acciones cometidas en cada una de las transacciones, así como medidas de validación mediante hashes. Sin embargo, esto puede acarrear penalizaciones en la eficiencia de la CS, ya que en la mayoría de ocasiones, se tratan de dispositivos con importantes limitaciones de recursos HW y SW, con poca capacidad de cómputo (p. ej., sistemas empotrados).

Por otro lado, los agentes SW pueden estar expuestos a actores maliciosos internos o externos, ya que las CS pueden estar desplegadas en dominios públicos. Como ya hemos analizado en la sección IV, estos actores pueden tener la habilidad para modificar los códigos fuentes de los agentes y su comportamiento, capacitándoles a realizar subsecuentes ataques que puede conllevar a un impacto mayor en la red EVSE y la disponibilidad de los servicios esenciales (la energía). Por tanto, se recomienda que aunque los agentes no puedan evitar la modificación de su propio código, es necesario que la CS intensifique las medidas de control acceso al sistema para proteger el acceso directo al código de los agentes SW, registrando cada acción cometida dentro de la misma para dar garantías de auditoría y responsabilidad (accountability), por lo que también se recomienda combinar las medidas de protección con el uso de tecnologías disruptivas como es la tecnología de blockchain para la trazabilidad [3]. Además de esto, también es aconsejable proteger frente a accesos no autorizados los archivos de registro y la información con la que operan los agentes.

Las amenazas que afectan a la disponibilidad de los agentes SW suelen ser difíciles de contrarrestar. Por lo general, un agente SW debe ser capaz de procesar y responder a las peticiones de otros agentes SW de la forma más eficiente posible, sin llevar a cabo cómputos muy costosos en recursos, de forma que pueda hacer frente al mayor número posible de peticiones en un período de tiempo. Por otro lado, esto se contradice con el resto de requisitos de seguridad que suelen consistir en comprobaciones en las que se realizan cálculos, por lo general, complejos. Por tanto, es conveniente que el sistema MAS integre otras herramientas de detección y monitorización de red que asistan a detectar y mitigar posibles amenazas, especialmente contra la DoS cuando se observen comportamientos de tráfico de red sospechosos, tales como IDS, IPS, SIEM, firewalls, etc. Evidentemente, estas soluciones deben estar bajo políticas de seguridad siguiendo marcos políticos y legales, en el que se deben cumplir las normativas existentes (p. ej., el Real Decreto 43/2021 [24]), especialmente cuando existes irregularidades que pueden afectar a otras organizaciones relacionadas. Por ello, es fundamental activar las medidas de inteligencia, y establecer contacto con los equipos de respuesta ante incidencias, como pueden ser el INCIBE-CERT [25] o el CCN-CERT [26].

Como ya se comentado, cada acción que realice un agente SW dentro del sistema y para cada evento que detecte, deberá quedar registrado de forma que se identifique inequívocamente el autor de esa entrada del registro. Esto puede lograrse mediante técnicas de firma digital y funciones hash. El objetivo es evitar que un agente bajo el control del atacante pueda negar haber realizado una acción (p. ej. enviar cierta información a otro agente SW). Es más, al tratarse de un sistema distribuido

y colaborativo, nuestro sistema MAS dispondrá de más de un archivo de registro, por lo que un mismo evento puede quedar registrado por dos agentes distintos. Esto mejora la protección frente a ataques de rechazo.

VI. Conclusiones y trabajo futuro

Este trabajo ha recopilado un análisis extendido sobre las posibles amenazas en sistemas multi-agente, desplegados en infraestructuras de carga de vehículos eléctricos teniendo presente la arquitectura estandarizada de OCPP. La infraestructura incluye no solo la comunicación específica de OCPP para el control y la administración de energía, sino que, además, incluye una red de información paralela basada en agentes SW para contribuir y precisar en las tareas de monitorización, mantenimiento y seguridad. A través de la interacción con los respectivos agentes, se busca la forma de maximizar la consciencia situacional y la protección proactiva mediante la retroalimentación de otros sistemas esenciales de seguridad, como IDS/IPS, SIEM o SOC. Sin embargo, en este escenario de aplicación es también de vital importancia analizar previamente las amenazas que pueden traer esta tecnología al contexto mencionado, priorizando aquellas que pueden impactar principalmente en la gestión y distribución de energía a los usuarios finales. Es por ello, que este artículo científico incluye una agrupación de agentes de acuerdo a los principales elementos de control según el protocolo OCPP y en base a las funciones e interacciones que pueden tener los agentes dentro de dicho escenario. En base a esta agrupación, se ha identificado diversos ataques siguiendo las recomendaciones dadas por el SP 800-19 definido por el NIST, y se ha modelado formalmente dichas amenazas en base a la metodología tradicional de árboles de ataques.

Concluimos que es fundamental considerar la influencia positiva del uso de agentes, pero priorizando los riesgos que puede conllevar su uso, ofreciendo a la comunidad científica y a la industria una base por la cual entender dichos riesgos e identificar posibles medidas de protección, también proporcionadas en este artículo. En el futuro, pretendemos, por un lado, integrar el MAS dentro de una infraestructura de carga real y dentro del proyecto "Smart and Secure EV Urban Lab II", y, por otro lado, automatizar la metodología para que los árboles de ataques puedan ser integrados como parte de la monitorización de esta infraestructura, extrayendo métricas y modos de evaluación.

Agradecimientos

Trabajo financiado por el proyecto "Smart and Secure EV Urban Lab II", perteneciente al II Plan Propio Smart Campus de la Universidad de Málaga, y por el proyecto SAVE (P18-TP-3724), perteneciente a la Junta de Andalucía.

Referencias

[1] Meticulous Research, "Electric Vehicle Market Worth $2,495.4 Billion and 233.9 Millions Units by 2027," https://www.meticulousresearch.com, 2021.

[2] Open Charge Alliance. Open Charge Point Prorocol (OCPP) 2.0.1. https://www.openchargealliance.org/protocols/ocpp-201/.

[3] H. ElHusseini, C. Assi, B. Moussa, R. Attallah, and A. Ghrayeb, "Blockchain, AI and Smart Grids: The three musketeers to a decentralized EV charging infrastructure," *IEEE Internet of Things Magazine*, vol. 3, no. 2, pp. 24–29, 2020.

[4] M. Pagani, W. Korosec, N. Chokani, and R. S. Abhari, "User behaviour and electric vehicle charging infrastructure: An agent-based model assessment," *Applied Energy*, vol. 254, p. 113680, 2019.

[5] J. Miranda, J. Borges, D. Valério, and M. J. Mendes, "Multi-agent management system for electric vehicle charging," *International Transactions on Electrical Energy Systems*, vol. 25, no. 5, pp. 770–788, 2015.

[6] C. B. Saner, A. Trivedi, and D. Srinivasan, "A Cooperative Hierarchical Multi-Agent System for EV Charging Scheduling in Presence of Multiple Charging Stations," *IEEE Transactions on Smart Grid*, 2022.

[7] K. Chaudhari, N. K. Kandasamy, A. Krishnan, A. Ukil, and H. B. Gooi, "Agent-based aggregated behavior modeling for electric vehicle charging load," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 856–868, 2018.

[8] E. Y. ElBanhawy, R. Dalton, E. M. Thompson, and R. Kotter, "A heuristic approach for investigating the integration of electric mobility charging infrastructure in metropolitan areas: An agent-based modeling simulation," in *2012 2nd International Symposium On Environment Friendly Energies And Applications*. IEEE, 2012, pp. 74–86.

[9] W. Jansen and T. Karygiannis, "Mobile agent security, SP 800-19," National Institute of Standards and Tecnology, Tech. Rep., 1998.

[10] AMPECO, "Enable innovation and cost efficiency with OCPP," https://www.ampeco.com, 2022.

[11] C. Alcaraz, J. Lopez, and S. Wolthunsen, "OCPP Protocol: Security Threats and Challenges," *IEEE Transactions on Smart Grid*, vol. 8, pp. 2452 – 2459, 02/2017 2017.

[12] J. E. Rubio, C. Alcaraz, and J. Lopez, "Addressing Security in OCPP: Protection Against Man-in-the-Middle Attacks," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2018, pp. 1–5.

[13] T. Jager, J. Schwenk, and J. Somorovsky, "On the Security of TLS 1.3 and QUIC Against Weaknesses in PKCS1 v1.5 Encryption," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1185–1196. [Online]. Available: https://doi.org/10.1145/2810103.2813657

[14] T. Tsar, R. Alexander, M. Dohler, V. Daza, A. Lozano, and M. Richardson, "A Security Threat Analysis for Routing Protocol for Low-power and lossy networks (RPL)," Routing Over Low-Power and Lossy Networks, IETF, Tech. Rep., 2014.

[15] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "STRIDE-based threat modeling for cyber-physical systems," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE, 2017, pp. 1–6.

[16] W. Young and N. G. Leveson, "An integrated approach to safety and security based on systems theory," *Communications of the ACM*, vol. 57, no. 2, pp. 31–35, 2014.

[17] C. Alberts, A. Dorofee, J. Stevens, and C. Woody, "Introduction to the OCTAVE Approach," Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, Tech. Rep., 2003.

[18] S. Haque, M. Keffeler, and T. Atkison, "An evolutionary approach of attack graphs and attack trees: A survey of attack modeling," in *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer, 2017, pp. 224–229.

[19] D. Vitkus, J. Salter, N. Goranin, and D. Čeponis, "Method for Attack Tree Data Transformation and Import Into IT Risk Analysis Expert Systems," *Applied Sciences*, vol. 10, no. 23, p. 8423, 2020.

[20] S. Pasandideh, L. Gomes, and P. Maló, "Improving attack trees analysis using Petri net modeling of cyber-attacks," in *28th International Symposium on Industrial Electronics (ISIE)*. IEEE, 2019, pp. 1644–1649.

[21] G. Dalton, R. F. Mills, J. M. Colombi, R. A. Raines *et al.*, "Analyzing attack trees using generalized stochastic petri nets," in *Information Assurance Workshop*. IEEE, 2006, pp. 116–123.

[22] D. Beaulaton, N. B. Said, I. Cristescu, and S. Sadou, "Security analysis of IoT systems using attack trees," in *International Workshop on Graphical Models for Security*. Springer, 2019, pp. 68–94.

[23] D. Kim, Y.-H. Kim, D. Shin, and D. Shin, "Fast attack detection system using log analysis and attack tree generation," *Cluster Computing*, vol. 22, no. 1, pp. 1827–1835, 2019.

[24] Agencia Estatal Boletín Oficial del Estado, "Real Decreto 43/2021, de 26 de enero, por el que se desarrolla el Real Decreto-ley 12/2018, de 7 de septiembre, de seguridad de las redes y sistemas de información," https://www.boe.es/diario_boe/txt.php?id=BOE-A-2021-1192, 2018.

[25] Instituto Nacional de Ciberseguridad de España, "INCIBE-CERT," https://www.incibe-cert.es, 2022.

[26] Centro Criptológico Nacional, "CCN-CERT," https://www.ccn-cert.cni.es, 2022.

Aleatorización de direcciones IP para mitigar ataques de reconocimiento de forma proactiva en sistemas de control industrial

Xabier Etxezarreta, Iñaki Garitano, Mikel Iturbe y Urko Zurutuza

Departamento de Electrónica e Informática
Escuela Politécnica Superior
Mondragon Unibertsitatea
Goiru 2, E-20500 Arrasate-Mondragón
Email: {xetxezarreta,igaritano,miturbe,uzurutuza}@mondragon.edu

***Resumen*—Los sistemas de control industrial se utilizan en una gran variedad de procesos físicos, incluidas las infraestructuras críticas, convirtiéndose en el principal objetivo de múltiples ataques de seguridad. Un ataque malintencionado y exitoso contra estas infraestructuras podría causar graves consecuencias económicas y ambientales, incluyendo la pérdida de vidas humanas. Las redes estáticas, que caracterizan a los sistemas de control industrial, suponen una ventaja para los atacantes, permitiéndola explorar en busca de dispositivos o servicios vulnerables antes de realizar el ataque. Identificar dispositivos activos suele ser el primer paso para muchos ataques. Este trabajo presenta un sistema de defensa ante reconocimientos de red que se basa en la aleatorización temporal de las direcciones de red. La distorsión de la información obtenida inhabilita el conocimiento adquirido por parte de los atacantes dificultando así cualquier ataque que se apoya en el direccionamiento de la red. La aleatorización temporal de las direcciones de red se realiza de forma adaptativa minimizando así la sobrecarga introducida en la red y evitando cualquier error y latencia en las comunicaciones. La implementación así como las pruebas se han realizado en un laboratorio con equipamiento industrial real, demostrando así la efectividad de la solución presentada.**

***Index Terms*—Sistemas de control industrial, Moving Target Defense, Redes definidas por software, Seguridad en redes industriales**

Tipo de contribución: *Investigación original (límite 8 páginas)*

I. Introducción

Sistema de control industrial (ICS por sus siglas en inglés) es un término general que cubre varios elementos especializados utilizados para la monitorización y control de procesos industriales [1]. Están compuestos por diversos elementos como sensores, actuadores, controladores lógicos programables (PLC) o sistemas de control supervisor y adquisición de datos (SCADA). Se pueden encontrar en todo tipo de industrias, incluidas en las infraestructuras críticas, convirtiéndose en elementos imprescindibles para el bienestar y desarrollo económico de la sociedad. Ejemplos de infraestructura crítica incluyen centrales nucleares, sistemas de transporte, redes eléctricas, presas hidroeléctricas y plantas de fabricación críticas.

Tradicionalmente, los ICS han estado desplegados en entornos aislados, utilizando protocolos de comunicación y hardware propietarios. El aislamiento y la oscuridad han sido los pilares en los que se ha basado la seguridad en ICS, pero la integración de tecnología de la información (IT) ha expuesto los originalmente aislados ICS a las redes corporativas, incluyendo Internet. Este cambio de tendencia ha hecho que las técnicas tradicionales de aislamiento y seguridad por oscuridad dejen de ser efectivas en estos entornos. La naturaleza de los ICS hace difícil que las soluciones de seguridad IT cumplan con los requisitos de estos sistemas. Esto hace que sea necesario desarrollar soluciones de seguridad específicas para estos entornos. De acuerdo con la publicación NIST SP 800-82 Rev 2 [1], los ICS se diferencian de los sistemas IT en los siguientes aspectos:

- Los ICS se utilizan para controlar y monitorizar procesos y dispositivos físicos.
- Una interrupción no es aceptable. La disponibilidad es prioritaria frente a la integridad y la confidencialidad.
- El tiempo es crítico en los ICS, la latencia en las comunicaciones tiene que ser mínima.
- El periodo de sustitución/actualización de los dispositivos que componen los ICS es muy largo en comparación con los sistemas IT.
- La aplicación de parches de seguridad muchas veces es pospuesta debido a las necesidades de disponibilidad y fiabilidad.
- En muchos casos los ICS no tienen capacidad de integrar mecanismos de seguridad.

Si lo comparamos con las redes IT, las topologías de red industriales son generalmente estáticas y el tráfico de control es por naturaleza repetitiva y predecible, debido a que la mayor parte del tráfico es creado por procesos automatizados [2]. Esta característica estática de las redes industriales supone un escenario ventajoso para el atacante, permitiendo explorar vulnerabilidades antes de realizar el ataque. Debido a este problema, una tendencia de soluciones de seguridad proactivas empezaron a desarrollarse en respuesta a estos sistemas estáticos bajo el nombre de Moving Target Defense (MTD). MTD se puede definir como un sistema en constante cambio que desplaza o reduce la superficie de ataque, dificultando que un atacante pueda explorar y explotar vulnerabilidades fácilmente.

Las redes definidas por software (SDN) se han convertido en una tecnología prometedora para ICS, tanto para desarrollar

técnicas de MTD [3] como para el desarrollo de herramientas de detección y respuesta a intrusiones en general [4]. En el RFC 7140 [5] se define SDN como un conjunto de técnicas utilizadas para facilitar el diseño, la entrega y la operación de servicios de red de una forma determinista, dinámica y escalable. Para esto, el plano de control es separado y centralizado, mientras que el plano de datos se mantiene en los dispositivos de red, centrando su funcionalidad al reenvío de paquetes. En la figura 1 se representan los principales planos de la arquitectura SDN. En primer lugar, el plano de datos es donde se aplican las decisiones de reenvío y se procesa el tráfico. En segundo lugar se encuentra el plano de control, utilizado para proporcionar lógica al plano de datos mediante el uso de controladores SDN. La comunicación entre el plano de datos y de control se realiza mediante la *interfaz southbound* con protocolos específicos en los que OpenFlow es el principal exponente, pero existen otros (ej. NETCONF). Por último, el plano de aplicación es utilizado para desarrollar aplicaciones de negocio que interaccionan con la red mediante la *interfaz northbound*.

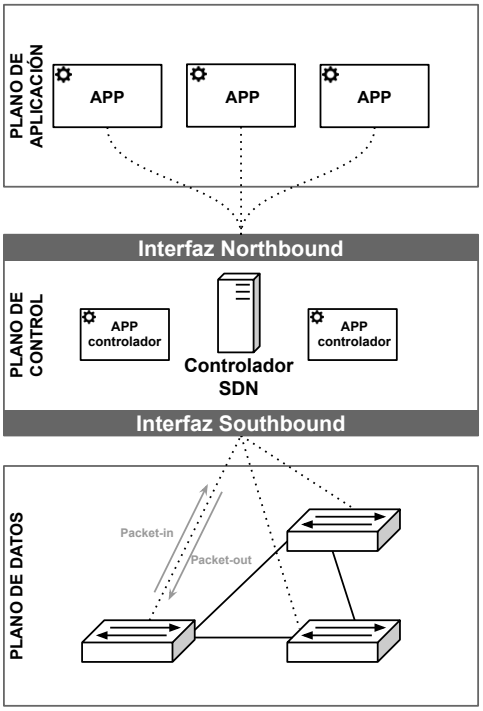


Figura 1: Arquitectura SDN.

Las redes tradicionales no están optimizadas para satisfacer las necesidades presentes y futuras de los ICS, que demandan más flexibilidad sin comprometer la calidad el servicio [6]. Desde el punto de vista del desarrollo de soluciones de detección y respuesta de intrusiones, las SDN ofrecen ventajas respecto a las redes tradicionales en los siguientes aspectos: (1) ofrecen una visión global sobre la red, (2) aumentan la programabilidad de la red integrando aplicaciones desarrolladas por el usuario y (3) permiten modificar el comportamiento de los flujos de red de forma dinámica.

En este artículo, se combinan los conceptos de SDN y MTD para desarrollar un mecanismo de defensa proactiva que sirve para mitigar ataques de reconocimiento en redes de control industrial. Las principales contribuciones de este trabajo se pueden resumir en los siguientes puntos:

- Desarrollo de un sistema MTD que aleatoriza las direcciones IP del tráfico de red en tiempo real, distorsionando la información obtenida por un atacante durante la fase de reconocimiento e impidiendo el acceso directo a los dispositivos mediante el uso de IPs reales. La aleatorización se realiza en los switches y es completamente transparente para los dispositivos finales de la red.
- La inicialización de las reglas de flujo que aleatorizan las direcciones IP y reenvían el tráfico a su destino se realiza mediante un *allowlist* predefinido. En este *allowlist* se define qué dispositivos están autorizados para comunicarse entre ellos. En base a esta información, se instalan las reglas de flujo pertinentes en los switches.
- La aleatorización de las direcciones IP se realiza de forma adaptativa, minimizando la latencia introducida y cumpliendo con los requisitos de tiempo real de las redes de control industrial. Para esto se hace uso de reglas de flujo de respaldo y el campo *priority* del protocolo OpenFlow.
- La solución se ha implementado y probado en un entorno con equipamiento industrial real, demostrando así la efectividad de la solución presentada.

II. Trabajos relacionados

Esta sección presenta una breve discusión acerca de los trabajos relacionados. Por un lado, se proporciona información acerca de las diferentes técnicas MTD de la literatura. Por otro lado, se discute la aplicabilidad de MTD en sistemas de control industrial.

II-A. Técnicas de Moving Target Defense

Las técnicas MTD tienen como objetivo cambiar la naturaleza estática de las redes modificando la superficie de ataque de forma dinámica. Estas técnicas pueden ser clasificadas en los siguientes cuatro grupos [7]:

MTD basado en *Shuffling*: Son técnicas que aleatorizan la configuración de la red para hacerla menos predecible y disminuir la superficie de ataque. En este grupo podemos encontrar técnicas que aleatorizan las direcciones IP [8], [9], [10], puertos [11], rutas por donde fluye el tráfico de red [12] o la cabecera de los paquetes [13]. También existen soluciones que combinan varias técnicas MTD a la vez basadas en *shuffling* [14].

MTD basado en Diversidad: Consiste en proporcionar servicios equivalentes pero con diferentes implementaciones. La diversidad de código tiene como objetivo dividir un programa en componentes que pueden ser implementados en diferentes entornos de ejecución [15]. Las técnicas de diversidad de software despliegan variantes equivalentes de servidores web, aplicaciones o servidores virtuales para mejorar la resiliencia de la red [16]. Por último, las técnicas de diversidad en lenguajes de programación permiten mitigar ataques de código o inyección SQL [17].

MTD basado en Redundancia: Consiste en desplegar réplicas que ofrecen la misma funcionalidad. Podemos encontrar técnicas que proporcionan redundancia en sesiones de red [18] o que despliegan réplicas de servidores con la misma funcionalidad [19].

MTD Híbrido: Estas técnicas combinan MTD basado en *Shuffling*, Diversidad y Redundancia [20], [21].

II-B. MTD en sistemas de control industrial

Las técnicas MTD han sido introducidas en sistemas de control industrial para revertir la naturaleza estática y predecible de estos sistemas. La investigación se ha centrado en desarrollar y adaptar técnicas MTD basadas en *shuffling*, especialmente técnicas que aleatorizan las direcciones IP y las rutas por donde fluye tráfico. Por un lado, las técnicas de aleatorización de direcciones IP para sistemas de control industrial de la literatura aprovechan la utilidad IPTables para realizar operaciones de traducción de direcciones. En los artículos [22] y [23] se puede observar que conforme el intervalo de aleatorización disminuye, el *Round Trip Time (RTT)* aumenta considerablemente, convirtiéndose en un problema para los sistemas que requieren unas comunicaciones con latencias bajas. Por otro lado, en relación con la aleatorización de rutas, los autores en [24] proponen una técnica para evitar que todo el tráfico vaya por el mismo camino, dispersando el tráfico por múltiples rutas para defenderse de intercepciones de tráfico no autorizadas. Como la transición a rutas alternativas no se realiza de forma adaptativa, los autores en [25] proponen utilizar el campo *hard-timeout* del protocolo OpenFlow para definir varias reglas de flujo a la vez y minimizar la latencia introducida en la red industrial basada en SDN.

A diferencia de las publicaciones existentes, este trabajo combina la tecnología SDN y la aleatorización de direcciones IP para desarrollar un mecanismo de defensa proactiva que puede ser implementado en entornos sensibles a la latencia o retardo en las comunicaciones como los sistemas de control industrial. Para esto, el proceso de aleatorización se implementa en los dispositivos de reenvío y la transición a diferentes direcciones IP aleatorias se realiza de forma adaptativa utilizando reglas de respaldo y el campo *priority* del protocolo OpenFlow.

III. Framework

En esta sección se presenta el mecanismo de defensa proactiva que proporciona una aleatorización de direcciones IP para una red de control industrial. Primero se detalla la arquitectura y las partes en la que está compuesta. En segundo lugar, se detalla la inicialización del sistema mediante el uso de un *allowlist*. Por último se define como se implementa de forma adaptativa la transición a nuevas direcciones IP aleatorias.

III-A. Arquitectura

La arquitectura está desarrollada para ser integrada y utilizada en un entorno industrial basado en SDN. En la figura 2 se representa la visión general de la arquitectura de aleatorización de direcciones IP. Los principales componentes de esta arquitectura son los siguientes:

- *Dispositivos finales:* Componen una gran variedad de dispositivos utilizados en entornos industriales como servidores SCADA, PLCs o estaciones de trabajo.
- *Switches OpenFlow:* Estos dispositivos de reenvío tienen como función enrutar el tráfico a su destino, en base a reglas de flujo existentes en sus tablas de enrutamiento. En el caso de la presente solución, estos dispositivos se utilizan para procesar los paquetes de red y aleatorizar las direcciones IP. Para esto, se instalan reglas de flujo que aplican acciones en paquetes en base a coincidencias.
- *Controlador SDN:* Es el responsable de la comunicación entre el plano de datos y plano de aplicación. El controlador obtiene las peticiones del módulo MTD y las transmite a los switches utilizando el protocolo OpenFlow.
- *Módulo MTD:* Es un programa desplegado en el plano de aplicación. Tiene como función inicializar y actualizar las reglas de flujo en los switches OpenFlow que realizan las traducciones entre direcciones IP reales y aleatorias. Esto se realiza utilizando la *interfaz northbound* del controlador SDN, en este caso una API REST.

III-B. Aleatorización de direcciones IP

En la fase de inicialización, las reglas de flujo iniciales que se instalan en los switches están basadas en un *allowlist* definido por el usuario. Aprovechando la naturaleza estática de las redes industriales y lo repetitivas y predecibles que son las comunicaciones del tráfico de control, en este *allowlist* se recogen las comunicaciones entre los dispositivos de la red que están autorizadas. Si una comunicación está autorizada, los dispositivos podrán comunicarse entre ellos utilizando IPs reales. Por el contrario, en comunicaciones no autorizadas, un dispositivo solo podrá comunicarse con otro dispositivo utilizando la dirección IP aleatoria asignada en ese momento al dispositivo de destino.

El primer paso consiste en generar y asignar una dirección IP aleatoria a cada dispositivo presente en la red. Estas direcciones IP asignadas solo serán válidas durante un intervalo MTD y serán reemplazadas por otras IPs aleatorias en el siguiente intervalo MTD. El intervalo MTD es definido por el usuario y tiene que ser adaptado para cada caso de uso. Al generar las IPs aleatorias se comprueba que no se hayan generado dos iguales, de esta manera se evita que haya colisiones y posibles errores en el funcionamiento del sistema. Hay que destacar que la configuración de red de los dispositivos finales no se cambia, las traducciones de direcciones IP se realizan en los switches y el proceso es completamente transparente.

Con las direcciones IP aleatorias generadas para cada dispositivo, se instalan reglas de flujo en los switches OpenFlow para que solo los dispositivos autorizados puedan comunicarse mediante el uso de direcciones IP reales. Existen dos opciones para cambiar o aleatorizar las direcciones IP de los paquetes: (1) enviar todos los paquetes al controlador SDN para que sean procesados de forma centralizada o (2) procesar los paquetes directamente en los switches. Para minimizar el retardo introducido en una comunicación, las reglas de flujo procesan el paquete directamente en el switch, evitando que cada paquete sea enviado al controlador SDN para su procesamiento. La tabla I representa una tabla de flujos de un switch OpenFlow. Para la comunicación entre dos dispositivos autorizados son necesarias ocho entradas en las tablas de flujo, cuatro en el switch de origen y cuatro en el switch de destino. Los paquetes IP y ARP son procesados con reglas de flujos independientes, ya que como se especifica en la especificación del protocolo OpenFlow [26], es necesario

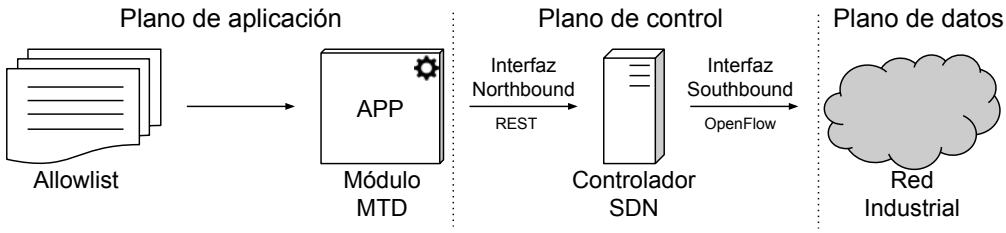


Figura 2: Visión general del sistema de aleatorización de direcciones IP.

definir el tipo de *frame ethernet* (*eth_type*) en la columna *match* para poder procesar paquetes con valores de la cabecera del protocolo IP. A cada tipo de paquete (IP o ARP) le pertenecen dos reglas de flujo en cada switch; uno para traducir las IPs reales a IPs aleatorias y otro para traducir las IPs aleatorias a IPs reales. Para todos los demás dispositivos no autorizados, solo son necesarios cuatro reglas de flujo que permiten la comunicación utilizando las direcciones IP aleatorias asignadas en ese momento.

Una vez que las reglas de flujo estén instaladas en los switches, el proceso que sigue cada paquete que va desde un origen a un destino está definido en el Algoritmo 1. Imaginemos una comunicación entre dos dispositivos finales h_1 y h_2. Cuando un paquete llega al switch OpenFlow de origen, es decir, al switch al que está conectado h_1, las direcciones IP reales (rIP) son cambiadas por IPs aleatorias (vIP) en caso de que la comunicación entre h_1 y h_2 esté autorizada. En caso contrario, solo se cambia la dirección IP de origen y se le da salida al paquete por un puerto del switch. Cuando el paquete llega al switch OpenFlow de destino, es decir, al switch al que está conectado h_2, las direcciones IP aleatorias vIP son traducidas a IPs reales rIP en caso de que la comunicación entre h_1 y h_2 esté autorizada. De lo contrario, si h_1 no tiene autorización para comunicarse con h_2 y la IP de destino utilizada por h_1 no coincide con la IP aleatoria asignada a h_2 en ese intervalo, el paquete es descartado evitando que llegue a su destino. Si la IP de destino utilizada por h_1 es la IP aleatoria asignada a h_2 en ese momento, el paquete es reenviado a su destino cambiando la IP aleatoria por la IP real del dispositivo de destino.

Algorithm 1 Proceso seguido por cada paquete para llegar a su destino.

for all packets pkt from h_1 to h_2 **do**
  if pkt is at src switch **then**
    set $pkt.src = vIP(h_1)$
    if h_1 is authorized to h_2 **then**
      set $pkt.dst = vIP(h_2)$
    end if
    $output \leftarrow out\_port$
  else if pkt is at dst switch **then**
    if h_1 is authorized to h_2 **then**
      set $pkt.src = rIP(h_1)$
      set $pkt.dst = rIP(h_2)$
      $output \leftarrow out\_port$
    else
      if $pkt.dst$ is $vIP(h_2)$ **then**
        set $pkt.dst = rIP(h_2)$
        $output \leftarrow out\_port$
      else
        $drop \leftarrow pkt$
      end if
    end if
  else
    $output \leftarrow out\_port$
  end if
end for

III-C. Transición adaptativa entre intervalos

Las direcciones IP aleatorias asignadas solo son válidas durante un intervalo de tiempo definido por el usuario. Al final de cada intervalo, las direcciones IP cambian por otras nuevas generadas aleatoriamente y son asignadas a cada dispositivo final, enrutando el tráfico utilizando las nuevas direcciones IP. Si las reglas de flujo que están siendo utilizadas son directamente borradas o actualizadas, las comunicaciones que aún estén utilizando las direcciones IPs aleatorias del intervalo anterior pueden ser enrutadas de forma incorrecta, generando una interrupción en la red. Para solventar este problema y evitar la introducción de retardos o interrupciones en la red, se ha diseñado un método de actualización de reglas de flujo que hace uso de reglas de respaldo y del campo *priority* del protocolo OpenFlow. De esta manera, conseguimos una transición adaptativa a unas direcciones IP aleatorias nuevas. La figura 3 muestra el proceso seguido al final de cada intervalo MTD y consta de cuatro fases.

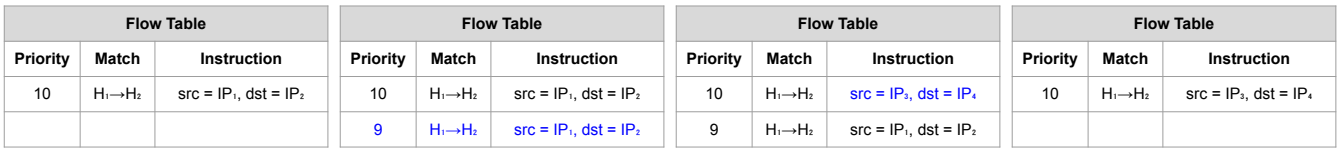
La primera fase, representada en la subfigura 3(a), hace referencia al estado de las reglas de flujo al final de cada intervalo MTD. En este estado, las reglas de flujo hacen traducciones entre IPs reales y aleatorias (y viceversa) asignadas en ese intervalo activo.

Cuando un intervalo MTD llega a su fin, se generan nuevas direcciones IP aleatorias que serán utilizadas en el siguiente intervalo. Si las reglas de flujo activas son actualizadas o eliminadas, el tráfico que aún esté utilizando las direcciones IP del intervalo anterior puede ser descartado o forzado a que pase por el controlador SDN. Esto se debe a que ese tráfico no dispone de ninguna regla de flujo para hacer *match*. En sistemas de control industrial donde el tiempo es crítico, interrumpir o introducir retardo en el tráfico no es aceptable. Para evitar estos problemas, al final de cada intervalo, se genera una regla de respaldo por cada regla de flujo activa en cada switch. Como se representa en la subfigura 3(b), la regla de respaldo es una copia de la regla de flujo activa pero con menor prioridad.

Con las reglas de respaldo, se procede a actualizar las reglas activas asignando las nuevas direcciones IP aleatorias

Tabla I: Reglas en una tabla de flujos de un switch OpenFlow.

Switch Flow Table		
Priority	**Match**	**Instruction**
10	src = $rIP(h_1)$, dst = $rIP(h_2)$, eth_type = IP	src = $vIP(h_1)$, dst = $vIP(h_2)$, ouput($port$)
10	src = $vIP(h_2)$, dst = $vIP(h_1)$, eth_type = IP	src = $rIP(h_2)$, dst = $rIP(h_1)$, ouput($port$)
10	src = $rIP(h_1)$, dst = $rIP(h_2)$, eth_type = ARP	src = $vIP(h_1)$, dst = $vIP(h_2)$, output($port$)
10	src = $vIP(h_2)$, dst = $vIP(h_1)$, eth_type = ARP	src = $rIP(h_2)$, dst = $rIP(h_1)$, output($port$)
5	src = $rIP(h_1)$, dst = $vIP(h_3)$, eth_type = IP	src = $vIP(h_1)$, output($port$)
5	src = $vIP(h_3)$, dst = $vIP(h_1)$, eth_type = IP	dst = $rIP(h_1)$, output($port$)
5	src = $rIP(h_1)$, dst = $vIP(h_3)$, eth_type = ARP	src = $vIP(h_1)$, output($port$)
5	src = $vIP(h_3)$, dst = $vIP(h_1)$, eth_type = ARP	dst = $rIP(h_1)$, output($port$)
0	any	drop



(a) Estado inicial.

(b) Se añade una regla de flujo respaldo con la misma instrucción, pero con menor prioridad.

(c) Se actualiza la regla con mayor prioridad asignando nuevas direcciones IP aleatorias.

(d) Se elimina la regla de flujo de respaldo.

Figura 3: Proceso de actualización de las reglas de flujo al final de un intervalo MTD.

generadas para el nuevo intervalo. Esta fase está representada en la subfigura 3(c). En este estado, el tráfico nuevo empezará a utilizar las direcciones IP aleatorias del nuevo intervalo, mientras que el tráfico generado en el intervalo anterior utilizará las reglas de flujo de respaldo. Con este método, se consigue una transición entre intervalos adaptativa, sin generar retardos ni pérdida de paquetes.

Por último, como se muestra en la subfigura 3(d), se eliminan las reglas de respaldo, volviendo al estado inicial. Este proceso se aplica de manera iterativa al final de cada intervalo, independientemente del tiempo de intervalo definido por el usuario.

IV. Resultados y discusión

Para probar la viabilidad del sistema de aleatorización de direcciones IP en un sistema de control industrial, se ha desplegado un entorno experimental industrial con equipamiento real: PLC AC800M y servidor SCADA, ambos de ABB. El banco de pruebas experimental está basado en una emulación de un sistema de entrada de un almacén. La entrada del almacén está formada por dos puertas correderas que se abren cuando un operario pasa por delante de un sensor de movimiento. Cuando el sensor de movimiento detecta el paso de un operario, las puertas se abren y se mantienen abiertas durante cinco segundos hasta que se vuelven a cerrar. Si el sensor detecta a un operario mientras la puerta está abierta, el tiempo de apertura de la puerta se prolonga cinco segundos más. Durante este proceso, se almacenan dos variables: (1) número de operarios que entran y (2) número de veces que se abre la puerta.

La figura 4 representa la topología del entorno de experimentación. En primer lugar, la emulación de la entrada del almacén se ejecuta en un PLC propietario de ABB AC800M ubicado en la red de control y está conectado directamente a un switch OpenFlow. En segundo lugar, en la red de supervisión, se ha desplegado un servidor SCADA ABB que está conectado a otro switch OpenFlow y solicita datos al PLC. La comunicación entre estos dispositivos ABB se realiza mediante el protocolo Manufacturing Message Specification (MMS). En tercer lugar, se ha desplegado un servidor para simular un potencial atacante. Como el resto de dispositivos, el servidor del atacante también está conectado a su propio switch. En último lugar, como controlador SDN se ha utilizado Ryu [27], la cual su *interfaz northbound* es utilizada por el módulo MTD para instalar y actualizar reglas de flujo en los switches OpenFlow.

IV-A. Rendimiento

Round Trip Time: Para ver la sobrecarga que introduce en la red el sistema de defensa MTD presentado en este artículo, se ha utilizado la medida *Round Trip Time (RTT)*. El RTT mide el tiempo necesario para que un paquete vaya y vuelva de un receptor. La solución se ha probado con la topología de experimentación en seis escenarios diferentes; por un lado, se han realizado las mediciones en una red estática sin aplicar MTD, simulando un estado normal, y por otro lado se ha utilizado la técnica de aleatorización de direcciones IP con intervalos de 60, 30, 10, 5 y 1 segundo. Para cada escenario, se han realizado 15 mediciones de 200 segundos de duración en cada una de ellas. En la tabla II se representan la media, desviación estándar y valores mínimo/máximo de RTT de los resultados de las mediciones.

Tabla II: RTT entre el servidor SCADA y el PLC con diferentes intervalos MTD.

		Intervalo MTD					
		Sin MTD	60s	30s	10s	5s	1s
RTT (ms)	avg	5.108	5.135	5.145	5.147	5.147	5.169
	stdv	0.091	0.099	0.107	0.122	0.148	0.217
	min	4.73	4.788	4.801	4.819	4.814	4.824
	max	5.666	5.799	6.076	6.023	6.41	6.762

Tamaño de las tablas de flujo: La cantidad de reglas de flujo que realizan las traducciones entre direcciones IP varía dependiendo de la cantidad de dispositivos autorizados que tenga definidas un dispositivo. Para la red MTD presentada en este artículo, son necesarias 4 reglas de flujo en el

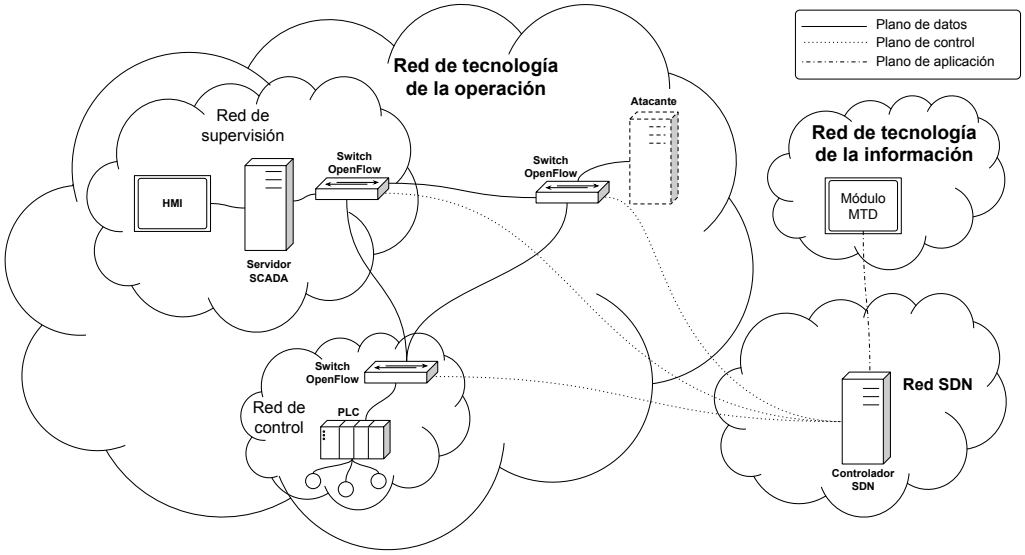


Figura 4: Topología utilizada para la evaluación.

switch de origen y el switch de destino que permitan realizar traducciones $rIP \rightarrow vIP$ y $vIP \rightarrow rIP$ para paquetes de tipo IP y ARP. En el caso de una red estática, si el reenvío de paquetes se realiza utilizando las direcciones IP/MAC de origen y destino de los dispositivos, serían suficientes 2 reglas de flujo por dispositivo autorizado.

Supongamos un conjunto de dispositivos de red $\delta \in \Delta$ conectados a un mismo switch S_w. N representa el numero de dispositivos disponibles en la red. El número necesario de reglas de flujo en una red estática y en una red MTD están definidos en las ecuaciones 1 y 2 respectivamente.

$$F_r(S_w) = 1 + \sum_{\delta \in \Delta} 2N \tag{1}$$

$$F_r^*(S_w) = 1 + \sum_{\delta \in \Delta} 4N \tag{2}$$

La figura 5 muestra la relación entre la cantidad de reglas de flujo necesarias en un switch por cada dispositivo conectado y el número de dispositivos presentes en la red.

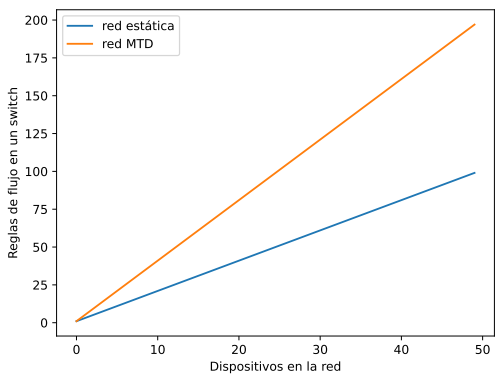


Figura 5: Relación entre número de reglas de flujo y número de dispositivos autorizados.

IV-B. Ataques de reconocimiento

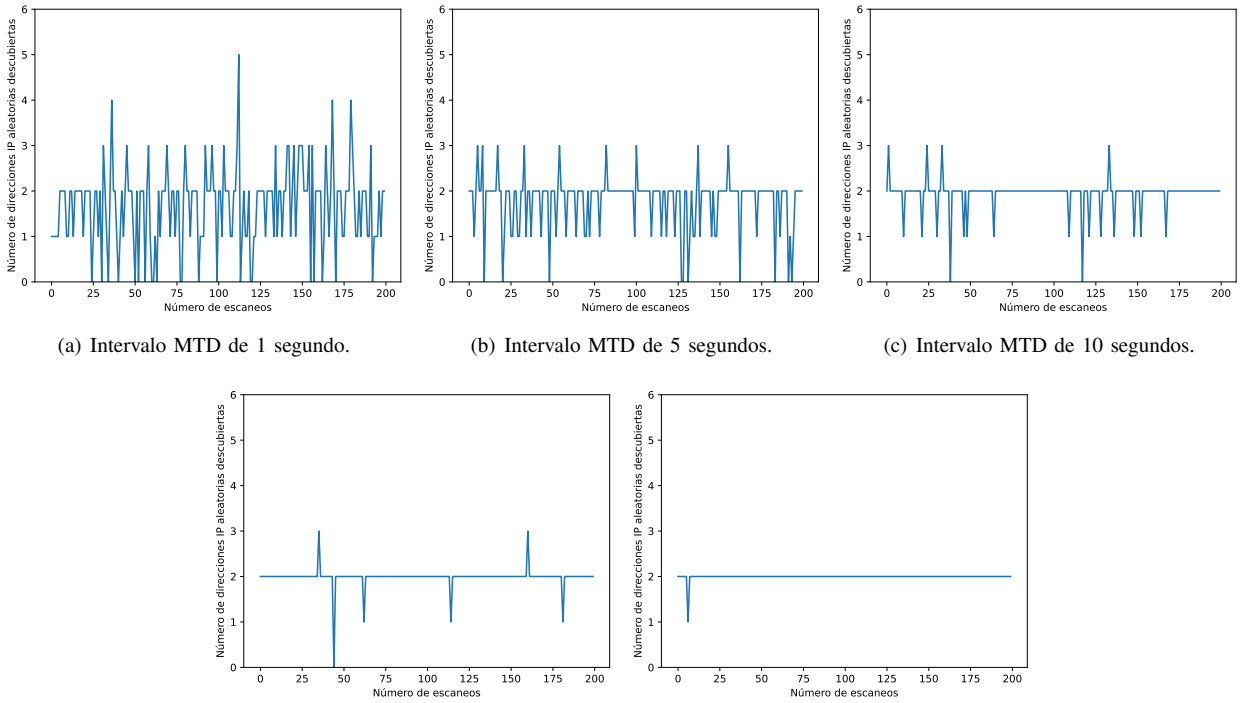
Para probar la eficacia de mitigar ataques de reconocimiento, se han realizado 200 escaneos consecutivos en nuestra red de tipo *clase A* con un espacio de direcciones IP de 2^8. Se ha utilizado la herramienta *nmap* [28] desde el servidor del atacante para realizar escaneos en busca de dispositivos activos en la red. La herramienta *nmap* dispone de múltiples métodos para descubrir dispositivos en una red. Entre estas técnicas se han utilizado y probado las siguientes: *TCP SYN Ping, TCP ACK Ping, ICMP Ping, IP Protocol Ping y ARP Ping*.

Por un lado, se han realizado escaneos en una red estática tradicional sin la defensa proactiva MTD activada. Escaneando todo el rango de direcciones IP, el atacante es capaz de identificar en cada escaneo todos los dispositivos activos de la red.

Por otro lado, con la defensa proactiva MTD activa, se han realizado 200 escaneos consecutivos con diferentes intervalos de aleatorización (1, 5, 10, 30 y 60 segundos). Las diferentes subfiguras representadas en la figura 6 reflejan el número de IPs aleatorias descubiertas por el atacante en 200 escaneos consecutivos y con diferentes intervalos MTD. Podemos observar que conforme el intervalo de aleatorización disminuye, la variación de direcciones IP aleatorias descubiertas entre diferentes escaneos es mayor. Esto se debe a que con intervalos de aleatorización mayores, aumenta la probabilidad de que un escaneo a todo el rango de direcciones IP de la red se realice dentro de los límites de ese intervalo. En intervalos de aleatorización más bajos, la probabilidad de que un escaneo completo se ejecute en más de un intervalo es mayor, aumentando la variabilidad de los resultados.

IV-C. Discusión

El rendimiento en términos de RTT es similar con diferentes intervalos MTD gracias a que la transición a nuevas IPs aleatorias se realiza de forma escalonada utilizando reglas de flujos de respaldo. El intervalo MTD a utilizar dependerá de la criticidad del sistema a defender y es un valor que tiene que ser adaptado para cada caso de uso. Con un intervalo de



(a) Intervalo MTD de 1 segundo. (b) Intervalo MTD de 5 segundos. (c) Intervalo MTD de 10 segundos.

(d) Intervalo MTD de 30 segundos. (e) Intervalo MTD de 60 segundos.

Figura 6: Número de direcciones IP aleatorias descubiertas con diferentes intervalos MTD.

aleatorización menor, la información obtenida por el atacante es más difusa y la información obtenida en reconocimientos anteriores se vuelve irrelevante de forma más frecuente. Al contrario que en intervalos más largos, la información que se obtiene es más constante, incluso se puede obtener la misma información en varios escaneos consecutivos si se realizan en un mismo intervalo MTD. Se podría cuantificar la criticidad de cada dispositivo y definir unos intervalos de aleatorización más frecuentes en las comunicaciones más críticas e intervalos menos frecuentes para comunicaciones menos críticas.

La cantidad de reglas de flujo necesarias en los switches encargadas de la traducción de las direcciones de red aumenta considerablemente en función del número de dispositivos con los que se puede comunicar un dispositivo en concreto. En estos casos, si una tabla de flujos de un switch tiene muchas entradas, el rendimiento de la red puede verse afectado negativamente y es algo a tener en cuenta especialmente en entornos sensibles al tiempo. Como alternativa y para optimizar el proceso en redes grandes donde son necesarias muchas reglas de flujo, se podría utilizar el procesamiento en *pipeline* que proporcionan los switches OpenFlow. Un switch OpenFlow puede contener varias tablas de flujo en cadena donde las reglas pueden ser instaladas en diferentes tablas. De esta forma, se podrían hacer grupos de reglas de flujo para evitar que los paquetes sean procesados por todas las entradas en una única tabla.

Por último, la seguridad por oscuridad es una técnica que utiliza el ocultamiento para proporcionar seguridad. MTD puede considerarse como una técnica de seguridad por oscuridad, especialmente las técnicas de *shuffling* que basan su seguridad en impedir que un atacante descubra posibles vectores de ataque ocultando la configuración, servicios o dispositivos disponibles en la red. Como se cita en el volumen 2 del SP 800-160 de NIST [29], la seguridad por oscuridad no puede ser el principal mecanismo de defensa. Este tipo de técnicas pueden ser utilizadas como una capa complementaria de seguridad en entornos seguros y resilientes.

V. Conclusiones y líneas futuras

Este artículo presenta un mecanismo que aleatoriza las direcciones IP en comunicaciones industriales utilizando el paradigma SDN. El objetivo principal de este sistema es mitigar ataques de reconocimiento y evitar que un equipo no autorizado pueda comunicarse con su objetivo de manera convencional al perder la información obtenida en ataques anteriores. Esto se consigue asignando una dirección IP aleatoria a cada dispositivo de la red que solo son válidas durante un periodo de tiempo limitado. Los resultados demuestran que este sistema ayuda a que la información obtenida a través de ataques de reconocimiento pierda relevancia debido a que solo es válida temporalmente. También, gracias a que la transición a nuevas direcciones IP aleatorias se realiza de forma adaptativa, el retardo introducido en comparación a una red estática tradicional es mínimo, permitiendo que la solución pueda ser implementada en sistemas donde el tiempo es crítico.

Como líneas futuras, tenemos como objetivo desarrollar un sistema que permita detectar comportamientos o comunicaciones sospechosas en sistemas de control industrial con la aleatorización de direcciones IP activa. También, queremos explorar la posibilidad de integración de *honeypots* industriales.

AGRADECIMIENTOS

Este trabajo ha sido desarrollado por el grupo de sistemas inteligentes para sistemas industriales apoyado por el Departamento de Educación, Política Lingüística y Cultura del Gobierno Vasco (IT1676-22). Ha sido parcialmente financiado por el proyecto REMEDY. Este proyecto ha recibido financiación del Departamento de Desarrollo Económico e Infraestructuras bajo el acuerdo de concesión KK-2021/00091.

REFERENCIAS

[1] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ics) security," *NIST special publication*, vol. 800, no. 82, pp. 16–16, 2015.

[2] M. Iturbe, I. Garitano, U. Zurutuza, and R. Uribeetxeberria, "Visualizing network flows and related anomalies in industrial networks using chord diagrams and whitelisting." in *VISIGRAPP (2: IVAPP)*, 2016, pp. 101–108.

[3] J. Zheng and A. S. Namin, "A survey on the moving target defense strategies: An architectural perspective," *Journal of Computer Science and Technology*, vol. 34, no. 1, pp. 207–233, 2019.

[4] M. Sainz, M. Iturbe, I. Garitano, and U. Zurutuza, "Software defined networking opportunities for intelligent security enhancement of industrial control systems," in *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 León, Spain, September 6–8, 2017, Proceeding*, H. Pérez García, J. Alfonso-Cendón, L. Sánchez González, H. Quintián, and E. Corchado, Eds. Cham: Springer International Publishing, 2018, pp. 577–586.

[5] M. Boucadair and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment," RFC 7149, Mar. 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7149

[6] E. Molina and E. Jacob, "Software-defined networking in cyber-physical systems: A survey," *Computers & Electrical Engineering*, vol. 66, pp. 407–419, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790617313368

[7] J.-H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore, D. S. Kim, H. Lim, and F. F. Nelson, "Toward proactive, adaptive defense: A survey on moving target defense," *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 709–745, 2020.

[8] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: Transparent moving target defense using software defined networking," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: Association for Computing Machinery, 2012, pp. 127–132. [Online]. Available: https://doi.org/10.1145/2342441.2342467

[9] D. P. Sharma, D. S. Kim, S. Yoon, H. Lim, J.-H. Cho, and T. J. Moore, "Frvm: Flexible random virtual ip multiplexing in software-defined networks," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 579–587.

[10] Y. Zhou, G. Cheng, and S. Yu, "An sdn-enabled proactive defense framework for ddos mitigation in iot networks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5366–5380, 2021.

[11] A. Chowdhary, A. Alshamrani, D. Huang, and H. Liang, "Mtd analysis and evaluation framework in software defined network (mason)," in *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, ser. SDN-NFV Sec'18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 43–48. [Online]. Available: https://doi.org/10.1145/3180465.3180473

[12] A. Aydeger, M. H. Manshaei, M. A. Rahman, and K. Akkaya, "Strategic defense against stealthy link flooding attacks: A signaling game approach," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 751–764, 2021.

[13] Y. Wang, Q. Chen, J. Yi, and J. Guo, "U-tri: Unlinkability through random identifier for sdn network," in *Proceedings of the 2017 Workshop on Moving Target Defense*, ser. MTD '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 3–15. [Online]. Available: https://doi.org/10.1145/3140549.3140554

[14] A. R. Chavez, W. M. Stout, and S. Peisert, "Techniques for the dynamic randomization of network attributes," in *2015 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2015, pp. 1–6.

[15] H. Koo, Y. Chen, L. Lu, V. P. Kemerlis, and M. Polychronakis, "Compiler-assisted code randomization," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 461–477.

[16] Y. Huang and A. K. Ghosh, "Introducing diversity and uncertainty to create moving attack surfaces for web services," in *Moving target defense*. Springer, 2011, pp. 131–151.

[17] M. Taguinod, A. Doupé, Z. Zhao, and G.-J. Ahn, "Toward a moving target defense for web applications," in *2015 IEEE International Conference on Information Reuse and Integration*, 2015, pp. 510–517.

[18] Y. Li, R. Dai, and J. Zhang, "Morphing communications of cyber-physical systems towards moving-target defense," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 592–598.

[19] A. Kanellopoulos and K. G. Vamvoudakis, "A moving target defense control framework for cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1029–1043, 2020.

[20] H. Alavizadeh, J. B. Hong, J. Jang-Jaccard, and D. S. Kim, "Comprehensive security assessment of combined mtd techniques for the cloud," in *Proceedings of the 5th ACM Workshop on Moving Target Defense*, ser. MTD '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 11–20. [Online]. Available: https://doi.org/10.1145/3268966.3268967

[21] H. Alavizadeh, J. Jang-Jaccard, and D. S. Kim, "Evaluation for combination of shuffle and diversity on moving target defense strategy for cloud computing," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 573–578.

[22] J. Ulrich, J. Drahos, and M. Govindarasu, "A symmetric address translation approach for a network layer moving target defense to secure power grid networks," in *2017 Resilience Week (RWS)*, 2017, pp. 163–169.

[23] A. C. Pappa, A. Ashok, and M. Govindarasu, "Moving target defense for securing smart grid communications: Architecture, implementation amp; evaluation," in *2017 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2017, pp. 1–5.

[24] E. Germano da Silva, L. A. Dias Knob, J. A. Wickboldt, L. P. Gaspary, L. Z. Granville, and A. Schaeffer-Filho, "Capitalizing on sdn-based scada systems: An anti-eavesdropping case-study," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 165–173.

[25] G. K. Ndonda and R. Sadre, "A low-delay sdn-based countermeasure to eavesdropping attacks in industrial control systems," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2017, pp. 1–7.

[26] O. N. Foundation. (2015) Openflow switch specification, version 1.3.5. [Online]. Available: https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.3.5.pdf

[27] Ryu sdn framework. [Online]. Available: https://ryu-sdn.org/

[28] Nmap: the network mapper - free security scanner. [Online]. Available: https://nmap.org/

[29] R. Ross, V. Pillitteri, R. Graubart, D. Bodeau, and R. McQuaid, "Developing cyber resilient systems: a systems security engineering approach," National Institute of Standards and Technology, Tech. Rep., 2019.

An ML and Behavior Fingerprinting-based Framework for Cyberattack Detection in IoT Crowdsensing Platforms

Pedro Miguel Sánchez[1,*], Alberto Huertas[2], Gérôme Bovet[3],
Gregorio Martínez Pérez[1], Burkhard Stiller[2]
[1]*Department of Information and Communications Engineering, University of Murcia*,
30100–Murcia, Spain. [pedromiguel.sanchez@um.es]
[2] *Communication Systems Group (CSG), Department of Informatics (IfI), University of Zurich UZH*,
CH–8050 Zürich, Switzerland
[3] *Cyber-Defence Campus, armasuisse Science & Technology*, CH–3602 Thun, Switzerland.

***Abstract*—Crowdsensing platforms enable novel applications based on data sharing. After aggregating and analyzing the exchanged data, advanced services and information can be offered to achieve common objectives. However, these platforms also are vulnerable to cyberattacks that must be prevented to guarantee data and services security. Nowadays, device behavior fingerprinting is a promising approach to detect and mitigate possible cyberattacks affecting resource-constrained devices. However, its application in novel crowdsensing platforms is unexplored. This work presents a novel framework combining behavior fingerprinting and Machine Learning to identify devices and detect anomalies produced by cyberattacks affecting crowdsensing sensors. The framework performance is evaluated in two research projects that deal with real spectrum sensors belonging to a crowdsensing platform affected by device spoofing and malware. The results showed that it is possible the identification of 25 identical Raspberry Pi devices and the detection of malware and data manipulation attacks.**

***Index Terms*—Crowdsensing, Device Behavior, Cybersecurity, Identification, Attack Detection**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

The emergence of new communications technologies, such as 5G, is bringing novel platforms that integrate IoT (Internet-of-Things) sensors and actuators into heterogeneous environments. In such scenario, data sharing and joint processing gain huge value as new insights can be extracted from the available data. One of the most promising types of collaborative platforms is based on crowdsensing. Crowdsensing platforms allow users to share data collected in local environments and analyze them, from a global perspective, to draw common conclusions. For examples, there exist successful platforms, such as Flightradar24 and OpenSky24, in the field of air traffic or ElectroSense [1] focused on analyzing the electromagnetic spectrum usage.

Crowdsensing-based applications present a lot of advantages over traditional ones based on isolated users or sensors. Some examples are flexibility, low deployment and maintenance costs, or wider data variety. However, they open the door to cybersecurity concerns that must be solved to provide secure and reliable platforms, services, and data. In this regard, one of the most critical cybersecurity concerns consists of exploiting well-known vulnerabilities of resource-constrained sensors and actuators used in crowdsensing platforms. Besides, spoofing and/or modification of devices with malicious functionality is a threat also present. Then, there is a requirement for new solutions capable of detecting and mitigating cyberattacks before a crowdsensing platform is compromised.

Recent related work propose the generation of fingerprints that model the internal behavior of IoT devices to detect possible cyberattacks. Several sources, such as hardware events, system logs, or clock skew, can be leveraged in order to model the "normal" behavior of devices and detect variations produced by cyberattacks. Two principal cybersecurity scenarios appear for behavior fingerprinting [2]. The first one identifies devices with different granularity levels, such as type, model, or individual device. Depending on the identification level, the resources that should be monitored differ. Network, resource consumption, or performance are related to the type and model identification, while low-level sources, such as clock skew and oscillator variations, are used to distinguish identical devices. The second scenario seeks to notice misbehavior caused by cyberattacks or device malfunctioning. Here, resources, such as network communications, resource usage, system calls, and logs, are monitored to deploy a Host-based Intrusion Detection System (HIDS).

Despite the advances made by related work, device identification and detection of cyberattacks are still open challenges in crowdsensing platforms. Among the main issues, the next challenges arise: (i) there is no solution measuring the performance and suitability of behavioral fingerprinting in resource-constrained spectrum sensors [3]; (ii) data sources and events precisely modeling normal and under attack behaviors of spectrum sensors have not been studied [4]; (iii) existing individual identification solutions are designed for traditional computers [5], being not suitable for IoT environments with software and hardware restrictions.

With the goal of improving these challenges, this work proposes an artificial intelligence-based security framework that uses device behavior fingerprinting to detect security threats present in sensors belonging to a crowdsensing platform focused on radio frequency monitoring. Concretely, the present work is divided into two research lines, RESERVE which focuses on the identification of identical devices of the

platform to detect spoofing cyberattacks, and CyberTracer which focuses on detecting behavior anomalies caused by cyberattacks affecting the sensors. The main contributions of this paper include:

- The design of a multipurpose security framework for IoT devices that combines device behavioral fingerprinting and ML techniques.
- The implementation, deployment and validation of the framework for identical device identification based on hardware manufacturing variations, describing the objectives of the *RESERVE* project.
- The implementation, deployment and validation of the framework for cyberattack detection in a spectrum crowdsensing platform, describing the objectives of the *CyberTracer* project.

The remainder of this paper is structured as follows. Section II sketches the design of the cyberattack detection framework. Sections III and IV describe the current results of the two research projects. Finally, Section V provides insights gained so far and outlines future steps.

II. Behavior-based Cyberattack detection Framework

This section introduces the framework combining behavior fingerprinting and ML to detect cyberattack affecting IoT spectrum sensors. The main objectives of the framework are: *(a)* identify identical devices belonging to crowdsensing platforms, solving threats such as device spoofing, and *(b)* detect heterogeneous cyberattacks affecting those devices. The framework can be deployed in a hybrid way, where the sensor hosts the behavior monitoring and fingerprinting functionalities, or the processing and ML/DL applications can be executed on sensors or on servers.

As Fig. 1 depicts, two main modules have been designed to identify identical devices and detect cyberattacks affecting devices of crowdsensing platforms. In detail, the *Behavior Fingerprinting module* is in charge of monitoring the behavior of the sensor. It periodically acquires internal metrics of each device, taking into account three resources: hardware events (such as CPU, GPU, and Hardware Performance Counters), systems calls, and resource usage (Memory, CPU, tasks, or network). Besides, it sends behavioral data to a processing server, if needed. After that, the *Cyberattacks Detection module* trains and evaluates ML-based models identifying devices and detecting cyberattacks. It is important to note that this architecture is generic to be deployed in other IoT scenarios.

III. RESERVE. Device Identification

The main goal of the RESERVE research project is to identify identical sensors in crowdsensing platforms to avoid security threats based on sensor impersonation or malicious sensor deployment. For that, the previous ML-based framework is implemented with the purpose of individual device identification based on hardware variations during manufacturing.

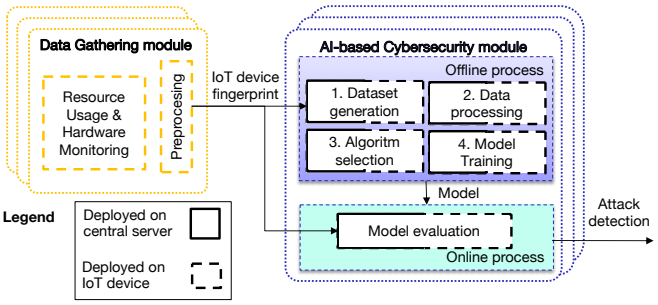


Fig. 1: Device Behavior Fingerprinting Architecture

A. Scenario

The proposed solution is validated in an ElectroSense deployment composed of identical Raspberry Pi devices. Concretely, 10 identical Raspberry Pi 3 Model B+ (RPi3) and 15 identical Raspberry Pi 4 Model B (RPi4) are available.

B. Framework Implementation

In this scenario, the framework should generate ML/DL models to identify the sensors as well as possible malicious elements affecting the identification process robustness. The implementation follows a hardware-based methodology where the device components to be fingerprinted are analyzed, trying to find the hardware elements that may reflect manufacturing imperfections [6]. These chip imperfections are then leveraged in order to generate an unique fingerprint for each device.

After hardware analysis, it is decided to use the skew between the CPU and the GPU cycle counters as source for the fingerprint generation. In this sense, a precise sleep function is executed during 120 seconds on the CPU while the GPU cycle counter is monitored. Then, the 400 values of the previous metric are used as raw device fingerprint. As preprocessing, an sliding window is applied in the 400 value list, generating the next features: average, standard deviation, minimum, maximum, median and mode. This process is repeated 10 times, generating a dataset per device containing 4000 samples. Besides, between each data collection, the device is rebooted to avoid noise introduced by other processes in the device in the cycle measurements.

For device identification, ML/DL classification algorithms are employed. The exact algorithms tested are: k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), XGBoost (XGB), Decision Tree (DT), Random Forest (RF) and Multi-Layer Perceptron (MLP).

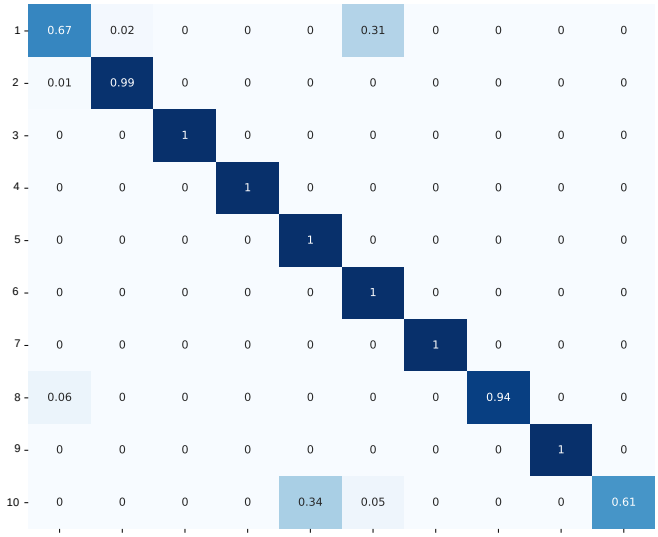
C. Results

TABLE I shows the average True Positive Rate (TPR) of each technique. It can be appreciated that RF and XGB are the best performing models, with +91% TPR. Concretely, XGB is the best one with 91.92% TPR.

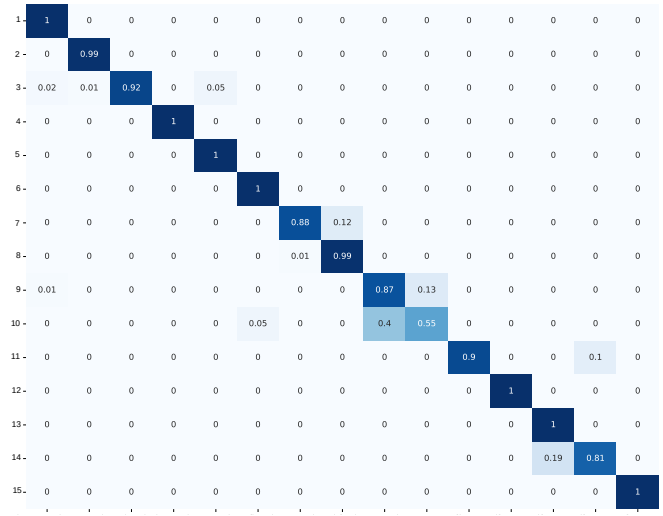
TABLE I: TPR per Classification Algorithm.

Algorithm	k-NN	SVM	XGB	DT	RF	MLP
Avg. % TPR	71.40	89.65	91.92	86.47	91.64	85.32

Fig. 2 shows the average confusion matrix for the fingerprints used for testing, using XGB as classifier. Using a 50% TPR threshold, all the devices can be correctly identified without any device erroneously identified as another one.



(a) 10 RPi3 classification confusion matrix.



(b) 15 RPi4 classification confusion matrix.

Fig. 2: Test confusion matrix for device identification using XGBoost.

IV. CyberTracer. Cyberattack Detection

The main goal of the CyberTracer research project is to detect anomalies produced by cyberattacks affecting resource-constrained devices of crowdsensing platforms [7]. For that, it implements, evaluates and validates the proposed framework in Raspberries Pi, used by the crowdsensing ElectroSense platform. In particular, the aim is to detect two different types of cyberattacks: (i) well-known malware, and (ii) Spectrum Sensing Data Falsification (SSDF) attacks affecting the integrity of spectrum data.

A. Scenario

Three RPi4 acting as ElectroSense sensors are deployed to detect anomalies produced by diverse cyberattacks. Dealing with malware, the following samples are considered: *Botnets* (Mirai and Bashlite); *Rootkits* (Beurk, Bdvl and Diamorphine); *Backdoors* (TheTick, HttpBackdoor, and a simple Python-based backdoor). Regarding SSDF attacks, seven different attacks are proposed based on the modifications made to the spectrum data: Noise, Spoof, Repeat, Confusion, Mimic, Freeze and Delay (more details in [7]).

B. Framework Implementation

As a starting point, a systematic literature analysis has been tackled to study the internal behavior dimensions and events available in Raspberry Pis. As a consequence, a *Behavior Fingerprinting* module has been designed and implemented to periodically acquire around 80 events belonging to the usage of resources, hardware, and software events produced in the RPi. Fig. 3 lists the collected events. These events are collected using *perf* Linux command in loops of 50 seconds, the time required for the ElectroSense processes to scan radio frequency spectrum (from 20 MHz to 1.6 GHz).

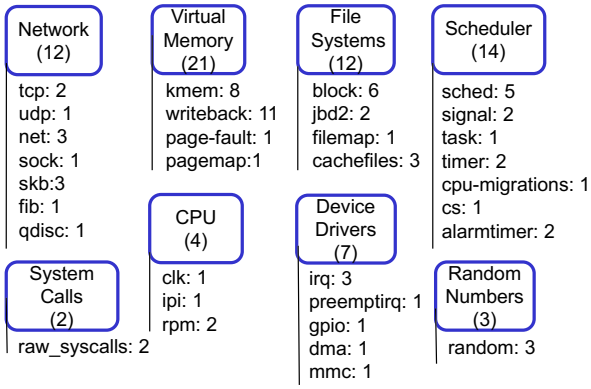


Fig. 3: Collected features for behavior monitoring.

After that, seven datasets with "normal" behavior of the ElectroSense sensors have been collected. These seven datasets represent the seven running configurations that are available in the sensor, modeling normal behaviors of a sensor. Each dataset contains 6 hours of sensor functioning, around 390 vectors. Once the normal datasets are collected, the different attacks to be detected are deployed in the sensors. Later, all the malware samples are monitored while executed in a passive way (without harmful actions being made, just running in foreground) and some of them performing command execution or data leakage (underscored names in Fig. 4). To detect attacks, ML/DL anomaly detection algorithms are employed. Specifically, Autoencoder, Isolation Forest (IF), Copula-Based Outlier Detection (COPOD), Local Outlier Factor (LOF), and One-Class Support Vector Machine (OC-SVM) are used.

C. Results

Fig. 4 shows the confusion matrix for the best performing algorithm (OC-SVM). More than 95% of samples belonging

to the different normal behaviors are correctly detected as "normal." Looking at the rootkits, the passive and innocuous behavior of Diamorphine is not detected, but when it establishes an SSH connection every five seconds, it is identified as malicious. In the case of passive behavior of Bdvl, it is detected only half of the time. In terms of Backdoors, 34% of the samples belonging to Data Leak behavior executed by TheTick are not detected correctly. Furthermore, the DNS behavior only is detected 27% of the time. At this point, it is important to mention that those two behaviors have a minimum impact on the sensor integrity and data confidentiality, as the leaked data is only a few Kb of sensitive data. Finally, it is important to highlight that the rest of the malicious behaviors are detected in an almost perfect fashion.

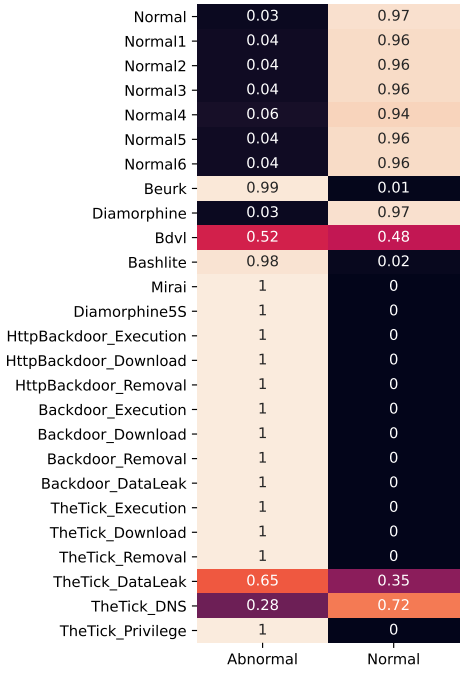


Fig. 4: OC-SVM Confusion Matrix for Anomaly Detection in FTTH Experiment

Fig. 5 shows the True Negative Rate (TNR) for normal behavior and the TPR evolution in the different SSDF attacks depending on the amount of spectrum data modified by the attack (X axis). OC-SVM models have the best performance (100% TPR) for all configurations of noise, spoof, confusion, mimic and delay. Repeat and Freeze attacks deserve special consideration since they are not adequately detected until the affected bandwidth reaches 80 and 160 MHz (only by Autoencoder).

V. Summary and Next Steps

This paper described relevant problems concerning device behavior fingerprinting to identify identical devices and detect anomalies produced by cyberattacks. The goals and current status of two ongoing research projects, RESERVE and CyberTracer, aligning with these research lines, were introduced.

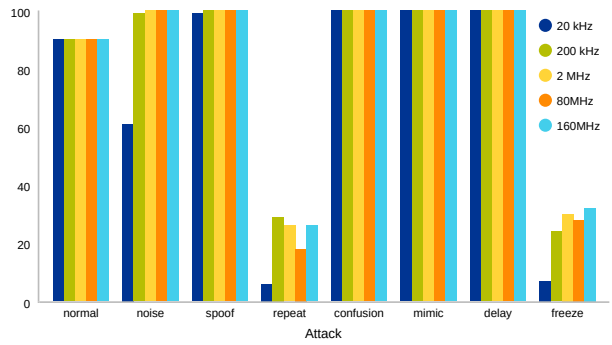


Fig. 5: Average TNR and TPR Performance of OC-SVM When Detecting SSDF Attacks.

In preliminary conclusions, both projects show promising results. RESERVE is able to perform individual identification of 10 RPi3 and 15 RPi4 with identical hardware and software configurations. Besides, CyberTracer has shown excellent results detecting different cyberattacks executed by SSDF and malware.

As future work, the validation of current results with more complex ElectroSense scenarios is key as well as improving the performance of the anomaly detection mechanism with new ML/DL algorithms. Besides, since the proposed platform is independent of the IoT scenario, it is planned to deploy it in other use cases, too. Additionally, further objectives and research questions related to the privacy management, seeking to apply Federated Learning for distributed model generation without data sharing between sensors.

Acknowledgments

This work has been supported by *(a)* the Swiss Federal Office for Defense Procurement (armasuisse) with the RESERVE and CyberTracer (CYD-C-2020003) projects and *(b)* the University of Zürich UZH.

References

[1] S. Rajendran, R. Calvo-Palomino, M. Fuchs, B. V. den Bergh, H. Cordobés, D. Giustiniano, S. Pollin, and V. Lenders, "Electrosense: Open and Big Spectrum Data," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 210–217, January 2018.

[2] P. M. S. Sánchez, J. M. J. Valero, A. H. Celdrán, G. Bovet, , M. G. Pérez, and G. M. Pérez, "A Survey on Device Behavior Fingerprinting: Data Sources, Techniques, Application Scenarios, and Datasets," *IEEE Communications Surveys & Tutorials*, In press.

[3] P. S. Chatterjee, "Systematic survey on ssdf attack and detection mechanism in cognitive wireless sensor network," in *2021 International Conference on Intelligent Technologies (CONIT)*, 2021, pp. 1–5.

[4] K. Yadav, S. D. Roy, and S. Kundu, "Defense against spectrum sensing data falsification attacker in cognitive radio networks," *Wireless Personal Communications*, pp. 1–14, 2020.

[5] I. Sanchez-Rola, I. Santos, and D. Balzarotti, "Clock around the clock: Time-based device fingerprinting," in *2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, p. 1502–1514.

[6] P. M. S. Sánchez, J. M. J. Valero, A. H. Celdrán, G. Bovet, M. G. Pérez, and G. M. Pérez, "Can evil iot twins be identified? now yes, a hardware behavioral fingerprinting methodology," *arXiv:2106.08209*, 2021.

[7] A. Huertas Celdrán, P. M. Sánchez Sánchez, G. Bovet, G. Martínez Pérez, and B. Stiller, "Cyberspec: Intelligent behavioral fingerprinting to detect attacks on crowdsensing spectrum sensors," *arXiv*, p. 2201, 2022.

Sesión VI: Mecanismos de protección del usuario

Automatizando RGPD 2016/679 mediante Procesos de Negocio: El caso del Artículo 32

Ángel Jesús Varela-Vaca, Rafael M. Gasca, María Teresa Gómez López,
Yolanda Morales Zamora
IDEA Research Group, Universidad de Sevilla
{ajvarela, gasca, maytegomez, ymorales}@us.es

***Abstract*—El correcto cumplimiento del Reglamento General de Protección de Datos (RGPD) es complejo, debido a la subjetividad en la interpretación de la norma dada y su dificultad en la implantación por parte de las organizaciones. El correcto cumplimiento requerirá una adaptación profunda de los procesos y tareas internas de las organizaciones, pero una incorrecta implantación podría derivar en problemas de seguridad y sanciones para estas. Nuestra propuesta, se centra en la utilización de las tecnologías y metodologías relacionadas con los Procesos de Negocio para digitalizar y automatizar los procesos y tareas que puedan dar soporte a los distintos artículos del RGPD, facilitando la implantación y automatización de estos procesos, a la vez que se crean las evidencias necesarias para demostrar el buen cumplimiento de la norma. Dada la amplitud del RGPD, en este artículo entramos a detallar una prueba de concepto relacionada con el proceso derivado del Art. 32 sobre la Seguridad del tratamiento, proponiendo el proceso concreto que le da soporte.**

***Index Terms*—Automatización, Digitalización, RGPD, Protección de datos personales, Proceso de Negocio**

Tipo de contribución: *Investigación original.*

I. Introducción

El Reglamento (UE) 2016/679 (RGPD) del Parlamento Europeo y ddel Consejo de 27 de abril de 2016 [1] ha generado novedades en la regulación de la protección de datos personales. Dicho reglamento es relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos, derogando la Directiva 95/46/CE. Dicho Reglamento es de obligado cumplimiento para todo tipo de entidades del ámbito privado y público que hagan tratamientos de datos personales. Para ello, las entidades deben incorporar o adaptar, dentro de sus procesos y tareas, aspectos que permitan demostrar el cumplimiento del Reglamento. Una correcta adaptación de las tareas y procesos internos de las entidades llevará, por un lado a un correcto cumplimiento de RGPD, pero también reportará muchos otros beneficios, sobre todo a nivel organizativo y técnico. Sin embargo, el incumplimiento parcial o total del Reglamento puede derivar en múltiples y diversos problemas de seguridad para las entidades, por ejemplo, la posible pérdida de los datos de los interesados y por tanto su privacidad, daño reputacional para la entidad por el incorrecto uso/tratamiento de los datos, y en sanciones para dichas entidades. Según [2], la media de las sanciones interpuestas en España por incumplimiento ascendió de media a 31.941€ en el año 2020, aunque países como Reino Unido alcanzó una media de 105.103.400€ de media en dicho año. En su mayoría, las sanciones están relacionadas con el incumplimiento de los artículos 5, 6 y 32, enfocados en: (i) Art. 5: Incumplimiento de los principios generales de tratamiento de datos. (ii) Art. 6: Privacidad base jurídica insuficiente para el tratamiento de datos. (iii) Art. 32: Medidas técnicas y organizativas insuficientes para garantizar la seguridad de la información.

El cumplimiento de RGPD [3] en las organizaciones requiere de una revisión y adaptación profunda de sus procedimientos y procesos internos. Sin embargo, entre los factores que impiden el correcto cumplimiento [4][5][6] se encuentra la complejidad y la extensión de la normativa, que requiere de interpretaciones subjetivas, así como de una alta inversión en recursos. Por lo que cuando una organización se enfrenta a la implantación de los mecanismos necesarios para cumplir la RGPD, debe definir exactamente qué procesos y qué conjunto de tareas mínimas se deben implementar para dar cabida a los aspectos de dicho Reglamento. Por lo tanto, cada organización debe adoptar las mejores estrategias y soluciones organizativas y tecnológicas adaptándolas de la manera más adecuada a su contexto [7].

La adaptación del RGPD se puede afrontar usando diferentes estrategias, por ejemplo, basadas en modelos como el del ciclo de vida del dato, en el ciclo de vida de la rendición de cuentas (accountability), o en estrategias basadas en estándares de seguridad de la información o privacidad.

Los procesos de negocio y la gestión de los mismos (BPM - Business Process Management) [8][9] representan la integración de un conjunto de tecnologías y metodologías que facilitan el modelado y despliegue de modelos de procesos, y permiten coordinar un conjunto de actividades para que las organizaciones alcancen sus objetivos empresariales. Estos procesos de negocio han sido típicamente utilizados para facilitar la descripción de las actividades que se tienen que realizar en las organizaciones, la automatización de los mismos mediante la ejecución del flujo de actividades, o el soporte a la toma de decisiones mediante técnicas de simulación [10]. Por esta razón, su aplicación en el cumplimiento de la RGPD puede tener grandes ventajas, avanzando hacia la automatización de los procesos y a la creación de las evidencias necesarias que vienen determinadas por la norma. La aplicación de un sistema de gestión de procesos de negocio (BPMS) permitirá:

- La digitalización del cumplimiento legal en una organización.
- La sistematización de las actividades de cumplimiento mediante procesos de negocio imperativos frente a textos normativos y jurídicos interpretables.
- El alineamiento de las actividades de cumplimiento con las actividades del negocio y reconocer el impacto de los

mismos en los demás procesos del negocio.
- La documentación sobre las actividades de cumplimiento para facilitar auditorías, revisiones y cambios.
- La optimización de los procesos de cumplimiento legal en la organización.
- Evitar sanciones por las autoridades de control debido al incumplimiento legal.
- Facilitar la certificaciones en otras normas de calidad, seguridad de la información, etc.

Nuestra experiencia de más de 20 años en el cumplimiento de la protección de datos personales en grandes organizaciones con tratamientos de muy diferentes tipos, nos ha permitido comprender que las entidades necesitan acompañamiento y una definición clara de las tareas y procesos que son necesarios implantar para adaptarse lo más rápidamente posible al cumplimiento de dicho Reglamento. Por tanto, en esta aproximación, nosotros pretendemos dotar a las organizaciones de una herramienta que sea práctica, útil, adaptable y automatizable para dar total soporte al cumplimiento del RGPD. Para ello, nuestra aportación pretende:

- Analizar el marco normativo de RGPD completo.
- Definir un catálogo de procesos de negocio que den cobertura al articulado de RGPD.
- Definir una matriz de rastreabilidad proceso-artículo, que de manera sencilla permita saber qué procesos y tareas debemos implementar para dar cobertura a las necesidades de cada organización.
- Realizar una prueba de concepto del Art. 32. sobre Seguridad del tratamiento de datos, donde daremos más detalles sobre el análisis de dicho artículo, sus dependencias y se presentará una propuesta de especificación de proceso de negocio para su posterior automatización mediante el correspondiente despliegue y ejecución.

El resto del artículo está organizado de la siguiente manera: Sección II describiremos el marco normativo asociado a RGPD 2016/679. En la Sección III haremos un análisis de los trabajos relacionados y una discusión acerca de las ventajas e inconvenientes de la propuesta y enfrentaremos nuestra propuesta a otras estrategias. En la Sección IV definiremos el catálogo de procesos y la matriz de rastreabilidad de proceso-artículo. La Sección V describe un caso práctico de proceso definido para el artículo 32. Finalmente en la Sección VI daremos unas conclusiones finales.

II. Marco Normativo: RGPD

El RGPD establecido en el marco de la Unión Europea contiene mejoras con respecto a regulaciones anteriores. En primer lugar, a través de este Reglamento, se consigue un régimen jurídico único en materia de Protección de Datos para todos los integrantes de la UE. Este tipo de regulación es obligatoria y directamente aplicable a todos los sujetos de derecho (Estados, instituciones, organismos, personas físicas y jurídicas) de la Unión Europea (UE), a diferencia de la directiva anterior que no tenía valor directo como tal, y que obligaba a la transposición previa de la norma por parte de cada Estado miembro, lo que desembocó en una regulación con diferentes resultados. Es cierto que el Reglamento habilita a que cada Estado miembro pueda desarrollar determinados puntos de la norma, como es el caso de la regulación de una autoridad de control, pero el régimen jurídico es común y las particularidades no afectan a la intencionalidad de la norma. Existe un régimen jurídico único como consecuencia de su regulación a través del mencionado Reglamento.

Algunos ejemplos de este régimen unficado son la responsabilidad de los intervinientes en la gestión de la protección de datos, sus obligaciones, posibles sanciones, posibles conflictos entre distintas autoridades de control de cada país, sistemas probatorios de cumplimiento y no menos importante, los derechos y obligaciones de los titulares de los datos personales que son protegidos a través de este tipo de normativa.

Además, el Reglamento amplía su ámbito de competencia territorial a los responsables y a los encargados del tratamiento no establecidos en la UE (Art. 27), cuando las actividades de tratamiento están relacionadas con la “oferta de bienes o servicios, o con el control del comportamiento, en la medida en que este tenga lugar en la Unión" (Art. 3 RGPD).

El RGPD incorpora, entre otras medidas a cumplir, la obligación de realizar una evaluación de impacto para las organizaciones que realicen tratamientos de datos, que impliquen un alto riesgo de los derechos y libertades de las personas físicas. En dichas evaluaciones se analiza el origen, la naturaleza, la particularidad y la gravedad de dicho riesgo (Art. 35). Además de la obligación de notificar la existencia de una violación de la seguridad de los datos personales a la autoridad de control competente en el plazo de 72 horas (Art. 33). Incluyendo, también, la obligación del registro de actividades de tratamiento de datos personales en determinados supuestos (Art. 30). En cuanto al consentimiento, éste debe ser libre, informado, específico e inequívoco, por el que el interesado acepta, ya sea mediante una declaración o una clara acción afirmativa, el tratamiento de datos personales que le conciernen, no habiendo cabida para el consentimiento tácito (Art. 4.11 y Art. 7). Igualmente, el Reglamento amplía los derechos de los interesados con el derecho de transparencia de la información (Art. 12), derecho de supresión (“Derecho al olvido") (Art. 17), derecho a la portabilidad de los datos (Art.20) o el derecho a la limitación de los datos (Art. 18).

Otras de las novedades del RGPD es el principio de responsabilidad proactiva (Art. 5.2), al que se le une la protección de datos desde el diseño (Accountability) (Art. 25). Incluso asigna nuevas y cualificadas competencias en materia de coordinación y control del cumplimiento de la normativa de protección de datos a la figura del Delegado de Protección de datos (Data Protection Officer) (Art. 37, Art. 38 y Art. 39). Facilita la adhesión de códigos de conducta o a mecanismos de certificación como mecanismos de verificación del cumplimiento de la norma. En cuanto a la seguridad, el Reglamento no distingue entre ficheros de nivel básico, medio o alto, sino que indica que deberán ser “medidas técnicas y organizativas adecuadas para garantizar un nivel de seguridad adecuado al riesgo" (Art. 32.1).

En este contexto, este régimen jurídico único y común, aporta una mayor seguridad jurídica, definiendo las bases para un sistema sancionador común, aunque pueda ser desarrollado por cada Estado miembro para ser adaptado a sus peculiaridades y por tanto, la automatización de los procesos que pueden devenir de la norma, y atendiendo a la misma aportará también los beneficios que la propia norma genera.

III. Trabajos relacionados & Discusión sobre estrategias de cumplimiento

Cuando una organización o empresa decide tomar en serio el cumplimiento del RGPD, resulta claro que necesita un buen sistema de gobierno y gestión de dicho cumplimiento [11], además de asegurar que todas las partes interesadas le confíen sus datos personales. Para llevar a cabo con cierta garantía de éxito este cumplimiento, tanto en los entornos empresariales como gubernamentales, existen una gran diversidad de estrategias de buen gobierno y gestión. Dichas estrategias las hemos organizado para la discusión en los siguientes tres bloques:

- Estrategias basadas en estándares de seguridad de la información.
- Estrategias basadas en estándares de privacidad y privacidad/seguridad de la información.
- Estrategias basadas en modelos.

Estrategias de cumplimiento basadas en estándares de seguridad de la información

Estas estrategias de buen gobierno y gestión se basan en un Marco de controles (Common Controls Framework CCF) que consiste en un conjunto integral de requisitos de control. Agregados, correlacionados y racionalizados obtenidos a partir un vasto conocimiento de la industria de la seguridad de la información. La utilización de este CCF permitirá a las organizaciones responder a los requisitos de seguridad que el cumplimiento RGPD exige y tener "controlado" los ciberriesgos relacionados con ello. Entre estas estrategias podemos citar el cumplimiento RGPD basado en los estándares ISO/IEC 27014:2015 e ISO/IEC 27001/27002:2013 sobre buen gobierno y gestión de la seguridad de la información. Y el basado en el Real Decreto 3/2010 del Gobierno de España, por el que se regula el Esquema Nacional de Seguridad (ENS) en el ámbito de la Administración Electrónica. Este último es la estrategia de cumplimiento que se adopta por ejemplo en la LOPDGDD del Gobierno de España.

Estas estrategias se centran principalmente en aspectos de seguridad de la información y escasamente en aspectos de interés para la protección de datos y la privacidad, por tanto, no son específicos para el cumplimiento exigido y buscan proporcionar una aproximación a la privacidad como un componente más de la seguridad de la información. Y lo que consideramos que es más importante que estas estrategias no establecen pautas o directrices claras para allanar el camino hacia la automatización/digitalización del cumplimiento RGPD. Además, muchas veces tardan en actualizarse a nuevas versiones, por ejemplo el estándar ISO 27002:2013 ha tardado 8 años en publicarse su nueva edición (ISO 27001/27002:2021) y la ISO 27014:2015 ha tardado 5 años en publicarse la siguiente (ISO/IEC 27014:2020), donde tanto en una como en otra ya se tiene en cuenta la privacidad de forma mucho más detallada. Respecto al ENS, después de 11 años se ha publicado recientemente el RD 311/2022 donde se ha aprobado su modificación. Por lo tanto, consideramos no hay una agilidad adecuada en las directrices estratégicas que proponen de acuerdo a los cambios tecnológicos y la evolución de las amenazas que se producen en nuestra sociedad.

Estrategias de cumplimiento basadas en estándares de privacidad o privacidad/seguridad

En este bloque de estrategias hemos incluido primeramente los estándares que incluyen la privacidad de forma única, tales como ISO/IEC 27701:2019, y en segundo lugar las estrategias que incluyen mejor balanceado de la seguridad y privacidad de forma conjunta, que como ya hemos indicado lo hacen ya la ISO/IEC 27014:2020, ISO 27001/27002:2021 y el NIST 800:53 (2020).

En el caso concreto del estándar ISO 27701:2019, proporciona un primer marco internacional para la protección de la privacidad mediante el establecimiento de los requisitos para implementar un Sistema de Gestión para la privacidad (PIMS) sobre el tratamiento de los datos personales, que son denominados como Personally Identifiable Information (PII). Este estándar establece un detallado conjunto de checklists operacionales que pueden ser adaptados a una gran variedad de regulaciones sobre la privacidad y que por supuesto pueden incluir el RGPD. Por tanto, este estándar ya facilita a las organizaciones y empresas las directrices mucho más específicas para la protección de datos personales y que están relacionadas con las actividades de documentación de políticas, procedimientos, protocolos y operaciones. Aunque mucho más específicas, todas estas estrategias allanan mucho el camino hacia la posible implementación operacional de las medidas técnicas y organizativas que requiere el RGPD, creemos que todavía se quedan en su especificación muy lejos de lo que sería una automatización o digitalización del cumplimiento del RGPD.

Estrategias de cumplimiento basadas en modelos

Dentro de este bloque de estrategias tenemos en cuenta todas aquellas que están basadas de alguna forma en modelos de procesos para la protección de datos personales.

Entre estos, se pueden incluir todas las estrategias de cumplimiento relacionadas con lo que son los ciclos de vida, entre los que destacan los ciclos de vida de los datos personales y los ciclos de vida relacionados con la rendición de cuentas (accountability principle).

En el modelo de ciclo de vida de los datos personales, se incluyen cinco tareas de forma cíclica, que son: Asegurar la privacidad por el diseño, Capturar (Obtener y registrar los datos), Almacenar (salvaguardarlo en formato electrónico o papel), Usar/Procesar los datos y Destruir de forma segura los datos. Según nuestro criterio, representa un modelo de cumplimiento muy sesgado hacia aspectos tecnológicos y que aborda escasamente los aspectos organizativos, además de no facilitar detalles sobre la automatización de dichas tareas.

En el modelo de ciclo de vida relacionado con la rendición de cuentas se incluye también un proceso de tres tareas que son preparar (incluye las actividades relacionadas con la preparación de la organización para el cumplimiento, entre las que se incluye la formación del personal), operar (incluye las actividades de definir y desplegar procedimientos que permitan al personal el uso/procesamiento de los datos personales de forma eficiente y conforme al RGPD y la gestión de incidentes de seguridad) y mantener (que incluye evaluar todas la implementación de las anteriores actividades y la toma de decisiones para mejorar dichas actividades en el

caso que sea necesario). También en este caso, consideramos que las especificaciones de este ciclo de vida se quedan a muy alto nivel y no facilitan la automatización de las actividades que proponen.

Finalmente, en este bloque de estrategias podemos considerar aquellas estrategias basadas en modelos que proceden de la academia tales como la aproximación "GuideMe" que fue propuesta en [12], y que representa una aproximación sistemática para ir desde las disposiciones legales bastante abstractas del RGPD a la descripción de requisitos funcionales y no funcionales que puedan implementarse en sistemas software. Es de destacar que para evitar la interpretación subjetiva de los principios de protección de datos personales (DPR) del RGPD, se expresan estos como plantillas de especificación de requisitos software. En esta misma línea de trabajo se encuentra otra propuesta [13], que usa los diagramas UML y restricciones OCL para modelar el articulado de la RGPD, facilitando la comprobación del cumplimiento de dicho Reglamento. Consideramos que ambos trabajos son un avance importante hacia la automatización del cumplimiento RGPD, pero resultan todavía demasiado genéricos, ya que hay un camino demasiado largo por recorrer desde lo que proponen hasta el total despliegue de herramientas de cumplimiento.

Otras aproximaciones se basan en modelos de razonamiento como el de las ontologías [14][13], se basan en generan un modelo de ontología o semántico para RPGD o similar y con dicho modelo dotar de capacidades de razonamiento que permitan la verificación y validación posteriores, como por ejemplo de una política de seguridad concreta. Otras aproximaciones usan modelos de razonamientos como el presentado en [15], donde se usa un modelo basado en objetivos (STS) para la definición de aspectos de privacidad con respecto al cumplimiento del RGPD.

IV. Catálogo de Procesos para cumplimiento de RGPD

Derivado de la complejidad que implica para las organizaciones dar soporte a los artículos mencionados, en esta sección pretendemos hacer una definición de un catálogo de procesos que den cobertura a toda la sección de artículos del RGPD para el cumplimiento de la protección de datos personales por parte de cada entidad. Para ello, proponemos definir una matriz de rastreabilidad donde, como filas aparecerán todos los artículos que dan cumplimiento a la protección de datos según el RGPD, y como columnas los procesos que vamos a proponer para su automatización. Dicha matriz se presenta en la Tabla I. El RGPD se compone de 99 artículos, donde un subconjunto de ellos son los que deben ser cumplidos por parte de cualquier entidad que realice el tratamiento de datos personales, los 26 incluidos en la Tabla I. En nuestra propuesta, estos 26 artículos pueden ser cubiertos con la implementación de 10 procesos de negocio.

El conjunto de procesos que se propone en el catálogo son los siguientes:

- Proceso de Licitud del Tratamiento (**LT**), abarca los artículos 5 y 6 del RGPD. Implica los Principios del tratamiento y la comprobación de todas y cada una de las condiciones de licitud.
- Proceso de Registro de Tratamiento (**RAT**), comprende el artículo 30, donde se determina en qué supuestos se llevará un registro de actividades de tratamiento bajo la responsabilidad de la figura del Responsable.
- Proceso de Transparencia y Recogida del consentimiento para el interesado (**TRC**), contiene los artículos 6, 7, 8 y 12. Se describe cómo cumplir con el Principio de Transparencia en cuanto a la información a facilitar al interesado y los requisitos para el consentimiento, teniendo en cuenta la minoría de edad del interesado.
- Proceso de Evaluación de Impacto Relativa a la Protección de Datos (**EIPA**), engloba el artículo 35. En este proceso se refleja cuándo aplica la Evaluación de Impacto (Privacy Impact Assessment) y cómo implementar dicha evaluación.
- Proceso de Consulta Previa a la Autoridad de Control (**CAC**), incluye el artículo 36. Este proceso se inicia si la Evaluación de Impacto Relativa a la Protección de Datos pudiese entrañar un alto riesgo si el responsable no toma medidas para mitigarlo.
- Proceso de Gestión de Encargado de Tratamiento (**GET**), contiene el artículo 28. Este proceso refleja los requisitos para asignar un encargado externo, las gestiones que le compete y en el caso de recurrir a otro encargado cómo proceder.
- Proceso de gestión de violaciones de la seguridad de los datos personales (**GVS**), comprende los artículos 33 y 34. Refleja la obligación de notificar a la autoridad de control en caso de violencia y comunicar la misma al interesado.
- Proceso de ejercicios de los derechos de los interesados (**EDI**), en el proceso se engloban los artículos 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 y 23. El proceso refleja cómo proceder en caso de que el interesado ejercite alguno de los derechos definidos en los artículos antes mencionados.
- Proceso de nombramiento de Delegado de Protección de Datos (**DPD**), relacionado con los artículos 37, 38 y 39. El proceso define en qué casos es necesaria la designación del Delegado de Protección de Datos y las correspondientes funciones a desempeñar por su parte, además de cómo debe desempeñar dichas funciones.
- Proceso de medidas de seguridad del tratamiento de datos

TABLE I
Matriz de rastreabilidad articulo-proceso.

Art./Proceso	LT	RAT	TRC	EIPA	CAC	GET	GVS	EDI	DPD	ST
Art. 5	×									
Art. 6	×		×							
Art. 7			×							
Art. 8			×							
Art. 12			×					×		
Art. 13								×		
Art. 14								×		
Art. 15								×		
Art. 16								×		
Art. 17								×		
Art. 18								×		
Art. 19								×		
Art. 20								×		
Art. 21								×		
Art. 22								×		
Art. 23								×		
Art. 28						×				
Art. 30		×								
Art. 32										×
Art. 33							×			
Art. 34							×			
Art. 35				×						
Art. 36					×					
Art. 37									×	
Art. 38									×	
Art. 39									×	

(**ST**), comprende el artículo 32. Este proceso desarrolla cómo debe estructurarse a nivel organizativo y técnico para dar cumplimiento y preservar las seguridad del tratamiento de datos

V. Prueba de concepto: Proceso del artículo 32

Como muestra la matriz de trazabilidad de la Tabla I, varios son los procesos necesarios para dar cumplimiento al tratamiento de datos personales por parte de las empresas. En este artículo mostramos en detalle solamente uno de ellos, el proceso ST, el cuál da soporte al Art. 32 y que consideramos podría ser de mayor interés.

A. Análisis de dependencias del articulado con el artículo 32

Antes de comenzar, vamos a realizar un análisis de dicho artículo, ayudándonos de la Fig. 1, la cual describe un grafo de relaciones del Art. 32 (nodo central) y otros derivados. La idea de usar este análisis para poder identificar mejor qué aspectos y tareas deberemos tener en cuenta para el proceso ST correspondiente a este artículo y que queremos especificar para su automatización.

En el grafo podemos observar arcos punteados, y de línea continua que relacionan el Art. 32 con otros artículos del RGPD. Los arcos de línea continua, por ejemplo, entre Art. 32 y Art. 4, intentan establecer una relación directa de aplicación, uso, o fundamento. Por otro lado, los arcos punteados nos ayudan a identificar relaciones indirectas o interrelaciones entre artículos, que deben ser tenidas en cuenta como una colaboración entre el proceso del Art. 32 y los de otros procesos. Entre las relaciones hemos identificado las siguientes:

- Artículo 4, este último establece las definiciones de la normativa, y entre ellas está la "seudonimización" a la que se hace referencia en el artículo central (como una de las medidas técnicas) y las definiciones de "Responsable del tratamiento" y "Encargado del tratamiento".
- Artículo 5, donde se establecen los principios relativos al tratamiento, entre los cuales está el principio de minimización, integridad, confidencialidad y responsabilidad proactiva. El primer principio indicado que se debe tener en cuenta en el proceso de negocio del Art. 32, en el momento de determinar los fines que deben ser realizados conforme a este principio ya que los datos personales a recoger deben ser adecuados, pertinentes y limitados a lo necesario en relación a los fines, en cuanto a la integridad y confidencialidad. El propio Art. 32 dispone que debe garantizarse estos dos conceptos a la hora de aplicar las medidas técnicas y organizativas. Y respecto a la responsabilidad proactiva, es implícita en el Art. 32 en cuanto hace referencia a que en las medidas, hay que implementar un proceso de verificación, evaluación y valoración regulares de la eficacia de las medidas técnicas y organizativas para poder garantizar la seguridad del tratamiento.
- Artículo 6, es el precepto que establece la licitud, y además en su apartado 4, se indica que si el tratamiento es distinto a los fines iniciales por los que se recogieron, el responsable del tratamiento para determinar si el tratamiento con otro fin es compatible con el fin inicial que recoge los datos personales, tendrá en cuenta entre otras situaciones, la existencia de garantías adecuadas, que podrán incluir, por ejemplo, el cifrado o la seudonimización.
- Artículos 12, 13 y 14, estos preceptos (reflejados en los procesos TRC y EDI) establecen la información a facilitar al destinatario, por lo que tiene que ser tenido en cuenta en el proceso del Art. 32 en cuanto a la seguridad del tratamiento de datos, puesto que deben ser respetados en el momento de establecer las medidas técnicas y organizativas.
- Artículos 15, 16, 17, 18, 19, 20, 21, 22 y 23 los derechos de acceso, rectificación, cancelación, oposición, portabilidad y limitaciones, deben ser tenidos en cuenta para no ser infringidos cuando se establecen las medidas técnicas y organizativas del Art. 32.
- Artículo 24, hace referencia al Responsable del tratamiento de datos, este artículo debe ser contemplado, para cumplir con la confidencialidad y responsabilidad del mismo, ya que es uno de los participantes del proceso de negocio del Art. 32.
- Artículo 25, Protección de datos desde el diseño y por defecto, una de las novedades de la normativa que obliga en todo momento a que al establecer los fines y las medidas, ya se prevea cómo se va a gestionar la seguridad del tratamiento de datos para su protección.
- Artículo 28, Encargado del tratamiento, debemos observar las funciones y la confidencialidad de este participante en el proceso de negocio del Art. 32.
- Artículo 29, cuando se gestionen las medidas técnicas y organizativas por cualquiera de los participantes en el proceso deben ser tenidas en cuenta las instrucciones del Responsable y del Encargado del tratamiento de datos.
- Artículo 30, el Responsable del tratamiento está obligado a establecer un registro de actividad de tratamiento de datos personales, el cual podríamos incluir dentro de las medidas organizativas del proceso en cuestión.
- Artículo 33 y 34, la notificación de la violación de la seguridad y la comunicación de la violación al interesado, deben de quedar implícitas en las tareas de detectar y responder que comprende el proceso de negocio del Art. 32, aunque ambos artículos tienen su proceso propio (GVS).

La importancia de este grafo en nuestra propuesta reside en que nos puede servir de información muy valiosa para entender todas y cada una de las tareas que deberemos incorporar al proceso de negocio **ST** del catálogo, y a comprender las posibles colaboraciones de dicho proceso con el resto de procesos de negocio del catálogo.

B. Especificación del proceso de negocio para la automatización

Para el desarrollo de la especificación del proceso y teniendo en cuenta lo analizado en el apartado anterior, tendremos que describir de la forma más precisa posible todos los detalles del mismo, con la idea de facilitar la posterior automatización del proceso de negocio a través de un BPMS. Para ellos vamos a usar una aproximación basada en plantillas [16], con la que daremos una documentación bastante detallada de todos los aspectos de dicho proceso. La plantilla del proceso viene definida de la siguiente manera:

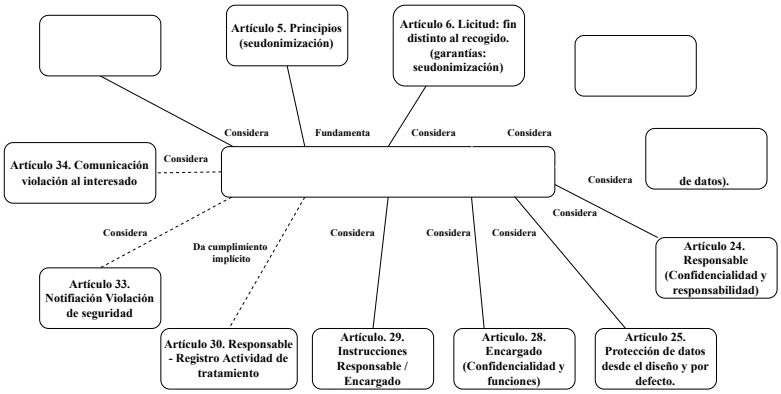


Fig. 1. Análisis de las relaciones del Art. 32 y artículos derivados del RGPD.

- **Nombre del Proceso:** ST (Art. 32.1 y 32.2 RGPD).
- **Propietario del Proceso:** Comisión de Protección de Datos Personales de la Organización.
- **Creador:** Responsable designado por la Comisión.
- **Fecha de creación y última modificación:** DD/MM/AAAA - DD/MM/AAAA.
- **Objetivos del Proceso de negocio** (Incluye por qué y cómo el proceso beneficiará a la organización): Dar cumplimiento al Art.32.1 y 32.2 del RGPD. Facilitar la automatización del cumplimiento.
- **Ámbito del Proceso de Negocio** (Descripción de lo que es incluido en el proceso y de lo que queda fuera del ámbito del proceso): Respecto al ámbito del proceso atiende exclusivamente a las medidas de carácter técnico y organizativo establecidas por el Art. 32.1 y punto 2 del RGPD para su correcto cumplimiento. Este proceso comprende la situación básica de una entidad para establecer la Seguridad del Tratamiento de Protección de Datos, quedando excluidas las situaciones de procesos que comprenden datos de categorías especiales o cesión de datos a terceras entidades. También quedan excluidos los tratamientos que requieran la figura de un Delegado/a de Protección de Datos
- **Roles/Participantes del Proceso:** 1. Responsable del tratamiento de Protección de Datos, persona física sobre la que recae la responsabilidad en atención a la normativa; 2. Responsable Delegado, persona física que realiza gestiones por delegación del Responsable, antes indicado; 3. Responsable Tecnológico, figura que realiza todas las tareas técnicas y/o operaciones necesarias en el proceso; 4. Auditoría interna, personal establecido por la entidad o empresa en cuestión, para comprobar todas las medidas implantadas, además de generar un informe como resultado de la auditoría.
- **Entradas del Proceso:** Partimos de la necesidad de la entidad o empresa que tiene por gestionar un conjunto determinado de datos personales dentro del ámbito especificado anteriormente y que se encuentran bajo su custodia. Los contratos de encargados de tratamiento. También como entrada estarán los nombres, capacidades y competencias de las personas físicas que podrán ser nombradas responsables para los diferentes *lanes* o *carriles* del proceso de negocio y las aplicaciones e infraestructuras tecnológica para el tratamiento de los datos personales.
- **Salida del proceso** Las Políticas de Seguridad internas y las evaluaciones positivas de los contratos de encargado de tratamiento.
- **Límites del Proceso** (Son las actividades o procesos inmediatamente anteriores a la entrada al proceso e inmediatamente posterior a la salida): Los procesos anteriores a la ejecución de este proceso de negocio son los que aparecen en el catálogo como **LT, RAT, TRC y EIPA** y en ciertas condiciones el proceso **CAC** y como procesos posteriores a este proceso de negocio podemos citar a los procesos del catálogo **GVS** y **EDI**
- **Diagrama del Proceso de Negocio:** Ver la Fig. 2 pudiéndose descargar libremente desde https://www.idea.us.es/automatizando-rgpd-bp/. Este modelo está definido usando la notación estándar BPMN 2.0 (Business Process Model Notation) [17] propuesta por la OMG para la descripción de los flujos de actividades. BPMN 2.0 es un lenguaje gráfico que define el flujo de actividades que se pueden ejecutar incluyendo puertas lógicas (AND, OR y XOR). Su instanciación comienza con la ejecución de un evento de inicio, y no termina hasta que la instancia no alcanza el evento final. El proceso propuesto está estructurado a través de cuatro carriles, que describen qué responsable gestiona (ver apartado Roles) cada una de las tareas que se incluyen en el carril.
- **Descripción del flujo del Proceso de Negocio:**
  1) El proceso se inicia en el carril asignado a la figura del Responsable, el flujo de secuencia se dirige hacia la acción *Designar los participantes* (o roles), que van a intervenir en el proceso y serán determinados por el Responsable, los cuales son: Responsable Delegado, Responsable Tecnológico y Auditores internos.
  2) Una vez realizada la asignación, el Responsable tiene dos acciones que llevar a cabo, la de *Publicitar políticas de seguridad internas*, y *Formar y concienciar al personal*.
  3) A continuación, el flujo de secuencia se paraleliza en dos ramas, la primera, que se encuentra en el carril del Responsable Delegado, donde éste tiene que *Analizar el ciberriesgo inicial*, y una vez finalizada esta tarea el flujo de secuencia entra en el carril del Responsable, quien va a desempeñar la acción de *Evaluar ciberriesgo inicial*. La segunda rama es ejecutada por

el Responsable Tecnológico, que tendrá que realizar toda la tarea que conlleva *Gestionar técnicamente el tratamiento de datos* (ejemplos: establecer usuarios, permisos o accesos, entre otros), terminadas las tareas de ambos caminos, estos confluyen en el carril del Responsable Delegado.

4) El flujo de secuencia se sitúa ante tres opciones. La primera de ellas, en el carril del Responsable con la tarea *Determinar los fines de los datos personales* (la finalidad para la cual los datos personales van a ser objeto de su tratamiento). La segunda opción dentro del carril Responsable Delegado, *Determinar medidas organizativas* (medidas que afectan a la estructura y a la toma de decisiones, adecuada gestión de los recursos humanos, formación y concienciación continua a cualquier integrante de la entidad, colaboradores, proveedores y clientes, entre otros tipos de medidas). Y en tercer lugar, en el carril del Responsable Tecnológico, se encuentra la acción *Determinar medidas técnicas* (como pueden ser la seudonimización de la información, cifrado, mecanismos de control y copias de seguridad, entre otros). Finalizadas las tres tareas en sus correspondientes carriles, las tres ramas del flujo se unen en el carril del Responsable Delegado. Debemos aclarar que el establecimiento de estas medidas organizativas, técnicas y sus garantías, desde el inicio del proceso hasta su finalización, tienen que ser establecidas atendiendo al estado de la técnica, el coste de la aplicación y la naturaleza, ámbito, contexto y fines del tratamiento, además de los riesgos de diversa probabilidad y gravedad que genera el tratamiento para los derechos y libertades de las personas físicas.
5) A continuación, el Responsable Delegado, decidirá si designa o no encargado. Si decide designar encargado externo, el flujo de secuencia alcanza la tarea Solicitar autorización previa, el Responsable autorizará o no al Responsable Delegado en la acción *Evaluar contrato de encargo de tratamiento*. Si la respuesta es negativa a la Evaluación del contrato de encargo de tratamiento por parte del Responsable, el flujo de secuencia se dirige al carril del Responsable Delegado para que pueda *Redefinir la propuesta de encargo de tratamiento*, volviendo a *Solicitar autorización previa*. Si la respuesta es afirmativa con respecto a la Evaluación del contrato de encargo de tratamiento por parte del Responsable, el tratamiento lo realizará el encargado de tratamiento externo y este se realizará a través del subproceso de *Encargo tratamiento externo* (Proceso externo, GET). Si no se designa encargado externo, el Responsable Tecnológico será la figura encargada de implementar las medidas, dicha implementación comprenderá cinco tareas de forma iterativa y que se corresponden con las actividades especificadas en el Framework de Ciberseguridad del NIST. Finalizado uno u otro camino del flujo de secuencia, se haya designado o no encargado, el proceso continua.
6) Nos situamos en la tarea *Recibir la información de gestión de las medidas tecnológicas*, dentro del carril correspondiente al Responsable Delegado, el cual va a recibir dicha información para la toma de decisiones que van a surgir durante el proceso. A continuación, éste último decide si habrá auditoría interna.
7) Si no existe auditoría, en la que el camino del flujo de secuencia dependerá de si hay que cambiar las medidas o no, si no hay que cambiar medidas (carril del Responsable Delegado), continua el flujo hacia la acción *Evaluar medidas técnicas y organizativas* cuyo competente es el Responsable y dará fin al proceso. Si las medidas tienen que ser cambiadas, el flujo de secuencia nos comunica con las tareas *Redefinir medidas organizativas* (dentro del carril competencia del Responsable Delegado) y *Redefinir medidas técnicas* (dentro del carril competencia del Responsable Tecnológico), tras la gestión de las dos tareas inclusivas, se unen y finalmente llegan al carril competencia del Responsable donde éste realiza la tarea de *Evaluar medidas técnicas y organizativas*, que da fin al proceso.
8) Si existe auditoría, el flujo de secuencia llega al carril correspondiente a la Auditoría interna (competencia del personal que conforma dicha Auditoría interna) nos conecta con las acciones secuenciales de *Verificar tratamiento de medidas*, *Tomar evidencias y valorar sobre medidas*, y *Realizar informe de auditoría*, terminadas las acciones, el flujo de secuencia alcanza al carril del Responsable, donde éste último tiene que *Evaluar el informe de la auditoría*, y en función de su resultado el flujo tiene dos opciones.
9) Si la respuesta de la evaluación del informe de auditoría no es conforme, el flujo de secuencia conecta con las dos tareas antes mencionadas. Si no se optaba por la auditoría, recordamos que estas medidas son *Redefinir medidas organizativas* (Responsable Delegado) y *Redefinir medidas técnicas* (Responsable Tecnológico). Tras la gestión de las dos tareas inclusivas, el flujo de secuencia alcanza el carril competencia del Responsable donde éste realiza la tarea de *Evaluar medidas técnicas y organizativas*, que da fin al proceso.
10) Si la respuesta de la evaluación es conforme, el flujo alcanza el carril del Responsable para la tarea de *Evaluar medidas técnicas y organizativas* dando fin al proceso.

VI. Conclusiones finales

Tras el profundo análisis de un conjunto de artículos del RGPD, se propone la aplicación de las metodologías y tecnologías relacionadas con los Procesos de Negocio para dar soporte a dichos artículos. Frente a todas estas propuestas consideramos que nuestra principal aportación, respecto a la automatización del cumplimiento del RGPD, es que avanza de forma significativa en ello ya que los modelos de BPMN que se proponen para los requisitos especificados son primeramente fruto de una extensa experiencia de decenas de años en el buen gobierno y gestión de la protección de datos personales en organizaciones. En segundo lugar, son fácilmente desplegables y ejecutables en un motor de ejecución de procesos de negocio (BPMS) y por tanto facilitan la optimización, simulación y mejora en general del cumplimiento

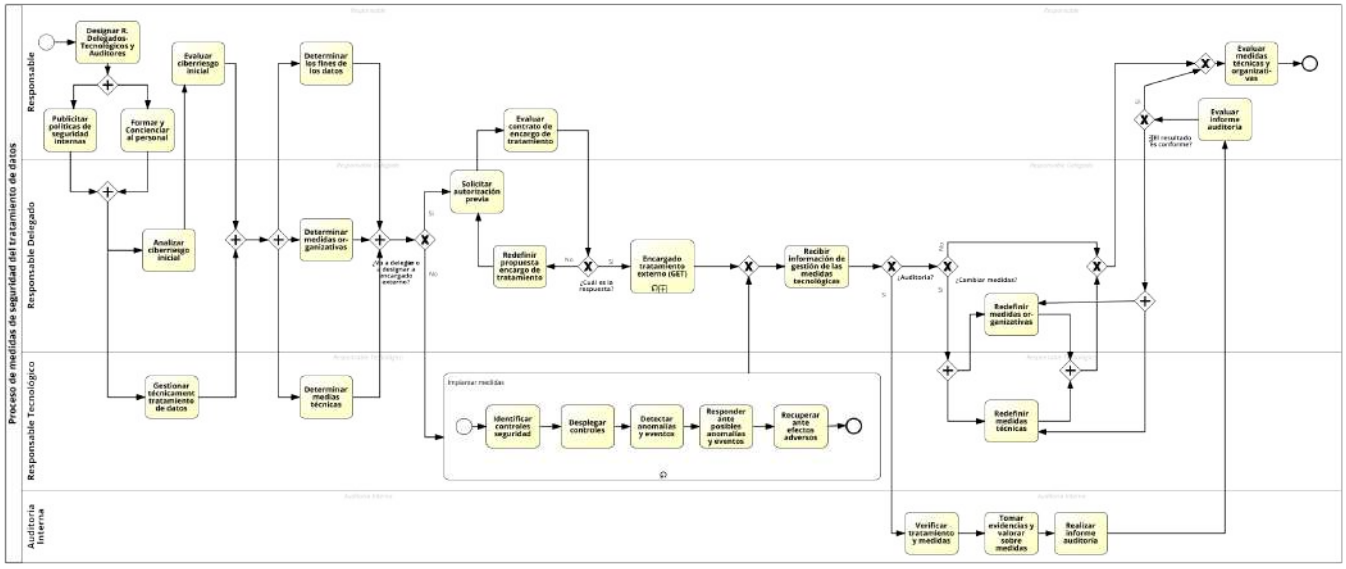


Fig. 2. Modelo de proceso de negocio (BPMN) del Art. 32 del RGPD.

del RGPD en organizaciones y empresas. Además, hemos hecho una aportación de manera más concreta mediante el análisis profundo del Art. 32 sobre Seguridad del tratamiento, del que hemos definido y formalizado un proceso disponible en https://www.idea.us.es/automatizando-rgpd-bp/.

Como actividades futuras, pretendemos continuar con la publicación del resto de procesos necesarios y descritos en este documento. Con dichos procesos, realizaremos un análisis de actuación en base a simulaciones de procesos, lo que ayudará en la toma de decisiones a las organizaciones sobre los recursos que aplicar en las distintas tareas y roles involucrados en los modelos. Esto hará que las empresas puedan predecir su comportamiento ante situaciones de tratamientos de la información.

AGRADECIMIENTOS

Este trabajo ha sido financiado por los proyectos AETHER-US (PID2020-112540RB-C44/AEI/10.13039/501100011033), COPERNICA (P20_01224) y METAMORFOSIS (US-1381375).

REFERENCES

[1] THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION. (2016) Regulation (EU) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/oj

[2] J. Wolff and N. Atallah, "Early GDPR penalties: Analysis of implementation and fines through may 2020," *Journal of Information Policy*, vol. 11, no. 1, pp. 63–103, Jan. 2021. [Online]. Available: https://doi.org/10.5325/jinfopoli.11.2021.0063

[3] M. Brodin, "A framework for GDPR compliance for small- and medium-sized enterprises," *European Journal for Security Research*, vol. 4, no. 2, pp. 243–264, Jun. 2019. [Online]. Available: https://doi.org/10.1007/s41125-019-00042-z

[4] G. A. Teixeira, M. M. da Silva, and R. Pereira, "The critical success factors of GDPR implementation: a systematic literature review," *Digital Policy, Regulation and Governance*, vol. 21, no. 4, pp. 402–418, Jun. 2019. [Online]. Available: https://doi.org/10.1108/dprg-01-2019-0007

[5] M. da Conceição Freitas and M. M. da Silva, "GDPR compliance in SMEs: There is much to be done," *Journal of Information Systems Engineering & Management*, vol. 3, no. 4, Nov. 2018. [Online]. Available: https://doi.org/10.20897/jisem/3941

[6] J. Garber, "Gdpr – compliance nightmare or business opportunity?" *Computer Fraud & Security*, vol. 2018, no. 6, pp. 14–15, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361372318300551

[7] C. Tikkinen-Piri, A. Rohunen, and J. Markkula, "EU general data protection regulation: Changes and implications for personal data collecting companies," *Computer Law & Security Review*, vol. 34, no. 1, pp. 134–153, Feb. 2018. [Online]. Available: https://doi.org/10.1016/j.clsr.2017.05.015

[8] M. Weske, *Business Process Management - Concepts, Languages, Architectures, 2nd Edition.* Springer, 2012.

[9] M. Dumas, M. L. Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*. Springer Publishing Company, Incorporated, 2013.

[10] J. Brocke, J. Mendling, and M. Rosemann, *Business Process Management Cases Vol. 2: Digital Transformation - Strategy, Processes and Execution*. Springer Berlin Heidelberg, 2021. [Online]. Available: https://books.google.es/books?id=zs47EAAAQBAJ

[11] D. Mikkelsen, H. Soller, M. Strandell-Jansson, and M. Wahlers, "Gdpr compliance since may 2018: a continuing challenge," *McKinsey & Company*, vol. 22, 2019.

[12] V. Ayala-Rivera and L. Pasquale, "The grace period has ended: An approach to operationalize gdpr requirements," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, 2018, pp. 136–146.

[13] D. Torre, G. Soltana, M. Sabetzadeh, L. C. Briand, Y. Auffinger, and P. Goes, "Using models to enable compliance checking against the gdpr: An experience report," in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*, 2019, pp. 1–11.

[14] L. Elluri, A. Nagar, and K. P. Joshi, "An integrated knowledge graph to automate gdpr and pci dss compliance," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 1266–1271.

[15] M. Robol, M. Salnitri, and P. Giorgini, "Toward gdpr-compliant socio-technical systems: Modeling language and reasoning framework," in *The Practice of Enterprise Modeling - 10th IFIP WG 8.1. Working Conference, PoEM 2017, Leuven, Belgium, November 22-24, 2017, Proceedings*, ser. Lecture Notes in Business Information Processing, G. Poels, F. Gailly, E. S. Asensio, and M. Snoeck, Eds., vol. 305. Springer, 2017, pp. 236–250. [Online]. Available: https://doi.org/10.1007/978-3-319-70241-4_16

[16] A. del Río-Ortega, M. Resinas Arias de Reyna, A. Durán Toro, and A. Ruiz-Cortés, "Defining process performance indicators by using templates and patterns," in *International Conference on Business Process Management*. Springer, 2012, pp. 223–228.

[17] OMG. (2017) Business process model and notation. [Online]. Available: http://www.omg.org/spec/BPMN/2.0

SignAir - In-air signature biometric system making use of machine learning techniques

Íñigo Turrientes
Vicomtech,Basque Research and Tech. Alliance (BRTA) & Uni. of Burgos student
Burgos, Spain
inigoturrientes@gmail.com

Chiara Lunerti
Vicomtech, Basque Research and Tech. Alliance (BRTA)
Donostia/San Sebastian, Spain
lunerti.chiara91@gmail.com

Daniel Urda
Gr. de Intel. Comp. Apl. (GICAP)
Dpto. de Ingeniería Informática, Esc. Politécnica Superior,
Universidad de Burgos (Spain).
durda@ubu.es

Raúl Orduna
Vicomtech, Basque Research and Tech. Alliance (BRTA)
Donostia/San Sebastian, Spain
rorduna@vicomtech.org

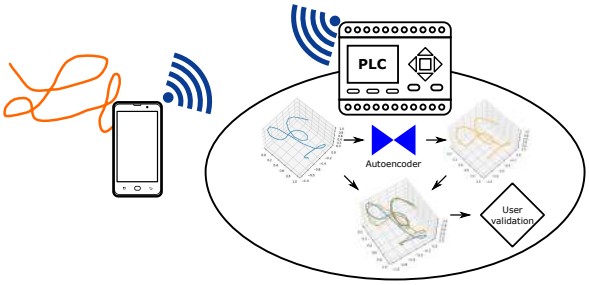


Figure 1. Graphical abstract.

***Abstract*—Signature is an effective method to identify or verify individuals. With the latest technologies, it has arised the possibility of using an "in-air signature", which is harder to forge than the classical handwritten signature thanks to the added dimension of depth. This project proposes a dynamic "in-air signature" biometric system that authenticates a user using a three dimensional signature captured by an Android device (making use of accelerometer and gyroscope). The in-air signature is then sent to the system in which the user will be authenticated, where it will be compared with its reconstruction, obtained by passing it through the user's corresponding autoencoder. The result of the comparison will authenticate or reject the user. The outcome of the project is a complete biometrical system, from the capture of the signature to the authentication or identification of the user. This system achieves an equal error rate of 0.075%.**

***Index Terms*—Biometry, In-Air Signature, Industry 4.0, Cybersecurity, Android App, Android, Accelerometer, IoT Authentication.**

Type of contribution: *Research in progress.*

I. Introduction

Biometry has been gaining ground in our routines, being present not only in access control systems, such as those present in most workplaces, but also in our pockets thanks to mobile devices.

The signature has been used to identify people since writing began to be mastered. Since the 1980s, with the rapid change in technology, the possibility to digitize it using a secure mechanism to prevent forgery has been studied.

Over time, systems for digitizing the signature have been implemented and improved. These include, tablets that take into account not only the shape, but also dynamic traits of the signature production process, like: time, speed and pressure exerted on each part of the signature, among other variables [1], [2], [3], [4]. Later more advanced and less intrusive systems have been used to capture the signature, such as cameras that follow the finger while the signature is made in the air or products that detect the position and movements made by a user with the hand [5], [6]. Initially, these in-air signatures were captured in two dimensions and processed as conventional signatures. More recently, the incorporation of the third dimension has provided valuable information: the 3D-treated signature provides additional value by adding reliability to the traditional signature [6], [7], [8].

The scope of application of this project is the industry, focusing its use in environments where other types of identification may be difficult if not impossible. The specific use case is the identification of an individual which has to wear protective gear and another type of identification, such as fingerprint, face or retina, is not possible. For instance, an operator that interacts with a machine that can work with corrosive elements, so the operator has to be protected with gloves and face protection. This paper proposes a three-dimensional in-air biometric signature system. It is a complete system, from the capture to the final decision of whether or not to identify an individual.

With the development carried out, the individual who wishes to be validated, will use an Android mobile device, smart band or watch, equipped with a gyroscope and an accelerometer, to capture the trajectory of the movement made, which can be either a signature or a chosen movement in the air. The phone then sends this captured data to the system in which the individual is going to be validated. In this system, the data is processed and a three-dimensional representation of the trajectory of the gesture or signature is obtained, followed by a filtering and normalization operation. This representation is needed in order to be fed to an autoencoder [8], [9]. This autoencoder will output a reconstruction of the signature which will then be compared to the original signature to decide if it is genuine or not.

An original aspect of this work is the application of intelligent techniques, more specifically, *Artificial Neural Networks (ANN)* [8], [10], [11]. The biometric solution proposed avail of various autoencoders, one per user, that generate a reconstruction of the signature submitted by an individual.

II. State of the Art

Previous works, to the authors' knowledge, are shown in this section to present the state of the art of in-air signature.

The works showed here are focused to in-air signature, we refer the reader to [1], [4], [12], [13], [14] for a detailed

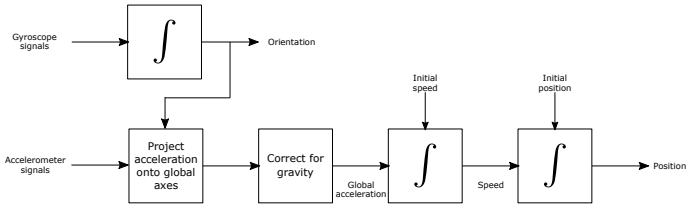


Figure 2. Representation of the capture and integration algorithm [16].

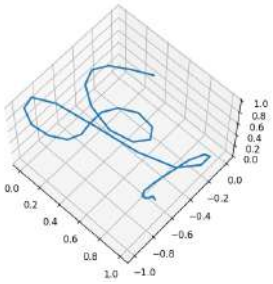


Figure 3. 3D signature representation.

literature review of signature verification.

As for 3D signature verification, The first accelerometer based verification system was proposed by Bailador et al. [7] performed a detailed analysis on the performance of different pattern recognition algorithms (HMM, DTW and Bayes Classifiers). They obtained the best results using DTW. They used an iPhone in order to capture the signature and discussed the importance of the hidden information of the third dimension (depth). Nguyen Ngoc Diep et al. [6], introduced a 3D signature-based authentication method for mobile users with accelerometers, using Support Vector Machines (SVM) in 2015.

Jameek Malik et al. [15] discussed in 2018 for the first time the importance of the depth of a three-dimensional signature in image based acquisition 3D signature authentication systems. As prior to this point, the signatures captured by image based systems were treated as two-dimensional, ignoring the depth. This work demonstrates that taking depth into account adds security. The verification is done with DTW in the three dimensions of the signature. In addition, this work provides a dataset with 600 three-dimensional signatures.

Jameel Malik et al. [8] proposed autoencoders as verification mechanism in 2020 for image based signature capture systems. This work is a continuation of [15] in which, in addition to providing a new and more complete dataset with 1,800 signatures (including genuine samples and forgeries) to train and test the neural networks. With the proposed architecture, an autoencoder is needed for each user. It proposes two different ways of representing the signatures, one in the format of a static image and the second, in the form of a dynamic gathered point cloud.

Elyoenai Guerra-Segura et al. [5] proposed an in-air signature verification system in 2021 using a Leap Motion device to capture the signature, this device is an optical hand-tracking module over which the user draws its signature. The authors then propose Least Squares Support Vector Machines to classify the signature. However most of the previous works need expensive equipment or complex camera jigs in order to capture the trajectory of the signature in the air [5], [8], [15]. We propose a combination between the development in verification mechanisms carried out by image based signature capture in [8] and previous works [15], and a simple and widely extended capture system, as an Android smartphone or wearable equipped with accelerometers and gyroscopes, as proposed in [6], [7].

We adapt an accurate deep learning based matching algorithm, that has not been previously used with accelerometer based signature capture, as an alternative to traditional DTW algorithm. We also add the identification process to the previous work, which only takes verification into account.

III. Experimental Design

In this section, we explain the signature acquisition system.

We have developed an Android app that that captures the trajectory followed by the in-air movement done by the user. The phone acts as an inertial measurement unit (IMU) that digitizes the movement. The movement used to identify the user can be a signature or any kind of repetitive and memorable motion. To digitize it, the app makes use of the embedded accelerometer and gyroscope. Given the ubiquity of this sensors, the capture system could be any device with this pair of sensors [6], [7], [16].

The capture app is able to digitize the signature's path drawn by the user in the air by making use of the accelerometer and gyroscope mounted in the Android device. The capture rate is fixed to $5ms$, this consistency in the capture encapsulates the hidden temporal information, so if 2 points are close together, is due to a slow movement, whereas if 2 points are far apart, it indicates that the movement was faster in that portion of the signature. The rotation suffered by the capture device is also recorded so the acceleration can be projected in the correct axis. Gravity's acceleration is also compensated to avoid signature's drifting in the direction in which gravity "pulls" [16]. At this point, the system has digitized the accelerations made by the user while "drawing" the in-air signature. From now on, the process is independent of the capture mechanism. These data are then filtered and normalized following the method described in [8]. Then, in order to obtain the three dimensional representation of the signature in the form of a 3D point-cloud, the accelerations captured by the phone are integrated 2 times, so the accelerations get converted to 3D coordinates. This whole process is represented in *Fig. 2*. After that, filtering and normalizing processes are applied to the signature as [8] states, in order to be able to use the signature with the autoencoders that we are going to discuss in the next section.

At this point, in a similar way of the dataset provided by [8] the signature is now represented as a 3 dimensional ordered, filtered and normalized point-cloud. To represent it, each point is connected to the next one by a segment as is shown in *Fig. 3*.

This app acts as the biometric capture subsystem, and after being digitized, the signature is ready to be sent to the matching and decision subsystems where it will be evaluated.

IV. Biometric System

In this section we are going to explain the whole biometric processes of this work, including enrolment, verification and

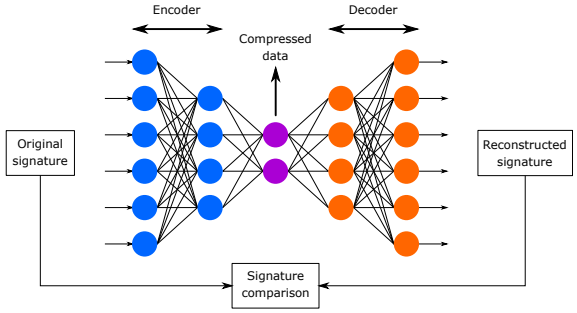


Figure 4. Representation of the signature being fed to the autoencoder and the corresponding reconstruction as output.

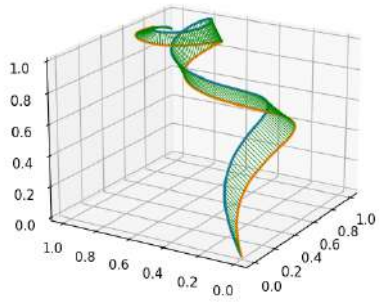


Figure 5. Representation of the euclidean distance. The original signature is represented in blue while the reconstructed signature is represented in orange.

identification methods. To achieve this we have replicated the verification system described in [8] for point-cloud representation and improved it so we are also capable to perform identification.

We use autoencoders in order to compare the point-cloud representation of the signature obtained by the system described in the previous section.

This system uses a personalized autoencoder per user, they are trained with a set of signatures from an user and used as classifiers. Each autoencoder can replicate the signature of the user with which they have been trained, but they are unable to replicate another signature, such as a forgery or a signature from another user. The entire architecture of the autoencoder is used, so the output is a reconstruction of a given signature.

A. Enrollment

The enrollment process needs 10 initial signatures from a new user to enroll it in the system. The autoencoder is then trained with these signatures. The loss function used is the euclidean distance, and it is defined in *Eq. (1)*, the autoencoder tends to minimize the distance between the input signature and the reconstructed signature in training, learning to reconstruct the original signatures, using the distance to evaluate success in training.

$$L_p = \frac{1}{2}\left\|X - \hat{X}\right\|^2 \tag{1}$$

When the autoencoder's training finishes, the trained model is saved and this is the way in which the system stores each user's signature, avoiding the storage of data linked with static information of the user, protecting its privacy.

B. Authentication

To authenticate a user the following process is applied:

As shown in *Fig. 4*, in order to validate a particular signature in a given autoencoder, the signature is fed to the trained autoencoder, which outputs another signature that is a reconstruction of the original one. Then, the reconstructed signature is compared to the input signature using the euclidean distance, defined in *Eq. (1)*. If the value obtained is lower than a threshold the signature is accepted, if not it is rejected. A visual representation of the euclidean distance is shown in *Fig. 5* the green segments between the original and the reconstructed signature represent the distance between each respective pair of points.

Then, depending if the aim of the system is to verify or identify the user different processes are applied:

In the case of **verification**, only the corresponding user's autoencoder is loaded and the signature is provided to it, then the signature is evaluated as previously described. The device identifies the user and it is used to locate the proper model. Finally, depending on the score and the threshold the system decides if the user is verified or not.

In the case of **identification** all of the autoencoders are loaded, the signature is fed to all of them, each time outputting a score, the user represented by the autoencoder that scores the minimum value is the one identified if its score is lower than the threshold. The identification adds a new feature to the work previously done by [8].

After each of these processes, the signature is deleted from the system in order to provide more security by avoiding possible leaks of genuine signatures.

V. Results

The results obtained by this project are discussed in this section. To evaluate the system we have used the dataset provided by [8] and signatures captured by ourselves.

For verification, the EER obtained by the system is 0.075 at a threshold of 3.09, the graphic plotting the false positives and false negatives rates for different thresholds is displayed in *Fig. 6*.

We use the Detection Error Tradeoff (DET) curve to evaluate the system, instead of the ROC curve used in [8], as a state of the art evaluation of biometric systems in *Fig. 7* [17].

In the case of identification, it depends on the threshold used:

Table I
Identification metrics: Identification Rate (IR), False Negative Identification Rate (FNIR) and False Positive Identification Rate (FPIR).

Threshold	IR	FNIR	FPIR
1	0.635	0.365	0.0
2	0.82	0.18	0.046
3	0.9025	0.0975	0.314
4	0.915	0.085	0.723

As expected, a stricter threshold helps to avoid false positive identifications, but it compromises the identification and the false negative identification rates as we can see in Table I.

The choice of a threshold is not trivial and requires a trade-off between security and user experience, so the choice of threshold will depend on the security requirements of the application in which the system will be deployed.

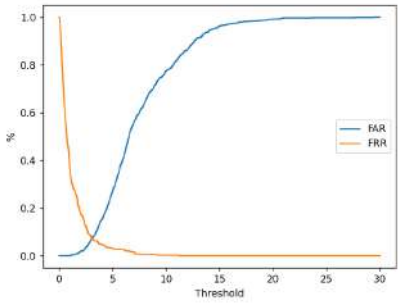


Figure 6. Graphic that displays the false positive and negative rates for different thresholds.

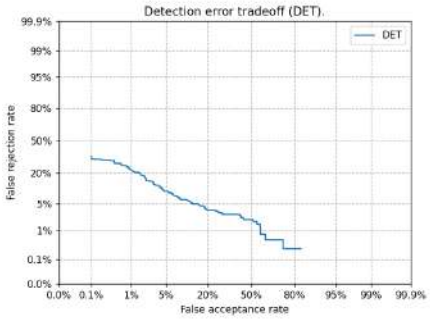


Figure 7. Graphic that displays the Detection Error Tradeoff (DET) curve.

VI. Conclusions and Future Work

A. Conclusions

This system is an alternative to other biometrics in the case that other methods cannot be used. Although right now it is weaker than other biometrics, it can be greatly improved by studying its possible use cases beyond weak systems. It is also capable to perform the identification or verification of a user that has been previously enrolled into the system successfully.

The lack of any physical evidence, as the signature can only be revealed while being drawn, is an improvement over the traditional handwritten signature. Also, the fact that signatures are not stored in the system, adds privacy to it.

This signature method is not being used on a daily basis, but with little effort, every user could develop muscle memory in order to perform the 3D signature or gesture. In addition, the system can be re-trained with already validated signatures, taking into account changes suffered by the signature due to injuries, ageing or other circumstances.

The addition of the identification process is an interesting topic for future research.

The capture system doesn't represent the signatures with great accuracy due to the physical limitations of the device and the size of the signature, which is often as big as a *20cm* sided cube; nevertheless, the signatures are always captured consistently. Accidentally, this adds security to this project, considering that even if an impostor obtains a 3D representation of a signature, the real trajectory will not be revealed.

B. Future work

As future work, a new way of automatically calculating the threshold would improve this project, as now the threshold is chosen manually.

The addition of a second authentication factor, by only allowing connections from previously authorized devices, can improve the security of this system.

Also, the comparison between 2 signatures can be enhanced by making use of intelligent methods, instead of the used equation *Eq. (1)*, for instance deeper autoencoder architectures. Or alternatively, trying systems that include temporal information such as those based on Long Short-Term Memory.

And finally exploring other capture devices, such as commercially available IMU integrated circuits or devices, could greatly improve the security of the system by getting more accurate representations of the signatures.

References

[1] Frank Leclerc and Réjean Plamondon: "AUTOMATIC SIGNATURE VERIFICATION: THE STATE OF THE ART—1989–1993" on *International Journal of Pattern Recognition and Artifitial Intelligence*, vol. 8, N. 3, pp. 643-660, 1994.

[2] Anil K. Jain, Friederike D. Griess and Scott D. Connell: "On-line signature verification" on *Pattern Recognition*, vol. 35, n. 12, pp. 2963-2972, 2002.

[3] McCabe, Alan, Trevathan, Jarrod, and Read, Wayne: "Neural network-based handwritten signature verification.", on *Journal of Computers*, vol. 3 n. 8, pp. 9-22, 2008.

[4] M. Stauffer, P. Maergner, A. Fischer, and K. Riesen: "A survey of state of the art methods employed in the offline signature verification process", on *New Trends in Business Information Systems and Technology*, pp. 17–30, 2021.

[5] Elyoenai Guerra-Segura, Aysse Ortega-Pérez, and Carlos M. Travieso: "In-air signature verification system using leap motion", on *Expert Systems with Applications*, vol. 165, pp. 1-14, 2021.

[6] Nguyen Ngoc Diep, Cuong Pham, and Tu Minh Phuong: "Sigver3d: Accelerometer based verification of 3-d signatures on mobile devices", on *Advances in Intelligent Systems and Computing*, vol. 326, pp. 353–365, 2015.

[7] G. Bailador, C. Sanchez-Avila, J. Guerra-Casanova, and A. de Santos Sierra: "Analysis of pattern recognition techniques for in-air signature biometrics", on *Pattern Recognit.*, vol. 44, n. 10-11, pp. 2468–2478, 2011.

[8] J. Malik, A. Elhayek, S. Guha, S. Ahmed, A. Gillani, D. Stricker: "Deepair-sig: End-to-end deep learning based in-air signature verification", on *IEEE Access* vol. 8, pp. 195832–195843, 2020.

[9] Ahrabian, K. and BabaAli, B: "Usage of autoencoders and Siamese networks for online handwritten signature verification", on *Neural Comput & Applic*, vol. 31, pp. 9321–9334, 2019.

[10] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Hoseini and M. Fathy: "Online signature verification based on feature representation", on *2015 The International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pp. 211-216, 2015.

[11] M. Viswanathan, Ganesh Babu Loganathan, and S. Srinivasan: "IKP based biometric authentication using artificial neural network", on *AIP Conference Proceedings*, vol. 2271, n. 030030, pp. 1-8, 2020.

[12] M. Diaz, M. A. Ferrer, D. Impedovo, M. I. Malik, G. Pirlo, and R. Plamondon: "A perspective analysis of handwritten signature technology", on *ACM Comput. Surv.*, vol. 51, n. 6, pp. 1–39, 2019.

[13] A. Kumar and K. Bhatia: "A survey on offline handwritten signature verification system using writer dependent and independent approaches", on *Proc. 2nd Int. Conf. Adv. Comput., Commun., Autom. (ICACCA) (Fall)*, pp. 1-6, 2016.

[14] S. Dalal and U. Jindal: "Performance of integrated signature verification approach", on *Proc. Int. Conf. Comput. Sustain. Global Develop. (INDIA-Com)*, pp. 3369–3373, 2016.

[15] Jameel Malik, Ahmed Elhayek, Suparna Guha, Sheraz Ahmed, Amna Gillani, and Didier Stricker: "3dairsig: A framework for enabling in-air signatures using a multi-modal depth sensor", on *Sensors (Switzerland)*, vol. 18, pp 1-16, 2018.

[16] Oliver J. Woodman: "An introduction to inertial navigation", on *University of Cambridge, Computer Laboratory, Technical Report*, n. 696 pp. 1-37, 2020.

[17] Shuaishuai, Zhu, Lv, Xiaobo, Feng, Xiaohua, Lin, Jz, Jin, Peng and Gao, Liang: "Plenoptic Face Presentation Attack Detection", on *IEEE Access*, vol. 3, pp. 1-1, 2020.

EEG Data for User Authentication with Multi–Class and One–Class Classifiers

Luis Hernández-Álvarez*,[1,2], Stefano Caputo[3], Lorenzo Mucchi[3], Luis Hernández Encinas[1]

[1]Institute of Physical and Information Technologies (ITEFI)
Spanish National Research Council (CSIC), Madrid, Spain
[2]Computer Security Lab (COSEC), Universidad Carlos III de Madrid, Madrid, Spain
[3]Department of Information Engineering, University of Florence, Florence, Italy
{luis.hdez.alvarez, luis}@iec.csic.es, {stefano.caputo, lorenzo.mucchi}@unifi.it

***Abstract*—Nowadays, the development of user authentication protocols is a hot topic, due to the importance of authentication mechanisms in online services as bank applications, online shopping or personal and professional document requests. Biometric information is commonly combined with Artificial Intelligence (Machine Learning and Deep Learning) methods to develop these systems. Nevertheless, they are usually based on Multi–Class classifiers, which need the impostor's information in order to be trained. The access to the impostor's information is an unrealistic assumption and, therefore, in this ongoing research we propose the construction of more realistic user authentication models using One–Class classifiers, and compare their performance with Multi–Class classifiers. Moreover, we also pretend to evaluate the contribution of different sensor locations and brain waves, and define the best model for a secure and a usable user authentication system.**

***Index Terms*—Artificial Intelligence, Biological Sensors, Electroencephalogram, Machine Learning, Multi–Class Classifiers, One–Class Classifiers.**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

Electroencephalography (EEG) is a method that registers the bioelectrical brain activity of a user and represents one of the body most unique physiological characteristic [1]. As a result, its use is suitable for the development of user biometric authentication systems, as it meets the requirements of these systems (i.e. changeability, shoulder–surfing resistance, theft protection and protection from user non–compliance) [2]. In this sense, it should be highlighted that EEG information enables user authentication, which is different and more convenient than device authentication, as it represents features inherent to the person. EEG is the natural feature for user authentication in noninvasive brain–computer interfaces, a technology that is experiencing a rapid development and becoming more portable, popular and accessible [3]. This technology can extract the intentions or emotions of a user, but it faces some security issues [4]. Thus, it is essential to protect the user information, being EEG the most appropriate feature.

The use of Artificial Intelligence (AI) techniques has gained notable importance, since they allow to identify patterns and structures in this type of data. Specifically, Machine Learning (ML) and Deep Learning (DL) have been shown to be particularly useful in this context, but Multi–Class (MC) classifiers are mainly used to perform user authentication [5]. Despite this type of model achieves excellent results, they need to learn from both, information of the legitimate user (positive data) and from an impostor (negative data). This is not the real case scenario, specially in the use of EEG data, where only information of the user of interest is usually available. For the works that focus on One-Class (OC) classifiers, they limit their study to One-Class Support Vector Machine (OC–SVM), obviating other models such as Isolation Forest (IF) and Local Outlier Factor (LOF). These models have been shown to produce comparable outcomes to those of OC–SVM in different applications and to be suitable for abnormality detection systems [6].

In this ongoing research, we propose to analyze different MC– and OC–ML classifiers to authenticate users based on their EEG signals. As principal novelty, we will study IF and LOF and compare them with OC–SVM which, to our knowledge, have not been explored before for user authentication. Moreover, we will also address how several dimensionality reduction techniques impact the performance of the models in order to improve their computational efficiency, but minimizing their effect in the authentication. In addition, we will distinguish between secure and usable systems, and propose the best solution for each one. In this work, a model is defined as *secure* if the probability of accepting an impostor is low or, in other words, the probability of an authenticated user being the legitimate one is high. On the other hand, a model is going to be *usable* if the number of attempts the legitimate owner must do to be authenticated is low. Therefore, this research will include the following contributions:

- Study of the suitability of OC classifiers for EEG–based user authentication. In particular, we will explore the usefulness of IF and LOF and compare their performance with OC–SVM and other MC–ML classifiers, as SVM and Random Forest (RF).
- Test the effect of different dimensionality reduction techniques in the efficiency of the classifiers, including Principal Component Analysis (PCA) and K–best selection with a χ^2 statistical test. Additionally, we will explore the contribution of each EEG sensor and frequency component to identify the sensors location and brain waves that contribute more in the authentication process of the user.
- Propose the best solution depending on its application in terms of security and usability.

The rest of this contribution is organized as follows: in Section II works related to the use of AI techniques with user authentication and EEG data are included. Then, in Section III and Section IV the process of data acquisition and the proposed models are described. In Section V the expected results and how they are going to be studied is detailed and, finally, in Section VI the conclusions are presented.

II. Related Work

The use of AI techniques for user identification and authentication has been widely investigated in the recent years. In this sense, we have organized the current related literature in the next Subsections.

A. Multi–Class EEG user authentication

A wide range of MC–ML and MC–DL algorithms for EEG–based user authentication have been proposed. SVM, Bayesian Network (BN) and Artificial Neural Network (ANN) models are suggested in [7], achieving an accuracy and a F_1–score of 85.49% and 79% for the SVM, of 85.97% and 85% for the BN, and of 92.89% and 88% for the ANN. Similar approaches and results are reported in [8], [9], and [10]. Other classifier that has been shown to be appropriated for this application is RF [11]. More options include Hidden Markov Model (HMM), K–Nearest Neighbors (KNN), and Gaussian Mixture Models (GMM) [5]. In the recent years, Convolutional Neural Networks (CNN) have been investigated in this application, being produced several articles with good results [12], [13].

B. One–Class EEG user authentication

Some works have considered OC–SVM for EEG user authentication. For example, in [14] the authors use a Support Vector Data Description (SVDD) to authenticate subjects based on their visual evoked potentials. They achieved good results (98.5% correctness) by combining the users samples in groups of three. In [15] the accuracy and False Positive Rate (FPR) of a OC–SVM is studied by increasing the number of blinks while measuring EEG signals. The best values obtained are 80% and 2.2%, respectively. A OC–SVM is also employed in [16] as a first security layer of a intruder detection/user identification system, and showed how its parameters affect the True Acceptance Rate (TAR) and True Rejection Rate (TRR). Similarly, in [17] a biometric authentication system using a OC-SVM and a CNN. The study presented in [18] uses a OC–SVM to extract unsupervised features of EEG signals and explores their robustness against intra–subject variability.

Alternatively to these methods, we propose the use of IF and LOF as new OC classifiers. In this way, we can work only with positive samples under realistic assumptions, and investigate the performance of these models in comparison with OC–SVM, an option proved to be suitable for user authentication. It should be mentioned that IF and LOF have been combined with EEG data for different applications, such as epileptic seizure detection [19] and artificial removal pipeline [20], but no for the design of user authentication systems.

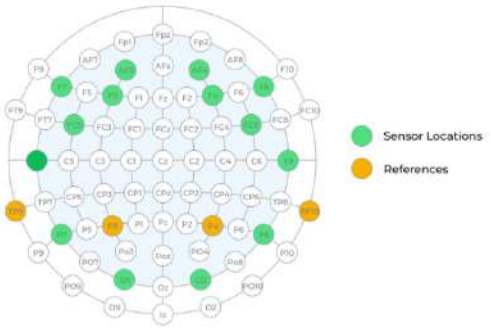


Figure 1. Measure Map of Emotiv Epoc+ Headset[2].

III. Data Acquisition and Processing

The EEG of 40 volunteers was measured with the *Emotiv Epoc+ V1.1* headset at a sampling rate of 256Hz. This instrument includes 14 EEG sensors plus 4 references, which in the International 10–20 system correspond to the following locations: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, and AF4 for sensors, and P3, P4, TP9, and TP10 for references[1]. A map of the location of these sensors is shown in Figure 1. The EEG signal of each subject, continuous signal $S_i(t)$, was recorded for approximately two minutes while presenting 8 different stimuli in the form of images. Nevertheless, due to issues in the data acquisition process, one of the subjects had to be discarded and, thus, the experiments are going to be performed with data of 39 different subjects. It should be clarified that privacy protection is not the focus of this work and, hence, we assume that the connection between the EEG sensors and the authentication model platform is secure.

Regarding the data processing, we defined the following procedure:

1) Divide of the EEG signal in time periods of 240 milliseconds. That is, the signal of user i, $S_i(t)$, is separated in j_i different smaller signals, $s_{i,j_i}(t)$, of 240 milliseconds each. It must be remarked that the number j is not the same for all users (from here the notation j_i), as the EEG signals were not measured for exactly the same time.
2) Compute the wavelet decomposition of each $s_{i,j_i}(t)$ for each $i \in \{1, \ldots, 39\}$ and $j_i \in \{1, \ldots, m_i\}$, where m_i is the latest signal division of user i. Specifically, we used a five–level wavelet decomposition using the order 2 Daubechies wavelet with Matlab, version $R2021b$. From this decomposition, we obtained the wavelet coefficients D1, D2, D3, D4, D5, and A5. In Table I the frequency content and corresponding brain wave of each coefficient are reported.

Table I
Correspondence between Wavelet Coefficients, Brain Waves and Frequency Content.

Wavelet Coeff.	D1	D2	D3	D4	D5	A5
Brain wave	γ	γ	β	α	θ	δ
Freq. (Hz)	0–4	4–8	8–16	16–32	32–64	64–128

[1]For more specifications, see: https://emotiv.gitbook.io/epoc-user-manual/introduction-1/technical_specifications

3) For each coefficient in each $s_{i,j_i}(t)$, the following 8 metrics were calculated: maximum, minimum, mean, standard deviation, variance, skewness, Shannon entropy, and average power.

Since the *Emotiv Epoc+* headset collects data from 14 sensors, 6 coefficients are computed from each sensor, and 8 metrics are calculated for each coefficient, the final number of features of our data is $14 \cdot 6 \cdot 8 = 672$.

We decided to use the wavelet decomposition for several reasons: 1) the implementation of the Fast Wavelet Transform is computationally fast, 2) it offers a simultaneous feature localization in time and frequency domain, 3) it is able to identify details of small parts of the signal, better than its general characteristics, and 4) it has been shown that its application to EEG data is suitable with AI models [21].

IV. Proposed Models

To perform user authentication, we will create a personalized model for each user. In the next subsections, the concrete configurations of the models that will be used in the experiments, as well as how we will construct their train and test sets are described.

A. Multi–Class Classifiers

In the case of Multi–Class classifiers we will perform a binary classification. This means that the models will have to decide if a novel data sample belongs to the legitimate user or not. Therefore, the train set is composed by both, positive (information from the legitimate user) and negative (information from an impostor) data in equal percentage. In our case, the negative data in the training of a model (i.e. user 1) will be formed by randomly selected samples of any of the other users (i.e. users 2–39). The same procedure will be followed to create the test set.

Support Vector Machine: It constructs a hyperplane in a high–dimensional space to separate two set of points. SVMs optimize the functional margin (distance to the nearest element of each class) to obtain the best hyperplane. We will use Radial Basis Function (RBF) SVM, optimizing the parameters *C* (indicates the size of the margin hyperplane) and *gamma* (relevance of each training samples) with a 10–fold cross–validation process [22].

Random Forest: Combination of several decision tree classifiers in a bagging method that outputs the most voted class. The RF will be defined with the "gini" criterion and exploring the number of estimators and the maximum depth of each tree [22].

B. One–Class Classifiers

One–Class classifiers function as abnormality detectors, since they can be trained only with positive data in the train set and detect samples that do not belong to the legitimate user in the test set. Hence, they represent the real case scenario, in which only information from the owner of the device is known.

One–Class Support Vector Machine: Proposed in [23], OC–SVM uses hyperplanes to separate regions withand without data. Its parameters are similar to the ones of SVMs, with the addition of the ν parameter, which represent a lower bound for the number of samples that are support vectors and an upper bound for the number of samples that are on the wrong side of the hyperplane. We will use a RBF–based OC–SVM, fixing the parameter ν.

Isolation Forest: Anomaly detection algorithm that uses binary tree classifiers to individually isolate points [24]. In this case, we will explore different values for the number of estimators and the contamination parameter (analogous to the ν parameter of OC–SVM), so that more secure/usable models can be created.

Local Outlier Factor: Uses the local deviation of each point with respect to its neighbors to detect anomalous samples [24]. Also in this case several contamination values will be tested.

C. Dimensionality Reduction

As commented in section III, our data has 672 features or dimensions. Thus, it would be interesting to reduce this dimensionality, improving the computational efficiency of the models, while avoiding the reduction of their performance.

Principal Component Analysis: We will explore the effect of performing PCA, obtaining new data with a lower number of features, that are linear combinations of the original ones.

χ^2 **Statistical Test**: This statistical test will allow us to define the most significant features, and we will evaluate the performance of the models considering only a number of those.

Important Channels/Waves: Similarly, we will investigate the results obtained with the data of each individual sensor and frequency component. Apart from reducing the dimensionality of the problem, this will enable us to identify if some sensing locations and brain waves have more importance in user authentication and extract its biological meaning.

V. Expected Results

To evaluate the authentication performance of the generated models, several metrics will be analyzed. As explained before, the models will be trained with a train set, while the test set is going to be used to predict a user with previously unseen data. The metrics we are going to use depend on these predictions on the test set, specifically on the number of True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN).

The number of TP, FP, FN and TN will allow us to calculate the precision, recall, F_1–Score, accuracy and False Positive Rate of the developed models. The definition of these metrics are included in Equations (1)–(5). Taking into account these definitions, precision (percentage of times a predicted positive was a true positive) and FPR (percentage of times an impostor was identified as the owner from his total number of attempts) should be interpreted as security measures. This means that, the higher the precision and the lower the FPR are, the more secure our model is. On the other hand, recall represents the usability of the system, as it is defined as the percentage of times the owner is correctly authenticated. Lastly, F_1–Score is a combined measure of precision and recall, while accuracy gives us a general idea of how well our model performs.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

[2]Obtained from: https://emotiv.gitbook.io/epoc-user-manual/introduction-1/technical_specifications

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F}_1\text{–Score} = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{4}$$

$$\text{False Positive Rate (FPR)} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{5}$$

As commonly found in security systems, we expect to find a trade–off between security and usability, meaning that the increase in one implies a decrease in the other. For example, we could design a very strict model that hardly recognizes the owner; this system would be very secure, but the legitimate user may have to repeat the authentication process several times in order to be identified. It is our goal to investigate the parameters of the proposed classifiers to find the best configuration for the three cases: a secure system, a usable system and a balanced system.

VI. Conclusion

The use of the proposed method could improve the security of the user authentication process by, under realistic assumptions (i.e. using only positive data), reduce the dimensionality of the problem with the selection of the proper EEG channels/waves. As a result, the computational effort and consumed time of the authentication procedure can be reduced, enhancing the efficiency while maintaining a reliable model.

Acknowledgements

This work was supported in part by the Spanish State Research Agency (AEI) of the Ministry of Science and Innovation (MCIN), project P2QProMeTe (PID2020-112586RB-I00/AEI/10.13039/501100011033); in part by Comunidad de Madrid (Spain) through Project CYNAMON, grant No. P2018/TCS-4566-CM, both co-funded by the European Regional Development Fund (ESF, FEDER and ERDF, EU); in part by the European Union's Horizon 2020 Research and Innovation Program under Grant 872752 and under Grant 101017141. L.H.A. would like to thank CSIC Project CAS-DiM for its support.

References

[1] D. Smit, D. Posthuma, D. Boomsma, and E. Geus, "Heritability of background EEG across the power spectrum," *Psychophysiology*, vol. 42, no. 6, pp. 691–697, 2005, https://doi.org/10.1111/j.1469-8986.2005.00352.x.

[2] W. Khalifa, A. Salem, M. Roushdy, and K. Revett, "A survey of EEG based user authentication schemes," in *2012 8th International Conference on Informatics and Systems (INFOS)*. IEEE, 2012, pp. 55–60.

[3] B. He, H. Yuan, M. Jianjun, and S. Gao, *Brain Computer Interfaces*, 2020, pp. 131–183, https://doi.org/10.1007/978-3-030-43395-6_4.

[4] Q. Li, D. Ding, and M. Conti, "Brain–computer interface applications: Security and privacy challenges." Proceedings of the 2015 IEEE Conference on Communications and Network Security (CNS), 2015, pp. 663–666, https://doi.org/10.1109/CNS.2015.7346884.

[5] A. Jalaly Bidgoly, H. Jalaly, and Z. Arezoumand, "A survey on methods and challenges in EEG based authentication," *Computers & Security*, vol. 93, pp. 1–16, 2020, https://doi.org/10.1016/j.cose.2020.101788.

[6] S. Jumpathong, K. Kriengket, P. Boonkwan, and T. Supnithi, "Anomaly detection in lexical definitions via one-class classification techniques," in *2021 16th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 2021, pp. 1–6, https://doi.org/10.1109/iSAI-NLP54397.2021.9678166.

[7] J. Hu and Z. Mu, "EEG authentication system based on auto-regression coefficients," in *Proceedings of the 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, pp. 1–5, https://doi.org/10.1109/ISCO.2016.7727122.

[8] Q. Gui, Z. Jin, and W. Xu, "Exploring EEG-based biometrics for user identification and authentication," *2014 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pp. 1–6, 2014, https://doi.org/10.1109/SPMB.2014.7002950.

[9] B. Kaur, D. Singh, and P. Roy, "A novel framework of EEG-based user identification by analyzing music-listening behavior," *Multimedia Tools and Applications*, vol. 76, no. 24, pp. 25 581–25 602, 2017, https://doi.org/10.1007/s11042-016-4232-2.

[10] T. Waili, M. G. Md Johar, K. Sidek, N. Nor, H. Yaacob, and M. Othman, "EEG based biometric identification using correlation and MLPNN models," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 15, no. 10, pp. 77–90, 2019, https://doi.org/10.3991/ijoe.v15i10.10880.

[11] A. Valsaraj, I. Madala, N. Garg, M. Patil, and V. Baths, "Motor imagery based multimodal biometric user authentication system using EEG," in *2020 International Conference on Cyberworlds (CW)*, 2020, pp. 272–279, https://doi.org/10.1109/CW49994.2020.00050.

[12] Q. Wu, Y. Zeng, C. Zhang, l. Tong, and B. Yan, "An EEG-based person authentication system with open-set capability combining eye blinking signals," vol. 18, no. 2, pp. 1–18, 2018.

[13] T. Schons, G. Moreira, P. Silva, V. Coelho, and E. Luz, *Convolutional Network for EEG-Based Biometric*, 2018, pp. 601–608, https://doi.org/10.1007/978-3-319-75193-1_72.

[14] A. Zuquete, B. Quintela, and J. P. Cunha, "Biometric authentication using brain responses to visual stimuli," in *Proceedings of the Third International Conference on Bio-inspired Systems and Signal Processing*. Biosignals, 2010, pp. 103–112.

[15] E. Gupta, M. Agarwal, and R. Sivakumar, "Blink to get in: Biometric authentication for mobile devices using EEG signals," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6, https://doi.org/10.1109/ICC40277.2020.9148741.

[16] L. Moctezuma and M. Molinas, "Multi-objective optimization for EEG channel selection and accurate intruder detection in an EEG-based subject identification system," *Scientific Reports*, vol. 10, no. 5850, pp. 1–12, 2020, https://doi.org/10.1038/s41598-020-62712-6.

[17] J. Haga, "Biometric system using EEG signals from resting–state and one–class classifiers," Ph.D. dissertation, Norwegian University of Science and Technology, 2020.

[18] T. Nishimoto, H. Higashi, H. Morioka, and S. Ishii, "EEG-based personal identification method using unsupervised feature extraction and its robustness against intra-subject variability," *Journal of Neural Engineering*, vol. 17, no. 2, pp. 1–16, 2020, https://doi.org/10.1088/1741-2552/ab6d89.

[19] Y. Guo, X. Jiangb, L. Tao, L. Mengc, C. Dai, X. Long, F. Wan, Y. Zhang, J. Van Dijk, R. M. Aarts, W. Chen, and C. Chen, "Epileptic seizure detection by cascading isolation forest-based anomaly screening and easyensemble," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 8, pp. 1–10, 2022, https://doi.org/10.1109/TNSRE.2022.3163503.

[20] V. Kumaravel, E. Farella, E. Parise, and M. Buiatti, "NEAR: An artifact removal pipeline for human newborn EEG data," *Developmental Cognitive Neuroscience*, vol. 54, pp. 1–14, 2022, https://doi.org/10.1016/j.dcn.2022.101068.

[21] A. K. Kumar Nitendra and A. H. Siddiqi, "Wavelet transform for classification of EEG signal using SVM and ANN," *Biomedical Pharmacology*, vol. 10, no. 4, pp. 1–9, 2017, https://doi.org/10.13005/bpj/1328.

[22] K. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[23] R. Scholkopf, A. Williamson, J. Smola, and J. Platt, "Support vector method for novelty detection," *Neural Information Processing Systems*, pp. 582–588, 2000, https://doi.org/10.5555/3009657.3009740.

[24] F. T. Liu, K. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2009, pp. 413–422, https://doi.org/10.1109/ICDM.2008.17.

Análisis de Problemas de Seguridad y Privacidad en Wearables Usados por Menores

Sonia Solera-Cotanilla[1], Jaime Fúster[2], Jaime Pérez[2], Rafael Palacios[2], Mario Vega-Barbas[1], Manuel Álvarez-Campana[1], Gregorio López[2]
[1]ETSI Telecomunicación, Universidad Politécnica de Madrid
{sonia.solera, mario.vega, manuel.alvarez-campana}@upm.es
[2]Instituto de Investigación Tecnológica, ICAI, Universidad Pontificia Comillas
jaimeff@alu.comillas.edu, {jaime.perez, rafael.palacios}@iit.comillas.edu, gllopez@comillas.edu

***Resumen*—El incremento de uso de los wearables en los últimos años ha fomentado un desarrollo tecnológico en este ámbito que ha permitido la evolución de estos dispositivos a lo que actualmente conocemos. El hecho de que cada vez hayan ido requiriendo mayor cantidad de datos personales del usuario convierte a los wearables en dispositivos atractivos para un atacante. Esta investigación propone una metodología para el análisis de la seguridad y privacidad de los wearables, especialmente los utilizados por menores de edad. La metodología parte de un escenario de pruebas específico, expone las herramientas de apoyo a la investigación, especifica el proceso de pruebas a seguir y se aplica a un conjunto de dispositivos comerciales puestos a prueba. Esta investigación pretende poner a prueba dispositivos comerciales de diferentes gamas y características técnicas para concienciar a fabricantes y usuarios del estado de seguridad y privacidad con el que cuentan sus dispositivos.**

***Index Terms*—Ciberseguridad, Internet de las Cosas, Privacidad, Seguridad, Wearables**

Tipo de contribución: *Contribución científica original*

I. Introducción

En los últimos años, se ha producido un notable aumento del mercado de dispositivos conectados, especialmente en el ámbito de los wearables, que ha multiplicado por 13 sus ventas en todo el mundo desde el 2015 al 2020 [1]. A diferencia de los wearables del pasado, los actuales recogen una amplia gama de datos que a menudo se almacenan en la nube y son gestionados por terceros. Estos datos del usuario suelen incluir información sensible como ubicación, dirección de correo electrónico, información sobre el ritmo cardíaco y otros relacionados con la salud.

A pesar de que el crecimiento acelerado del mercado de los wearables pueda favorecer el progreso tecnológico en el desarrollo de estos dispositivos, también supone el riesgo de que su producción crezca sin el control y la regulación adecuados para garantizar unos niveles mínimos de seguridad y privacidad. Una supervisión insuficiente o ineficaz de la producción de estos dispositivos puede favorecer la salida al mercado de productos inseguros que prioricen la usabilidad sobre la seguridad. Esta amenaza es especialmente preocupante en el caso de los wearables para menores debido al general desconocimiento del uso correcto de estos dispositivos y de la cantidad de datos personales que manejan. El Consejo de Consumidores de Noruega [2] detectó importantes fallos de seguridad en los smartwatches para menores, lo que llevó a algunos organismos, como la Agencia Federal de Telecomunicaciones de Alemania en 2017 a prohibir la venta de smartwatches para niños, calificándolos de herramientas de espionaje y llegando a instar a los padres a destruir los dispositivos de sus hijos [3].

Aunque algunas empresas han centrado recientemente sus estrategias en la venta de productos que garantizan la seguridad y privacidad de sus usuarios, los resultados de una búsqueda en los principales mercados en línea muestran que una parte importante del mercado está formada por productos sin tales garantías. Estos productos suelen ser dispositivos de bajo coste que generan altas cifras de ventas pero que priorizan el precio y la usabilidad frente a la seguridad. Aunque marcas conocidas como Fitbit o Apple han acaparado más del cincuenta por ciento del mercado de los wearables a lo largo de los años, las marcas menos conocidas parecen estar en alza [1]. Ante este panorama, parece oportuno estudiar la situación actual del mercado analizando la seguridad y la privacidad de estos dispositivos.

Este artículo se centra en el desarrollo de una metodología para evaluar los riesgos de seguridad y privacidad en los wearables, así como el estudio del caso de estudio de wearables comercializados para menores de edad. Utilizando esta metodología, se identifica el origen de estas amenazas y sus posibles contramedidas, aprovechando herramientas de fácil y libre acceso que se aplican a escenarios similares de la Internet de las Cosas (IoT).

Otros trabajos en la literatura ya han abordado los problemas de ciberseguridad de los dispositivos IoT y wearables [4], [5], [6], [7]. A diferencia de estos trabajos, esta investigación se centra en el desarrollo de una metodología específica para las pruebas de vulnerabilidad de los wearables que abarque todos los escenarios de comunicación que intervienen en el funcionamiento de dichos dispositivos. Además, como apoyo a la metodología, este documento expone un conjunto de herramientas para el análisis de vulnerabilidades de seguridad y privacidad en el contexto de wearables.

El resto del documento está organizado de la siguiente manera. En la sección II se analiza la literatura relacionada con la seguridad y la privacidad de los wearables y las aplicaciones móviles. Esta sección describe brevemente el protocolo Bluetooth Low Energy (BLE), junto con las vulnerabilidades relevantes encontradas en la literatura. En la sección III, se define y expone la metodología propuesta para las pruebas de vulnerabilidad de los wearables, definiendo el escenario de ataque, las categorías de ataque y los procedimientos de prueba, así como las herramientas de apoyo utilizadas. En

la sección IV se describen los resultados obtenidos de la aplicación de la metodología en un conjunto de wearables comerciales utilizados por menores. En la sección V, se discuten dichos resultados y se analizan las posibles mitigaciones a los riesgos observados y algunos otros hallazgos relevantes sobre la seguridad y privacidad de BLE que pueden dar lugar a futuras investigaciones. Además, se exponen las conclusiones extraídas de esta investigación.

II. Antecedentes

Como base de nuestra investigación se ha realizado un análisis de la literatura existente acerca de las vulnerabilidades de seguridad y privacidad existentes en los wearables y las aplicaciones. También se han estudiado características de seguridad, métodos de emparejamientos y vulnerabilidades más importantes a considerar en BLE.

En este sentido, S. Seneviratne *y al.* [7] ofrecen un estudio exhaustivo y una clasificación de los wearables disponibles en el mercado, las amenazas a la seguridad de las comunicaciones, y algunas soluciones a estos problemas encontradas en la literatura. El estudio analiza las amenazas en términos de confidencialidad, integridad y disponibilidad de la información que manejan los dispositivos. En cuanto a las amenazas a la confidencialidad, debido al uso de BLE como principal medio de comunicación, la mayoría de los wearables son vulnerables a tres tipos de ataques:

- *Espionaje*: Interceptación no autorizada en tiempo real de una comunicación confidencial.
- *Análisis de tráfico*: Monitorización del tráfico intercambiado entre los wearables y su base y/o servidor para hacer inferencias a partir de los patrones de comunicación.
- *Recogida de información* transferida entre el dispositivo y su base (frecuentemente un smartphone).

La mayoría de los ataques de espionaje y análisis de tráfico están relacionados con implementaciones inadecuadas del proceso de publicación de BLE o con el uso de direcciones estáticas de los dispositivos. Por otro lado, los ataques de recopilación de información suelen implicar la ruptura del proceso de intercambio de claves durante el emparejamiento BLE o la recopilación de información sobre otros dispositivos, como los smartphones [7]. Aunque no son tan comunes como las amenazas a la confidencialidad, los principales ataques que amenazan la integridad de estos dispositivos son:

- Ataques que modifican la información transmitida por el dispositivo.
- Ataques de repetición de paquetes para suplantar la identidad del usuario o corromper los datos.
- Ataques de enmascaramiento, en los que el atacante se hace pasar por un dispositivo autenticado para robar datos o inyectar información falsa en el sistema.

Todas las vulnerabilidades encontradas en el contexto de los ataques a la integridad se deben a métodos de autenticación débiles o a la ausencia de cifrado en las comunicaciones entre dispositivos. Por último, los ataques de Denegación de Servicio (DoS) son los más frecuentes contra los wearables, aunque son menos utilizados que otras categorías [7]. Al igual que con las otras amenazas, los ataques contra la disponibilidad son posibles debido a las deficiencias de implementación de los fabricantes.

Desde el punto de vista de la evaluación de la vulnerabilidad, M. Langone *et al.* [8] describen una metodología para realizar una evaluación de vulnerabilidad en wearables, con el fin de analizar e identificar problemas de seguridad en tres wearables diferentes que se comunican vía BLE con un smartphone: Easy Fit de Cellular Line, Fitbit Charge y Fitbit Alta de Fitbit. Los resultados del análisis muestran que el uso de métodos débiles de generación de claves a corto plazo (STK), como los métodos Just Work o Passkey Entry, y la falta de un proceso de emparejamiento y vinculación entre el dispositivo y el smartphone son las principales vulnerabilidades que afectan a estas tecnologías. Tanto Easy Fit como Fitbit Charge presentan problemas relacionados con estas vulnerabilidades, lo que permite a un actor malicioso interceptar información sensible del usuario intercambiada entre el dispositivo y el smartphone.

A lo largo de los años han surgido diversas vulnerabilidades, problemas de seguridad y ataques contra BLE. Los más relevantes son:

- *Ausencia de protección contra ataques Man in the Middle (MitM) y escuchas*. Los atacantes pueden capturar y manipular los datos intercambiados entre dispositivos de confianza [9]. En las conexiones heredadas de BLE, Just Works es vulnerable a las escuchas y al MitM ya que el TK es conocido [10]. Esto ocurre en las versiones 4.0, 4.1 y 4.2 de BLE.
- *Vulnerabilidades en el protocolo de generación de claves*. Passkey es vulnerable a los ataques de fuerza bruta [10] [11]. Existente en todas las versiones de BLE.
- *Vulnerabilidades en el algoritmo de intercambio de claves*. En el pareo de la Passkey de Bluetooth Secure Connections, dado que la contraseña se transmite bit a bit y cada bit es confirmado por el periférico cada vez que se recibe, un atacante podría adivinar fácilmente la contraseña probando cada bit de las reconexiones con el periférico [12] [13]. Existente en todas las versiones de BLE.
- *Ausencia de autenticación de usuario*. La especificación de Bluetooth solo proporciona autenticación de dispositivos [9]. Existente en todas las versiones de BLE.

Aunque algunas de estas vulnerabilidades sólo aparecen en las especificaciones BLE más antiguas y parecen haberse corregido en versiones posteriores, la mayoría de los dispositivos actuales siguen implementando Bluetooth 4.0, 4.1 y 4.2, por lo que las debilidades siguen siendo relevantes.

Con respecto a las aplicaciones subyacentes, en el sistema operativo Android algunas aplicaciones pueden eludir el sistema de permisos utilizando canales encubiertos o canales laterales. J. Reardon *et al.* [14] demuestran que, con suficientes permisos, las aplicaciones de Android podían utilizar la tarjeta SD como canal encubierto para compartir la Identidad Internacional de Equipo Móvil (IMEI) del teléfono, un valor numérico que identifica a los teléfonos móviles de forma exclusiva, con otras apps no autorizadas. Además, algunas aplicaciones utilizaban otros canales para estimar y compartir la ubicación del usuario a través de la dirección MAC del dispositivo, la caché ARP o los metadatos de las imágenes.

En lo que respecta a las aplicaciones de control parental, A. Feal *et al.* [15] realizaron un estudio en profundidad del ecosistema de estas aplicaciones en Android desde el punto de vista de la privacidad y la normativa. En este estudio se distingue entre las aplicaciones de supervisión, que permiten a los padres controlar el comportamiento de los niños, y las aplicaciones de restricción, que permiten a los padres filtrar contenidos y definir reglas de uso para limitar las acciones de los niños. En lo que respecta al uso de permisos, A. Feal *et al.* [15] mostraron que las aplicaciones de control parental solicitan 27 permisos de media, 9 de ellos etiquetados como peligrosos. Estos permisos se utilizaron para filtrar datos a servidores remotos en muchos casos. Algunas aplicaciones analizadas en [15] también utilizan permisos personalizados para obtener funcionalidades expuestas por otros desarrolladores o proveedores de teléfonos, lo que revela asociaciones (comerciales) entre ellos. Sólo la mitad de las aplicaciones analizadas informaban claramente a los usuarios sobre sus prácticas de recogida y tratamiento de datos. Mientras que el 59 % de las aplicaciones admitía el uso de datos sensibles por parte de terceros, sólo el 24 % revelaba la lista completa de terceros integrados en el software.

En lo que respecta al cumplimiento de la normativa, I. Reyes *et al.* [16] presentaron un marco para la evaluación automática de los comportamientos de privacidad de las aplicaciones de Android. Dicho marco analizó el cumplimiento de la COPPA (Children's Online Privacy Protection Act) de multitud de las aplicaciones infantiles gratuitas más populares. Este análisis mostró que la mayoría de las aplicaciones examinadas infringían potencialmente la COPPA, principalmente debido al uso de SDK de terceros. Además, el estudio descubrió que más de la mitad de las aplicaciones no utilizaban TLS en al menos una transmisión que contenía identificadores u otra información sensible.

III. Metodología

III-A. Escenario de pruebas

El esquema de comunicación comúnmente utilizado por los wearables actuales se muestra en la Figura 1. Un elemento con mayor capacidad de computación (por ejemplo, un smartphone) hace de intermediario (hub, configurador, etc.) entre el wearable y los servidores externos. La tecnología de comunicación más común entre estos dispositivos es BLE.

En el escenario se observan tres áreas de comunicación potenciales de análisis: (i) la primera centrada en la interacción usuario-dispositivo conectado; (ii) la segunda entre el wearable y el hub de comunicación; (iii) y una tercera entre el hub de comunicación y servidores externos o aplicaciones de terceros. En este trabajo, no analizaremos el caso de las conexiones móviles Long-Term Evolution (LTE) entre el hub de comunicación (por ejemplo, un smartphone) y los servidores externos. Nos centraremos en el escenario en el que se utiliza una conexión Wi-Fi en esta interfaz.

III-B. Herramientas de apoyo

La metodología de las pruebas se centra en el análisis de los paquetes de información emitidos por los dispositivos implicados. Para ello, hemos utilizado la herramienta de software Wireshark para analizar las comunicaciones BLE y Wi-Fi. Wireshark es un analizador de paquetes de comunicación

Figura 1. Escenario de pruebas.

ampliamente utilizado. Además, es una herramienta de código abierto y multiplataforma, lo que facilita su adaptación a diferentes herramientas de hardware y sistemas operativos y nos proporciona una gran versatilidad. A continuación, se describen el resto de las herramientas de hardware y software utilizadas para realizar las pruebas previstas.

Utilizamos el dispositivo sniffer nRF52 DK de Nordic Semiconductor para interceptar los paquetes de comunicación BLE. Este dispositivo es compatible con la herramienta Wireshark, es programable y soporta comunicaciones BLE, Bluetooth Mesh, Near Field Communication (NFC) y ANT. Además, para simular algunos ataques (por ejemplo, el ping de la muerte), hemos utilizado BlueZ, el software oficial de la pila de protocolos Bluetooth de Linux, en una Raspberry Pi 4 Modelo B.

Para interceptar los paquetes de comunicación Wi-Fi, hemos utilizado las antenas TP-Link TL-WN722N y Alfa AWUS036ACH. Además, para rastrear los paquetes enviados entre la aplicación del smartphone (del wearable) y el servidor externo, hemos utilizado Mitmproxy, una herramienta de código abierto que proporciona un proxy interactivo con capacidad SSL/TLS para interceptar HTTP/1, HTTP/2 y WebSockets, creando un proxy HTTP para las conexiones del smartphone.

Para proteger la integridad y confidencialidad de los datos transmitidos, HTTPS utiliza el protocolo TLS/SSL para cifrar los datos. Por lo tanto, para interceptar con éxito el tráfico HTTPS transmitido entre un smartphone y un servidor externo, es necesario instalar un certificado raíz personalizado en el dispositivo. Mitmproxy utiliza un certificado creado por él mismo en el que confía el smartphone analizado, implementando un ataque MitM contra la aplicación. Así, el contenido cifrado de los mensajes intercambiados puede ser capturado en texto plano.

Para controlar el entorno de comunicación basado en Wi-Fi se ha creado un hotspot virtual, tal y como se muestra en la Figura 2. Para ello se hace uso de la antena TP-Link TL-WN722N y un sistema operativo Ubuntu 20.04.2.0 virtualizado mediante VirtualBox.

III-C. Categorías de ataque

La siguiente lista muestra los escenarios analizados y las pruebas realizadas en los dispositivos considerados.

- *Autenticación* La aplicación asociada al wearable implementa un método para autenticar la identidad del usuario.
- *Método de emparejamiento inseguro* El enlace entre el wearable y el smartphone utiliza un método de emparejamiento considerado inseguro o ineficaz contra los

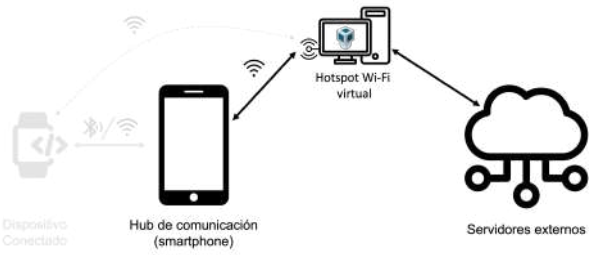


Figura 2. Escenario de hotspot virtualizado.

ataques MitM o los ataques de escucha pasiva y carece de salvaguardias de privacidad.

- *Comunicaciones no encriptadas* Las comunicaciones BLE entre el wearable y el smartphone no están cifradas.
- *Captura de la clave de cifrado* Durante el proceso de emparejamiento, el dispositivo portátil y el móvil intercambian claves de cifrado en un formato que el rastreador BLE puede capturar y procesar fácilmente.
- *Dirección MAC estática* El wearable utiliza una dirección MAC estática (es decir, no cambia cuando el dispositivo se apaga o se reinicia y no cambia periódicamente), lo que lo expone a ataques de seguimiento e identificación de usuarios.
- *Transmisión de información sensible a servidores de terceros* La aplicación envía información sensible del usuario a servidores de terceros.
- *Envío de información y actualizaciones de firmware a través de HTTP* La aplicación recibe actualizaciones de firmware y envía solicitudes con información sensible mediante HTTP sin cifrado TLS.

III-D. Procedimiento de pruebas

Cada categoría de dispositivo conectado sigue un procedimiento operativo diferente, pero que puede generalizarse. De este modo, es posible sistematizar el proceso de adquisición de pruebas durante la ejecución de cada prueba definida para cada dispositivo, en función de la categoría a la que pertenece. Así, es posible obtener un conjunto uniforme de resultados y evitar la exclusión de datos y pruebas relevantes.

- Encendido del wearable y del dispositivo móvil.
- Conexión del wearable con nRF52 DK y Wireshark.
- Registro/Inicio de sesión en la aplicación.
- Proceso de emparejamiento entre wearable y smartphone.
- Actividades de recogida de datos BLE:
  - Realización de actividades físicas como caminar, correr, etc.
  - Sincronización de datos con el wearable.
  - Desconexión con el wearable.
  - Reconexión con el wearable.
- Actividades de recogida de datos HTTP:
  - Edición del perfil de usuario.
  - Sincronización de los datos con los servidores de la nube.
  - Cierre de sesión.
  - Inicio de sesión.
- Desconexión.

III-E. Selección de dispositivos

Para la fase de aplicación de nuestra investigación, se seleccionaron wearables para incluir marcas de alta gama como Fitbit o Garmin, así como dispositivos mucho menos costosos, aunque frecuentes que se encuentran en mercados como Amazon y Alibaba. Además, se hizo un esfuerzo por incluir modelos diseñados específicamente para niños. Los dispositivos seleccionados para el análisis son los siguientes, por marca (entre paréntesis la aplicación de configuración analizada):

- Garmin Vívofit jr. (Garmin jr.)
- Fitbit Ace 3 e Inspire 2 (Fitbit)
- Mi Band 5 (Mi Fit)
- Amazfit Band 5 (Zepp)
- Honor Band 5 y Watch ES (Huawei Health)
- TOOBUR Smartwatch y Smart Band (VeryFitPro)
- BIGGERFIVE Fitness y Vigor (VeryFitPro)
- Apple Watch Series 6

IV. Análisis de resultados

En esta sección, se describen los resultados obtenidos de la aplicación de nuestra metodología y de las pruebas descritas en la sección anterior sobre los wearables comerciales descritos. Estos se encuentran resumidos en la Tabla I.

IV-A. Autenticación

La mayoría de las aplicaciones usadas por estos dispositivos incluyen métodos para la autenticación del usuario, aunque no todas aseguran su uso o hacen que el usuario se registre antes de utilizar dicha aplicación. En este sentido, tanto Mi Band 5 como Amazfit Band 5 requieren que el usuario se conecte a través de las aplicaciones de Huami (Mi Fit y Zepp, respectivamente). Estas aplicaciones requieren una validación, mediante una "cuenta Mi" o una cuenta de terceros como Google, Apple, Mi-Xioami o Facebook.

Del mismo modo, otros wearables de gama alta requieren que los usuarios utilicen aplicaciones propias o específicas. El Garmin Vívofit jr. 2 exige que el usuario registre el dispositivo en la aplicación Garmin Jr. y el Fitbit Ace 3 y el Fitbit Inspire 2 requieren la aplicación Fitbit y una cuenta Fitbit. Ambos procedimientos de registro pueden realizarse desde cuentas de terceros como Google o Apple. En el caso del Fitbit Ace 3, es necesario crear una cuenta familiar. Una vez registrado, la aplicación permite al usuario cambiar entre diferentes vistas para niño/adulto validando con la contraseña de la cuenta.

Honor Band 5 y Honor Watch ES utilizan Huawei Health con un Huawei ID que requiere un teléfono y una dirección de correo electrónico para el registro. Tanto los dispositivos de BIGGERFIVE (Fitness y Vigor) como los de TOOBUR (Smart band y Smartwatch) recomiendan el uso de una aplicación de terceros, VeryFitPro, que no requiere ninguna autenticación ni registro, aunque se puede crear una cuenta de usuario.

IV-B. Emparejamiento y encriptación

Los wearables de Huami y Huawei se conectan al dispositivo central del smartphone sin encriptación, por lo que las comunicaciones en el segmento Bluetooth no están encriptadas. Sin embargo, Zepp, Mi Fit y Huawei Health establecen una conexión entre la banda y los servidores de cada

Tabla I
RESULTADOS DE LOS DISPOSITIVOS ANALIZADOS.

		Autenticación	Métodos de emparej. seguro	Comunicaciones encriptadas	Envío de claves de encriptación encriptadas	Direcciones MAC dinámicas	Comunicaciones y actualización de firmware sobre HTTP
Gama alta	**Amazfit Band 5**	✓	✓	×	Sin encriptación	×	✓
	Apple Watch Series 6	✓	✓	✓	✓	✓	✓
	Fitbit Ace 3	✓	✓	✓	✓	×	✓
	Fitbit Inspire 2	✓	✓	✓	✓	×	✓
	Garmin Vívofit jr. 2	✓	✓	✓	×	×	✓
	Honor Band 5	✓	×	×	Sin encriptación	×	✓
	Honor Watch ES	✓	×	×	Sin encriptación	×	✓
	Mi Band 5	✓	✓	×	Sin encriptación	×	✓
Gama baja	**BIGGERFIVE Fitness**	×	×	×	Sin encriptación	×	×
	BIGGERFIVE Vigor	×	×	×	Sin encriptación	×	×
	TOOBUR Smartwatch	×	×	×	×	×	×
	TOOBUR Smart Band	×	×	×	×	×	×

compañía, ocultando las comunicaciones mediante el uso de servicios propietarios de la compañía e impidiendo el uso de otras aplicaciones. Las aplicaciones autentican y emparejan el teléfono con los servidores de Huami o Huawei y ocultan la Auth Key en el sistema de archivos del teléfono para que otras aplicaciones no puedan utilizarla.

Aunque no es fácil identificar inmediatamente qué información se está intercambiando, dado que las comunicaciones no están cifradas, un atacante podría entender el funcionamiento de los servicios propietarios de Huami o Huawei y obtener los datos del usuario. Varios sitios web demuestran cómo sortear esta limitación [17] [18].

Garmin Vívofit jr.2 utiliza el método Passkey para el emparejamiento, por el que el usuario debe introducir en la aplicación un número que aparece en la pantalla del wearable. Hay cifrado, pero, aunque el dispositivo utiliza la versión 4.2 de BLE, la conexión se establece con el emparejamiento legado de LE en lugar de las conexiones seguras de LE, lo que permite a un husmeador descifrar los paquetes que se intercambian gracias a que la clave a largo plazo (LTK) se envía en texto claro.

Fitbit Ace 3 y Fitbit Inspire 2 implementan Conexiones Seguras BLE y cuentan con el procedimiento de emparejamiento más seguro, cifrando las comunicaciones con una clave pública y Criptografía de Curva Elíptica (ECC). La implementación de un algoritmo de intercambio de claves de curva elíptica Diffie Hellman (ECDH), hace imposible descifrar la comunicación una vez que los dispositivos están emparejados. El método de emparejamiento utilizado es Passkey, con una clave de 4 dígitos en lugar de 6.

Los wearables TOOBUR y BIGGERFIVE analizados se emparejan con el smartphone directamente, sin encriptación, lo que permite que el dispositivo se conecte sin problemas con cualquier otro dispositivo una vez que ha perdido la conectividad con el hub central de comunicaciones (por ejemplo, el smartphone del usuario). Si se empareja desde fuera de VeryFitPro, el LTK de TOOBUR Smartwacth se envía en texto plano, de modo que un sniffer puede interceptar los paquetes intercambiados.

En el caso de la información enviada desde la aplicación a servidores externos a través de Wi-Fi e Internet, es posible observar la información encriptada (a través de HTTPS), pero parte de la información también se envía en texto plano (HTTP). En nuestras pruebas se ha comprobado que esta información transmitida en texto plano contiene datos sensibles, como el sexo del usuario o la dirección MAC. Esta información se envía a servidores externos cuando el usuario intenta actualizar el firmware del dispositivo. El proceso de emparejamiento en el Apple Watch Series 6 es robusto y seguro (dejando de lado los problemas inherentes al Bluetooth, como el BIAS o el KNOB).

IV-C. Dirección MAC

En este caso, salvo el Apple Watch Series 6, ninguno de los dispositivos analizados utilizaba direcciones MAC dinámicas. Si la dirección MAC de un dispositivo es estática, es decir, no cambia al reiniciar o periódicamente, y se anuncia constantemente cuando no está emparejado, un atacante podría identificar fácilmente el dispositivo, lo que supondría un riesgo para la privacidad del usuario.

IV-D. Privacidad

Las aplicaciones Zepp y Mi Fit de Huami solicitan constantemente al usuario que conceda permisos para la localización, los datos de salud y el acceso al álbum de fotos, el contenido multimedia y otros archivos. Del mismo modo, Huawei Health solicita acceso a la ubicación, los contactos, las llamadas, las notificaciones, las fotos, la cámara y el sistema de archivos. Garmin Jr. debe gestionarse desde una cuenta controlada por un adulto. Sin embargo, el método utilizado para identificar si el usuario que registra la cuenta es un adulto está sujeto a simples preguntas de opción múltiple

como "¿Cuál de estos (cuatro) entrenamientos es aeróbico?". La aplicación solicita permisos de localización para utilizar el Bluetooth. Por otro lado, la aplicación Fitbit debe utilizarse desde una cuenta controlada por los padres del niño y permite al usuario cambiar entre dos vistas (menor y adulto), cuyo acceso está protegido por la contraseña de la cuenta. Todas las aplicaciones mencionadas anteriormente utilizan Certificate Pinning para evitar certificados fraudulentos, por lo que es imposible capturar el tráfico HTTPS empleando Mitmproxy.

Por su parte, VeryFitPro, utilizada por los dispositivos BIGGERFIVE y TOOBUR estudiados, solicita permisos para el seguimiento de la ubicación, el acceso a los contactos y mensajes, y el acceso al álbum de fotos y a la cámara. La política de privacidad establece que la aplicación recoge información personal como el IMEI del dispositivo y la ubicación exacta y que dicha información puede ser compartida con terceros. Aunque los dispositivos BIGGERFIVE y TOOBUR analizados están dirigidos a menores de edad, la política de privacidad de la compañía especifica que la aplicación no está pensada para su uso por parte de menores. Utilizando la herramienta Mitmproxy, se ha podido interceptar el tráfico HTTP/HTTPS de la aplicación VeryFitPro, observando que envía información sensible como la edad del usuario, la localización y la dirección MAC del wearable a través de HTTP. Además, la clave de usuario "Invitado" se envía en texto claro, sin cifrar a través de HTTP. En cuanto a la privacidad del Apple Watch Series 6, el dispositivo se rige por los acuerdos base de Apple y toda la información sensible que se maneja se procesa de forma segura.

V. Discusión y conclusiones

En este trabajo se presenta una metodología para analizar los riesgos de seguridad y privacidad en wearables. Utilizando dicha metodología, se evalúan los problemas asociados a los wearables prevalentes, especialmente los dirigidos a niños y jóvenes. Los resultados obtenidos durante las pruebas muestran que la mayoría de los dispositivos de bajo coste, dispositivos de marcas menos conocidas como BIGGERFIVE o TOOBUR, presentan más vulnerabilidades de seguridad y privacidad que los de gama alta, dispositivos de marcas más conocidas como Fitbit, Garmin o Apple. En general, desde el punto de vista de la seguridad, estos dispositivos de bajo coste carecen de los medios o herramientas de autenticación y/o cifrado necesarios para garantizar la integridad de los propios dispositivos o de los datos que manejan. Por ello, la privacidad de sus usuarios se ve comprometida, tanto por el posible acceso a la información sensible que manejan estos dispositivos como porque dicha información se transfiere a través de conexiones inseguras con servidores en la nube y se comparte con terceras empresas.

No obstante, a pesar de que los dispositivos de marcas conocidas tiendan a aplicar más medidas de seguridad y privacidad que los dispositivos de empresas más pequeñas, no se encuentran exentos de estos problemas. Muchos de ellos directamente no cifran las comunicaciones BLE o implementan métodos de emparejamiento insuficientes para garantizar la confidencialidad de los datos personales. Es el caso de Garmin Vívofit jr. 2, Mi Band 5, Honor Band 5 y Honor Watch ES. Aunque intentan ofuscar sus comunicaciones utilizando servicios y atributos BLE propietarios, se ha comprobado en varias ocasiones que estos métodos habían sido vulnerados mediante ingeniería inversa, y existe información de acceso público que describe su funcionamiento.

Curiosamente, los únicos wearables que pueden evitar los ataques MitM y de escucha son el Fitbit Ace 3 y el Apple Watch Series 6, ya que ambos implementan conexiones seguras BLE con intercambio de claves ECDH o métodos de intercambio propietarios seguros. Todos los demás usan versiones obsoletas de BLE, con métodos de emparejamiento heredados como Just Works permitiendo al atacante interceptar las claves y acceder al tráfico descifrado. No obstante, todos los dispositivos son susceptibles de ser atacados por KNOB o BIAS, debido a una vulnerabilidad en la arquitectura de Bluetooth en la versión 5 o anterior.

En cuanto a la privacidad que ofrecen las aplicaciones asociadas al wearable, en general, todas parecen indicar que recogen información sensible del usuario en sus políticas de privacidad. Sin embargo, existe una clara diferencia entre gamas de dispositivos. Los dispositivos de gama alta suelen contar con aplicaciones propietarias de la empresa, mientras que los de gama baja suelen vincularse a aplicaciones de terceros. Las soluciones propietarias implementan Certificate Pinning en HTTPS/TLS evitando así ataques MitM y de espionaje con herramientas como Mitmproxy. Mientras que las de terceros envían sus datos a través de canales inseguros sobre HTTP y no suelen requerir autenticación de usuario, este es el ejemplo de VeryFitPro (utilizada por BIGGERFIVE Fitness y TOOBUR Smartwatch).

Al no cifrar ni la conexión BLE ni las peticiones enviadas por HTTP, VeryFitPro es la aplicación más insegura y menos privada de las analizadas. Al inspeccionar el tráfico BLE intercambiado entre VeryFitPro y los wearables conectados, descubrimos que su funcionamiento es vulnerable a ataques de ingeniería inversa, independientemente del dispositivo conectado. Resulta especialmente preocupante que los dispositivos BIGGERFIVE y TOOBUR Smartwatch, diseñados específicamente para menores, indiquen en sus cajas y manuales que las pulseras deben utilizarse con la aplicación VeryFitPro.

Finalmente, uno de los hallazgos más preocupantes de esta investigación es que todos los dispositivos analizados utilizan direcciones MAC estáticas, excepto el Apple Watch Series 6. La dirección MAC de un dispositivo periférico BLE se anuncia constantemente sin cifrar cuando se desconecta de su controlador central, lo que lo hace vulnerable a ser rastreado e identificado por un atacante. Como solución a esta brecha de seguridad se propone hacer uso de direcciones privadas y aleatorias que cambian periódicamente y que es soportado por BLE.

A lo largo de esta investigación y, especialmente, en el proceso de realización de pruebas y evaluación de las vulnerabilidades encontradas no se han tenido en cuenta aspectos relacionados con el tipo de dispositivo, precio, fabricación u origen. Los resultados extraídos muestran claras brechas de seguridad y privacidad en wearables comúnmente usados entre la población. Por lo que esta investigación espera servir de apoyo para futuras mejoras técnicas por parte de los fabricantes y como concienciación para los usuarios de dichos dispositivos de cara a las existentes vulnerabilidades encontradas.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el programa Horizon 2020 de la Unión Europea a través del proyecto RAYUELA (nº de contrato 882828). El contenido del artículo refleja solo el punto de vista de sus autores. La Comisión Europea no es responsable del uso que se pueda hacer de la información que contiene.

REFERENCIAS

[1] F. Laricchia, "Wearables unit shipments worldwide by vendor from 1st quarter 2014 to 3rd quarter 2021," https://www.statista.com/statistics/435933/quarterly-wearables-shipments-worldwide-by-vendor/, February 2022, [Accessed 18 February 2022].

[2] Forbrukerrådet, "Watchout: Analysis of smartwatches for children," 2017. [Online]. Available: https://fil.forbrukerradet.no/wp-content/uploads/2017/10/watchout-rapport-october-2017.pdf

[3] Bundesnetzagentur, "Bundesnetzagentur takes action against children'swatches with "eavesdropping" function," 2017. [Online]. Available: https://www.bundesnetzagentur.de/SharedDocs/Downloads/EN/BNetzA/PressSection/PressReleases/2017/17112017_Verbraucherschutz.pdf?__blob=publicationFile&v=4

[4] J. K. Andrew Hilts, Christopher Parsons, "Every step you fake: A comparative analysis of fitness tracker privacy and security," 2016. [Online]. Available: https://openeffect.ca/reports/Every_Step_You_Fake.pdf

[5] C. Zuo, H. Wen, and Y. Zhang, "Automatic fingerprinting of vulnerable ble iot devices with static uuids from mobile apps," 11 2019, pp. 1469–1483.

[6] A. Das, P. Pathak, C.-N. Chuah, and P. Mohapatra, "Uncovering privacy leakage in ble network traffic of wearable fitness trackers," 02 2016, pp. 99–104.

[7] S. Senevirante, Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, and A. Seneviratne, "A survey of wearable devices and challenges," *IEEE COMMUNICATIONS SURVEYS and TUTORIALS*, vol. 19, no. 4, 2017.

[8] M. Langone, R. Setola, and J. López, "Cybersecurity of wearable devices: an experimental analysis and a vulnerability assessment method," *IEEE 41st Annual Computer Software and Applications Conference*, 2017.

[9] J. Padgette, J. Bahr, M. Batra, M. Holtmann, R. Smithbey, L. Chen, and K. Scarfone, "Guide to bluetooth security," 2017-05-08 00:05:00 2017.

[10] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[11] M. Ryan, "Bluetooth: With low energy comes low security," 08 2013, pp. 4–4.

[12] W. Zegeye, R. Dean, F. Moazzami, and Y. Astatke, "Exploiting bluetooth low energy pairing vulnerability in telemedicine," 10 2015.

[13] T. Rosa, "Bypassing passkey authentication in bluetooth low energy," 05 2013.

[14] J. Reardon, Feal, P. Wijesekera, A. Elazari Bar On, N. Vallina-Rodriguez, and S. Egelman, "50 ways to leak your data: An exploration of apps' circumvention of the android permissions system," *28th USENIX Security Symposium*, 2019.

[15] A. Feal, P. Calciati, N. Vallina-Rodríguez, C. Troncoso, and A. Gorla, "Angel or devil? a privacy study of mobile parental control apps," *Privacy Enhancing Technologies Symposium (PETS)*, 2020.

[16] I. Reyes, P. Wijesekera, J. Reardon, A. Elazari, B. On, A. Razaghpanah, N. Vallina Rodriguez, and S. Egelman, "Wont somebody think of the children? examining coppa compliance at scale," *Privacy Enhancing Technologies Symposium (PETS)*, 2018.

[17] Y. Ojha. I hacked miband 3, and here is how i did it. part i. [Online]. Available: https://medium.com/@yogeshojha/i-hacked-xiaomi-miband-3-and-here-is-how-i-did-it-43d68c272391

[18] How to use mi band 5 without the mi fit app. [Online]. Available: https://techwiser.com/use-mi-band-without-the-mi-fit-app

OpenUEBA – A systematic approach to learn behavioural patterns

Albert Calvo
i2CAT Foundation
08034, Barcelona
albert.calvo@i2cat

Nil Ortiz
i2CAT Foundation
08034, Barcelona
nil.ortiz@i2cat.net

Jordi Guijarro
i2cat Foundation
08034, Barcelona
jordi.guijarro@i2cat.net

Shuaib Siddiqui
i2cat Foundation
08034, Barcelona
shuaib.siddiqui@i2cat.net

***Abstract*—For years, Security Operations Centers (SOC) have resorted to SIEM and IDS tools as the core defence shield, offering reactive detection capabilities against latent threats. Despite the effectiveness of the tools described above, cyber-criminal groups have professionalized themselves by launching very sophisticated campaigns that unfortunately, go unnoticed by current detection tools. In order to revolutionize the current range of security tools, we present our vision and advances in openUEBA; An open-source framework focused on the study of the behaviour of users and entities on the network; Where through state-of-the-art Artificial Intelligence techniques are learn behavioural patterns of those users who later fall into cyber attacks. With the learnt knowledge, the tool calculates the user exposure; in other words, it predicts which users will be victims of latent threats, allowing the analyst to make preventive decisions.**

***Index Terms*—Cybersecurity, Artificial Intelligence, Behavioral Analytics,**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

Data-driven applications are disrupting our modern culture; changing our careers, routines and habits. The cybersecurity sector is not lagging, being Artificial Intelligence (AI) adopted in the nearby 80s as a paradigm to automatize expert knowledge. Even the great performance of these technologies, being the core of current Intrusion detection Systems (IDS) and Firewalls, is not enough to detect new multilayered attacks in the actual hyper-hybrid environments [1].

The growing opportunity of AI to lead the next cybersecurity defensive tools is significant and will disrupt the market, changing the current paradigm from expert systems to data-driven systems, allowing to enhance the decision making: reducing the response time and learning from historical facts. User and Entity Behaviour Analytics (UEBA) is a latent research field targeted to model and analyse the users and entities behaviour within a network through Artificial Intelligence [2].

In fact, network and endpoint monitoring as per security concerns have been around for a while already, developing a fast-growing industry revolving around security operations centres, with a centralized model of log collection, event correlation and analysis using SIEM technologies. The previously mentioned scheme left one part of the security scope uncovered, the user itself, which can be both the object and vector of a cybersecurity attack. Artificial intelligence has been dealing with user profiling and behaviour analytics for some time already in other areas like marketing, sales and overall business intelligence, hence the knowledge in this area is already mature enough to be transferred to other fields like cybersecurity, integrating it with existing threat intelligence tools to provide a user or entity threat profile based on its behaviour and the behaviour of known and unknown threats, adding another layer of visibility to the security landscape.

The advantages of UEBA tools rely on the ability to analyze large amounts of data in a time-effective manner, allowing to enhance visibility within the network; Learning the routines of each entity and flag anomalous activities that are potentially linked to an encompassing palette of threats by calculating the risk to threats. However, designing and integrating UEBA frameworks is challenging. One of the major issues relies on homogenising multiple sources of data to enrich the user and entity profiles allowing the detection of multi-layered attacks.

This paper introduces our advances towards *openUEBA* - an open-source framework targeted to estimate the user and entity exposition degree against specific threats, allowing stakeholders to take counterfactual preventive measures. In detail, the framework resorts to Artificial Intelligence techniques to learn behavioural patterns from clients with evidence of compromise. Then, the discovered patterns are inferred, computing the behaviour likelihood of entities producing a ranked entity list.

To the best of the author's knowledge, the exposition analysis has not been addressed in the literature, making this work novel. Further, we believe the following aspects enhance the novelty of our work:

- **Multimodal data** The proposed framework exploits several heterogeneous data sources allowing to build rich entity profiles.
- **Threat Intelligence alignment** The framework aligns with Open-source intelligence sources (OSINT) to enhance the profiles and allow the risk calculation of incoming threats and vulnerabilities.
- **Real-world validation** The framework is designed under a mission-driven project and validated under two real-life test-bed. The first of them uses live data from a Spanish university and the second one uses regional governmental data.

The remainder of this paper has the following structure: Section II-A provides a state-of-the-art overview. The proposed framework is introduced in section III. Lastly, sections IV and V are left for discussion of open challenges, conclusions and our intended future work.

II. Foundations of User and Entity Behaviour Analytics

The foundations of Behaviour analytics rely on psychology, marketing and biology studies where the behaviour is modelled to understand interactions in order to achieve objectives. In the cybersecurity domain, behaviour analytics profiles the baseline behaviour of users and entities in the network and outliers or abnormal behaviours are pinpointed as potential threats [3].

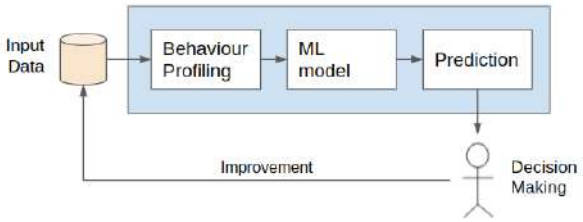


Fig. 1. UEBA workflow diagram. The input data is profiled to learn a ML-model and build predictions for decision making.

The UEBA methodologies are conceptualized and modelled as data-driven projects: The first step of any UEBA framework is behaviour Profiling, in this step the different data sources are modelled in feature vectors used to represent the properties or characteristics of the users. Later, the calculated feature vectors are fitted into a Machine Learning model used to learn the *Historical Behaviour*; describing what is the baseline activities of the user and also the *Peer Behaviour*; the similarity of the user amongst other users. Finally, the risk analysis module generates valuable knowledge for the stakeholder.

A. Related work

Anomaly and threat detection can be divided into two groups depending on how the analysis is performed. The first approach is signature-based detection which is based on building a knowledge database using the characterization of previous threats and comparing the signatures with current network traffic. The signature-based systems are categorized into three subcategories: Misuse detection, which uses known signatures from threat intelligence sources. The anomaly-based detection defines a set of thresholds from historical incidents. Finally, the hybrid method is a trade-off between using a knowledge system (misuse based system) and the patterns defined from historical incidents.

The second approach is Behaviour analytics, that focus on determining the user baseline behaviour and comparing it in two dimensions: historically and amongst peers. In comparison to a signature-based detection system, the analytics does not require a knowledge database providing flexibility, thus allowing the detection of zero-day attacks. Behaviour analytics have attracted a lot of attention to security companies but the real-world implementations are not publicly disclosed, thus not included in the comparison. The successful survey carried out by [3] includes the most relevant works in behavioral analytics and denotes the increasing interest on the domain.

One of the first efforts to develop Behaviour-based analytics in the cybersecurity domain studies is in [4], where the authors conceptualize an intrusion detection system based on the comparison of user profiles against the historic. The authors study the viability of the model proposed using an experimental setup on a limited users spectrum and includes OS and applications metadata. In a later publication - [5], the authors under the same hypothesis, develop an intrusion detection system through behavioural analytics. In detail, the authors use unsupervised techniques such as DBSCAN and hierarchical agglomerative clustering. In [2] the authors propose to use Singular Value Decomposition (SVD) to compare the client traffic behaviour against time and with other users.

III. Our proposed framework

OpenUEBA aims to compute the user exposition against latent threats by learning the behaviour of the different entities in the network. Due to the magnitude of the challenge, we initially resort to threat intelligence sources to identify suspicious entities by gathering Indicators of Compromise (IoCs), which are actionable pieces of evidence of potential threats in the network.

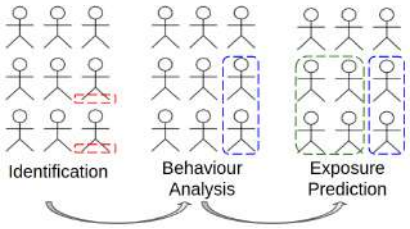


Fig. 2. Our analysis resorts on identification, behavior analysis and exposure prediction

Once identified the suspicious entities in the network. The framework determines which entities have unusual patterns and might be compromised. This step is introduced since it is a sufficient but not necessary condition; A user with activity related to an IoC, is not necessarily compromised. For instance, a user might open a link from a phishing campaign but might not experience any impact.

The next step is to extract behavioural patterns from the previously identified entities which characterize the habits and biases. The behavioural patterns, as opposed to misuse systems, are resilient and allow abstracting TTPs, which can be used to predict the exposition of entities on a network. I.e. If a user has similar behavioural patterns related to a threat, the exposition score to the threat will be high.

A. Methodology

The stated framework is being developed under the CRISP-DM methodology and using real data from test-beds. In detail, we extract the data via network sensors and aggregate it using Elasticsearch, which is used to query and transform data into event sequences for the subsequent analysis (see subsection III-A3). In detail, the event sequences are built using the following sources:

- **Activity data** Data generated by the users and devices connected on the network, comprising security events, application logs and network traces.
- **Identity data** LDAP and other user inventory related data which can characterize a user or device within the network.

- **Threat intelligence data** Data from threat feeds and threat reports generated by external entities related to threats seen in the wild and external networks.
- **Historical Incident data** Data from documented incidents on the network, where known local entities were affected by threats during a specific time period, usually managed via ticketing systems.

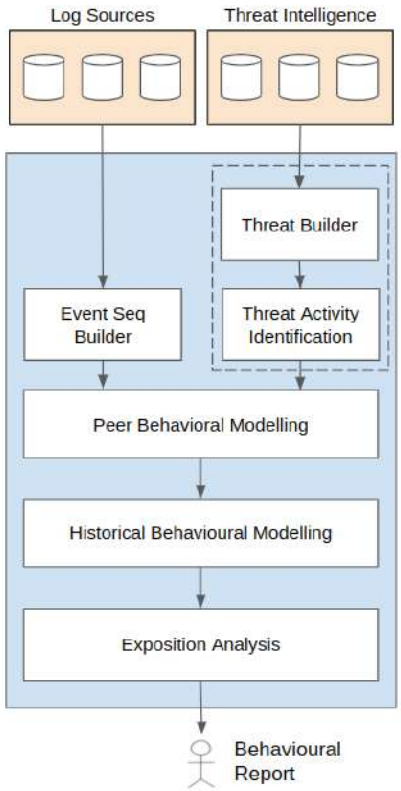


Fig. 3. Our framework allows us to learn Behavioral patterns from historical data and predict exposition

Once the event sequences are computed, the next step is to identify which users have associated IoCs by matching the extracted information from Threat Intelligence sources with the entities activity. The peer behavioural modelling module allows, from the previous suspicious clients, to select the clients with high statistical evidence of real threat impact resorting to clustering and forecasting techniques (see subsection III-A4). Then, the Historical Behavioral Modelling module (see subsection III-A5) allows us to identify and extract the common patterns in the entities subset. Finally, the Exposition Analysis module (see subsection III-A6) determines users with similar behaviour to those determined in the behavioral patterns. A flow chart of the framework is included in figure 5.

1) Threat Builder: We aggregate threat intelligence data from IoC feeds and threat reports and sample them into campaigns and incidents, enriching the campaign with information about TTPs from the MITRE ATTCK framework and contextual data of potential threat actors related to the identified TTPs. This module will also be retro alimented with the behaviour indicators identified on later stages of the analytics process.

2) Threat Activity Identification: We match the data from the documented threat objects with the activity data generated by users and devices to identify entities within the network with a similar behaviour to the threat.

3) Event Sequences Builder: In order to perform the corresponding analysis are introduced event sequences : $\mathcal{S}_{te}$, where are a contiguous sequence of n features for a given entity (e) at the timestamp (t)with fixed length k. Being $historic$: $\{\mathcal{S}_{te} | \ t < today\}$ $\forall$e $\in$ E, the historical set of event sequences and $actual$: $\{\mathcal{S}_{te} | \ t = today\}$ $\forall$e $\in$ E the sequences in current day. The proposed abstraction allows knowing the specific activities of the user through time. Instead of performing the analysis directly to log data, we observe the main benefits: (1) **Complexity reduction**, the event sequences contain an aggregation of the log data in an interval, drastically reducing the amount of data in the later analysis. (2) **Multimodal data**, the proposed abstraction allows a straightforward method to aggregate and correlate data from different sources. Finally, it offers (3) **Enhanced visibility**, the event sequences allows comparing entities from a historical and peer perspective a simpler task.

Even if the list of the features is not publicly disclosed yet, since our framework is still under modelling and validation phases. Some of the features are defined from simple statistical indicators (the mean HTTP frequency, number of SSL connections using specific protocols or the header length of received emails) to complex and non-trivial features (determine if the received email contains phishing content, ...) or features that involve threat intelligence feeds (similarity between entity DNS queries and published URLs in threat intelligence feeds).

4) Peer Behavioral Modelling: The Behavioural Modelling module determines entities with statistical evidence of a security incident. Given the subset of event sequences matching Threat activity identified in the corresponding module, it is used to build the following two-step methodology: The first step, characterize the entities into neighbourhoods reducing the variance in the posterior step. It is applied clustering techniques to place similar entities in the same region. We are evaluating the application of hierarchical clustering techniques during this stage. Later, for each discovered neighbourhood, it is training a forecasting model using the historical event sequences evaluating the fitness of Arima and additive models. Finally, for each entity is measured the error between the actual data (tag) and the forecasted values being possible to rank the entities.

5) Historical Behavioral Modelling: From the depicted entities is analyzed the historical behaviour to determine shared patterns. These patterns characterize the habits and aspects of users that have been compromised. For instance, the tendency to open shortened URLs or blindly follow e-mail URLs. To this end, we are considering a large umbrella of Machine Learning techniques ranging from statistical analysis: z-score and wavelet analysis to state-of-the-art techniques such as Transformers.

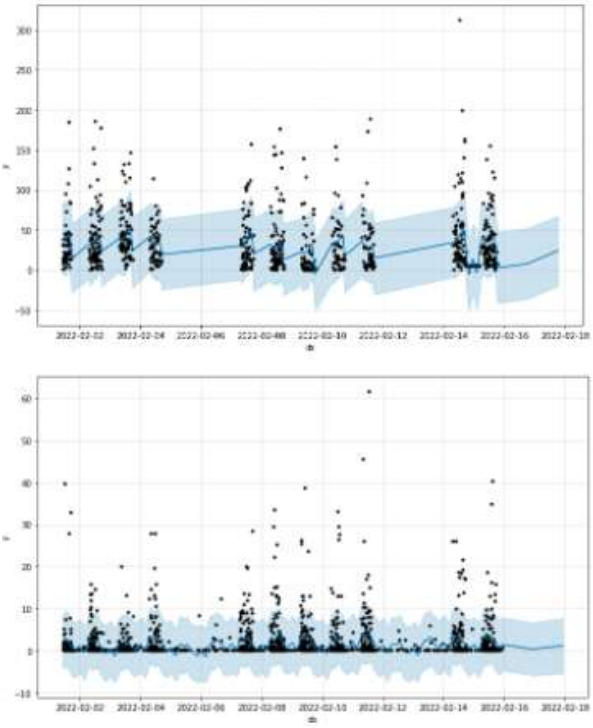


Fig. 4. The historical event sequences of each neighbourhood are fitted into a forecasting model. From top to bottom are show the uncertainty intervals of the historical values of two neighbourhoods, each sample represents the behaviour at a specific time.

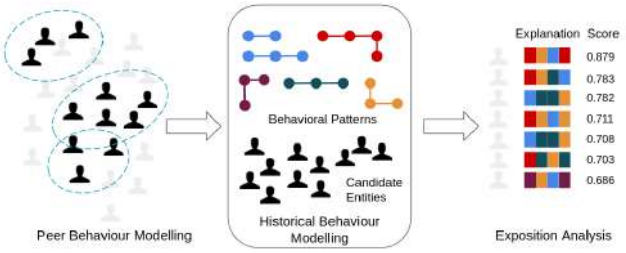


Fig. 5. The Historical Behavioral Modelling module takes as input the candidate users identified in the previous module generating a set of behavioural patterns. Later, the patterns are inferred by computing the exposition score and a corresponding explanation.

6) Exposition Analysis: Finally, the extracted behavioral patterns are inferred to unseen entities calculating the exposition score, the statistical probability of being compromised and an explanation, the justification to the previous calculus. The inference is performed by analyzing the similarity of the historical to the behavioral patterns - the samples with a historical activity similar to behavioral patterns, their exposition will be higher.

IV. Open challenges

Throughout the manuscript is presented a systematic approach to discovering behavioral patterns. Even the notable significance of our work we expose some open challenges for discussion:

- **Data Volume and privacy issues** The current test-bed generates a data flow of nearby 200Gb/day, being necessary to keep a subset to characterize each entity, and a real challenge to define it. Further, the storage of personal data raises privacy issues. In this sense we are elaborating a policy matrix that will allow users to quantify the impact of each data source on the capabilities of the framework, to gauge the need and usage of such data.
- **Ground-truth generation and validation** The validation of data-driven applications in the cybersecurity domain is a common issue. From our experience, the current open datasets simulating attacks are not easily extrapolating to behaviour analysis. To this end, we resort to IoC data as a novel approach for the ground-truth generation and the later validation in the domain.
- **Alignment between logs and Cyber Threat Intelligence (CTI)** Even using standardised data models such as STIX, the alignment between threat data and log sources is not automatic. To this end, we are focusing on adding the threat intelligence knowledge as a pillar in the event sequence builder procedure.
- **Stakeholder-in-the-loop** As seen, our framework automatise the decision making is necessary to provide enough mechanisms to justify each prediction provided. To this end, we will consider transparency and interpretability as a metric during the later evaluation.

V. Conclusions and future work

We presented, hereby, the main components of the framework *openUEBA*. We believe that there is a sufficient basis to show our approach and discuss the current work in progress. The next steps of our proposed work are to keep validating the framework, propose quantifiable metrics to measure the effectiveness of our tools and address the open challenges stated in previous sections.

Acknowledgments

This work has been supported by Smart Catalonia actions in collaboration with the Cybersecurity Agency of Catalonia, as well as the valuable support of all i2CAT team.

References

[1] S. Nayyar, *Borderless Behavior Analytics - Second Edition: Who's Inside? What're They Doing?*, 2nd ed. North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2018.

[2] M. Shashanka, M. Y. Shen, and J. Wang, "User and entity behavior analytics for enterprise security," in *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*. Institute of Electrical and Electronics Engineers Inc., 2016, pp. 1867–1874.

[3] A. G. Martín, A. Fernández-Isabel, I. Martín de Diego, and M. Beltrán, "A survey for user behavior analysis based on machine learning techniques: current models and applications," *Applied Intelligence*, vol. 51, no. 8, pp. 6029–6055, 8 2021. [Online]. Available: https://link-springer-com.recursos.biblioteca.upc.edu/article/10.1007/s10489-020-02160-x

[4] G. Pannell and H. Ashman, "Anomaly detection over user profiles for intrusion detection," in *Proceedings of the 8th Australian Information Security Management Conference*, 2010, pp. 81–94. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.258.660

[5] M. Garchery and M. Granitzer, "Identifying and Clustering Users for Unsupervised Intrusion Detection in Corporate Audit Sessions," in *Proceedings - 2019 IEEE International Conference on Cognitive Computing, ICCC 2019 - Part of the 2019 IEEE World Congress on Services*. Institute of Electrical and Electronics Engineers Inc., 7 2019, pp. 19–27.

Evaluando la Seguridad y Privacidad de los Asistentes Personales Inteligentes: ¡Ojo con el Juguete!

Cayetano Valero[1], Jaime Pérez[1], Sonia Solera-Cotanilla[2], Mario Vega-Barbas[2], Guillermo Suárez[3], Gregorio López[1], Manuel Álvarez-Campana[2]
[1]Instituto de Investigación Tecnológica, ICAI, Universidad Pontificia Comillas
cayetanova@alu.comillas.edu, jaime.perez@iit.comillas.edu, gllopez@comillas.edu
[2]ETSI Telecomunicación, Universidad Politécnica de Madrid
{sonia.solera, mario.vega, manuel.alvarez-campana}@upm.es
[3]IMDEA Networks Institute
guillermo.suarez-tangil@imdea.org

***Resumen*—Los avances en los campos de la Internet de las Cosas, el Reconocimiento del Habla y la Inteligencia Artificial han facilitado el desarrollo de los Asistentes Personales Inteligentes. Esta familia de aplicaciones y dispositivos permiten, a través de la voz, solicitar una amplia gama de tareas de manera natural e intuitiva. La enorme popularidad alcanzada los ha convertido también en un blanco muy codiciado para los ciberataques. Así, en los últimos años han aparecido numerosos informes y estudios cuestionando su seguridad y privacidad. En este trabajo se profundiza en el análisis de las vulnerabilidades que presentan los asistentes personales más utilizados (Alexa, Google Assistant y Siri). Esto se lleva a cabo mediante el desarrollo de una metodología general para la realización de pruebas de seguridad y privacidad, aplicándola a una muestra de los asistentes comerciales más extendidos actualmente. Los resultados permiten corroborar la existencia de múltiples vulnerabilidades y, en consecuencia, la necesidad de avanzar en la mejora de la seguridad y la privacidad de estos dispositivos. En este sentido, se proponen algunas recomendaciones tanto para los usuarios como para los fabricantes.**

***Index Terms*—Asistentes Personal Inteligente, Ciberseguridad, Hogar Inteligente, Internet de las Cosas, Metodología de Pruebas, Privacidad**

Tipo de contribución: *Contribución científica original*

I. Introducción

En los últimos años, los avances en el ámbito de la Inteligencia Artificial (IA) han dado lugar a la aparición de dispositivos y aplicaciones que permiten la interacción con los usuarios mediante la voz. Esta manera de comunicarse resulta mucho más intuitiva, natural y cómoda que el uso de teclados o mandos de control remoto [1].

Uno de los casos más significativos y con gran aceptación por el público general es el de los Asistentes Personales Inteligentes (SPAs, por sus siglas en inglés). Estos dispositivos permiten al usuario solicitar mediante órdenes vocales la realización de innumerables tareas: consultar el tiempo o el estado de tráfico, escuchar una canción, hacer una llamada, realizar compras, controlar un dispositivo remoto (por ejemplo, una bombilla o un termostato), etc. [2].

Los SPAs se diferencian de los sistemas tradicionales de interacción de voz en la manera de procesar las órdenes del usuario. Mientras que estos últimos solo admiten un repertorio rígido de órdenes predefinidas, los SPAs se apoyan en los avances del procesado natural del lenguaje para tratar de entender lo que el usuario realmente quiere y responderle de la mejor manera posible.

El amplio ecosistema de aplicaciones de terceros (*skills*), surgido entorno a los SPAs y la compatibilidad con numerosos dispositivos del hogar, hace que las posibilidades sean prácticamente ilimitadas. Todas estas ventajas los han convertido en uno de los dispositivos más populares, presentes en la actualidad en más de 200 millones de hogares y con previsiones de superar los 500 millones en 2030 [3][4]. Desafortunadamente, esta popularidad los ha puesto también en el punto de mira de los ciberdelincuentes, tratando de explotar las vulnerabilidades de seguridad y privacidad presentes en los SPAs.

La privacidad en los SPAs resulta especialmente delicada por su capacidad para recopilar información sobre lo que ocurre en los hogares a través de lo que captan sus micrófonos. Esto los convierte también en objeto de deseo para las empresas que se nutren de los datos personales para el marketing dirigido (como *Google*, *Apple*, *Amazon* o *Facebook*) [5].

El funcionamiento de estos dispositivos requiere que estén siempre escuchando, lo que plantea problemas de privacidad para sus usuarios. Según el Reglamento General de Protección de Datos (RGPD) vigente en la Unión Europea, la voz y los datos se consideran confidenciales. Esto conlleva a la obligatoriedad de solicitar la autorización explícita del usuario para su tratamiento, así como de aplicar el enfoque de privacidad por diseño, algo que, en general, no cumplen los dispositivos comerciales [6].

Los problemas se agravan si se tiene en cuenta que entre el colectivo de usuarios de los SPAs se encuentran niños y adolescentes, facilitando una mayor exposición de estos ante potenciales estafadores y acosadores [7]. Es necesario, en definitiva, ser conscientes de los problemas de privacidad y seguridad a los que se exponen los usuarios de los SPAs.

En este trabajo se analiza la seguridad y privacidad en los SPAs comerciales más relevantes desde el punto de vista del usuario. Con ello, se persiguen dos objetivos principales: (i) aumentar la transparencia y la conciencia de los usuarios sobre el funcionamiento de los SPAs desde las perspectivas de la seguridad y la gestión de datos; y (ii) impulsar a los fabricantes a mejorar la seguridad de estos dispositivos, y

especialmente la protección de los colectivos de usuarios más vulnerables, como los niños.

En concreto, las principales aportaciones del trabajo son:

- El desarrollo de una metodología para la realización de pruebas de vulnerabilidades de seguridad y privacidad en SPAs comerciales, definiendo el escenario de pruebas, los pasos a seguir, las herramientas hardware y software a utilizar, y el procedimiento de verificación de los resultados.
- La aplicación de la metodología de pruebas a los SPAs comerciales más relevantes en la actualidad, a fin de analizar qué niveles de seguridad y de privacidad ofrecen y contrastar los resultados de los diferentes dispositivos.

El resto del artículo se organiza como sigue. La sección II proporciona una panorámica del ecosistema SPA, haciendo énfasis en la seguridad. La sección III describe la metodología de pruebas. En la sección IV se analizan y comparan los resultados de la aplicación de la metodología al conjunto de SPAs comerciales seleccionados. A la vista de los mismos, en la sección V se proponen una serie de recomendaciones y precauciones a seguir a la hora de utilizar los SPAs más populares actualmente y se discuten las líneas de investigación futuras. Además, se incluye un resumen del artículo y las principales conclusiones del trabajo.

II. Antecedentes

II-A. Ecosistema SPA

Un SPA incluye los componentes hardware y software para la grabación, procesado y análisis de la voz, si bien para su correcto funcionamiento requiere la comunicación con otros elementos del ecosistema SPA, tal como se ilustra en la Figura 1. En dicho ecosistema, el usuario interactúa con los siguientes componentes básicos:

- Asistente Personal o Asistente Virtual. Pueden encontrarse en varias familias de dispositivos electrónicos como altavoces, smartphones, PCs o wearables. Se encarga de captar las frases pronunciadas por el usuario y enviarlas a la nube del proveedor de servicio SPA, así como de reproducir las respuestas vocales provenientes de éste.
- Nube del Proveedor de Servicio SPA. En este componente reside la inteligencia del asistente, siendo donde se procesan las órdenes del usuario para inferir sus intenciones y decidir cuál es la respuesta más adecuada.
- Nubes de Terceros y Accesorios. Permite la interacción del usuario con otros elementos ajenos al proveedor de servicio SPA, como *skills* de terceros o dispositivos inteligentes de otros fabricantes. Para ello se requiere la existencia de conectividad entre dichos elementos externos y la nube del proveedor de servicio SPA.

Por su facilidad de uso y capacidades de orquestación, el SPA se ha convertido en el elemento central (*hub*) del entorno del hogar inteligente (*Smart Home*), actuando como intermediario entre el usuario y el resto de los dispositivos conectados. Cada vez son más los fabricantes de dispositivos inteligentes que integran sus productos en los ecosistemas SPA [8][9][10]. Este rol central ha convertido a los SPAs en un objetivo claro para los actores maliciosos, ya que su toma de control permitiría acceder a los datos sensibles que maneja y a la interacción con otros dispositivos inteligentes.

La información que recopila y procesa un SPA es uno de los principales activos a proteger, ya sea por el valor intrínseco de los datos o porque contengan las credenciales de acceso al dispositivo. Entre la información que maneja un SPA se incluye los datos sobre la cuenta de usuario, datos de compras, registros de las interacciones con el dispositivo, o las propias grabaciones de voz del usuario [11][12][13].

II-B. Seguridad de los SPAs

Las vulnerabilidades presentes en los SPAs están relacionadas con su arquitectura y operación. Son varios los estudios en los que se analizan estas vulnerabilidades, cómo pueden ser explotadas por un actor malicioso y las consecuencias que esto supone para los usuarios. Los principales riesgos de seguridad radican en la naturaleza abierta del canal de comunicación entre el usuario y el asistente, el uso de aplicaciones de terceros y la posibilidad de utilizar técnicas de IA para generar órdenes suplantando la voz del usuario y solicitando la realización de tareas en su nombre.

Recientemente, Cheng and Roedig [14] han propuesto una taxonomía de problemas de seguridad y privacidad en los SPAs, definiendo cuatro categorías: Control de Acceso, Denegación de Servicio Acústico, Privacidad de la Voz y Monitorización de Actividad Acústica. Este artículo se centra en la categoría de Control de Acceso, en donde se concentran los problemas de seguridad más críticos de los SPAs. Dentro de esta categoría cabe identificar cuatro subcategorías especialmente relevantes: Debilidad de Activación, Debilidad de Autenticación, Ataques de Adversario con IA y Uso de *Skills* de Terceros.

II-B1. Debilidad en Activación: El funcionamiento de un SPA conlleva un modo de escucha permanente, quedando a la espera de que el usuario pronuncie la frase clave de activación seguida de las instrucciones que expresan la tarea a realizar. Al reconocer la frase clave, el dispositivo entra en el modo de operación activo en el que asistente procede a grabar la voz del usuario para acto seguido procesarla y generar la respuesta apropiada. En otras palabras, el único mecanismo de autenticación utilizado por un SPA es la frase de activación. Esta puede escogerse entre un conjunto reducido de frases predefinidas por cada fabricante [15], lo que facilita que un atacante pueda adivinarlo e interactuar con el dispositivo como si fuera su usuario legítimo. Con el asistente continuamente a la escucha, la ventana temporal en la que se puede perpetrar el ataque es enorme, pudiendo incluso utilizar un dispositivo cercano al SPA para reproducir comandos de voz sin necesidad de estar físicamente en las inmediaciones, solicitando todo tipo de acciones: abrir una

Figura 1. Ecosistema completo del SPA. (Iconos creados con Flaticon)

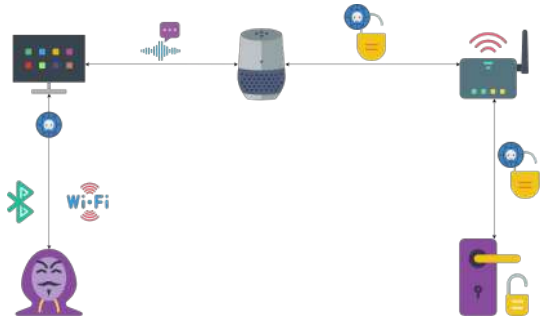


Figura 2. Ejemplo de ataque remoto a un SPA.

puerta (Figura 2), efectuar una compra en nombre del usuario [16], etc.)

II-B2. Debilidad en Autorización: Algunos SPAs comerciales, como *Alexa* o *Google Assistant*, permiten distinguir la voz de sus usuarios aplicando técnicas de reconocimiento de voz al analizar la frase de activación. Los fabricantes no consideran esta característica como un mecanismo de seguridad extra del dispositivo, sino de personalización. Ello se debe a la elevada tasa de falsos positivos que arrojan, pudiendo dar lugar a que una persona con rasgos de voz similares pueda suplantar al usuario [17]. Según los permisos concedidos por el usuario al SPA, esto podría facilitar el acceso de un tercero a sus datos personales, consultar su agenda, realizar una compra, mandar un e-mail, etc.

II-B3. Ataques de Adversario con IA: El uso de IA en los SPAs conlleva ciertos riesgos inherentes a esta tecnología, como la posibilidad de aplicar IA también para atacar los modelos de reconocimiento de voz [18]. Los entornos en los que se usan dichos modelos se suponen seguros, pero debido a la naturaleza abierta del canal de comunicación entre el usuario y el SPA, cabe la posibilidad de que un actor malicioso pudiera inyectar muestras especialmente concebidas para causar un comportamiento no deseado en el dispositivo, permitiendo realizar acciones maliciosas como la descarga de *malware* o provocar una denegación de servicio [19].

II-C. Uso de Skills de Terceros

Los *skills* de terceros permiten ampliar las funcionalidades de los SPAs, de modo que estos puedan invocar aplicaciones de terceros como si de una funcionalidad nativa del dispositivo se tratara. Esto supone, sin embargo, aumentar también la exposición al riesgo del usuario, ya que implica una mayor superficie de ataque ante un potencial actor malicioso. De este modo, un atacante podría crear *skills* maliciosas cuya pronunciación sea similar a la de otra legítima, secuestrando órdenes del usuario dirigidas a una *skill* legítima. Esto puede permitir la toma de control de servicios normalmente reservados para el SPA sin que el usuario sea consciente de ello. Un ejemplo común de ello es aquellas *skills* maliciosas que engañan al usuario haciéndole creer que el dispositivo ha ejecutado la orden y dejado de escuchar cuando en realidad se encuentra activo grabando y almacenando la información para el atacante [20].

III. Metodología propuesta

III-A. Desarrollo de la metodología

Como se ha indicado, uno de los principales objetivos de este trabajo es del desarrollo de una metodología para analizar las vulnerabilidades de seguridad y privacidad en SPAs comerciales. La Figura 3 muestra el procedimiento seguido para el desarrollo y validación de la metodología.

En el planteamiento de la metodología se estableció como requisito que fuera fácilmente aplicable a la mayoría de los SPAs disponibles en el mercado. Por ello, durante el desarrollo de la metodología se efectuó un análisis detallado del funcionamiento de los principales fabricantes de SPAs. Estos dispositivos, a su vez, han sido los utilizados para la realización de las pruebas de validación de la metodología. Los dispositivos seleccionados y sus principales características se describen en la sección III-D.

Una vez seleccionados los dispositivos comerciales, se procedió a analizar los principales tipos de interacciones que pueden entablarse entre el usuario y los SPAs. En base a este estudio, se identificaron cuatro categorías de procedimientos que caracterizan el uso de esta familia de dispositivos: (i) instalación de SPA, accesorios y *skills*; (ii) gestión de interacciones; (iii) gestión de funcionalidades y (iv) mecanismos de seguridad y privacidad. Seguidamente, en base al análisis de la literatura sobre vulnerabilidades de SPAs (ver sección II-B) y diversos experimentos *ad-hoc* efectuados sobre los dispositivos seleccionados, se procedió a definir las pruebas a efectuar para cada categoría de procedimientos, todas ellas orientadas a evaluar la seguridad y privacidad de datos de los SPAs. En la sección III-B se proporciona el detalle de las pruebas definidas.

La realización de las pruebas requiere como paso previo la configuración y arranque del dispositivo a fin de establecer el estado de partida concreto. Esto es esencial también de cara a facilitar la eventual replicación de los experimentos. El procedimiento general elaborado para este fin se describe en la sección III-C.

III-B. Tipos de pruebas y aspectos evaluados

A continuación, se describen el tipo de pruebas definido para cada categoría de procedimientos y los aspectos de seguridad y privacidad que pretenden evaluar:

- **Instalación**. Estos procedimientos incluyen la instalación del dispositivo SPA propiamente dicho, así como el registro en el mismo de objetos conectados y *skills* de terceros. Las pruebas en esta categoría pretenden evaluar la seguridad y privacidad relativa a:
  - El intercambio de información entre el SPA durante el emparejamiento con el punto de acceso Wi-Fi que le proporciona conectividad a Internet, así como con la aplicación móvil que normalmente se requiere para la instalación del asistente.
  - Las opciones relativas a la configuración de permisos de uso, el soporte de múltiples usuarios y la personalización de perfiles de uso.
  - La posibilidad de establecer permisos y restricciones de uso de los dispositivos conectados y *skills* d terceros utilizados por el SPA.

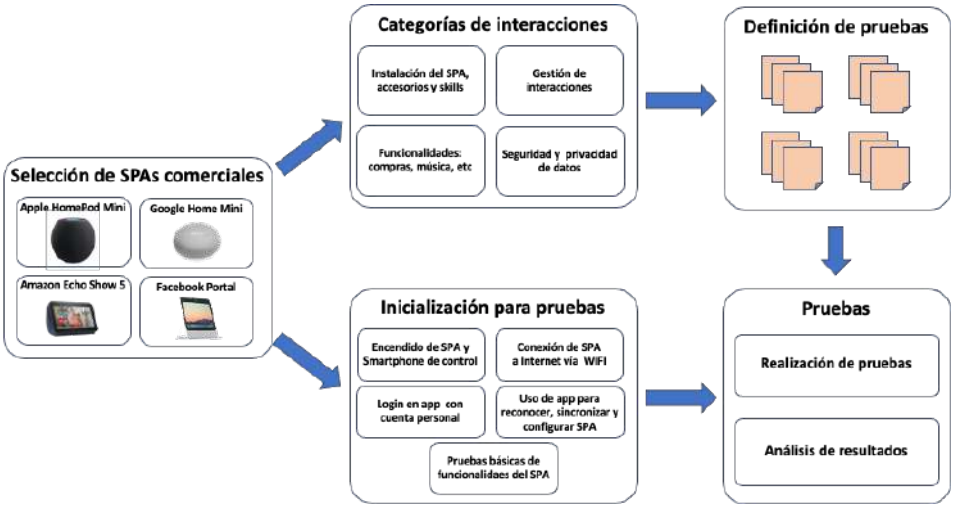


Figura 3. Metodología propuesta

- **Gestión de interacciones**. En esta categoría se incluyen los procedimientos que permiten al usuario gestionar las interacciones del SPA con los dispositivos conectados y los *skills* de terceros.
- **Gestión de funcionalidades**. Engloba el conjunto de procedimientos que permiten al usuario gestionar las funcionalidades del asistente, como, por ejemplo, pagos o reproducción de contenido multimedia. Es necesario verificar que estas acciones solo pueden efectuar por el usuario que dispone permiso para ellas, si se cumplen las restricciones que pueda haber definido. Igualmente, un aspecto de interés es analizar si es posible crear un perfil "seguro" para menores para, por ejemplo, controlar el contenido multimedia que puede reproducir, los accesorios con los que puede interactuar y los pagos que puede hacer.
- **Seguridad y gestión de datos**. Esta categoría abarca los procedimientos específicos que el SPA utiliza para garantizar la seguridad del dispositivo y la privacidad de los datos personales de sus usuarios. Dentro de los test definidos para esta categoría se contempla el estudio de:
  - Los métodos de autenticación usados y su robustez.
  - El control de la privacidad de las respuestas que contienen información personal.
  - La capacidad de evitar la activación del SPA mediante voz generada sintéticamente.
  - Las opciones controlar el acceso al historial de conversaciones con el asistente.

III-C. Inicialización del SPA para las pruebas

El procedimiento a seguir para establecer el estado de partida para las pruebas depende de cada modelo SPA. No obstante, es conveniente definir un procedimiento general que permita sistematizar el proceso de adquisición de evidencias durante la ejecución de las pruebas. De este modo, se posibilita la obtención de un conjunto uniforme de resultados, evitando la exclusión de datos y evidencias relevantes. En base a ello, el procedimiento general de inicialización de un SPA para la realización de las pruebas definidas en el apartado anterior es el siguiente:

- Encendido del SPA y del dispositivo auxiliar utilizado para su control (típicamente un smartphone).
- Conexión del SPA a Internet vía Wi-Fi.
- Registro/login a la aplicación de gestión del SPA, para lo cual es necesario que el usuario disponga de una cuenta personal (*Google*, *Amazon*, *iCloud*, *Facebook*, etc.)
- Uso de la aplicación para efectuar el procedimiento de reconocimiento, sincronización y configuración del SPA.
- Pruebas básicas de interacción con el SPA a fin de comprobar que el dispositivo, aparentemente, funciona.

III-D. Selección de dispositivos

Excluyendo los asistentes personales específicos para el mercado chino, los SPAs más utilizados son *Amazon Alexa*, *Google Assistant* y *Apple Siri*, con una cuota de mercado en 2019 estimada del 31.7 %, el 31.4 % y el 6 %, respectivamente [21]. Este último, no obstante, está instalado en número muy elevado de dispositivos por ser el que utilizan los smartphones y tablets de Apple [22]. En base a este análisis, para el desarrollo y validación de la metodología se ha seleccionado un conjunto representativo de SPAs basados en los tres asistentes personales mencionados. Los dispositivos concretos seleccionados son: *Apple HomePod Mini*, *Google Home Mini*, *Google Nest Audio*, *Amazon Echo Show 5*, *Amazon Echo Dot 4* y *Facebook Portal*. Sus características principales se resumen en la tabla I.

IV. Análisis de resultados

En este apartado se describen los detalles de las pruebas realizadas para cada uno de los dispositivos seleccionados, así como los resultados obtenidos.

IV-A. Instalación del SPA

El proceso de instalación del SPA es similar en todos los dispositivos. Para utilizar el SPA, es obligatorio disponer de una cuenta del proveedor (*Apple*, *Google*, *Amazon* o *Facebook*), que se puede crear de forma gratuita en todos los casos. Cabe destacar que todos los dispositivos probados excepto el *Apple HomePod Mini* pueden instalarse con dispositivos móviles con diferentes sistemas operativos. Este requiere un dispositivo móvil *Apple* para ser utilizado. A diferencia de los

Tabla I
COMPARACIÓN DE LAS CARACTERÍSTICAS DE LOS ASISTENTES PERSONALES INTELIGENTES SELECCIONADOS.

Características	*Apple HomePod Mini*	*Google Home Mini*	*Google Nest Audio*	*Amazon Echo Show 5*	*Amazon Echo Dot 4*	*Facebook Portal*
Modelo y fecha	1ªGen., 2020	1ªGen., 2017	-	1ªGen., 2019	4ªGen., 2020	2ªGen., 2019
Asistente Personal	*Siri*	*Google Assistant*		*Amazon Alexa*		
Aplicación	*Home* App	*Google Home App*		*Amazon Alexa App*		
Skills de terceros	No	Sí				
SO soportado	iOS	iOS and Android				
Palabra de activación	Se puede apagar, pero la palabra de activación no se puede cambiar	No se puede apagar y la palabra de activación no se puede cambiar		No se puede apagar y la palabra de activación se puede seleccionar de un conjunto de tres predefinidas		
Micrófono	No se puede desactivar	Se puede desactivar				
Cámara	No			Sí y puede ser desactivada	No	Sí y puede ser desactivada
Reconocimiento de voz	Sí					

anteriores, *Facebook Portal* no requiere de otro dispositivo para realizar la instalación, el propio dispositivo permite al usuario realizar este proceso a través de la pantalla táctil. De hecho, facilita este proceso de asociación de la cuenta de usuario con el portal mediante un código de paridad.

IV-B. Instalación de accesorios

El procedimiento para instalar nuevos dispositivos difiere considerablemente según el fabricante. El enfoque de *Apple HomePod Mini* es el único que combina usabilidad y seguridad, permitiendo diferenciar qué usuarios pueden instalar o editar dispositivos desde la aplicación. En el *Amazon Echo Show 5* y *Amazon Echo Dot 4* sólo el usuario administrador puede añadir o editar accesorios y usuarios. Sin embargo, en el *Google Home Mini* y *Google Nest Audio*, al añadir un nuevo usuario al hogar, éste se convierte en administrador, pudiendo instalar, editar y eliminar dispositivos conectados, sin posibilidad de definir permisos restrictivos a determinados usuarios. Del mismo modo, cualquier persona puede habilitar el uso de *skills* y *plugins* desde el portal de software de *Facebook Portal* sin necesidad de confirmación, e incluso reactivar complementos previamente desactivados por el usuario principal.

IV-C. Instalación de skills de terceros

La evaluación del proceso de instalación de *skills* de terceros en el SPA ha proporcionado resultados que han corroborado la importancia de los estudios realizados sobre la seguridad de esta funcionalidad [20]. A excepción del *Apple HomePod Mini*, que no tiene soporte para *skills* de terceros, el resto de los dispositivos analizados sí permiten su uso. Si bien el proceso de instalación de estas *skills* difiere en sus especificaciones, todas conducen al mismo resultado: un proceso de instalación de *skills* inseguro que no permite controlar quién puede añadir desarrollos de terceros al SPA. En el *Google Home Mini* y *Google Nest Audio* no existe un proceso de instalación de *skills*, conociendo la frase de activación de la *skill* se puede activar sin controles adicionales. En los *Amazon Echo Show 5* y *Amazon Echo Dot 4* existe un proceso de instalación a realizar desde la aplicación móvil, donde también es posible ver qué *skills* están activadas. Pero este proceso de control pierde su sentido si se interactúa con el SPA de manera directa, ya que de esta forma no se comprueba si una *skill* ha sido activado o desactivado por el administrador. Lo mismo ocurre con *Google Home Mini* y *Google Nest Audio*, basta con conocer la frase de activación para utilizar la *skill*. Aunque el administrador lo haya desactivado desde la aplicación móvil, se puede reactivar desde el SPA simplemente invocando de nuevo la *skill*. En el caso del *Facebook Portal*, se permite por restringido al portal de software del dispositivo.

IV-D. Interacción con el SPA y dispositivos conectados

A través de estas pruebas se ha evaluado el nivel de control de interacción con el SPA que se puede establecer. En el *Apple HomePod Mini* es posible habilitar y deshabilitar la interacción por voz o el control táctil, así como deshabilitar la reproducción de contenidos multimedia para determinados usuarios y el uso de ciertos dispositivos. En el *Google Home Mini* y *Google Nest Audio* hay un botón físico para desactivar los micrófonos del dispositivo. Se puede desactivar la reproducción de medios en el dispositivo, pero es necesario crear un grupo en una aplicación externa para incorporar a los usuarios que se desea controlar. Dicho control se configura en forma de filtros, que se aplican por grupos de usuarios, y no es posible establecer controles individuales. En el *Amazon Echo Show 5*, *Amazon Echo Dot 4* y *Facebook Portal* se proporcionan controles físicos para activar y desactivar la cámara, si la hay, y los micrófonos, pero al no existir un registro de usuarios, no es posible definir permisos de interacción para usuarios concretos.

IV-E. Interacción con skills de terceros

Como se ha comentado anteriormente, no es posible utilizar *skills* de terceros en el *HomePod Mini*. En el *Google Home Mini*, al no existir controles individuales de usuario, no es posible distinguir el uso de *skills* en función del usuario. La aplicación de filtros en el dispositivo permite desactivar el uso de *skills* de terceros para todos los usuarios o miembros registrados del hogar. Existen controles en el *Amazon Echo Show 5* y *Amazon Echo Dot 4* para activar y desactivar las *skills*, pero los controles no tienen efecto si se interactúa directamente con el SPA, ya que no se verifica qué usuario está intentando activar la *skill*. En el caso de *Facebook Portal*, el administrador puede permitir el uso de *skills* desde el portal de software de *Facebook Portal* antes de que se utilicen

en el dispositivo, pero cualquier usuario puede activar un complemento sin confirmación.

IV-F. Pagos y transacciones

La posibilidad de realizar pagos desde el SPA no está disponible en el *Apple HomePod Mini*. En el *Google Home Mini* y el *Google Nest Audio*, esta opción está desactivada por defecto, y se puede configurar desde la aplicación móvil. Solo los propios usuarios pueden habilitar los pagos con el SPA, ya que necesitan introducir un método de pago vinculado a su cuenta de *Google*, pero una vez configurado, pueden ser utilizados por cualquier usuario. Para protegerse del uso indebido, incluye la posibilidad de autenticación de dos factores en el dispositivo móvil al que se ha vinculado la cuenta, utilizando las capacidades del propio dispositivo móvil, como la huella dactilar o el reconocimiento facial. El *Amazon Echo Show 5* y el *Amazon Echo Dot 4* también ofrecen la posibilidad de realizar compras, para lo que es necesario activar el método de pago *Amazon 1 Click* desde la cuenta en la web de *Amazon*. Se dispone de mecanismos de autenticación, basados en un código de cuatro dígitos que será requerido por el SPA o un perfil de voz del usuario que realiza la compra. Si el perfil de voz coincide con el del usuario que ha configurado su modelo de voz en la aplicación, la compra será aceptada.

IV-G. Posibilidad de crear perfiles "seguros" para menores

En el momento de la redacción de este artículo y en España, no existe la posibilidad de crear perfiles para menores con uso restringido de forma sencilla y rápida en ninguno de los SPAs. Para restringir el uso de funcionalidades en el *Apple HomePod Mini* es necesario ajustar las opciones una a una, existiendo casos en los que es necesario deshabilitar una funcionalidad para todos los usuarios del dispositivo, por ejemplo, la reproducción de contenido multimedia explícito tiene que estar deshabilitada para todas las peticiones realizadas al *Apple HomePod Mini*, no pudiendo diferenciarse por usuarios. Desde la aplicación móvil es posible restringir el uso de los accesorios conectados para los miembros de la familia, para evitar que los menores interactúen con elementos como puertas o ventanas desde su dispositivo móvil, pero nada impide que actúen sobre los dispositivos con comandos de voz, para los que no hay control más allá de desactivar completamente la activación del SPA. En el *Google Home Mini* y el *Google Nest Audio* la opción de incluir a los menores en el grupo de usuarios del asistente solo está disponible en *Android*. En el resto del dispositivo es necesario establecer filtros, que, como se ha visto anteriormente, afectan a todos los usuarios del asistente por igual. Los filtros pueden utilizarse para restringir la reproducción de contenidos multimedia, las llamadas y el uso de *skills* de terceros. Sin embargo, los pagos siguen activos en el SPA a no ser que se desactiven para todos los usuarios. El *Amazon Echo Show 5* y el *Amazon Echo Dot 4* tampoco tienen la capacidad de establecer perfiles restringidos para los menores, por lo que las diferentes configuraciones tienen que ser ajustadas manualmente. Las restricciones se configuran desde el propio dispositivo, teniendo que introducir la contraseña de la cuenta de *Amazon* vinculada cada vez que se realicen cambios. Es posible restringir el uso del navegador web, la reproducción de contenidos multimedia o los pagos. En cuanto a los accesorios conectados, al igual que el resto de los dispositivos, no es posible restringir su uso.

IV-H. Control de respuestas con información personal

El tratamiento de las respuestas que contienen información personal puede dividirse en dos grupos, formados por el *Apple HomePod Mini*, *Google Home Mini* y *Google Nest Audio* en un caso y por el *Amazon Echo Show 5*, *Amazon Echo Dot 4* y *Facebook Portal* en el otro. En el *Apple HomePod Mini*, *Google Home Mini* y *Google Nest Audio* las respuestas personales están relacionadas con la configuración de reconocimiento de voz seleccionada. Si el usuario tiene el perfil de voz guardado y el reconocimiento de voz está activo, cuando al SPA se le pida información como eventos del calendario, mensajes o recordatorios, entre otros ejemplos, comprobará qué usuario inició la consulta para dar una respuesta acorde a los datos de esa persona. Si otro usuario realiza una pregunta sobre información personal, recibirá una respuesta con su propia información si su perfil de voz está almacenado, o se le negará el acceso a la información de la respuesta si se trata de un usuario desconocido para el SPA. A diferencia de estos dos dispositivos, no existe ningún método para controlar las respuestas que contienen información personal en el *Amazon Echo Show 5*, *Amazon Echo Dot 4* y *Facebook Portal*. La única opción para evitar que usuarios externos accedan a dicha información es no utilizar el servicio que puede generarla.

IV-I. Métodos de autenticación

La autenticación en los SPAs analizados se realiza mediante reconocimiento de voz. En el *Apple HomePod Mini*, la autenticación por reconocimiento de voz no está disponible en español, mientras que en el resto de los dispositivos analizados se ofrece esta posibilidad. En condiciones normales de uso, los dispositivos son capaces de distinguir entre diferentes voces y reconocer a los usuarios, denegando el acceso a la información personal entre usuarios de forma satisfactoria. En general, la autenticación no se utiliza como una medida de seguridad, sino como una funcionalidad auxiliar para personalizar la experiencia con el SPA. Esto se debe a la inseguridad de la función de reconocimiento de voz, tal y como se constata explícitamente en el análisis y como se recoge en la documentación de *Google* [17]. Como se comenta a continuación, todos los SPA son susceptibles de ser activados con mensajes grabados o procedentes de sistemas TTS. Este problema, combinado con un comportamiento de autenticación deficiente, que solo se activa durante la frase de activación y no durante todo el mensaje, lleva a una situación en la que cualquiera que tenga una grabación del usuario legítimo que pronuncia el mensaje de activación puede hacerse pasar por él.

El proceso de suplantación de un usuario se ha realizado en dos fases. En la primera, el usuario legítimo realiza una petición al SPA, solicitando cierta información personal. Con estas pruebas se puede comprobar que el sistema de reconocimiento de voz del SPA es capaz de distinguir a los usuarios con éxito. En la segunda fase, se simuló un ataque por parte de un posible actor malicioso en posesión de una grabación del usuario pronunciando el mensaje de activación del SPA. Dentro de esta fase, se pueden distinguir dos etapas. En la primera etapa, el actor malicioso debe obtener la grabación

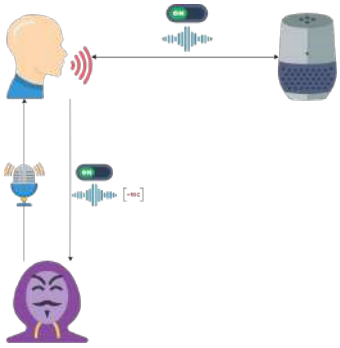


Figura 4. Grabación del mensaje de activación por un actor malicioso.

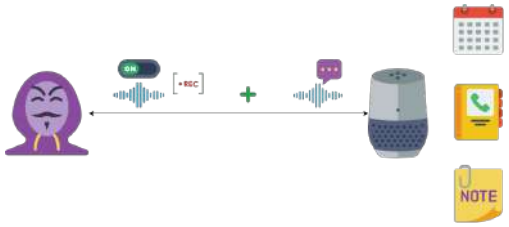


Figura 5. Ataque de suplantación de identidad.

de voz del usuario legítimo activando el SPA, tal y como se muestra en la Figura 4. En la segunda etapa, el atacante utiliza la grabación del usuario legítimo para activar el SPA y enviar solicitudes de acceso a la información personal del usuario, saltándose la autenticación de voz, como se muestra en la Figura 5.

La información personal a la que se puede acceder desde el SPA dependerá de la configuración de la cuenta del usuario y de la información que decida compartir con el proveedor.

El procedimiento de suplantación descrito ha sido probado en todos los SPAs, resultando satisfactorio en todos los dispositivos, pero con diferencias. Mientras que en el *Google Home Mini*, *Google Nest Audio*, *Amazon Echo Show 5*, *Amazon Echo Dot 4* y *Facebook Portal* el proceso de suplantación de un usuario fue totalmente exitoso, pudiendo suplantar a un usuario y acceder a su información personal, el *Apple HomePod Mini* cuenta con una medida de seguridad adicional que impide que un usuario que interactúe directamente con el *Siri* a través del *Apple HomePod Mini* acceda a información personal, teniendo que pasar por un proceso de verificación adicional basado en el desbloqueo del dispositivo móvil asociado al *Apple HomePod Mini*.

IV-J. Filtrado de voz no humana

Los SPAs analizados han demostrado ser susceptibles de ser activados por voces generadas artificialmente, como grabaciones y sistemas de SPAs. No implementan ningún tipo de medida de protección contra las voces sintéticas más allá de apagar los micrófonos de los dispositivos cuando no están siendo controlados físicamente. Esta constatación, que por sí misma puede no parecer un problema grave, combinada con un proceso de autenticación deficiente, como se ha detallado en el apartado anterior, lleva a la posibilidad de que un actor malicioso con una grabación del mensaje de activación del usuario pueda realizar cualquier consulta al SPA, saltándose los controles de reconocimiento de voz y pudiendo extraer información personal del dispositivo, interactuar con los accesorios conectados en el hogar, etc.

IV-K. Interacción con el historial de conversaciones

Las opciones para interactuar con el historial de conversaciones son diferentes en cada uno de los dispositivos. El *Apple HomePod Mini* es el que menos posibilidades ofrece, pudiendo únicamente enviar una solicitud de borrado de las grabaciones de conversaciones almacenadas en los servidores de *Apple*. El *Google Home Mini* ofrece una sección de configuración de privacidad completa para ver y eliminar las grabaciones de voz, así como para configurar un borrado automático de las grabaciones y pausar el almacenamiento de estas. Las opciones que *Amazon* incorpora en el *Amazon Echo Show 5* y *Amazon Echo Dot 4* son similares a las de *Google*, ofreciendo un completo conjunto de opciones para revisar las conversaciones con el SPA, eliminarlas y configurar un borrado automático de datos. En el caso de *Facebook Portal*, al utilizar *Amazon Alexa* como agente de voz, ocurre algo similar. De hecho, *Facebook Portal* graba clips de voz cuando los usuarios activan el asistente inteligente diciendo “Hey portal” y envía de vuelta a *Facebook* estos clips. Asimismo, estos clips de voz se graban y se envían a *Amazon*. Los datos extraídos de las conversaciones identificadas por *Facebook Portal* se utilizan para mostrar publicidad en *Facebook*. La empresa también puede compartir datos demográficos y de participación de la audiencia con anunciantes y socios de análisis.

V. Resumen y conclusiones

El SPA se ha consolidado como un dispositivo muy popular en los hogares y en nuestras vidas, motivo por el que también se ha convertido en un objetivo muy preciado para los ciberatacantes. La interacción por voz entre el usuario y el dispositivo da lugar a una nueva familia de potenciales vulnerabilidades y ciberataques. Entre estos destacan especialmente los relacionados con el control de acceso, mediante el que un atacante puede lograr interaccionar con el SPA para acceder a información sensible del usuario o incluso realizar compras u otras acciones en su nombre.

Mediante la revisión de la literatura reciente en este ámbito, se ha constatado la existencia de una gran variedad de vulnerabilidades en los SPAs comerciales. A fin de verificar la existencia de estos y otros problemas, se ha desarrollado una metodología para evaluar la seguridad y privacidad de esta familia de dispositivos. Para el desarrollo y validación de la metodología, se han seleccionado un conjunto significativo de SPAs comerciales de los principales fabricantes, los cuales han sido sometidos a un conjunto de pruebas exhaustivo.

En la metodología desarrollada se han considerado los distintos tipos de interacciones en los que pueden presentarse problemas de seguridad y privacidad, incluyendo los procedimientos de instalación y configuración del dispositivo, el acceso y control de objetos y aplicaciones de terceros (*skills*), y el uso propiamente del dispositivo para la invocación de tareas. También se ha hecho especial énfasis en los mecanismos de seguridad y privacidad implementados en los propios SPAs.

A partir de los resultados de las pruebas, se pueden extraer una serie de conclusiones y recomendaciones importantes.

Entre ellas, cabe destacar la debilidad de los mecanismos de autenticación utilizados por los asistentes, con medidas de protección deficientes frente a intentos de suplantación mediante la grabación de voz del usuario legítimo. Esto puede dar lugar a que un tercero pueda solicitar al SPA la realización de una tarea sin necesidad siquiera de fingir la voz del usuario legítimo. Las pruebas realizadas demuestran la posibilidad incluso de utilizar voz sintética para efectuar este tipo de ataques. A este respecto, se considera fundamental investigar y desarrollar nuevos sistemas de autenticación más robustos para los SPAs. En este sentido, algunas de las ideas que se han sugerido son el reconocimiento continuo del habla, y no solo durante el análisis de la frase de activación, como sucede en la actualidad. Así mismo, para dificultar la suplantación del usuario, parece lógico implementar un mecanismo capaz de detectar el uso grabaciones o voz sintética.

Los usuarios deben ser conscientes de que los SPAs actuales distan de ser todo los confiables que debieran, prestando atención a la hora de configurarlos para evitar el acceso indebido a sus datos personales. En este sentido, puede ser de ayuda el empleo de mecanismos de autenticación en dos pasos. Por otro lado, los fabricantes deberían garantizar que al configurar las opciones de seguridad del SPA se advierta al usuario de las implicaciones y potenciales peligros a los que puede quedar expuesto, así como cumplir con los requisitos de las regulaciones vigentes de protección de datos (por ejemplo, la RGPD en Unión Europea [23]).

Teniendo en cuanta que los SPAs son utilizados con frecuencia por menores, se considera fundamental la posibilidad de definir perfiles específicos para este colectivo de usuarios, a fin de restringir los contenidos y funcionalidades a las que tienen acceso y protegiéndoles, y minimizar su exposición frente a fraudes y acosos. Este tipo de facilidades no se ofrecen como tales en ninguno de los SPAs que se han estudiado (si bien en algunos casos se pueden establecer manualmente configuraciones que cumplan estos objetivos). Contrástese esta carencia con, por ejemplo, el soporte explícito de perfiles para menores en videoconsolas o en servicios de streaming de vídeo.

Agradecimientos

Este trabajo ha sido financiado por el programa Horizon 2020 de la Unión Europea a través del proyecto RAYUELA (nº de contrato 882828). El contenido del artículo refleja solo el punto de vista de sus autores. La Comisión Europea no es responsable del uso que se pueda hacer de la información que contiene.

Referencias

[1] S. Ruan, J. O. Wobbrock, K. Liou, A. Ng, and J. A. Landay, "Comparing speech and keyboard text entry for short messages in two languages on touchscreen phones," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, p. 1–23, January 2018.

[2] Amazon, "Alexa features." https://www.amazon.com/b?ie=UTF8&node=21576558011, 2021. Accessed: March 2021.

[3] Canalys, "Global smart speaker installed base to top 200 million by end of 2019." https://www.canalys.com/newsroom/canalys-global-smart-speaker-installed-base-to-top-200-million-by-end-of-2019, 2019. Accessed: March 2022.

[4] S. Liu, "Smart home in the united states - statistics & facts." https://www.statista.com/topics/6201/smart-home-in-the-united-states/, January 2021. Accessed: March 2021.

[5] N. Abdi, K. M. Ramokapane, and J. M. Such, "More than smart speakers: Security and privacy perceptions of smart home personal assistants," in *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, (Santa Clara, CA), pp. 451–466, USENIX Association, Aug. 2019.

[6] N. Abdi, X. Zhan, K. M. Ramokapane, and J. Such, "Privacy norms for smart home personal assistants," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, (New York, NY, USA), Association for Computing Machinery, 2021.

[7] S. Solera-Cotanilla et al., "Análisis de seguridad y privacidad en dispositivos de la internet de las cosas usados por jóvenes," in *Actas de las VI Jornadas Nacionales de Investigación en Ciberseguridad (JNIC)*, June 2021.

[8] Statista, "Total number of brands compatible with amazon alexa from january 2018 to july 2020." https://www.statista.com/statistics/912903/amazon-alexa-brand-use-growth/, January 2021. Accessed: March 2021.

[9] Statista, "Total number of smart home devices that are compatible with amazon's alexa as of july 2020." https://www.statista.com/statistics/912893/amazon-alexa-smart-home-compatible/, July 2020. Accessed: March 2021.

[10] Statista, "Number of smart home devices supported by google assistant worldwide from january 2018 to may 2019." https://www.statista.com/statistics/933532/worldwide-google-assistant-device-support/, May 2019. Accessed: March 2021.

[11] J. S. Edu, J. M. Such, and G. Suarez-Tangil, "Smart home personal assistants: A security and privacy review," *CoRR*, vol. abs/1903.05593, 2019.

[12] Alexa, "Alexa internet privacy notice." https://www.alexa.com/help/privacy, July 2020. Accessed: March 2021.

[13] Mozilla Foundation, "*privacy not included." https://foundation.mozilla.org/en/privacynotincluded/amazon-echo-dot/, February 2020. Accessed: March 2021.

[14] P. Cheng and U. Roedig, "Personal voice assistant security and privacy–a survey," *Proceedings of the IEEE*, pp. 1–32, 2022.

[15] Amazon, "Alexa accesibility - how to change your wake word." https://www.amazon.com/-/es/b?ie=UTF8&node=21341305011, 2021. Accessed: June 2021.

[16] X. Yuan, Y. Chen, A. Wang, K. Chen, S. Zhang, H. Huang, and I. M. Molloy, "All your alexa are belong to us: A remote voice control attack against echo," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2018.

[17] Google, "Link your voice to your devices with voice match." https://support.google.com/assistant/answer/9071681#vm_pr, March 2021. Accessed: March 2021.

[18] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7, 2018.

[19] J. B. Li, S. Qu, X. Li, J. Szurley, J. Z. Kolter, and F. Metze, "Adversarial music: Real world audio adversary against wake-word detection system," 2019.

[20] N. Zhang, X. Mi, X. Feng, X. Wang, Y. Tian, and F. Qian, "Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 1381–1396, 2019.

[21] Statista, "Smart speaker with intelligent personal assistant market share in 2018 and 2019, by platform." https://www.statista.com/statistics/1005558/worldwide-smart-speaker-market-share/, May 2019. Accessed: March 2021.

[22] Statista, "Digital voice assistant installed base worldwide in 2019, by brand." https://www.statista.com/statistics/967412/worldwide-digital-voice-assistant-installed-base-brand/, May 2020. Accesed: June 2021.

[23] E. Commission, "General data protection regulation (gdpr)." https://gdpr-info.eu/, 2018.

Conditional Generative Adversarial Network for keystroke presentation attack

Idoia Eizaguirre-Peral
Vicomtech, Basque Research and Technology Alliance (BRTA)
20009 Donostia/San Sebastian, Spain
ieizagirre@vicomtech.org

Lander Segurola-Gil
Vicomtech, Basque Research and Technology Alliance (BRTA)
20009 Donostia/San Sebastian, Spain
lsegurola@vicomtech.org

Francesco Zola
Vicomtech, Basque Research and Technology Alliance (BRTA)
20009 Donostia/San Sebastian, Spain
fzola@vicomtech.org

***Abstract*—Cybersecurity is a crucial step in data protection to ensure user security and personal data privacy. In this sense, many companies have started to control and restrict access to their data using authentication systems. However, these traditional authentication methods, are not enough for ensuring data protection, and for this reason, behavioral biometrics have gained importance. Despite their promising results and the wide range of applications, biometric systems have shown to be vulnerable to malicious attacks, such as Presentation Attacks. For this reason, in this work, we propose to study a new approach aiming to deploy a presentation attack towards a keystroke authentication system. Our idea is to use Conditional Generative Adversarial Networks (cGAN) for generating synthetic keystroke data that can be used for impersonating an authorized user. These synthetic data are generated following two different real use cases, one in which the order of the typed words is known (ordered dynamic) and the other in which this order is unknown (no-ordered dynamic). Finally, both keystroke dynamics (ordered and no-ordered) are validated using an external keystroke authentication system. Results indicate that the cGAN can effectively generate keystroke dynamics patterns that can be used for deceiving keystroke authentication systems.**

***Index Terms*—Keystroke dynamics, Users Behaviour, Conditional Generative Adversarial Networks**

Contribution type: *Research in development*

I. Introduction

In data protection, cybersecurity plays a key role in authenticating users and giving them access to personal and untransferable information. A failure in the user authentication system can lead to big economic, reputational, and social damage [1], and therefore, authentication systems are becoming more and more robust. In this scenario, biometry is playing an important role because it enables a universal, singular, permanent in time and measurable system of authenticating users [2].

In the last years, user authentication systems based on biometry are being used in very diverse scenarios, such as in airport scanners, banking, military access control, smartphones, and forensics among others [3], [4]. These systems, usually based on Machine Learning techniques, extract feature measurements and decide whether they correspond to the characteristics of the user that is requesting access. Biometrics is divided into two subfields: physical biometrics and behavioral biometrics. However, this study is focused on behavioral biometrics.

Despite presenting optimistic results in a wide range of applications, behavioral biometrics can be attacked in different ways [5]. The most common attack on a user authentication system is the Presentation Attack (PA), which consists of an attack on the biometric sensor that captures the individual's measurements [6]. Keystroke dynamics is a type of behavioral biometrics that refers to the way a user types on a keyboard taking into account the speed, the rhythm, the common mistakes and the times spent pressing and releasing each keycode. It is thought that each user has its typing ID and it can be very difficult to be imitated by another user or bot [7]

To date, little literature can be found on PA on keystroke dynamics data. Furthermore, to the best of out knowledge no literature can be found on keystroke behavior data generation using cGANs. In what keystroke dynamics studies concern, all major achievements have been made with fixed text, and, as far as we are aware, free-text dynamics have not been studied at a large scale yet. For this reason, in this project, we propose a new approach that uses free-text dynamics for learning and generalizing user keystroke behavior. In addition, we study how to use such information to implement a keystroke presentation attack on a biometric authenticator. In particular, we propose to use conditional Generative Adversarial Networks (cGAN) for learning keystroke dynamics and generate synthetic patterns that can deceive keystroke authentication systems. To the best of our knowledge, this is the first study in which keystroke dynamics behavior's synthetic data generation is done using cGANs.

The rest of the paper is organized as follows. In Section II, concepts regarding keystroke dynamics, generative models, and related work are introduced. In Section III, the proposed methodology is detailed and the data used, the metrics, the experiments, and the validation process carried out in this study are presented. In Section IV, results are reported and last, Section V, provides conclusions and guidelines for future work.

II. Preliminaries

In this section, some background concepts for understanding this study are explained. In section II-A, keystroke dynamics are presented, and in section II-B GANs and an extension of these are presented (Conditional Generative Adversarial Networks, cGAN), and their benefits and drawbacks are explained. Last, the dataset used in this study is presented in section II-C.

A. Keystroke dynamics

Keystroke dynamics are a type of behavioral biometric that measure the speed, the rhythm, the common errors, and

the typing times for pressing and releasing each keycode to identify patterns of typing IDs to identify or verify a user's identity. It has been proved that each person has their typing ID due to the habit of typing certain sequences always in a similar way, such as the user's name, password, etc [8]. The main advantages of using keystroke dynamics in user authentication systems are that there is no need for any special hardware because a keyboard in a device is the only requirement [9] and it enables continuous user authentication without disturbing the user experience while using such device [10].

In literature about keystroke dynamics, the robustness of keystroke dynamics against synthetic falsification attacks [11] and the robustness against replay attacks (an attack that collects and re-sends data to try to spoof the system) [12] have been studied and it has been shown that keystroke dynamics can be used to increase user authentication reliability using different modeling techniques such as SVM and NN [13].

Acien et al. [14] presented a Siamese Network capable of distinguishing whether two given keystroke sequences belonged to the same user or belonged to two different users. A Siamese Network consists of a type of NN composed of several subnetworks that are identical in architecture and weights but have got different inputs whose outputs are combined in a common loss function in charge of measuring the similarity between inputs. In this work, this model will be referenced as TypeNet.

B. Generative Adversarial Network

Generative Adversarial Networks (GAN) are a type of generative modeling from deep learning introduced by Ian Goodfellow [15]. They make use of two Neural Networks that compete with each other during their training: the generator (G) and the discriminator (D). The idea is that, through this adversarial training, both networks improve their performances. These networks play a zero-sum or min-max game, that is, the two networks play an adversarial game, one loses when the other wins and the other way round. This min-max game may not always reach an equilibrium, such a problem is known as *non-convergence* problem. Apart from the *non-convergence problem*, training a GAN may also result in *mode collapse* [16] and *vanishing gradient* which refer to a problem in the output of the generator and a problem with the value of the gradient during training respectively.

GANs are widely used to create or generate new samples of data that are as realistic as the samples in the original dataset. Since this new framework was proposed, GANs have been used for very diverse applications, and due to the difficulty when training, many extensions of GANs have been developed. They have been used for image creation [17] employing Deep Convolutional Generative Adversarial Networks (DCGAN), data augmentation using GANs [18], image-to-image translation utilizing Conditional Generative Adversarial Networks (cGAN) [19] and CycleGANs [20], melody generation from lyrics with Conditional LSTM-GANs [21]. What is more, in 2016, Zhang et al. proposed the StackGAN, another extension of GANs that transforms text descriptions into realistic images. In 2017, Karras et al. proposed a new methodology for GANs to improve training stability and speed it up through a Progressive Growing of GANs [22]. In the last years, new GAN extensions have been also used to generate synthetic sequential data. In 2017, Esteban et al. proposed the Recurrent GAN and the Recurrent Conditional GAN to generate synthetic multi-dimensional time series [23]. Among these GAN structures, in this study, we implement the Conditional GAN (or cGAN).

Conditional Generative Adversarial Networks (cGAN) are an extension of the GANs presented by Mehdi Mirza in 2014, [24]. The main difference is that cGANs can generate data from a previously specified category or class. Both the generator and the discriminator in cGANs have got an extra input, the extra information that will condition the generated sample. This condition can be a class label, a numerical value, or an embedding vector. Therefore, the generator will try to generate samples from the corresponding data distribution of that class and the discriminator will learn to decide whether the given pair (sample and condition) is a matching real sample or has been artificially generated by the generator.

The cGAN designed, implemented, and validated in this study aims to generate synthetic data on keystroke dynamics starting with two inputs: a latent vector and a character embedding. An embedding is a numerical representation of some high dimensional vector or text data into a low dimensional vector. The use of embeddings in Machine Learning (ML) models is widespread due to their contribution to making model training easier and more efficient. In this study, a pre-trained character embedding known as *char2vec* has been used.

C. Dataset

This study has been carried out using a dataset provided by the Aalto University in Finland [25], in which they collected data from $168,000$ volunteers in an online study. This dataset is a large-scale dataset that contains information on the keystroke patterns of each of the volunteers. It contains keystroke times of 15 sentences for each volunteer. For each keystroke pressed to type the sentence, the pressing and releasing times were registered for more than 136 million of keystrokes.

However, in this study, as the aim is to impersonate a single user and validate it with an external model, a subset of the whole dataset has been used. On the one hand, to train the external model, TypeNet, a dataset composed of data about 25 volunteers have been used. In this study, it will be referenced as TypeNet dataset. On the other hand, to train the cGAN, a dataset of a single user has been used, which will be referenced as single-user dataset. It is important to outline that despite having the records of the pressing and releasing timestamps of the keycodes in the dataset, in line with the literature in the area, four variables were extracted to build the ML models: Hold Latency (HL), Press Latency (PL), Release Latency (RL) and Inter-Key Latency (IL).

III. Methodology

In this section, first, a global view of the methodology is explained, next, in section III-B, the architecture of the cGAN is detailed. In section III-C, some important aspects of the experimental framework are presented. In section III-D, the metrics used to validate the results are presented, and finally, in section III-E, the validation procedure is described.

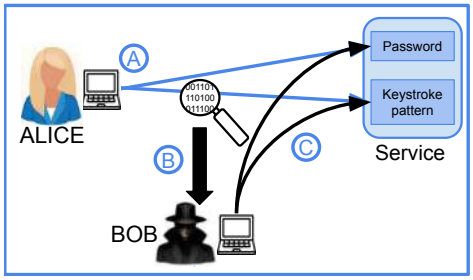


Figure 1. Steps of our Presentation Attack.

A. Proposal

This study aims to generate synthetic data on keystroke dynamics to impersonate a user and develop a presentation attack. To that end, we propose a cGAN able to learn the keystroke dynamics of a specific user and generate realistic data samples. These generated data samples will be able to fool an external model, TypeNet, used as a biometric authenticator to access a determined service.

Let us assume that Alice is a registered user of concrete service. To access this service, a double verification is carried out. the user that wants to access the service, not only has to type its password, but the system also checks the keystroke dynamics of the user typing it and verifies the user's typing ID to enable access or deny, see step A in Fig. 1. However, a non-registered user, Bob, wants to access the service with Alice's credentials. To do so, let us assume Bob already knows Alice's password, see step B in Fig. 1. To access the service, he needs to imitate Alice's typing behavior. In other words, he needs to impersonate her by generating synthetic behavioral biometric data to fool the biometric authenticator, see step C in Fig. 1.

In this context, we propose a new methodology to deploy a keystroke PA using cGAN technology. Our methodology is composed of three phases. In phase 1, the aim is to train the biometric authenticator following the process indicated in [14], which will be in charge of deciding whether two given 15-character input sequences belong to a single user or belong to two different users. In phase 2, making use of the single-user dataset and considering training data by words, the cGAN has been trained to be capable of generating realistic behavioral data on Alice's keystroke dynamics. Last, in phase 3, the PA is carried out, hence, the synthetic data is reconstructed into synthetic sequences of several words concatenated together and are introduced in the initial biometric authenticator to impersonate Alice. This last phase has been developed in two different real use cases, one in which the order of the words in the reconstructed sequence was identical to the order in the original dataset, and the other in which words were concatenated randomly considering a space between two words.

B. The model

The cGAN we propose is called Vanilla-cGAN because the internal architecture of the generator and the discriminator in the model are NN. The generator generates synthetic data based on a latent vector with 500 random values that follow a Gaussian distribution and a 100 dimensional character embedding that encodes the word whose typing times are generated. The discriminator distinguishes whether a given data sample is real or synthetic based on the typing times and the keycode values (15×5) for a given word and its corresponding 100 dimensional character embedding.

C. Experimental framework

As it has been mentioned previously, data on keystroke dynamics is usually treated using 5 variables that describe the typing behaviour of the users: *HL*, *PL*, *RL*, *IL* and the *KEYCODE* in ASCII code. In this study, both models, TypeNet and the cGAN, use these variables as training data. The difference lies in the sequences that each model receives in the training, and consequently, in the validation process. Note that the biometric authenticator considers sequences of 15 characters whereas the cGAN considers the training by words stored in sequences with the same length but with extra padding when the word has less than 15 characters.

The training process is carried out until reaching concrete stopping criteria related to the precision of the GAN discriminator. In particular, every 50 epoch, the training has been paused and 5 subsets of 32 real and 32 synthetic samples have been used as input for the discriminator to measure its performance in classifying the given samples. If the accuracy of the discriminator has been above 85% for both real and synthetic samples, the training process is stopped, otherwise, it is resumed until the next validation (on the next 50 epochs).

Once the training of the cGAN is completed, in phase 3, to perform a presentation attack, two use cases or conditions have been considered:

- Condition 1: in which the order of the words in the reconstructed sequence corresponds to the same order in the single-user dataset;
- Conditions 2: in which the order of the words in the reconstructed sequence is done randomly considering a space between every other word in the sequence.

D. Metrics

Let us consider the confusion matrix for a binary classification composed of the following values: True Negatives (TN), False Negatives, (FN) False Positives (FP), and True Positives (TP).

In this study the evaluation metric used to determine the quality of the obtained results is the following:

- Accuracy:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

E. Validation

As described in phase 3 of our methodology in section III, the validation has been done using the biometric authenticator, TypeNet, employing three tests:

- Test 1: Real vs. Fake (*Acc_rf*): 20 real sequences and 20 generated sequences from the same user are validated to try to fool the external model.
- Test 2: Fake vs. Fake (*Acc_ff*): 20 generated sequences used in test 1 and another 20 generated sequences from the same user are validated with the external model. Test 3: Real other vs. Fake (*Acc_rof*). 20 generated sequences used in test 1 and 20 real sequences that belong to other users (the other 24 users for which TypeNet has been trained) are validated with the external model.

IV. Results

Tab I show the accuracy values of the three tests in both use cases. It can be observed that in general, the values are high for all tests. Note that as the values in tests 1 and 2 are similar, it can be concluded that the synthetic samples are realistic, and the biometric authentication is not able to detect differences at all. With the results in test 3, it can be concluded that the generated samples for Alice are not similar to any other user in the TypeNet dataset. In what differences between conditions concern, it can be concluded that the order of the words does not affect at all the typing behavior of the users.

	Test 1: Real vs. Fake	**Test 2: Fake vs. Fake**	**Test 3: Real other vs. Fake**
Condition 1	0.945	0.945	0.975
Condition 2	0.950	0.960	0.955

Table I
Accuracy of the classification of TypeNet for the generated data.

V. Conclusions

The main goal of this work has been to generate synthetic data on keystroke dynamics in an adversarial way and to deploy a Presentation Attack. To validate the results, the performance of the discriminator has been measured and an external model has been used. Results indicate that keystroke dynamics patterns can be adversarially generated and promising results can be obtained in the two different real scenarios. This project shows the relevance of typing behavior data generation using adversarial networks with several approaches. It has been proved that valid data to impersonate users can be generated using generative modeling from ML.

Further research is needed to determine whether there is any other GAN extension that would reduce training time and improve results' performance.

References

[1] G. W. Clark, M. V. Doran, and T. R. Andel, "Cybersecurity issues in robotics," in *2017 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, 2017, pp. 1–5.

[2] V. Matyáš and Z. Říha, *Biometric Authentication — Security and Usability*. Boston, MA: Springer US, 2002, pp. 227–239. [Online]. Available: https://doi.org/10.1007/978-0-387-35612-9_17

[3] A. Muley and V. Kute, "Prospective solution to bank card system using fingerprint," in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, 2018, pp. 898–902.

[4] A. Bud, "Facing the future: The impact of apple faceid," *Biometric technology today*, vol. 2018, no. 1, pp. 5–7, 2018.

[5] S. Marcel, M. S. Nixon, and S. Z. Li, *Handbook of biometric anti-spoofing*. Springer, 2014, vol. 1.

[6] J. Ness, "Presentation attack and detection in keystroke dynamics," Master thesis.

[7] Z. Jin, A. B. J. Teoh, T. S. Ong, and C. Tee, "Typing dynamics biometric authentication through fuzzy logic," in *2008 International Symposium on Information Technology*, vol. 3, 2008, pp. 1–6.

[8] F. Bergadano, D. Gunetti, and C. Picardi, "User authentication through keystroke dynamics," *ACM Transactions on Information and System Security (TISSEC)*, vol. 5, no. 4, pp. 367–397, 2002.

[9] B. Hassan, K. Fouad, and M. Hassan, *Keystroke Dynamics Authentication in Cloud Computing*, 01 2019, pp. 923–945.

[10] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation computer systems*, vol. 16, no. 4, pp. 351–359, 2000.

[11] D. Stefan, X. Shu, and D. D. Yao, "Robustness of keystroke-dynamics based biometrics against synthetic forgeries," *computers & security*, vol. 31, no. 1, pp. 109–121, 2012.

[12] I. Hazan, O. Margalit, and L. Rokach, "Securing keystroke dynamics from replay attacks," *Applied Soft Computing*, vol. 85, p. 105798, 2019.

[13] E. Yu and S. Cho, "Keystroke dynamics identity verification—its problems and practical solutions," *Computers & Security*, vol. 23, no. 5, pp. 428–440, 2004.

[14] A. Acien, A. Morales, R. Vera-Rodriguez, J. Fierrez, and J. V. Monaco, "Typenet: Scaling up keystroke biometrics," in *2020 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2020, pp. 1–7. [Online]. Available: https://arxiv.org/pdf/2004.03627.pdf

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," 2014, vol. 27.

[16] D. Saxena and J. Cao, "Generative adversarial networks (gans): Challenges, solutions, and future directions," *ACM Comput. Surv.*, vol. 54, no. 3, 2021. [Online]. Available: https://doi.org/10.1145/3446374

[17] S. C. Alec Radford, Luke Metz, "Unsupervised representation learning with deep convolutional generative adversarial networks." [Online]. Available: https://arxiv.org/pdf/1511.06434.pdf

[18] F. Zola, J. L. Bruse, X. Etxeberria Barrio, M. Galar, and R. Orduna Urrutia, "Generative adversarial networks for bitcoin data augmentation." [Online]. Available: https://arxiv.org/pdf/2005.13369.pdf

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/papers/Isola_Image-To-Image_Translation_With_CVPR_2017_paper.pdf

[20] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2017/papers/Zhu_Unpaired_Image-To-Image_Translation_ICCV_2017_paper.pdf

[21] S. C. Yi Yu, Abhishek Srivastava, "Conditional lstm-gan for melody generation from lyrics." [Online]. Available: https://arxiv.org/pdf/1908.05551.pdf

[22] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *CoRR*, vol. abs/1710.10196, 2017. [Online]. Available: http://arxiv.org/abs/1710.10196

[23] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans," 2017.

[24] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[25] V. Dhakal, A. M. Feit, P. O. Kristensson, and A. Oulasvirta, *Observations on Typing from 136 Million Keystrokes*. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–12. [Online]. Available: https://doi.org/10.1145/3173574.3174220

Sesión VII: Criptografía y herramientas matemáticas

Análisis estadístico seguro para ensayos clínicos

Alicia Quirós
Universidad de León
Departamento de Matemáticas
Campus de Vegazana, León, España
alicia.quiros@unileon.es

Diego Simón
Universidad de León
Departamento de Matemáticas
Campus de Vegazana, León, España
dsimog01@estudiantes.unileon.es

Adriana Suárez Corona
Universidad de León
Departamento de Matemáticas
Campus de Vegazana, León, España
asuac@unileon.es

***Resumen*—En este trabajo se plantea la comparación de proporción de eventos adversos para dos grupos de tratamiento, con un enfoque bayesiano, en el contexto de un ensayo clínico en el que participan varios hospitales. Con el objetivo de preservar la confidencialidad de los datos, utilizamos un esquema de cifrado homomórfico, que permite realizar cálculos sobre los datos cifrados. La viabilidad de este diseño se ilustra con el análisis de datos simulados a través de la implementación de un prototipo de sistema cliente-servidor, desarrollado en el lenguaje C++ y capaz de realizar operaciones homomórficas mediante la biblioteca Microsoft SEAL. La incorporación de la inferencia bayesiana presenta ventajas como el procesado de grandes conjuntos de datos, la reducción del tiempo global y la posibilidad de incorporar un diseño adaptativo en ensayos clínicos con compartición de datos segura. Se ha demostrado la viabilidad, mostrándose los tiempos de ejecución.**

***Index Terms*—Cifrado homomórfico, Cuaderno de recogida de datos electrónico, Inferencia bayesiana.**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

El tratamiento de datos sensibles hace imprescindible contar con mecanismos que garanticen su privacidad, de forma que sólo los usuarios autorizados puedan acceder a los mismos. Las herramientas criptográficas son esenciales en estos casos, pudiendo garantizar, en base a un modelo matemático, que se satisfacen las nociones de seguridad requeridas en cada situación [1]. Esto puede ser crucial cuando los datos obtenidos para poder realizar un diagnóstico médico deben compartirse con otros especialistas que no se encuentren físicamente cerca, cuando el promotor de un estudio requiere que varios investigadores pongan en común sus datos sin desvelar más información de la necesaria para hacer un análisis conjunto o cuando se desea compartir el historial médico de un paciente al ser transferido de un centro a otro.

Cuando el volumen de los datos se hace inmanejable, puede ser necesario el uso de arquitecturas de *cloud computing* que permitan externalizar cálculos sobre estos datos u ofrecer disponibilidad, de forma que cualquier cliente solamente tendrá que conectarse a él para efectuar alguna operación. En estos casos, es conveniente que los datos se cifren antes de enviarlos al servicio *cloud*, de forma que ese tercero no tenga acceso a los datos en claro, estos estén protegidos y se cumpla así con las normativas relativas a la protección de datos.

Para el almacenamiento y la compartición segura de los datos, es suficiente con utilizar esquemas de cifrado clásicos. Sin embargo, si es necesario realizar cálculos sobre los mismos, la herramienta más conveniente es un esquema de cifrado homomórfico, que permite realizar el cálculo sobre los datos cifrados, manteniendo la confidencialidad frente al servidor externo [2].

El contexto de aplicación de este trabajo son los ensayos clínicos en los que participan varios hospitales. De acuerdo con el Real Decreto 1090/2015, cuando se trate de investigación clínica sin ánimo comercial "la propiedad de los datos de la investigación pertenece al promotor" y tiene que acordar con el investigador el tratamiento de datos. El investigador, a su vez, "es el responsable de garantizar la veracidad de los datos" y de "garantizar la confidencialidad acerca de los sujetos del ensayo y la protección de datos de carácter personal".

En concreto, el tipo de ensayos clínicos que conforman el contexto de este trabajo tienen como objetivo principal comparar la proporción de eventos adversos entre dos grupos de tratamiento.

Para abordar el análisis del objetivo principal utilizaremos la inferencia bayesiana con el modelo beta-binomial [3]. La inferencia bayesiana resuelve el problema de comparación de proporciones incluso cuando no se observa ningún caso para alguno de los posibles resultados en algún grupo de tratamiento, gracias a que puede incorporar información experta o asumir ignorancia a priori sobre las proporciones. En este último caso, el resultado obtenido es equivalente al proporcionado por la inferencia frecuentista. La inferencia bayesiana, además, proporciona resultados en términos de probabilidad.

I-A. Contribuciones

Nuestra principal contribución es el diseño y la implementación de un caso de estudio que permite calcular las distribuciones a posteriori de las proporciones de cada grupo sobre datos cifrados con cifrado homomórfico.

En nuestro escenario, varios hospitales participan en un estudio clínico en el que el promotor dispone de un servicio basado en la nube como cuaderno de recogida de datos electrónico (eCRD). Gracias al cifrado homomórfico de los datos, los hospitales pueden compartir sus datos y el promotor puede externalizar el almacenamiento y el análisis de los datos a este servicio basado en la nube sin revelar, en ningún momento, el contenido de dichos datos.

II. Preliminares

II-A. Comparación de proporciones bayesiana

En el marco de la inferencia bayesiana, uno de los modelos más utilizados para abordar el problema de comparación de dos proporciones es el beta-binomial. Sean p_1 y p_2 las proporciones de pacientes que sufren un evento adverso para cada grupo de tratamiento.

A priori, asumimos que

$$p_i \sim \mathcal{B}(a_i, b_i) \text{ para } i = 1, 2,$$

es decir, que la distribución de ambas proporciones es beta de parámetros $\alpha = a_i > 0$ y $\beta = b_i > 0$, con a_i, b_i reales. En caso de haberlas, estos parámetros a_i y b_i pueden escogerse de forma que las distribuciones a priori correspondientes reflejen nuestras creencias sobre las proporciones, p_i, antes de observar los datos, D. Para asumir ignorancia a priori sobre las proporciones debemos escoger $a_i = b_i = 1$ para $i = 1, 2$, puesto que la distribución $\mathcal{B}(1, 1)$ es equivalente a una uniforme en el intervalo $(0, 1)$.

Si definimos las variables aleatorias, X_1 y X_2, como el número de pacientes que experimentan un evento adverso grave en cada grupo de tratamiento, podemos afirmar que

$$X_i \sim \text{Binomial}(n_i, p_i), \quad i = 1, 2,$$

donde n_i es el número de pacientes incluidos en el grupo experimental i, con $i = 1, 2$.

La teoría de inferencia bayesiana afirma que la distribución a posteriori de las proporciones p_1 y p_2 es

$$p_i|D \sim \mathcal{B}(a_i + x_i, b_i + n_i - x_i), \text{ con } i = 1, 2,$$

donde x_i es el número observado de pacientes que han experimentado un evento en el grupo i.

Este análisis puede hacerse de forma secuencial de forma que, al observar nuevos datos, $D^\star$, podemos tomar $p_i|D$ como distribución a priori y actualizar los parámetros de la distribución con $D^\star$, para obtener una nueva distribución a posteriori, $p_i|D, D^\star$.

Una vez que conocemos la distribución a posteriori, podemos proporcionar, para cada proporción, la estimación puntual; un intervalo de credibilidad al 95 % (a, b) tal que $P(a < p_i < b) = 0.95$; o la gráfica de la distribución a posteriori; además de la $P(p_2 - p_1 > 0|D)$, como evaluación del objetivo principal.

II-B. Cifrado homomórfico

Los esquemas de cifrado homomórfico permiten realizar operaciones sobre textos cifrados, de forma que al descifrar el resultado obtenido es el mismo que se obtendría al aplicar la operación a los datos sin cifrar. Es decir,

$$\text{Enc}(m_1) * \text{Enc}(m_2) = \text{Enc}(m_1 * m_2),$$

donde $*$ es la operación para la que el cifrado tiene la propiedad homomórfica.

Pueden distinguirse esquemas de cifrado *parcialmente homomórfico*, en los que se cumple la propiedad homomórfica para una sola operación, como el cifrado de Pallier [4] o RSA [5]; los esquemas *en cierto modo homomórficos*, que permiten realizar distintas operaciones sobre los datos cifrados, pero un número limitado de veces, como los propuestos por Sander, Young y Yung [6] o el propuesto por Boneh, Eu-Jin y Kobbi [7]; y los esquemas *completamente homomórficos* que permiten realizar sumas y multiplicaciones un número ilimitado de veces, como el esquema de Gentry [8]. Tras esta propuesta, se han presentado distintos esquemas de cifrado completamente homomórfico, y existen distintas bibliotecas escritas en C++, como HELib, que implementa el esquema BGV, de Brakerski, Gentry y Vaikuntanathan [9] o Microsoft SEAL [10], que implementa, tanto el esquema CKKS, propuesto por Cheon et al. [11], como el BFV (Brakerski/Fan-Vercauteren) [21].

En este caso usaremos el esquema CKKS puesto que permite trabajar con números tipo *float*. Como se explicó en la sección II-A, esto permite utilizar distribuciones a priori informativas, i.e. definidas por expertos.

III. Trabajos relacionados

Existe una biblioteca de R relacionada con el análisis estadístico seguro [12], aunque se ciñe a los tests frecuentistas y no paramétricos más utilizados y utiliza esquemas de computación multiparte, como muchos de los trabajos que cita. Otras dos bibliotecas "hermanas" de R implementan un sistema de cifrado homomórfico [13] y la adaptación de dos herramientas de *machine learning* [14]. En esta línea, se han realizado diversos estudios sobre clasificadores que preservan la privacidad [15], en particular el clasificador Naïve Bayes [16], [17], [18]. Estos estudios suelen utilizar esquemas de cifrado homomórfico o esquemas de computación multiparte que permiten el procesamiento confidencial de los datos. A diferencia de todas estas propuestas, en nuestro trabajo, el objetivo no es la clasificación, sino la inferencia bayesiana sobre datos cifrados con un esquema homomórfico.

IV. Caso de uso y resultados experimentales

Ilustraremos la aplicación de este trabajo con un diseño que asume que un promotor lleva a cabo un ensayo clínico con tres hospitales participantes. Estos comparten con el promotor del estudio sus datos de forma confidencial enviándolos al servicio *cloud*, de forma que se puedan realizar los cálculos necesarios sobre ellos sin poner en peligro su confidencialidad.

Utilizaremos un esquema de cifrado completamente homomórfico de forma que el servicio *cloud* realice los cálculos de los parámetros de las distribuciones a posteriori de las proporciones de cada grupo de tratamiento, que denominaremos:

$$\alpha_i = 1 + x_i \quad \beta_i = 1 + n_i - x_i,$$

para $i = 1, 2$, de una forma secuencial, a medida que vaya recibiendo los datos.

IV-A. Datos

El conjunto de datos usado en este trabajo es una simulación de datos basada en los resultados publicados del estudio *Synergy between PCI with Taxus and cardiac surgery* (SYNTAX) [19]. El objetivo del estudio fue comparar dos estrategias de tratamiento –la cirugía de revascularización miocárdica (CABG) y la angioplastia coronaria (PCI)– en pacientes con lesiones de 3 vasos, lesiones del tronco o de ambos. La comparación se realizó en términos de eventos adversos cardíacos y cerebrovasculares graves (MACCE) observados durante el primer año después de la intervención.

El conjunto de datos simulados contiene 1740 casos, 849 en el grupo CABG y 891 en PCI. La tabla I resume los datos simulados.

Aunque en el estudio SYNTAX participaron 85 hospitales, en los datos simulados hemos repartido aleatoriamente los datos en 3 subconjuntos.

Tabla I
NÚMERO Y PORCENTAJE DE CASOS CON ALGÚN EVENTO ADVERSO GRAVE (MACCE) EN LOS DATOS SIMULADOS, PARA CADA GRUPO DE TRATAMIENTO Y EN TODOS LOS DATOS.

	CABG	PCI	Total
	$n = 849$	$n = 891$	$n = 1740$
MACCE	105 (12.4 %)	159 (17.8 %)	264 (15.2 %)

IV-B. Prototipo

Se ha creado un prototipo siguiendo el modelo cliente-servidor. En el servicio *cloud* se ejecuta una aplicación servidora, que es la encargada de realizar las operaciones matemáticas con los datos previamente cifrados mediante un esquema homomórfico. Tanto cada uno de los hospitales como el promotor ejecutan en sus dispositivos la aplicación cliente, que muestra diferentes menús dependiendo de qué tipo de usuario inicie sesión (investigador o promotor). El servidor tendrá que validar las credenciales de cada uno de ellos en este paso, por medio de comprobación de usuario y contraseña.

El promotor será el único poseedor de la clave privada (*SK*) capaz de descifrar los datos enviados al servidor. Este tendrá la responsabilidad de enviar al servidor la correspondiente clave pública (*PK*) que se utilizará para cifrar los datos que forman parte del estudio. Además, podrá solicitar al servidor los resultados obtenidos a partir de los datos cifrado enviados por los hospitales que hayan participado en el estudio. Aparte, tendrá la posibilidad de manejar las altas y bajas de investigadores que forman parte del estudio.

El investigador responsable autorizado de cada uno de los hospitales participantes tendrá que indicar a la aplicación la ruta al fichero CSV en el que está almacenada la información a incluir. Dichos datos se cifran y se envían al servidor para que éste actualice los parámetros de las distribuciones beta.

El servidor dispone de una base de datos MySQL en la que almacena tanto la clave pública del promotor como los valores de los parámetros de las distribuciones beta calculados tras las diferentes interacciones con los hospitales participantes. Mediante el uso de la biblioteca CROW, se encarga de procesar y responder a las peticiones del promotor y los investigadores, realizadas a través del protocolo HTTP.

El prototipo se ha desarrollado en C++, empleando como base la biblioteca para cifrado totalmente homomórfico Microsoft SEAL, con el esquema CKKS.

La figura 1 muestra la arquitectura del sistema. Por un lado, se representa la aplicación cliente y, por otro, la aplicación servidora que se ejecuta en el servicio *cloud*.

Para realizar los cálculos de los parámetros de las distribuciones a posteriori de p_i, se realizan los siguientes pasos:

1. El promotor genera un par de claves (*PK*, *SK*) y envía *PK* al servidor para que la almacene en su base de datos.
2. El investigador responsable de un hospital selecciona el fichero CSV en el que se incluyen los datos a enviar. Estos se extraen como dos vectores de enteros en claro que después se cifran (dato a dato) con la *PK* del promotor, que le facilita el servidor. Los vectores cifrados se envían al servidor en formato *JSON* a través de un POST HTTP.
3. El servidor recibe el vector de datos cifrados y realiza los cálculos necesarios para actualizar los parámetros de las distribuciones beta que tiene almacenadas en su base de datos (en caso de ser el primer envío de datos, se parte de la distribución a priori).
4. Se repiten los pasos 2 y 3 tantas veces como lotes de datos en los que divida cada hospital su conjunto de datos.
5. El promotor solicita al servidor los valorescifrado de α_i y β_i de las distribuciones beta a posteriori asociadas a cada tipo de tratamiento (CABG y PCI) y los descifra.

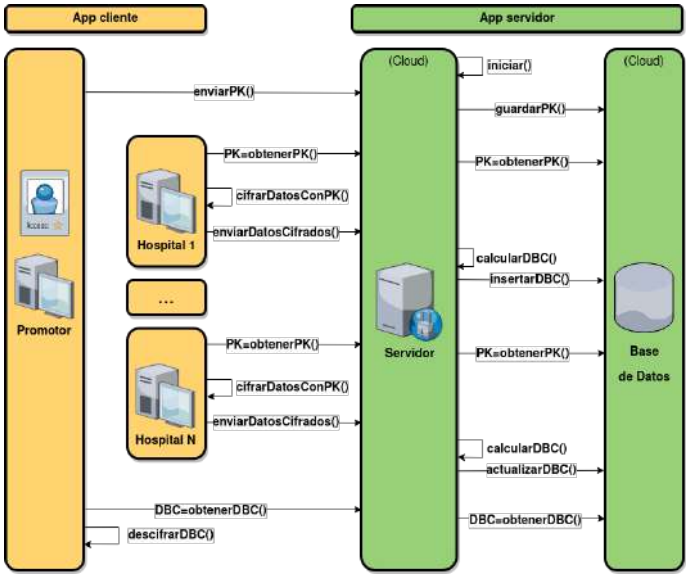


Figura 1. Diagrama de la arquitectura del sistema. En amarillo se representa la aplicación cliente y, en verde, la aplicación servidora que se ejecuta en el servicio *cloud*. Las flechas describen las interacciones entre los actores. Las siglas DBC se refieren a los parámetros de las distribuciones beta cifradas.

IV-C. Resultados

Todas las pruebas de rendimiento del prototipo se han llevado a cabo en un dispositivo con Ubuntu Desktop 21.04 como sistema operativo, equipado con 16GB de memoria RAM y CPU i7-8750H. Para la evaluación experimental del desempeño del prototipo hemos dividido el conjunto de datos en partes de diferente tamaño (100, 500, 1000 y todos los datos) con el objetivo de cuantificar el tiempo de ejecución. En concreto, se han medido los siguientes tiempos (en s):

- cifrado en la aplicación cliente (Enc),
- la conversión de datos cifrados a JSON en la aplicación cliente (Enc $\rightarrow$ JSON),
- la conversión de JSON a datos cifrados en la aplicación servidora (JSON $\rightarrow$ Enc),
- cálculos para actualizar α_i y β_i sobre datos cifrados (Comp).

Adicionalmente, se han calculado estos tiempos para cada uno de los tres hospitales. Todas las simulaciones se han repetido tres veces. El cálculo del tiempo total (para todos los datos) se ha calculado sumando el tiempo de los tres hospitales, puesto que el requerimiento de memoria del cifrado no permite el procesamiento de todos los datos. La tabla II muestra las medias y desviaciones típicas de los tiempos de ejecución.

Se ha comprobado que no hay error al realizar las operaciones sobre cifrados, en comparación con los mismos cálculos sobre datos en claro. La figura 2 muestra las distribuciones a

Tabla II
TIEMPOS DE EJECUCIÓN

	App Cliente		App Servidor	
n	Enc	Enc → JSON	JSON → Enc	Comp
100	3.2 ± 0.1	4.4 ± 0.1	5.0 ± 0.1	3.4 ± 0.0
500	18.3 ± 2.4	23.9 ± 3.5	25.0 ± 0.3	17.0 ± 0.5
1000	40.0 ± 7.1	56.8 ± 10.0	56.9 ± 4.5	42.9 ± 0.4
560 (h_1)	19.1 ± 1.0	25.8 ± 0.2	31.8 ± 2.9	20.6 ± 1.4
577 (h_2)	18.5 ± 0.1	26.4 ± 0.1	29.9 ± 0.7	20.9 ± 1.1
603 (h_3)	19.2 ± 0.1	27.4 ± 0.2	31.6 ± 1.4	22.3 ± 1.4
1740	56.8 ± 0.9	79.6 ± 0.3	93.3 ± 2.3	63.9 ± 3.0

Todos los tiempos se expresan en segundos y se describen mediante media ± sd, correspondientes a las 3 repeticiones de las simulaciones.

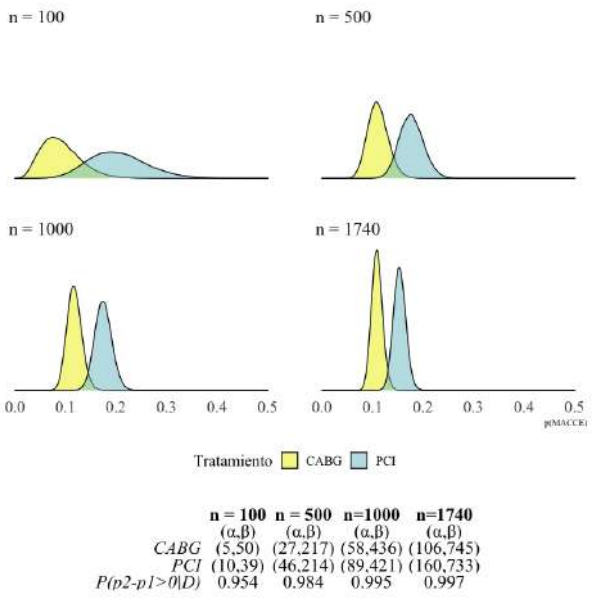


Figura 2. Distribución y parámetros de $p_i|D$ para cada grupo cuando se han observado 100, 500, 1000 y todos los datos.

posteriori de las proporciones, p_1 y p_2 y los parámetros de la mismas, para cada grupo cuando se han observado 100, 500, 1000 y todos los datos.

Una vez conocidos los parámetros de las distribuciones a posteriori, el promotor podría calcular la $P(p_2 - p_1 > 0|D)$ (ver tabla en la figura 2) a modo de evaluación de la hipótesis principal del estudio. A la vista de estos resultados, se alcanza una $P(p_2 - p_1 > 0|D) > 0.99$ con 1000 datos, por lo que se podría haber reducido el tamaño de la muestra de haberse propuesto un diseño adaptativo [20] para este estudio.

V. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo presentamos un prototipo de eCRD en el que los hospitales participantes en un ensayo clínico comparten sus datos cifrados. Se elige un cifrado homomórfico para que se puedan calcular los parámetros de las distribuciones a posteriori correspondientes con los que el promotor realizará el análisis del objetivo principal del estudio.

Los resultados experimentales muestran que es factible el uso de un esquema de cifrado homomórfico para realizar estos cálculos con garantías de seguridad, y con tiempos de ejecución razonables tanto para el cliente como el servidor.

El enfoque bayesiano permite una actualización secuencial de las distribuciones a posteriori a medida que los investigadores disponen de nuevos datos, lo que permite el procesado de grandes conjuntos de datos, aminora el tiempo de ejecución global y, por otro lado, facilita la implementación de diseños adaptativos en los ensayos clínicos con compartición de datos segura.

Como trabajo futuro, sería interesante comparar la implementación con otros esquemas de cifrado homomórfico, como el esquema BFV que implementa SEAL, e incorporar otros modelos de análisis de datos.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por: el proyecto MTM2017-83506-C2-2-P, financiado por el Ministerio de Economía y Competitividad; el proyecto PID2019-104790GB-I00, financiado por el Ministerio de Ciencia e Investigación; y las becas de colaboración del Ministerio de Educación y Formación Profesional.

V-A. Referencias

REFERENCIAS

[1] T. Baignères: “Provable security in cryptography”, pp. 28, 2017.
[2] J.W. Bos, K. Lauter, M. Naehrig: “Private predictive analysis on encrypted medical data”, en *Journal of Biomedical Informatics*, vol. 50, pp. 234-243, 2014.
[3] P.D. Hoff: “A First Course in Bayesian Statistical Methods”, Springer, 2009.
[4] P. Paillier: “Public-key cryptosystems based on composite degree residuosity classes”, in *EUROCRYPT, ser. LNCS*, vol. 1592. Springer, pp. 223–238, 1999.
[5] R.L. Rivest, A. Shamir, L. Adleman: “ A method for obtaining digital signatures and public-key cryptosystems”, en *Communications of the ACM*, num. 21, pp. 120–126, 1978.
[6] T. Sander, A. Young, M. Yung: “Non-interactive cryptocomputing for nc1”, en *40th Annual Symposium on Foundations of Computer Science*, pp 554–567, 1999.
[7] D. Boneh, E.-J. Goh, K. Nissim: “Evaluating 2-dnf formulas on ciphertexts”, en *Theory of Cryptography*, pp 325–341, 2005.
[8] C. Gentry: “A fully homomorphic encryption scheme”, PhD thesis, Stanford University, 2009.
[9] Z. Brakerski, C. Gentry, V. Vaikuntanathan: “(Leveled) Fully Homomorphic Encryption without Bootstrapping”, *ACM Trans. Comput. Theory*, vol. 6, n. 3, pp. 13:1-13:36, 2014.
[10] “Microsoft SEAL (release 3.0)”, http://sealcrypto.org, Oct. 2018, Microsoft Research, Redmond, WA.
[11] J.H. Cheon et al.: “Homomorphic encryption for arithmetic of approximate numbers”, in *ASIACRYPT, ser. LNCS*, vol. 10624. Springer, pp. 409–437, 2017.
[12] D. Bogdanov et al.: “Rmind: A Tool for Cryptographically Secure Statistical Analysis”, en *IEEE Trans. Dependable Secur. Comput.*, vol. 15, n. 3, pp. 481-495, 2018.
[13] L.J.M. Aslett, P.M. Esperança, C.C. Holmes: “A review of homomorphic encryption and software tools for encrypted statistical machine learning”, en *University of Oxford Technical report*, 2015.
[14] L.J.M. Aslett, P.M. Esperança, C.C. Holmes: “Encrypted statistical machine learning: new privacy preserving methods”, en *University of Oxford Technical report*, 2015.
[15] R. Bost et al.: “Machine Learning Classification over Encrypted Data”, en *NDSS*, 2015.
[16] S. Kim et al.: “Privacy-Preserving Naive Bayes Classification Using Fully Homomorphic Encryption”, en *ICONIP*, vol. 4, pp. 349-358, 2018.
[17] J. Chen et al.: “Non-interactive Privacy-Preserving Naïve Bayes Classifier Using Homomorphic Encryption”, en *SPNCE 2021, LNCS*, vol 423. Springer, 2022.
[18] Y. Yasumura, Y. Ishimaki, H. Yamana: “Secure Naïve Bayes Classification Protocol over Encrypted Data Using Fully Homomorphic Encryption”, en *iiWAS 2019*, pp. 45-54, 2019.
[19] P.W. Serruys et al.: “Percutaneous Coronary Intervention versus Coronary-Artery Bypass Grafting for Severe Coronary Artery Disease”, en *N Engl J Med*, vol. 360, pp. 961-972, 2009.
[20] S.M. Berry et al.: “Bayesian Adaptive Methods for Clinical Trials”, en *CRC press*, 2010.
[21] J. Fan, F. Vercauteren: “Somewhat Practical Fully Homomorphic Encryption”, en *Cryptology ePrint Archive, Report 2012/144*, 2012.

A Security and Trust Framework for Decentralized 5G Marketplaces

José María Jorquera Valero, Manuel Gil Pérez, and Gregorio Martínez Pérez
Faculty of Computer Science, University of Murcia, 30100 Murcia, Spain
Email: {josemaria.jorquera, mgilperez, gregorio}@um.es

***Abstract*—5G networks intend to cover user demands through multi-party collaborations in a secure and trustworthy manner. To this end, marketplaces play a pivotal role as enablers for network service consumers and infrastructure providers to offer, negotiate, and purchase 5G resources and services. Nevertheless, marketplaces often do not ensure trustworthy networking by analyzing the security and trust of their members and offers. This paper presents a security and trust framework to enable the selection of reliable third-party providers based on their history and reputation. In addition, it also introduces a reward and punishment mechanism to continuously update trust scores according to security events. Finally, we showcase a real use case in which the security and trust framework is being applied.**

***Index Terms*—Security, Trust Management, 5G, Trustworthy Relationships**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

The fifth generation of mobile networks (5G) entails a continuously increasing figure of interconnected end-users as well as relationships among entities. Thereby, 5G resource, service, and infrastructure providers are constantly working on innovative solutions to deal with the huge demand from tenants and users for data capacity, bandwidth, network coverage, and latency. One of the most promising approaches is *decentralized marketplaces*. A marketplace is normally described as a centralized or decentralized repository in which thousand of offers are advertised by resource and infrastructure providers. The trading of such resources and services is carried out by consumers who need to cover dynamic requirements and high Quality-of-Service (QoS); for instance, high speed, low energy consumption, and automatically satisfy user demands. In this context, marketplaces boost the generation of service chains across operators with security and trustworthiness [1].

Conventionally, trust has been employed as a mechanism capable of determining the trustworthiness level that a trustor has in a trustee. In this regard, there are multiple models for determining a trust level such as PKI-based, role-based, or reputation-based, among others. The latter is one of the most considered models, since it not only allows the trust to be calculated based on entity's behavior in previous interactions but also allows gathering feedback from reliable recommenders who also interacted with the entity. Yet, reputation-based trust models also open the door to key challenges to be addressed whether they want to be one of the technologies that await to support next-generation solutions (5G and beyond) [2].

Especially, trust models should guarantee a minimum set of features in order to be integrated with promising 5G solutions. First and foremost, trust models need to provide highly dynamic and context-dependence solutions as 5G ecosystems tend to support scenarios where main actors scale and migrate flexibly. In this sense, trust models should continuously collect information from the principal actors involved in the relationship through automatic mechanisms triggered by events, rules, time, etc. By means of these mechanisms, the trust score can be rapidly adapted in real-time, and in consequence, adjust the participating entities, if necessary. Besides, trust models should ensure reliable end-to-end establishments, therefore any intermediate entities must be analyzed. Another pivotal aspect to be covered is the automated management following a zero-touch approach. On the one hand, trust models should leverage tools that enable the automatization of network and service management via high-level policies, and artificial intelligence algorithms. On the other hand, they should empower a flexible integration with other 5G essential services and their workflows. Lastly, trust models should fulfill the zero trust principle [3]. It attempts not to attribute implicit trust to an entity regardless of whether the trustor and trustee had previously a relationship or whether both entities belong to the same domain. Therefore, trust models would avoid utilizing an outdated trust value without being updated at a given time as well as assign trust-by-default to entities under our reliable zone.

Despite the progress of literature, more efforts are needed to enhance the trust and reputation models in 5G scenarios as prior trust models did not contemplate all requirements set out above. Furthermore, the trust concept also includes security aspects, therefore considering security-related features for profiling actor behavior would help to detect feasible threats as well as to have a broader view when determining trust. In this regard, this paper presents a security and trust framework capable of ensuring a trustworthy network inside of a decentralized marketplace. In particular, such a framework enables consumers to establish trustworthy end-to-end connections across domains, at the same time it covers the aforementioned challenges. To this end, the security and trust framework analyzes a set of product offers, published by resource and infrastructure providers in a decentralized marketplace, in order to predict trust scores from previous interactions as well as recommendations from trustworthy third parties. It should be pointed out that such a framework is being developed in the 5GZORRO H2020 European project [4], which enables a secure, flexible, and distributed multi-stakeholder combination and composition of resources and services in 5G networks. Additionally, the security and trust framework also introduces a reward and punishment mechanism to continuously update trust levels from multiple security events obtained in real-time. Finally, the paper introduces a real use case to demonstrate how the framework is integrated with other 5G services and helps to perform a smart resource and service discovery.

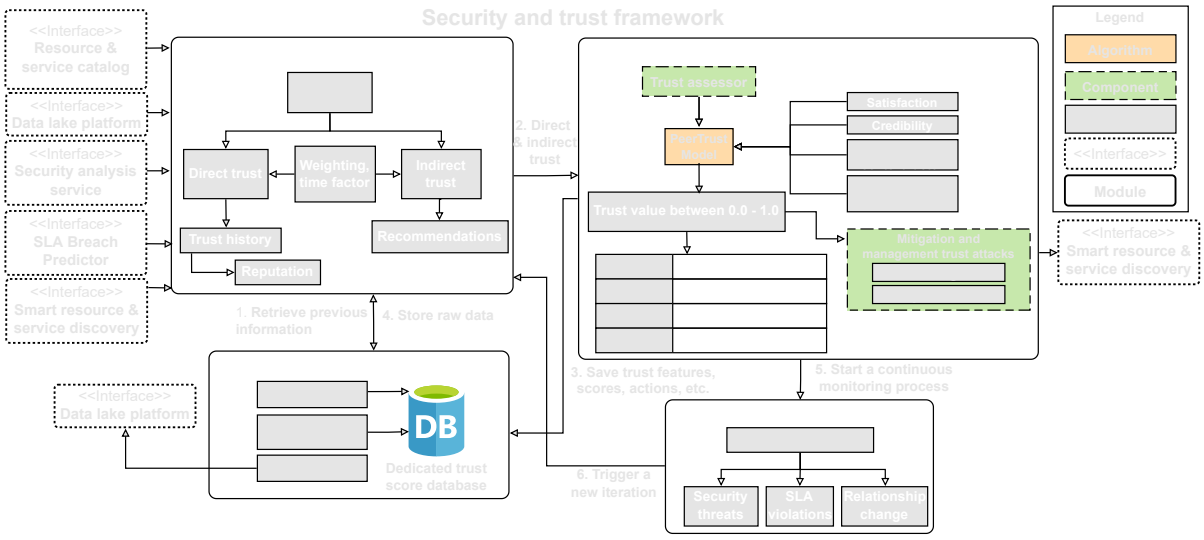


Figure 1. Modular architecture of the proposed security and trust framework

The remainder of this paper is structured as follows. Section II presents an overview of our security and trust framework, detailing the principal modules and the main actions to be performed. Section III introduces the use case in which our framework is being applied. Finally, Section IV expounds some conclusions an open perspectives for future work.

II. The Security and Trust Framework

This section describes a modular architecture of the proposed security and trust framework which enables to cater to a trustworthy ecosystem among multiple domains and operators. As Fig. 1 sketches, the security and trust framework is made up of four principal modules: the *information gathering and sharing*, *trust computation*, *trust storage*, and *continuous updates* modules. In this vein, this section is going to thoroughly explain them as well as the capital actions carries out.

A. Information gathering and sharing module

One of the uppermost important steps in a security and trust framework is the collection of meaningful features to afterward evaluate entity behavior. In this sense, it is pivotal to find out a set of information sources from which the trustor can retrieve information to ascertain the credibility and effectiveness of the assigned trust scores. Due to the fact that this framework is designed bearing in mind the 5GZORRO ecosystem, we have selected four principal software modules from which we will infer features (see interfaces in Fig. 1). On one side, we consider the Catalog module [5] which stores the Product Offerings (POs) published by the stakeholders registered in the 5GZORRO system. Note that there are seven types of product offers defined in the Catalog: radio access network (RAN), spectrum, virtual/container network function (VNF, CNF), slice, network service, cloud, and edge. By means of this dimension, the security and trust framework is capable of deriving statistic features such as the number of POs available per provider, per location, per type of resource, or what POs were available but are not currently, among other features. On another side, we contemplate the Data Lake platform [5] which stores all the interactions performed by stakeholders in the past. Thereby, a consumer would be able to know what stakeholders previously interacted with a specific resource and service provider and, if necessary, requests recommendations (indirect trust) about the provider. This information source plays an important pillar for the security and trust framework as this is based on a reputation model. Thence, recommendations from third parties are considered one of the two types of trust (direct and indirect trust) to compute a final trust score. Concerning direct trust, it is formulated from the previous interactions with the target stakeholder as well as new features inferred from Catalog.

In addition, the security and trust framework also gathers information from the Security Analysis Service (SAS) [5] which is instantiated once an end-to-end relationship is set up. Hence, it is not leveraged when the framework is analyzing a set of POs but when a final PO is selected. Through the SAS component, the framework can detect security threats or misbehavior in the network traffic as the SAS monitors, through the Zeek tool [6], network packets sent by consumers and providers. Moreover, the SAS component has activated several security policy rules that allow triggering alerts when unusual behaviors happen in real-time. Last but not least, the security and trust framework also recaps information from the Intelligent Service Level Agreement Breach Predictor (ISBP) module [5]. In this case, the framework leverages events published by the ISBP so as to predict the provider's reputation and how it can be affected based on current events such as Service Level Agreement (SLA) violations.

B. Trust computation module

Once the framework collects raw data from the aforementioned information sources, it is time to determine a trust score per each PO to be analyzed. Note that this step is performed regardless of whether stakeholders have a previous trust relationship for a while or whether they belong to the same domain (zero trust). Since the 5GZORRO ecosystem boosts a decentralized philosophy where peer-to-peer communications are carried out, we have selected the PeerTrust model [7] as a statistic algorithm to prognosticate a trust score. The PeerTrust model is an algorithm based on the reputation that has been

utilized in decentralized scenarios for ages. Concretely, the PeerTrust model determines trust scores from four primary pillars: Satisfaction (S), Credibility (Cr), Transaction Factor (TF), and Community Factor (CF), as depicted in Eq. (1). Note that the PeerTrust model only introduces the above-mentioned generic pillars but each researcher should find out how each one will be formulated. Thus, our security and trust framework has carried out an adapted PeerTrust model.

$$T(u) = \alpha \cdot \sum_{i=1}^{I(u)} S(u,i) \cdot Cr(p(u,i)) \cdot TF(u,i) + \beta \cdot CF(u), \quad (1)$$

Since each pillar is composed of multiple equations, this paper does not display the 19 equations formulated to cover all pillars. Instead, we broadly explain below how each pillar is computed as well as presenting some of the utmost important equations. In the case of *Satisfaction* (S), the framework predicts the provider's satisfaction and the offer's satisfaction. The main difference between them is the former considers all assets linked to a provider whereas the latter only envisages a given type of PO. Thence, the provider's and the offer's satisfactions are formulated from the reputation, a set of recommendations about the target, and the last trust score we have for each recommender in the previous set. When it comes to *Credibility* (Cr), the adapted PeerTrust model leverages a Personalized Similarity Metric (PSM) through which we can discover how similar two stakeholders are when they are evaluating the same targets. Thus, the PSM allows us to know the distance of credibility of a set of evaluated stakeholders as well as contrast their opinions.

Additionally, the adapted PeerTrust model also takes into account two context factors: the *Transaction Factor* (TF) and the *Community Factor* (CF). Concerning the TF, it intends to predict a trust value linked to the current interaction, with a particular stakeholder or product offer, from the number of feedbacks published in the Data Lake from different time windows. As a result, a higher number of feedbacks published in the Data Lake, a higher number of recommenders to be contemplated for finally computing a stable reputation. The TF then rewards stakeholders who publish their interactions with others in the Data Lake since it spurs future stakeholders to look into the Data Lake, request recommendations to other stakeholders, and grow the community. Instead, the CF intends to assess the stakeholder participation within the community to give greater or lesser weight to their recommendations over time. To this end, the CF collects multiple feedback from j-th trustworthy opinions on a target stakeholder u for computation, see Eq. (2), and notices dishonest recommendations as rated by the confidence from recommenders (CR) on u. By CF, we can thus detect conventional attacks in trust models such as bad-mouthing attacks.

$$CF(u) = \frac{\frac{R(u)}{I(u)} + \bigoplus_{j=1}^{n} CR(v,j,u) \cdot Inf(v,j)}{2}, \quad (2)$$

where $R(u)$ denotes the number of published recommendations and $I(u)$ the total number of interactions of u; and $Inf(v,j)$ the recommender's influence on all contemplated recommendations $Rec(j,u)$. Furthermore, Eq. (3) shows how to measure the confidence of a j-th recommender on u, in accordance to the trust on that recommender $T(v,j)$ and the recommendation trust $RT(v,j)$.

$$CR(v,j,u) = \alpha \cdot T(v,j) + (1-\alpha) \cdot \big(RT(v,j) \cdot Rec(j,u)\big), \quad (3)$$

Moreover, the CF also measures the number of interactions that a specific stakeholder had in the community through the contribution of services or resources with other stakeholders. In the end, multiple recommendations together with the credibility of the recommender are contemplated through an aggregation function to achieve the general reputation of the community about a target stakeholder.

C. Trust storage module

Another capital step of the security and trust framework is the storage of non-public data, raw data, and new interactions between stakeholders. In this vein, the framework employs two kinds of storage sources. On the one hand, the security and trust framework manages both raw data and all features inferred from them which are stored in a private database per domain. In this case, we are handling information which must not be public to other stakeholders. On the other side of the coin, the 5GZORRO ecosystem takes advantage of a Data Lake platform that allows users to push any type of information. By means of the Data Lake, the framework informs other stakeholders about the interactions. Concretely, after establishing a trust relationship the trustor pushes a new object into the Data Lake with information on the parties involved, the start date, the number of interactions between them, etc. In this way, newcomers can look at the Data Lake and identify what trustworthy recommenders had interaction with a particular target, and consequently, request their feedback about their relationships. In addition to that, the Data Lake introduces pivotal characteristics such as decentralization, a long-term reputation reflection, and security.

D. Continuous update module

Parallel to the trust storage module, the next step is to continuously monitor an ongoing trust relationship in order to adapt a previous trust score to the events occurring in real-time. Hence, this module plays an essential role since it not only enables early threat identification but also enhances the security capabilities of the framework through dynamicity, context-dependence, and end-to-end automatization. This module includes a reward and punishment mechanism which allows collecting security-based network monitoring events so as to increase or dwindle a trust score. This mechanism follows a time-driven approach so we need to declare pre-established time windows in which the previous trust score will be updated using the network monitoring events. To update the previous trust scores, the security and trust framework makes use of four main log files gathered by Zeek: *conn.log*, that gathers the tracking of general information regarding TCP, UDP, and ICMP traffics; *notice.log*, that collects likely monitoring events which are potentially odd; *weird.log*, that takes action when unusual or exceptional activity appears; and *stat.log*, that obtains memory, packet, and log statistics. These four weighted logs are shown in Eq. (4) to compute reward and punishment (RP) values.

$$RP(v,u) = \alpha \cdot Conn(v,u) + \beta \cdot Notice(v,u) + \psi \cdot Weird(v,u) + \phi \cdot Stat(v,u) \quad (4)$$

Finally, depending on the RP value obtained for the current time window, the framework will update the prior trust value (O_{ts}) to a new trust score (N_{ts}) accordingly, as given in Eq. (5). Note that a value close to the extremes will result in a larger increase or decrease than if the RP value is around 0.5.

$$N_{ts}(v,u) = \begin{cases} O_{ts}(v,u) + (RP(v,u) - 0.5) \cdot \frac{\left(1 - O_{ts}(v,u)\right)}{10}, & \text{if } RP(v,u) \geq 0.5 \\ O_{ts}(v,u) - (0.5 - RP(v,u)) \cdot \frac{\left(1 - O_{ts}(v,u)\right)}{10}, & \text{if } RP(v,u) < 0.5 \end{cases} \quad (5)$$

III. 5GZORRO Use Case

This section describes how the security and trust framework can assist in the process of selecting trustworthy resources and services available in a distributed marketplace (see Fig. 2).

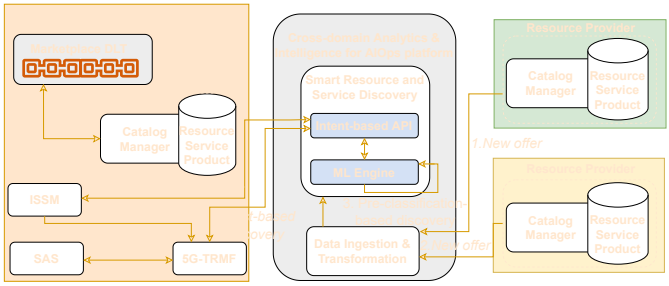


Figure 2. Trust-based resource discovery

Since the generation and registering of offers are not the main focus of our use case, we have simplified the sub-tasks of these steps in order to fully focus on the consumer side. In this case, the consumer detects a lack of resource capabilities to cover the agreed QoS and the dynamic requirements. To overcome the above problems, the consumer decides to extend its current capabilities to satisfy a signed contract and its SLA. At this point, the 5GZORRO marketplace [5] aims at facilitating multi-party collaboration in dynamic 5G environments where consumers and service providers often need to employ third-party resources. As a result, resource providers usually offer their resources available by promoting them through the 5GZORRO marketplace. In general terms, the distributed marketplace allows creating and acquiring product offers that cover a variety of telco digital assets.

To cope with this, the resource consumer leverages the Intelligent Slice and Service Manager (ISSM) [5] to request new resource capabilities as well as looking at the available offers in the . In this sense, two different discovery processes take place via the Smart Resource and Service Discovery (SRSD) [5]. On the one hand, the SRSD conducts a pre-classification of new POs based on imperative constraints such as category, geographic location, price, etc. On the other hand, the ISSM launches an intent-based discovery request employing priorities such as performance, proximity, cost, etc. Afterward, the SRSD determines trust scores and sends back to the ISSM a ranked list of product offers based on the reputation of both the provider and its type of offer. Therefore, the security and trust framework contributes directly to the intelligent discovery process classifying the most trustworthy POs which previously matched the intent-based criteria.

Afterward, the ISSM carries out additional optimization steps to find out the best offer from the list ranked by trust scores. As a next step, the ISSM informs the security and trust framework, called 5G-enabled Trust and Reputation Management Framework (5G-TRMF) in the 5GZORRO ecosystem [5], about the final offer selected by the consumer. To close the lifecycle of trust-based resource discovery, the security and trust framework enables, as a background process, ongoing monitoring of network traffic between the resource consumer and the new third-party via the Security Analysis Service. As shown in Fig 2, our framework can perfectly work with other pivotal modules such as the ISSM, Catalog, and SRSD to help consumers to address day-to-day problems such as resource shortages, while respecting automation and ensuring security and trustworthiness throughout the process.

IV. Conclusion

This paper presents a security and trust framework capable of ensuring a reliable ecosystem where stakeholders can set up trustworthy end-to-end connections across domains. In this vein, a modular architecture of the proposed framework has been explained, as well as the capital steps under each module. At the same time, the framework also introduces a reward and punishment mechanism to continuously update an ongoing trust relationship via security-based monitoring events generated in real-time.

As future work, we will carry out the validation of the current security and trust framework in real testbeds such as 5GBarcelona and 5TONIC. Besides, the continuous update module will be enhanced by adding a novel event-driven mechanism to adjust trust scores from SLA prediction notifications. Additionally, further computation models will be contemplated to analyze their performance and accuracy, as well as enlarging the resilience of the framework to other conventional trust attacks such as on-off or Sybil attacks.

Acknowledgment

This work has been supported by the European Commission through 5GZORRO project (grant no. 871533) part of the 5G PPP in Horizon 2020.

References

[1] A. Fernández-Fernández et al.: "Multi-party collaboration in 5G networks via DLT-enabled marketplaces: A pragmatic approach," in *Joint European Conference on Networks and Communications & 6G Summit*, pp. 550–555, 2021.

[2] J. M. Jorquera Valero et al.: "Toward pre-standardization of reputation-based trust models beyond 5G," *Computer Standards & Interfaces*, 81, 103596, 2022.

[3] S. Rose et al.: "Zero trust architecture," *NIST Special Publication (SP) 800-207*, National Institute of Standards and Technology, 2020.

[4] G. Carrozzo et al.: "AI-driven zero-touch operations, security and trust in multi-operator 5G networks: A conceptual architecture," in *European Conference on Networks and Communications*, pp. 254–258, 2020.

[5] D2.3: Update design of the 5GZORRO platform for security & trust, https://www.5gzorro.eu/deliverables/, [Online; accessed 13-April-2022] (2021).

[6] S. Haas et al.: "Zeek-osquery: Host-network correlation for advanced monitoring and intrusion detection," in *IFIP International Conference on ICT Systems Security and Privacy Protection*, pp. 248–262, 2020.

[7] L. Xiong, & L. Liu: "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE transactions on Knowledge and Data Engineering*, 16(7), pp. 843–857, 2004.

Privacidad y seguridad en Federated Learning: Combinando ambas mediante SCALE-MAMBA

Idoia Gamiz Ugarte[1], Cristina Regueiro Senderos[2], Oscar Lage Serrano[2], Eduardo Jacob Taquet[1]
[1]Department of Communications Engineering. University of the Basque Country (UPV/EHU), 48013 Bilbao, Bizkaia, Spain
{idoia.gamiz, eduardo.jacob}@ehu.eus
[2] TECNALIA, BRTA. Bizkaia Science and Technology Park, 700, E-48160 Derio, Bizkaia, Spain
{cristina.regueiro, oscar.lage}@tecnalia.com

***Resumen*—La privacidad es una de las mayores prioridades de la sociedad actual y a menudo se ve comprometida por la centralización de datos necesaria para el entrenamiento de *Machine Learning*. *Federated Learning* es una técnica ampliamente utilizada para evitar esta centralización, pero se han encontrado limitaciones en cuanto a la privacidad de los datos y la seguridad del modelo. El objetivo es encontrar una técnica capaz de abordar ambas simultáneamente. Para ello, la contribución principal es un análisis de la literatura sobre diferentes técnicas empleadas para proteger la privacidad y la seguridad de manera independiente, la elección de aquellas que puedan ser complementarias y la propuesta de un sistema completo en base a dicha elección. Se selecciona *Secure Multiparty Computation* para abordar el problema de privacidad, realizando la agregación mediante SCALE-MAMBA, y de entre las técnicas para proteger la seguridad se propone *Krum*, por su adaptabilidad a la herramienta seleccionada.**

***Index Terms*—Federated Learning, privacidad, seguridad, Secure Multiparty Computation, SCALE-MAMBA, Krum, Machine Learning, IoT**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

El volumen de datos generados crece exponencialmente [1] y cada vez son más las tecnologías *IoT* que recogen información para su posterior aplicación en entrenamientos de modelos de *Machine Learning(ML)*. La inteligencia artificial ha traído grandes ventajas a la sociedad actual, pero para obtener un modelo lo más generalista posible y evitar caer en la sobrestimación es imprescindible disponer de un número de datos apropiado para la complejidad del modelo. Sin embargo, la centralización de los mismos a menudo supone sobrepasar los límites de la privacidad y en la Unión Europea, el Reglamento General de Protección de Datos [2] ha impuesto el consentimiento y el control de los ciudadanos sobre sus datos como un derecho fundamental. *Federated Learning(FL)* [3] es una de las técnicas más utilizadas para evitar esta centralización, pero es importante tener en cuenta sus limitaciones en cuanto a privacidad de datos y seguridad del modelo [7], [8].

El objetivo principal de la descentralización es precisamente proteger la privacidad de los datos, pero se ha demostrado que existen varios ataques capaces de reconstruir información de los clientes a través de las actualizaciones de los parámetros [4], [7], [8], [9]. Se han estudiado diferentes técnicas para protegerse contra estos ataques, pero es importante tomar conciencia de la importancia de mantener la seguridad del modelo. Con seguridad del modelo se entiende que el modelo converja adecuadamente. El modelo puede no converger por falta de complejidad o por diversos motivos que no son necesariamente causados por clientes maliciosos. Por un lado, clientes honestos pueden enviar actualizaciones defectuosas al servidor, ya sea por errores de comunicación o porque sus datos tienen una distribución legítimamente diferente a la del resto de clientes. Por otro lado, existen ataques en los canales de comunicación para modificar la información transmitida o clientes maliciosos que modifican los parámetros intencionadamente. Se debe encontrar un método capaz de detectar estas situaciones. Además, en los entrenamientos de *ML* interesa recompensar a aquellos participantes que hayan ayudado en el proceso para motivarlos a participar. Esta compensación, que puede ser tanto económica como de reputación, debe basarse en hechos objetivos.

El objetivo principal de este trabajo es presentar un sistema capaz de mantener la privacidad de los datos y la seguridad del modelo simultáneamente. Para ello, se deben estudiar técnicas para solucionar cada una de ellas de manera independiente y seleccionar las que puedan ser complementarias. La mayor prioridad es mantener la privacidad total de los datos, al mismo tiempo que se minimice los efectos causados por un cliente que se desvía del protocolo enviando actualizaciones incorrectas, ya sea de manera accidental o intencionada.

El resto del artículo esta organizado de la siguiente manera: en la Sección II se introduce el caso de uso que se quiere afrontar. En la Sección III se detallan los principales problemas de privacidad y seguridad en *FL*. En la Sección IV se presenta una propuesta de implementación de un sistema completo capaz de abordar los problemas de privacidad y seguridad simultáneamente. Finalmente, en la Sección V se resumen las conclusiones y se propone el trabajo futuro.

II. Caso de uso

Un servidor, P_0, dispone de un modelo de *ML* que desea entrenar. El resto de participantes, $P_1, ..., P_n$, son los clientes que disponen de datos para entrenar dicho modelo. Para mantener los datos privados se realiza un entrenamiento descentralizado basado en la técnica de *FL*. El proceso se divide en múltiples rondas donde el servidor central se encarga de actualizar el vector de parámetros global, habitualmente mediante la media ponderada [3]. Los clientes realizan las actualizaciones locales en base a la porción de datos que les pertenece. Sin embargo, como se estudia más en profundidad en la Sección III, este método tiene limitaciones desde el punto de vista de la ciberseguridad. En el presente caso de uso, suponemos que el servidor dispone de datos públicos o

información suficiente de los participantes como para desanonimizar o apuntar al propietario de un modelo, aunque solo reciba el vector de parámetros local de dicho cliente.

III. Privacidad y seguridad en FL

A la hora de entrenar un modelo de *FL* hay dos factores que se deben tener en cuenta desde el punto de vista de la ciberseguridad: la privacidad de los datos y la seguridad del modelo. En [7], [8] explican detalladamente los distintos ataques en estos dos ámbitos y aportan referencias para todos ellos. Está sección recoge los ataques más recurrentes y se seleccionan de entre las soluciones encontradas en la literatura las más relevantes para cumplir el objetivo mencionado.

III-A. Ataques de privacidad

- *Attribute Inference attacks:* Los modelos de los clientes pueden desvelar atributos de sus datos privados.
- *Membership attacks:* Permiten descubrir si un dato concreto ha sido utilizado en el proceso de entrenamiento.
- *Reconstruction attacks:* Permite muestrear puntos de datos con la distribución de los utilizados durante el proceso de entrenamiento.

Solución: Emplear técnicas criptográficas como *Secure Multiparty Computation (SMPC)* [5] o *Differential Privacy (DP)* [6] antes de realizar la agregación de los parámetros para que el servidor no reciba los modelos en claro. Adicionalmente, en [7] presentan otra alternativa que consiste en definir un protocolo en el que se pierda el vínculo entre los clientes y sus actualizaciones, de tal forma que el servidor reciba los modelos locales, pero no sepa de quién los esta recibiendo. De aquí en adelante el artículo se refiere a este último método como *desvinculación*. Dependiendo de las características de los participantes cada una de estas técnicas solucionará alguno o todos los ataques aquí presentados. En la sección IV-A, se analiza más en profundidad para este caso de uso.

III-B. Ataques de seguridad

Según el objetivo, los ataques se agrupan en dos grandes bloques.

- *Byzantine attacks:* El objetivo es provocar que el modelo no converja. Habitualmente, la agregación del federado se realiza mediante la media, pero en [10] demuestran que ningún tipo de combinación lineal de los parámetros puede tolerar a un solo adversario de este tipo.
- *Model poisoning attacks:* El objetivo es que converja a un modelo erróneo, de tal forma que afecte al rendimiento del modelo sin que se note.

Solución: Encontrar un método que verifique la autenticidad de cada actualización local. Para ello, el servidor debe poder operar sobre cada una de las actualizaciones locales de manera independiente, ya sea de forma directa o cifrada (siempre que esta última permita evaluar los modelos y medir la contribución de los mismos al modelo global). En la siguiente sección se resumen métodos para realizar esta evaluación.

III-C. Evaluación de modelos locales

Las soluciones encontradas en la literatura se pueden agrupar en dos grandes ramas. Esta sección recoge las que podrían ser más adaptables al presente caso de uso.

Identificación de adversarios: Sirven para identificar adversarios maliciosos y aportan un valor a cada participante. Las más importantes se basan en métricas del modelo.

- *Federated Leave-one-out(LOO)* [11]. Mide la diferencia entre el comportamiento del modelo con y sin el participante mediante un conjunto de validación.
- *Federated Shapley Value (SV)* [11]. Es una de las más utilizadas y consiste en un método de distribución de riquezas en la teoría de juegos cooperativos.

Agregación robusta: No siempre asignan un valor a cada participante, pero realizan la agregación omitiendo aquellas actualizaciones que no benefician al modelo global. Se buscan técnicas que minimicen los efectos de los ataques.

- Utilizar métodos que se vean menos influenciados que la media por valores atípicos [12]. Por ejemplo, *Coordinate-wise median* calcula la mediana coordenada a coordenada y *Coordinate-wise trimmed mean* ordena las contribuciones para eliminar una fracción β de las contribuciones más pequeñas y más grandes.
- *Krum aggregation rule* [10]. Modifica el proceso para hacerlo seguro contra f pares maliciosos, eligiendo la actualización local más cercana de sus $n-f-2$ vecinos.

IV. Propuesta de implementación

Es importante recalcar la necesidad de encontrar una solución que aborde los problemas de privacidad de datos y seguridad del modelo simultáneamente.

En [8] se calcula la gravedad de cada una de las amenazas en función de la probabilidad de que el adversario se aproveche de la vulnerabilidad y lance una ataque. Argumentan que los *poisoning attacks* (modelos), causados por cambios en los datos o en los parámetros de los clientes, y los *inference attacks* (datos) son los más perjudiciales. Sin embargo, presentan las soluciones de manera independiente, sin abordar técnicas concretas que consigan proteger tanto la privacidad como la seguridad de manera sincronizada. Muchas de las técnicas que protegen la seguridad exigen tener acceso a los parámetros individuales, perjudicando la privacidad. Por lo tanto, es importante encontrar un equilibrio.

En este trabajo se pretende implementar una solución que mantenga la privacidad total de los datos, al mismo tiempo que minimice los efectos causados por un cliente que se desvía del protocolo enviando actualizaciones incorrectas.

IV-A. Privacidad: SMPC mediante SCALE-MAMBA

Para mantener la privacidad de los datos se han estudiado diferentes opciones en función de cuatro características: privacidad de los datos, adaptabilidad a técnicas para mantener los modelos seguros, precisión del modelo obtenido y eficiencia del proceso. La clasificación final se muestra en la *Tabla I* y se explica a continuación. La primera fila presenta el mecanismo que se quiere mejorar, FL, para visualizar mejor la comparación.

- *FL utilizando la media para la agregación:* No mantiene la privacidad de los datos porque es vulnerable a los ataques explicados en la Sección III-A, pero permite aplicar técnicas contra los ataques de seguridad como las mencionadas en III-C. La precisión del modelo no se ve alterada y tampoco exige un coste extra de comunicación, más haya del propio de *FL*.

- *FL utilizando SMPC para la agregación:* Consigue mantener la privacidad de los datos, cifrando las actualizaciones locales antes de su agregación. Sin embargo,el hecho de tener acceso cifrado a las actualizaciones, limita las técnicas de preservación de seguridad como las de la Sección III-C a aquellas que se puedan realizar mediante *SMPC*. Además, supone una perdida de precisión y un coste de comunicación extra entre los clientes, aunque puede ser aceptable para muchos casos de uso.
- *FL utilizando DP para la agregación:* Al añadir ruido a las actualizaciones, los datos se mantienen privados. Sin embargo, el ruido causa limitaciones para la evaluación de los modelos locales y la perdida de precisión que provoca puede llegar a impedir que el modelo converja. Exige un coste extra para añadir el ruido, pero es mínimo comparado con el de *SMPC*.
- *FL mediante un protocolo de desvinculación:* Es la mejor opción en cuanto a adaptabilidad a técnicas para preservar la seguridad y precisión. Sin embargo, para el caso de uso en el que los participantes se conozcan y dispongan de información para vincular la actualización con su dueño, puede no ser suficiente en términos de privacidad. Dependiendo del protocolo puede exigir un coste de comunicación extra.

Tabla I
ALTERNATIVAS DE FL: PRIVACIDAD Y SU ADAPTABILIDAD A TÉCNICAS PARA PRESERVAR LA SEGURIDAD DEL MODELO

	Privacidad (datos)	Seguridad (modelo)	Precisión (modelo)	Eficiencia
FL	No	Sí	Sí	Sí
FL+SMPC	Sí	Parcial	Parcial	Parcial
FL+DP	Sí	No	No	Sí
FL+desvinculación	Parcial	Sí	Sí	Parcial

El objetivo principal de este trabajo es mantener la privacidad total de los datos, incluso para el caso en el que los participantes puedan disponer de información suficiente como para vincular un modelo con un cliente, por lo que se descartan los protocolos de desvinculación para este caso de uso concreto. También se excluye el uso de *DP* por su pérdida de precisión y por las limitaciones que conlleva el hecho de añadir ruido para la evaluación de los modelos locales. Por tanto, la técnica seleccionada para realizar esta implementación es *SMPC*. Si bien es cierto que trae consigo un coste de comunicación extra significativo, es una buena alternativa para mantener la privacidad total de los datos y permite la realización de algunos métodos de validación dentro del mismo programa de *SMPC*, como se estudiará más en profundidad en la Sección IV-B .

Para la realización de *SMPC* se ha seleccionado la herramienta de SCALE-MAMBA [13], en base a las ventajas y características mencionadas en [14], el tipo de documentación y su adaptabilidad al caso de uso requerido. Las ventajas principales son: (i) flexibilidad en cuanto a tipo de operaciones y definición de entrada y salida de datos, lo que permite llevar acabo el diagrama presentado en la Sección IV-C, (ii) seguridad para adversarios activos, (iii) admisión de un número ilimitado de participantes y (iv) acceso *open-source*.

Implementa diferentes protocolos de *SMPC*, dependiendo del tipo de adversario y de la pérdida de eficiencia que se quiera afrontar. La comunidad detrás del proyecto lleva muchos años investigando en este campo y es muy activa a día de hoy. Actualmente, la librería se encuentra en fase de migración para reescribir los programas del lenguaje *Mamba*, propio de SCALE-MAMBA, al lenguaje *Rust*. Esto dotaría al proyecto de mayor impacto, siendo *Rust* un lenguaje más conocido y con ventajas en términos de eficiencia. Por el momento, este trabajo se ha realizado utilizando *Mamba*.

IV-B. Seguridad: Krum aggregation rule

Una vez seleccionada la aproximación para la protección total de datos y seleccionada la herramienta para llevarlo a cabo, se debe seleccionar de entre los métodos propuestos en la Sección III-C los que mejor se adaptan a las condiciones y al tipo de operaciones que se pueden realizar mediante SCALE-MAMBA. Una de sus mayores desventajas es su ineficiencia en cuanto a tiempo de ejecución y la falta de matrices de tamaño variable. Por tanto, se debe encontrar un algoritmo que se adapte a estos requisitos.

En [7] presentan *Krum aggregation rule [10]* como uno de los mejores candidatos por su coste computacional bajo y señalan sus ventajas en cuanto a resultados para grandes coaliciones. Además, no necesita un conjunto de validación para realizar la operación, lo que hace que pueda ser adaptable a más casos de uso. También en [8] lo presentan como una buena alternativa para la detección de anomalías. En cuanto al tipo de operaciones, se ha realizado un amplio análisis sobre las que se pueden realizar con el lenguaje *Mamba* y se ha confirmado la viabilidad de implementar este algoritmo en dicho lenguaje. Por tanto, se ha seleccionado *Krum* para la implementación y se ha escrito un código en *Mamba*, Krum.mpc, a modo de prueba de concepto para su posterior ejecución (disponible online: https://doi.org/10.5281/zenodo.6610424). Es resistente hasta f *Byzantine clients*, asumiendo que $f < \frac{n-2}{2}$.

Para una ronda t del proceso de FL, la regla *Krum* calcula un valor $s_t(P_i)$ para cada participante P_i:

1. Calcula la distancia de su parámetro local, ω_i^t, a cada uno de los parámetros locales del resto de clientes ω_j^t, para todo $i \neq j$.
2. Ordena las distancias para tomar los $n - f - 2$ más cercanos a ω_i^t.
3. Suma las distancias seleccionadas en el punto 2 y lo adjudica como valor del cliente P_i, $s_t(P_i)$.

Finalmente, se toma como valor de *Krum* la actualización local ω_{i*}^t, siendo P_{i*} el cliente que minimiza el valor, $s_t(P_{i*}) \leq s_t(P_i)$ para todo $i \in \{1, ..., n\}$, donde n es el número total de clientes. Se utiliza para obtener el modelo global para la próxima iteración: ω^{t+1}.

IV-C. Esquema de implementación

En esta sección se presenta la implementación teórica que se quiere llevar a cabo para la fusión de privacidad de datos y seguridad del modelo. Sea P_0 el servidor que posee un modelo de *ML* y sean $P_1, ..., P_n$ los clientes que contienen datos para entrenar ese modelo. El entrenamiento federado se realiza mediante un proceso iterativo de T rondas, donde $t = 1, ..., T$ es la variable indicativa de la ronda actual. El diagrama del proceso se presenta en la *Figura 1* y se resume en los siguientes pasos:

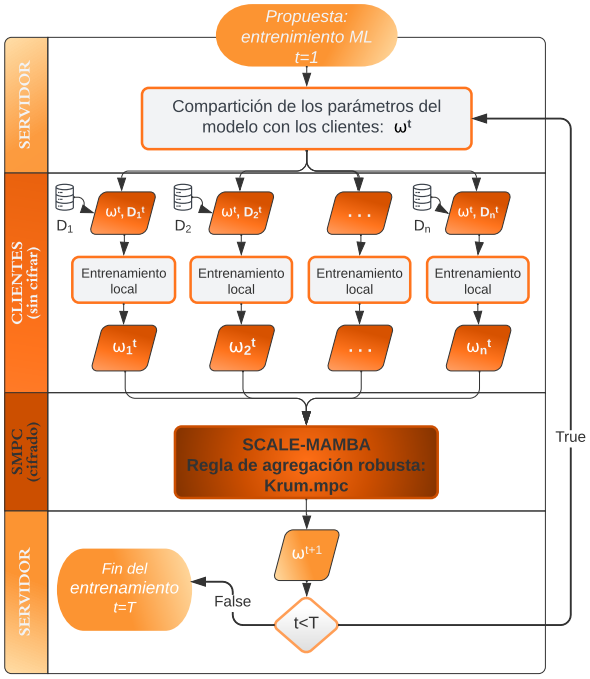


Figura 1. Esquema de implementación

1. El servidor propone un entrenamiento y se pone en contacto con los clientes para iniciar el proceso de *FL* y crear la red de SCALE-MAMBA.
2. Comienza el proceso iterativo. El servidor envía el vector de parámetros global de la ronda actual, ω^t, a los clientes.
3. Cada cliente P_i, para todo $i \in \{1, ..., n\}$, entrena localmente el modelo en base al vector de parámetros recibido, ω^t, y a la porción de datos de la que dispone para la ronda t, D_i^t . El resultado es el vector de parámetros local de dicho cliente en la ronda t: ω_i^t.
4. Una vez finalizado el entrenamiento local, comienza la fase de la agregación mediante SCALE-MAMBA. Los clientes aportan la entrada de datos al programa de *SMPC*, los parámetros locales $\omega_1^t, ..., \omega_n^t$, y el servidor es el único que recibe el resultado de la función, es decir, el vector de parámetros global resultante tras aplicar *Krum* sobre los parámetros locales: ω^{t+1}.
5. Se vuelve al paso 2, hasta que $t = T$ y finaliza el proceso.

V. Conclusiones y trabajo futuro

El objetivo principal es mantener la privacidad total de datos al mismo tiempo que se minimiza el impacto de los ataques contra la seguridad del modelo. El uso de *SMPC* para la agregación de los parámetros permite mantener la privacidad de estos, mientras que emplear *Krum* en lugar de la media ponderada para la agregación hace que el algoritmo sea tolerante a clientes maliciosos que deseen perjudicar el modelo final. Esto permite construir un sistema seguro contra hasta f *Byzantine attacks*, siendo $f < \frac{n-2}{2}$.

Hay varios puntos que se deben seguir investigando como trabajo futuro:

- La definición completa de un sistema de *FL* adaptable a la herramienta de SCALE-MAMBA. Dicho sistema supone un protocolo para habilitar la comunicación entre los participantes y la compartición segura de parámetros y una fusión de ese protocolo con la red de SCALE-MAMBA.
- La ejecución del programa Krum.mpc en dicho sistema y su equivalente en el lenguaje Rust para su comparación.
- La investigación de diferentes alternativas de valoración de modelos y el análisis de las ventajas y desventajas de cada una de ellas. Resultan de especial interés aquellas que permitan obtener una lista con los posibles adversarios para crear un sistema de recompensas.
- La implementación de un protocolo de desvinculación, la comparación con este sistema y la identificación de los posibles casos de uso para cada uno de ellos, en función del contexto.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Gobierno Vasco mediante el proyecto con referencia KK-2021/00026, y por TECNALIA y la Universidad del País Vasco mediante el contrato con referencia PIFTEC21/05.

Referencias

[1] Padhi, B.K., Nayak, S. and Biswal, B.: "Machine learning for big data processing: A literature review", en *Int. J. Innov. Res. Technol*, vol. 5, n. 7, pp. 359-368, 2018.
[2] GDPR: "General Data Protection Regulation (GDPR)". https://gdpr-info.eu/(2018)
[3] McMahan, B., Moore, E., Ramage, D., Hampson, S. and y Arcas, B.A.: "Communication-efficient learning of deep networks from decentralized data", en *Artificial intelligence and statistics*, pp. 1273-1282, 2017.
[4] Melis, L., Song, C., De Cristofaro, E. and Shmatikov, V.: "Exploiting unintended feature leakage in collaborative learning", en *In 2019 IEEE Symposium on Security and Privacy (SP)*, pp. 691-706, 2019.
[5] Canetti, R., Feige, U., Goldreich, O. and Naor, M.: "Adaptively Secure Multi-party Computation", en *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* , pp. 639-648, 1996.
[6] Dwork, C.: "Differential privacy", en *H.C.A. van Tilborg, S. Jajodia (Eds.), Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.
[7] Blanco-Justicia, A., Domingo-Ferrer, J., Martínez, S., Sánchez, D., Flanagan, A. and Tan, K.E.: "Achieving security and privacy in federated learning systems: Survey, research challenges and future directions", en *Engineering Applications of Artificial Intelligence*, vol. 106, p. 104468, 2021.
[8] Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A. and Srivastava, G.: "A survey on security and privacy of federated learning", en *Future Generation Computer Systems*, vol. 115, pp. 619-640, 2021.
[9] Narayanan, A. and Shmatikov, V.: "How to break anonymity of the netflix prize dataset", en *arXiv preprint cs/0610105*, 2006.
[10] Blanchard, P., El Mhamdi, E.M., Guerraoui, R. and Stainer, J.: "Machine learning with adversaries: Byzantine tolerant gradient descent", en *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[11] Wang, T., Rausch, J., Zhang, C., Jia, R. and Song, D.: "A principled approach to data valuation for federated learning", en *Federated Learning*, pp. 153-167, 2020.
[12] Yin, D., Chen, Y., Kannan, R. and Bartlett, P.: "Byzantine-robust distributed learning: Towards optimal statistical rates", en *In International Conference on Machine Learning*, vol. x, n. y, pp. 5650-5659, 2018.
[13] Aly, A., Cong, K., Cozzo, D., Keller, M., Orsini, E., Rotaru, D., Scherer, O., Scholl, P., Smart, N., Tanguy, T. and Wood, T.: "SCALE-MAMBA v1. 2: Documentation", 2018.
[14] Hastings, M., Hemenway, B., Noble, D. and Zdancewic, S.: "Sok: General purpose compilers for secure multi-party computation", en *2019 IEEE symposium on security and privacy (SP)*, pp. 1220-1237, 2019.

Estudio comparativo de rendimiento de frameworks de criptografía homomórfica

Iñaki Seco Aguirre[1], Cristina Regueiro[1], Mª Carmen Palacios[1],
Maite Álvarez Piernavieja[1] y Eduardo Jacob[2]

[1]TECNALIA, Basque Research and Technology Alliance (BRTA)
[2]University of the Basque Country UPV/EHU

***Resumen*—La criptografía homomórfica (HE, por sus siglas en inglés) es una tecnología que permite realizar operaciones computacionales sobre datos numéricos que están cifrados, y que por tanto, se mantienen privados para quien opera con ellos. Recientemente, este sistema criptográfico ha acaparado mucha importancia debido a ciertos avances tecnológicos, y se está empezando a utilizar con frecuencia en combinación con otras tecnologías, como la Inteligencia Artificial (IA) o la Computación Multi-Parte (MPC, por sus siglas en inglés).**

En este artículo se presenta un estudio que compara tres de las herramientas más importantes en la actualidad, SEAL, PALISADE y LATTIGO, con las que es posible implementar criptografía homomórfica, más concretamente, el esquema propuesto por Cheon, Kim, Kim y Song (CKKS). Los resultados obtenidos muestran las ventajas y limitaciones que tienen cada uno de ellos, permitiendo una mejor caracterización de sus potenciales casos de aplicación.

***Index Terms*—Criptografía Homomórfica, Ciberseguridad, Privacidad, Tiempo, Dimensión, Precisión, PALISADE, SEAL, LATTIGO**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

A medida que la tecnología avanza, la posibilidad de recoger grandes cantidades de datos de los usuarios está cada vez más al alcance de las empresas y las instituciones públicas, habitualmente con el objetivo de utilizarlos en su propio beneficio. Esta práctica es cada vez más común, y ocurre en detrimento de la privacidad de los que hacen uso de la dicha tecnología, algo prácticamente inevitable hoy en día [1]. Para hacer frente a este problema, están surgiendo una gran variedad de nuevas formas de preservar la privacidad, apoyándose en las herramientas de seguridad digital existentes en la actualidad, y una de estas herramientas es la criptografía homomórfica (HE) [2]. El problema es que están emergiendo varias soluciones que implementan esta tecnología, pero existe muy poca información que permita valorar qué alternativa se ajusta mejor a las necesidades particulares de cada caso [3].

El objetivo de esta publicación es estudiar el rendimiento de algunas de las plataformas más comunes de criptografía homomórfica, como son SEAL [4], PALISADE [5] y LATTIGO [6], con el fin de identificar sus fortalezas y debilidades.

Este artículo se divide en una sección II que recoge los antecedentes, la sección III donde se recoge la metodología que se ha aplicado para lograr los resultados de la sección IV, en la sección V se exponen las conclusiones obtenidas a partir de los resultados, y finalmente, en la sección VI se exponen las futuras líneas de trabajo.

II. Antecedentes

II-A. Criptografía Homomórfica

HE es un sistema criptográfico cuya principal característica es que permite aplicar operaciones aritméticas sobre los textos cifrados (CT, por sus siglas en inglés), o entre ellos. Es decir, permite obtener resultados a partir de un conjunto de elementos cifrados, sin llegar a tener acceso a la fuente de datos en claro.

Para lograr esto HE se apoya en el uso de los homomorfismos, que son funciones que preservan las operaciones que se pueden aplicar sobre una serie de objetos, siempre y cuando mantengan la misma estructura algebraica [7].

Esta tecnología existe desde hace años pero tenía limitaciones computacionales que no permitían su aplicación [8], hasta que en 2009, Craig Gentry propuso un nuevo mecanismo de cifrado en su tesis doctoral y lo llamó Fully Homomorphic Encryption (FHE) [9] [10]. A partir de ahí, han surgido varios esquemas de cifrado homomórficos, entre los que se encuentra Cheon, Kim, Kim y Song (CKKS) [11] [12], que es el que se ha utilizado para realizar el estudio presentado en este artículo debido a su capacidad para operar con valores reales.

II-B. Trabajos relacionados

Existen comparativas de frameworks realizados previamente, pero ponen el foco principalmente en los múltiples esquemas de cifrado homomórficos que se han desarrollado en lo últimos años. En 2016 se redactó un estado del arte que analizaba todos ellos [13], haciendo un estudio centrado en las capacidades teóricas, pero no aplicadas, de todos los esquemas hasta ese momento. Cuando comenzaron a surgir librerías de código abierto que implementaban los mecanismos de cifrado, se analizaron algunos de ellos. Así, en 2018, se publicó un artículo donde hacía una comparación centrándose en el coste computacional y el ruido introducido por los algoritmos de criptografía [14].

Otros artículos más recientes sí hacen comparaciones orientadas a la implementación de los esquemas, por ejemplo, el artículo [15] se limita a analizar los desarrollos de Microsoft en SEAL pero sin involucrar a otros desarrolladores, o [16], donde es posible ver un estudio de las librerías desarrolladas en C++: SEAL, PALISADE y HElib, centrado mayormente en los tiempos de ejecución para distintas dimensiones del anillo de los textos cifrados. Sin embargo, aún no se ha realizado un estudio comparativo que profundice en aspectos como el tamaño o la precisión de los frameworks, además, ninguno de ellos incluye LATTIGO como alternativa.

III. Metodología

Es importante mencionar que, la criptografía homomórfica actual está basada en retículos (lattice) donde los espacios para los CT se estructuran en anillos [17]. Estos anillos, programáticamente, se representan como arrays de valores complejos o de tipo float, donde se introducen los valores normalizados antes de ser cifrados. Los anillos tienen siempre una dimensión finita, que se preconfigura en el contexto criptográfico, y por lo general, los huecos del anillo donde no se introducen datos se completan con valores nulos o ceros.

Para realizar la comparativa se ha decidido partir de tres de las características más relevantes a la hora de valorar un sistema de cifrado:

- Los **tiempos** de cifrado, descifrado y evaluación. Son importantes para conocer los requisitos computacionales y qué se puede esperar de estos sistemas de cifrado.
- El **tamaño** de los textos cifrados tiene un gran impacto en los requisitos de almacenamiento de una aplicación que utilice HE.
- La **precisión** de los datos descifrados. Es trascendental para obtener unos resultados rigurosos, un dato impreciso puede desencadenar errores en el futuro.

La Figura 1 muestra la infraestructura empleada para la realización de las pruebas.

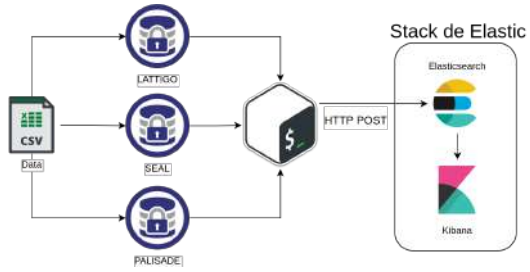


Figura 1. Infraestructura utilizada para ejecutar las pruebas.

Dicha infraestructura incluye los siguientes componentes:

- Fichero de formato **CSV como entrada de datos**. El dataset utilizado es el CIC-IDS2017 [18], acotado a 10.000 filas, cada fila contiene 172 columnas con valores entre 0.0 y 1.0, que son los utilizados para operar.
- Conjunto de **ejecutables** binarios. Se han realizado las mismas pruebas para lograr el mismo número de registros en cada uno de los frameworks, y poder así obtener unas medidas en igualdad de condiciones para evaluarlos de forma homogénea.
- Grupo de herramientas de Elastic [19] para su procesamiento y visualización.

Para una toma de medidas fiable, se han diseñado e implementado una serie de ejecutables por cada framework, todos ellos con un contexto criptográfico similar y una dimensión del anillo de 4096 huecos, de forma que se garantice una toma de medidas homogénea. Además, todas las pruebas se han ejecutado sobre el mismo equipo en las mismas condiciones: Dell Latitude 5590 de 64 bits, Linux OS, 16GB RAM, procesador Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz.

Cabe destacar que las operaciones homomórficas que se han efectuado durante el estudio son de dos tipos: suma (*EvalAdd*) y multiplicación (*EvalMult*).

IV. Resultados

En esta sección se van a mostrar los resultados obtenidos a partir a las medidas tomadas mediantes los procesos explicados en las secciones previas.

IV-A. Tiempo

Para analizar los tiempos, se han dividido los resultados en tres tipos, dependiendo del tipo de acción que se ejecute: cifrado, descifrado y evaluación.

Como se puede apreciar en la Figura 2, tanto PALISADE como SEAL tienen un promedio de tiempo de cifrado muy similar, ambos por debajo de los 12 ms. No obstante, el primero de ellos tiene una menor mediana, esto es debido a que, como se ve en la gráfica, la mayoría de muestras de PALISADE no superan los 10 ms, sin embargo, tiene más tendencia a sufrir desviaciones, ya que bastantes de sus tiempos de cifrado superan los 20 ms, de hecho, el tiempo máximo de PALISADE es 175 ms, muy superior al de las otras alternativas, incluso al de LATTIGO, el framework con tiempos de cifrado más elevados en mediana y promedio.

	Nº Registros	Promedio (µs)	Mediana (µs)	Máx. (µs)	Mín. (µs)
lattigo	35,000	16,145	12,613	43,165	6,775
palisade	35,000	11,717	6,773	175,685	4,196
seal	35,000	11,837	10,209	30,269	3,820

Figura 2. Tiempos de cifrado.

En cambio, durante descifrado de los textos cifrados, es PALISADE la que obtiene peores resultados, ya que sus tiempos llegan a ser hasta 41 veces superior a los de sus competidores, que no superan el milisegundo. Estos se pueden observar en la tabla de la Figura 3.

	Nº Registros	Promedio (µs)	Mediana (µs)	Máx. (µs)	Mín. (µs)
lattigo	12,602	491	361	2,721	214
palisade	12,602	20,509	10,459	176,776	4,612
seal	12,602	491	330	2,608	94

Figura 3. Tiempos de descifrado.

En cuanto a las medidas tomadas durante la evaluación de las operaciones homomórficas, los resultados se asemejan a los tiempos de descifrado. En la gráfica de la Figura 4 es posible ver que SEAL y LATTIGO son los más rápidos, con una significante diferencia, y PALISADE es la más lenta, donde la mayoría de sus operaciones se realizan en un intervalo de 0,4 ms a 0,6 ms, aunque muchas se disparan por encima de 1,2 ms.

Cabe destacar que el valor más alto tomado en una única evaluación es extremadamente superior en el caso de PALISADE, de hasta 450 ms, como se ve en la tabla de la Figura 4. Para entender este dato es necesario contextualizar las

	Nº Registros	Promedio (µs)	Mediana (µs)	Máx. (µs)	Mín. (µs)
lattigo	22,500	333	235	3,378	133
palisade	22,500	2,064	505	450,215	376
seal	22,500	418	302	3,403	149

Figura 4. Tiempos de evaluación.

muestras en función del tipo de operaciones que se ejecutan ya que, como se aprecia en la Figura 5, los productos tienen un promedio muy superior al resto de combinaciones, de hecho, de las 67.500 muestras tomadas en total en esta categoría, solo cuatro de ellas superan los 110 ms, y las cuatro son productos realizados en PALISADE.

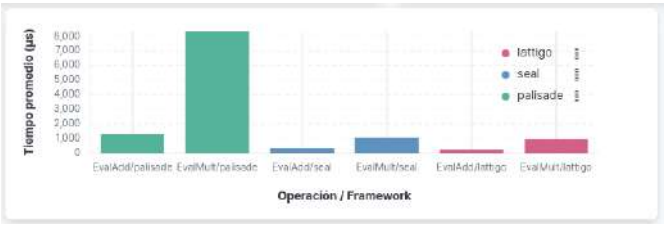


Figura 5. Tiempos de evaluación en función de la operación.

IV-B. Tamaño de los CT

En esta sección se va a analizar el tamaño de los CT, para ello se han tenido en cuenta dos cosas: los textos inmediatamente después de ser cifrados, antes de ser evaluados, y los CT después de realizar las operaciones correspondientes.

En el primero de los casos se ha observado que, independientemente del número de elementos que se introduzcan en el anillo, el tamaño de los CT es siempre el mismo para cada una de las plataformas, únicamente varía en base a la dimensión del anillo, que en este caso es siempre de 4096, siendo LATTIGO el framework con el CT más pesado (768KB), seguido por el de SEAL (326KB), y en último lugar el más ligero, que es PALISADE con 259KB.

Si se tienen en cuenta únicamente los textos cifrados tras aplicar alguna de las operaciones homomórficas sobre ellos, hay dos conclusiones interesantes que se pueden extraer. La primera es que, independientemente del número de veces que haya aplicado la operación de suma sobre el texto cifrado, el tamaño de este no varía en ninguna de las tres plataformas, y además, es el mismo que el de un CT al que no se le haya aplicado ninguna operación, es decir, que la operación de suma no afecta al tamaño de los textos cifrados.

La segunda conclusión que se puede extraer es que, cuando se multiplica un CT, el tamaño sí varía, como se puede apreciar en la Figura 6, aunque en este caso se ha valorado sobre una única iteración, ya que el producto en un contexto homomórfico tiene varias limitaciones de escalabilidad. En LATTIGO el peso aumenta un 33 % y en SEAL un 50 %, no obstante, en PALISADE el tamaño resultante tras aplicar la operación *EvalMult* no se ve afectado sino que se mantiene en 259KB.

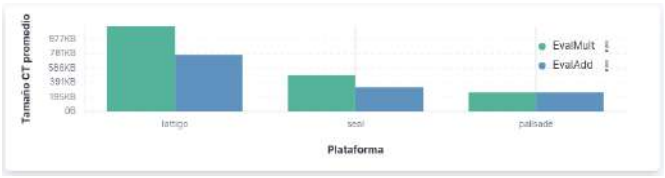


Figura 6. Tamaño del CT promedio después de operar.

IV-C. Precisión

Para terminar con esta sección, se va a estudiar la precisión de los elementos descifrados comparándolos con los originales, o en su defecto, los valores que realmente tendrían que haberse obtenido si las operaciones fueran exactas.

En la Figura 7 se muestra una gráfica que contiene todas las muestras de desviación recogidas en los descifrados, donde se puede intuir que PALISADE es la herramienta que menos error comete de las tres analizadas, LATTIGO, por el contrario, es la que comete errores de mayor nivel. Se debe tener en cuenta, además, que en la gráfica cada intervalo del eje horizontal disminuye exponencialmente, por lo que el error que comete LATTIGO se puede considerar mucho mayor que los de los otros dos frameworks.

	Nº Registros	Promedio	Promedio EvalAdd	Promedio EvalMult
lattigo	12,602	0.00213574	0.26240917	0.0000071
palisade	12,602	1.4577849952886558e-11	7.271529012770824e-10	1.3251539146379165e-12
seal	12,602	1.1902708490302512e-9	5.816705860763621e-8	9.509652129785496e-11

Figura 7. Distribución de errores.

Entre los factores que afectan a la precisión que se mencionaban anteriormente está el número de iteraciones que se realiza sobre un texto cifrado. Como se observa en la tabla de la Figura 7, el error cometido al realizar un producto es inferior al de la suma, sin embargo, este es un resultado preliminar que necesita un análisis más profundo, ya que la función *EvalAdd* se aplica múltiples veces sobre los textos cifrados, mientras que *EvalMult* se opera una única vez, debido a las limitaciones de escalabilidad existentes. En la Figura 8 se puede apreciar cómo afecta el número de iteraciones de una suma sobre el aumento del error obtenido en el resultado. Se puede deducir que el error en la multiplicación podría aumentar con el número de iteraciones igualmente, y es por eso que es necesario profundizar en este estudio.

Otro de los factores detectado en LATTIGO que influye a la desviación del descifrado, y que también afecta en el significativo error que obtiene este framework en la tabla de la Figura 7, es la posición en la que se encuentre el valor en el anillo del que se obtiene el texto cifrado, y es que, a medida que aumenta la posición en el array de valores, también se atenúa el error recibido. En la Figura 9 es posible ver un ejemplo de los errores obtenidos en un registro donde se ha

Figura 8. Incremento del error en función del número de iteraciones.

operado 10.000 veces la función *EvalAdd* sobre un anillo con 172 elementos.

deviation.samples

```
9.507, 8.952, 1.295, 0.21, 0.004, 0.01, 0.024, 0.001, 0.003, 0.004, 0.006, 0.0
07, 0.008, 0.004, 0.023, 0.006, 0.008, 0.001, 0.016, 0.007, 0.001, 0.002, 0.00
2, 0.003, 0.003, 0.008, 0.003, 0.003, 0, 0.021, 0.011, 0.003, 0.001, 0.004, 0.
005, 0.003, 0.009, 0.004, 0.001, 0.001, 0.013, 0.004, 0.005, 0.007, 0.01, 0.00
1, 0.004, 0.007, 0.029, 0.02, 0, 0, 0.003, 0, 0.008, 0.007, 0.001, 0.052, 0.00
5, 0.001, 0.007, 0.031, 0.001, 0.004, 0.002, 0.005, 0.001, 0.004, 0.004, 0.00
1, 0.005, 0.001, 0.001, 0.002, 0.002, 0.002, 0.008, 0.086, 0.014, 0.003, 0.00
```

Figura 9. Ejemplo de los errores cometidos en un descifrado.

Existen varios factores que pueden afectar al error obtenido cuando se descifra un valor, y varían dependiendo de la plataforma que se esté utilizando. En este artículo se ha hecho un análisis preliminar teniendo en cuenta estos factores, pero se está trabajando para acotarlos lo máximo posible, y poder comprender cómo afectan en cada caso.

V. Conclusiones

Un resumen de las conclusiones obtenidas a partir de los resultados podría ser:

- Los **tiempos de SEAL, por lo general, son los mejores**. En cifrado PALISADE también tiene un promedio muy bajo, pero puede sufrir desviaciones muy altas. Y LATTIGO es muy similar a SEAL a la hora de evaluar y descifrar, pero su promedio de cifrado es inferior.
- **PALISADE tiene textos cifrados más ligeros**, casi tres veces más pequeños que LATTIGO, que es el más pesado. Además, el tamaño del CT resultante de la multiplicación no aumenta en PALISADE, a diferencia de los de sus competidores. También se ha podido observar que el peso del CT es independiente del número de valores que se introduzcan en el anillo, pero sí le afecta si se modifica la dimensión de este.
- En cuanto al error acumulado tras el descifrado, **PALISADE es el más preciso**, seguido muy de cerca por SEAL. LATTIGO, por el contrario, tiene una imprecisión bastante importante que a la larga puede afectar a los resultados.

VI. Lineas futuras

Este trabajo de investigación se continuará en el futuro para profundizar en el conocimiento de las implementaciones de los sistemas criptográficos homomórficos actuales, permitiendo así optimizar su uso según los requisitos de diferentes casos de uso que requieran confidencialidad de la información, y para ello, se han propuesto una serie de metas:

- Tomar otras medidas que se consideran relevantes para el uso de un sistema FHE, como métricas de consumo computacional o la energía consumida.
- Introducir otras alternativas para implementar FHE como puede ser HELib.
- Analizar otros esquemas de cifrado homomórficos como BGV y TFHE.
- Comprobar qué pérdidas de eficacia existen frente a las operaciones en claro tradicionales.
- Analizar flujos de ejecución más complejos a medida que se vayan implementando en las distintas plataformas.
- Profundizar en el estudio del error cometido en función del número de multiplicaciones que se apliquen sobre un CT, del que únicamente se han obtenido resultados preliminares con una única iteración.

Agradecimientos

Este trabajo ha sido parcialmente financiado por Izertis a través del proyecto SECUREWORLD, cofinanciado por el Centro para el Desarrollo Tecnológico Industrial (CDTI) en el marco del programa estratégico CIEN (nº de expediente: CIEN-20200001).

Referencias

[1] N. Kshetri, "Big data's impact on privacy, security and consumer welfare," *Telecommunications Policy*, vol. 38, no. 11, pp. 1134–1145, 2014.
[2] X. Yi, R. Paulet, and E. Bertino, "Homomorphic encryption," in *Homomorphic encryption and applications*. Springer, 2014, pp. 27–46.
[3] M. Alloghani, M. M. Alani, D. Al-Jumeily, T. Baker, J. Mustafina, A. Hussain, and A. J. Aljaaf, "A systematic review on the status and progress of homomorphic encryption technologies," *Journal of Information Security and Applications*, vol. 48, p. 102362, 2019.
[4] "Microsoft SEAL (release 4.0)," Online: https://github.com/Microsoft/SEAL, Microsoft Research, Redmond, WA., Mar. 2022.
[5] "PALISADE (release 1.11.6)," Online: https://gitlab.com/palisade/palisade-release, PALISADE Team, Jan. 2022.
[6] "LATTIGO (release v3.0.1)," Online: https://github.com/tuneinsight/lattigo, EPFL-LDS, Tune Insight SA, Feb. 2022.
[7] B. Rossman, "Homomorphism preservation theorems," *Journal of the ACM (JACM)*, vol. 55, no. 3, pp. 1–53, 2008.
[8] R. L. Rivest, L. Adleman, M. L. Dertouzos *et al.*, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.
[9] C. Gentry, *A fully homomorphic encryption scheme*. Stanford university, 2009.
[10] ——, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.
[11] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2017, pp. 409–437.
[12] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, "A full rns variant of approximate homomorphic encryption," in *International Conference on Selected Areas in Cryptography*. Springer, 2018, pp. 347–368.
[13] E.-Y. Ahmed and M. D. Elkettani, "Fully homomorphic encryption: state of art and comparison," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 14, no. 4, 2016.
[14] C. Aguilar Melchor, M.-O. Kilijian, C. Lefebvre, and T. Ricosset, "A comparison of the homomorphic encryption libraries helib, seal and fv-nfllib," in *International Conference on Security for Information Technology and Communications*. Springer, 2018, pp. 425–442.
[15] S. M. Fawaz, N. Belal, A. ElRefaey, and M. W. Fakhr, "A comparative study of homomorphic encryption schemes using microsoft seal," in *Journal of Physics: Conference Series*, vol. 2128, no. 1. IOP Publishing, 2021, p. 012021.
[16] G. Đorđević, "Performance comparison of homomorphic encryption scheme implementations," in *International Conference on Electrical, Electronic and Computing Engineering (IcETRAN)*, 2021.
[17] M. Barcanau and V. Pasol, "Ring homomorphic encryption schemes," *Cryptology ePrint Archive*, 2018.
[18] "Intrusion Detection Evaluation Dataset (CIC-IDS2017)," Online: https://www.unb.ca/cic/datasets/ids-2017.html.
[19] "Elastic (release 7.16.3)," Online: https://www.elastic.co, Elastic NV, Jan. 2022.

Experiments on modeling and verification of post-quantum protocols using Maude

Víctor García
VRAIN, Universitat Politècnica de València
Camí de Vera, Valencia, Spain
vicgarv2@upv.es

Santiago Escobar
VRAIN, Universitat Politècnica de València
Camí de Vera, Valencia, Spain
sescobar@upv.es

***Abstract*—Current state of the art in quantum information processing and quantum computing sooner or later will challenge today's cryptography protocols, algorithms and cryptosystems. Many classical cryptography protocols are based on computational problems hard to solve for classical algorithms, but most of these problems can be solved easily using algorithms on quantum computers. This document presents the intentions on the formal specification and verification of properties for some post-quantum cryptography protocols. We present and explain the specification of the CRYSTALS-KYBER protocol, which we aim to build a model using the Maude rewriting language. Maude is a high-performance programming, modeling and verification language that has been used in countless systems, from communication protocols to real-time systems, including cyber-physical, hybrid and embedded systems. Using the specification, we also pretend to identify necessary properties.**

***Index Terms*—Maude, Rewriting logic, Post-quantum protocols, Key Encapsulation Mechanisms, Model Checking**

Type of contribution: *Research in development*

I. Introduction

Research on the quantum field in the past years has been active, proposing new algorithms that could put in danger the security principles of current cryptosystems and cryptography protocols. Most of the security of the protocols we use today relay on three types of problems that are hard to solve under classic computing: the integer factorization problem, the discrete logarithm problem and the elliptic-curve discrete logarithm problem. Some of the most popular asymmetric (or public key) algorithms, which relay on integer factorization, will become insecure under quantum computers using Shor's algorithm [1]. Another example is the Grover's search algorithm [2] for unstructured databases, which in principle one could ask what does it has to do with cryptography. It has been shown in that same paper that this algorithm makes possible to reduce the complexity of the integer factorization problem to a quadratic cost.

In order to face the threat quantum advances could make to the principles of today's security schemes some actions and solutions are being proposed. One of the main contributors to this cause is the National Institute of Standards and Technology (NIST). On 2017 NIST started a project called Post-Quantum Cryptography[1] to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptography algorithms. The proyect started with 69 candidate algorithms that met both the minimum acceptance criteria and submission requirements. As of 2022, the project is on round 3 and the third-round finalist on public-key encryption and key-establishment algorithms are Classic McEliece [3], CRYSTALS-KYBER [4], NTRU [5], and SABER [6]. The third-round finalists for digital signatures are CRYSTALS-DILITHIUM [7], FALCON [8], and Rainbow [9]. In this paper, we focus on public-key encryption and key-establishment algorithms because some attacks have been found, such as Man-In-The-Middle or Meet-In-The-Middle Attacks.

Kyber, more specifically CRYSTALS-KYBER [4], is one of the finalists of the Post-Quantum Cryptography project proposed by NIST. The chosen version for this paper is 3.01, released on the 21st of January in 2021 as a response to the recommendations and comments received during round 2. Kyber specifically is one of several submitted Key Encapsulation Mechanisms (KEM) to the NIST program. These KEM main goal is to securely share a given key between two participants of a network where channels are not safe from intruders. Such goal is interesting for conventional cryptography, also known as Symmetric Cryptosystem, which uses a secret key to encrypt a message. The same key is required to decrypt the encrypted message. Therefore, the sender and the receiver of an encrypted message either had to agree on the secret key before encrypting the message, or the receiver had to learn the secret key before decrypting it. The same secret key could not be used when sending a message to a different receiver.

When we talk about the analysis of security systems and protocols there are two kinds of approaches that can be taken: computational security and symbolic security. The former is based on mathematical proofs on a computational model, where messages are bitstrings and the adversary is any probabilistic Turing machine. Length of keys are determined by a value called security parameter, and the adversary runtime is supposed to be polynomial under such security parameter. This is the approach generally used by cryptographers and the authors of Kyber have already covered this approach. The later is based on the use of symbols, where the cryptography primitives are function symbols acting as black boxes. Messages of symbolic models are terms on these primitives and the adversary is restricted to use the defined primitives. It is important to note that these models assume perfect cryptography, i.e., ciphertexts cannot be broken without the proper key. Despite the fact that the computational model is closer to reality, it complicates the proofs and is hard to understand for non experts of cryptography. On the other hand, symbolic models are suitable for automation and easier to understand, so in this paper, our experiments belong to the later. It is important to mention that this approach not only can be applied to the selected protocol but also to any other

[1] https://csrc.nist.gov/projects/post-quantum-cryptography

scheme or mechanism on the round 3 of the Post-Quantum Cryptography project.

The chosen technology to achieve symbolic security is the language Maude [10]. It is a reflective language and specification tool used to represent many kinds of systems. It supports equational and rewriting logic specification and programming, providing the required tools to specify any kind of syntax that could represent a target system. Maude also has a model checker that uses LTL formulas, and a reachability analyzer with the search command. Maude inherits many features from the language known as OBJ3 [11]. Maude also provides us with a flexible language able to construct a symbolic model and the necessary tools to reason on the model.

Related work: Current advances in security of protocol analysis have been made. One interesting idea is the one proposed at [12] where the author explains several examples on the formal specification of protocols, where the symbolic and computational models are explained. One protocol analysis tool is Maude-NPA [13], related to the programming language Maude [10]. Maude-NPA has a theoretical basis on rewriting logic, unification and narrowing, and performs backwards search from a final attack state to determine whether or not it is reachable from an initial state. Other tools such as ProVerif [14] are based on an abstract representation of a protocol using Horn clauses and the verification of security properties is done by reasoning on these representative clauses. Another tools such as Tamarin [15] are based on constraint solving to perform an exhaustive, symbolic search for executions traces. And other tools such as Scyther [16] or CPSA [17] attempt to enumerate all the essential parts of the different possible executions of a protocol.

We have provided a first symbolic security analysis of Kyber in [18] using Maude, where a man in the middle (MITM) attack is found. But in this paper we attempt to provide a higher level or more abstract model of Kyber and a symbolic analysis. A more abstract model is needed in order to better understand and reason on KEMs for non experts of cryptography. The symbolic analysis aims to demonstrate not only that a MITM attack is present, but that our model considers all protocol's possible behaviours. In [19], we have used the method of [18] for the KEM known as SABER, a close relative of Kyber. It may be of interest to apply the same methodology in this paper to SABER.

Roadmap: The paper is structured as follows: Section II expands the previous introduction about Maude, explaining key concepts to understand the language selected. Next, a first overview explanation of Kyber KEM and a more detailed insight is given in Section III. Finally, Section IV contains final thoughts and proposed work to be done, including the next steps towards our main goal.

II. MAUDE

In this section we make a brief introduction to the Maude language and the features we have used in this preliminary work.

Maude is based on rewriting logic, a logic ideally suited to specify and execute computational systems in a simple and natural way. Since nowadays most computational systems are concurrent, rewriting logic is particularly well suited to specify concurrent systems without making any a priori commitments about the model of concurrency in question, which can be synchronous or asynchronous, and can vary widely in its shape and nature: from a Petri net [20] to a process calculus [21], from an object-based system [22] to asynchronous hardware [23], from a mobile ad hoc network protocol [24] to a cloud-based storage system [25], from a web browser [26] to a programming language with threads [27], or from a distributed control system [28] to a model of mammalian cell pathways [29], [30]. And all *without any encoding*: what you see and get is a direct definition of the system itself, without any artificial encoding.

Maude is based on rewriting logic but rewriting logic has a sublogic, called membership equational logic, for those deterministic or functional parts of a system. That is, an equational program is a functional program in which a functional expression (called a term) is evaluated using the equations in the program as left-to-right rewrite rules, which are assumed confluent and terminating. If such an evaluation terminates, it returns a unique computed value (determinism), namely, its normal form after simplifying it with the (oriented) equations.

In contrast, Maude system modules represent concurrent systems as conditional rewrite theories that model a non-deterministic system which may never terminate and where the notion of a computed value may be meaningless. In this concurrent system, the membership equational subtheory defines the states of such a system as the elements of an algebraic data type, for example terms in an equivalence class associated to cryptographic properties. We can call this aspect the static part of the specification. Instead, its dynamics, i.e., how states evolve, is described by the transition rules, which specify the possible local concurrent transitions of the system thus specified. The system's concurrency is naturally modeled by the fact that in a given state several transition rules may be applied concurrently to different subparts, producing several concurrent local state changes, and that rewriting logic itself models those concurrent transitions as logical deductions [31].

The most basic form of system analysis, in the form of explicit-state model checking, is illustrated by the use of the `search` command in Maude that performs reachability analysis from an initial state to a target state. Reachability can be used to both verify invariants or to find violations of invariants in the following sense. We can search for a violation of an invariant. If the invariant fails to hold, it will do so for some finite sequence of transitions from the initial state, and this will be uncovered by the above `search` command since all reachable sates are explored in a breadth-first manner. If, instead, the invariant does hold, we may be lucky and have a finite state system, in which case the `search` command will report failure to find a violation of the invariant. But if there is an infinite number of states reachable from the initial state, search will never terminate. A standard approach is to bound the analysis, e.g. by specifying a depth bound for the search command. Another approach is to perform an over or under approximation of the search space, so that the system becomes finite-state and the invariant can be verified.

Under the assumption that the set of states reachable from an initial state is finite, Maude also supports explicit-state model checking verification of any properties in linear time

temporal logic (LTL) through its LTL model checker.

III. Kyber

Kyber is an IND-CCA2-secure key encapsulation mechanism (KEM), whose security is based on the hardness of solving the learning-with-errors (LWE) problem over module lattices. To explain Kyber, first we give a simple explanation of the basic message exchange of Kyber assuming two participants Alice and Bob, following standard reasons. Then we will do a brief introduction to the specific algorithms and later explain an abstraction of them easy to understand. Finally, all technical information of the protocol is omitted because such details are beyond the scope of the paper. All KEM follow the same behaviour:

$$\begin{aligned} Alice &\rightarrow Bob : pk \\ Alice &\leftarrow Bob : c \end{aligned} \tag{1}$$

The first interaction between participants is Alice sending the public key to Bob, as can be seen in *Eq. (1)* where the message carries the public key generated by Alice. Then there is a second transaction, also visible in *Eq. (1)*, where Bob sends to Alice the ciphered text constructed using the previously received public key and a message M generated by himself. As the reader may have guessed the complexity of the network is as simple as it can get. The mechanism only requires two messages for two participants, for example, to establish a shared key between them.

Now let's take a look on a more formal specification. Figures 1 and 2 were extracted from [18] which are an abstraction of the algorithms exposed in [4]. Fig. 1 shows the intrinsic operations that happen on the previous transactions and participants. Operations KeyGen and Dec occur on Alice's side, while operation Enc occurs on Bob's side. These algorithms can be represented as black boxes as the *Eq. (2)* shows, each one with its corresponding inputs and outputs.

$$\begin{aligned} KeyGen() &\rightarrow (pk, sk) \\ Enc(pk) &\rightarrow c \\ Dec(c) &\rightarrow k \end{aligned} \tag{2}$$

Moreover, Fig. 2 specifies a high level representation of the operations inside the previous algorithms *CPAPKE.KeyGen, CPAPKE.Enc* and *CPAPKE.Dec*. The procedure is as follows: First a participant, in this case Alice because tradition states it, performs the operation *KeyGen()* that outputs a pair of keys. The former, known as the public key *pk*, will be sent to any other participant in the network, making it public so any other participant knows it. The later, known only by the participant that generates it, is the secret key. Once the target participant, Bob, receives the message containing the public key of the other participant it performs operation *Enc(pk)*. The operation returns a ciphered text c which contains a message (a key k for example) that is then sent to the origin of the first message, Alice in this case. Once Alice receives the message from Bob, she performs the operation *Dec(c)*. This operation reverts the transformations that originated c to obtain the original message or content, in our case the shared key k. Then Alice and Bob have agreed on a shared key in order to perform symmetric encryption. It is important to mention that other operations take place such as *generate*, *sampleCBD*, *Compress*, *Decompress*, *encode* and *decode*. These are operations necessary for the main three operations we have described before but are not explained in this paper because of space limitations (see [4] for details).

IV. Conclusion

In this short abstract we presented our intentions on doing a more detailed approach on the specification and verification of properties for public key cryptography schemes. We are currently constructing a model in Maude that can represent the KEM Kyber beyond [18]. Later on we will try to apply model checking and reachability analysis to verify properties on the model. Some of the properties we have in mind could confirm that the model is well constructed such as that only one key is shared between two participants. Another interesting property could be that a key cannot be stolen from a participant or intercepted from a message of the network. In future works we could even extend our model and use tools such as Maude-NPA to verify some properties hold for an unbounded number of sessions of the protocol. We could also extend this work by applying symbolic security analysis for the rest of candidates of the Post-Quantum project from NIST. Our main goal is to make a framework which can be used to model and verify this and other kinds of post-quantum protocols.

Acknowledgment

This paper has been partially supported by the grant RTI2018-094403-B-C32 funded by MCIN/AEI/10.13039/501100011033 and ERDF A way of making Europe, by the grant PROMETEO/2019/098 funded by Generalitat Valenciana, and by the grant PCI2020-120708-2 funded by MICIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR.

References

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134.

[2] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, "Applying grover's algorithm to aes: quantum resource estimates," in *Post-Quantum Cryptography*. Springer, 2016, pp. 29–43.

[3] T. Chou, C. Cid, S. UiB, J. Gilcher, T. Lange, V. Maram, R. Misoczki, R. Niederhagen, K. Paterson, and E. Persichetti, "Classic mceliece: conservative code-based cryptography, 10 october 2020," 2020.

[4] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-kyber algorithm specifications and supporting documentation," *NIST PQC Round*, vol. 2, no. 4, 2019.

[5] C. Chen, O. Danba, J. Hoffstein, A. Hülsing, J. Rijneveld, J. M. Schanck, P. Schwabe, W. Whyte, and Z. Zhang, "Algorithm specifications and supporting documentation," *Brown University and Onboard security company, Wilmington USA*, 2019.

[6] J.-P. D'Anvers, A. Karmakar, S. Sinha Roy, and F. Vercauteren, "Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure kem," in *International Conference on Cryptology in Africa*. Springer, 2018, pp. 282–305.

[7] L. Beckwith, D. T. Nguyen, and K. Gaj, "High-performance hardware implementation of crystals-dilithium," in *2021 International Conference on Field-Programmable Technology (ICFPT)*. IEEE, 2021, pp. 1–10.

[8] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, "Falcon: Fast-fourier lattice-based compact signatures over ntru," *Submission to the NIST's post-quantum cryptography standardization process*, vol. 36, no. 5, 2018.

[9] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *International conference on applied cryptography and network security*. Springer, 2005, pp. 164–175.

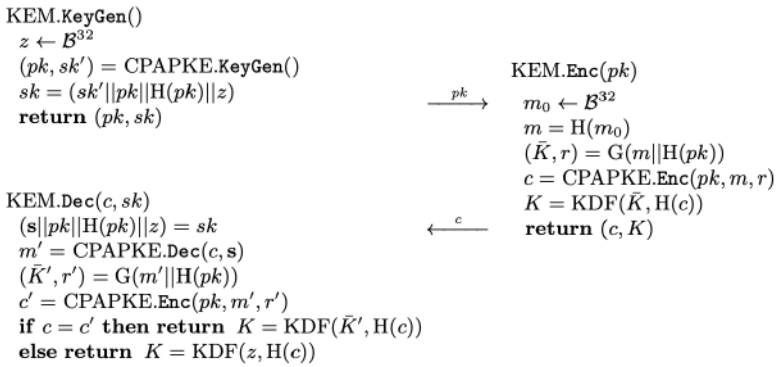


Figure 1. Algorithm of Kyber Key Encapsulation Mechanism.

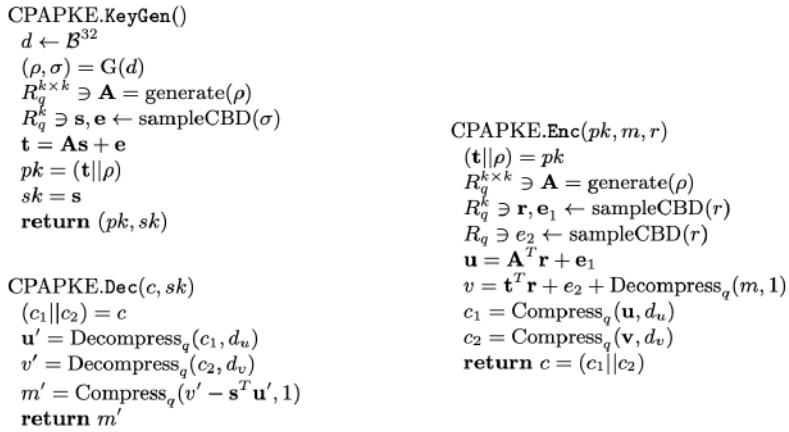


Figure 2. Internal algorithms of the Kyber Key Encapsulation Mechanism.

[10] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott, *All About Maude-A High-Performance Logical Framework: How to Specify, Program, and Verify Systems in Rewriting Logic*. Springer, 2007, vol. 4350.

[11] J. Goguen, C. Kirchner, H. Kirchner, A. Mégrelis, J. Meseguer, and T. Winkler, "An introduction to obj 3," in *International Workshop on Conditional Term Rewriting Systems*. Springer, 1987, pp. 258–263.

[12] B. Blanchet, "Security protocol verification: Symbolic and computational models," in *International Conference on Principles of Security and Trust*. Springer, 2012, pp. 3–29.

[13] S. Escobar, C. Meadows, and J. Meseguer, "Maude-npa: Cryptographic protocol analysis modulo equational properties," in *Foundations of Security Analysis and Design V*. Springer, 2009, pp. 1–50.

[14] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "Proverif 2.00: automatic cryptographic protocol verifier, user manual and tutorial," *Version from*, pp. 05–16, 2018.

[15] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The tamarin prover for the symbolic analysis of security protocols," in *International conference on computer aided verification*. Springer, 2013, pp. 696–701.

[16] C. J. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in *International conference on computer aided verification*. Springer, 2008, pp. 414–418.

[17] J. Ramsdell and J. Guttman, "CPSA4: A cryptographic protocol shapes analyzer," https://github.com/mitre/cpsaexp, The MITRE Corporation, 2018.

[18] D. D. Tran, K. Ogata, S. Escobar, S. Akleylek, and A. Otmani, "Formal specification and model checking of lattice-based key encapsulation mechanisms in maude," in *Rewriting Logic and its Applications 14th International Workshop, WRLA 2022*, p. 26.

[19] ——, "Formal specification and model checking of saber lattice-based key encapsulation mechanism in maude," in *Proceedings of the 34th International Conference on Software Engineering and Knowledge Engineering*, 2022, to appear.

[20] M.-O. Stehr, J. Meseguer, and P. C. Ölveczky, "Rewriting logic as a unifying framework for Petri nets," pp. 250–303.

[21] A. Verdejo and N. Martí-Oliet, "Implementing CCS in Maude," pp. 351–366.

[22] J. Meseguer, "A logical theory of concurrent objects and its realization in the Maude language," in *Research Directions in Concurrent Object-Oriented Programming*, G. Agha, P. Wegner, and A. Yonezawa, Eds. MIT Press, 1993, pp. 314–390.

[23] M. Katelman, S. Keller, and J. Meseguer, "Rewriting semantics of production rule sets," *Journal of Logic and Algebraic Programming*, vol. 81, no. 7-8, pp. 929–956, 2012.

[24] S. Liu, P. C. Ölveczky, and J. Meseguer, "Modeling and analyzing mobile ad hoc networks in Real-Time Maude," *Journal of Logical and Algebraic Methods in Programming*, 2015.

[25] R. Bobba, J. Grov, I. Gupta, S. Liu, J. Meseguer, P. Ölveczky, and S. Skeirik, "Design, formal modeling, and validation of cloud storage systems using Maude," in *Assured Cloud Computing*, R. H. Campbell, C. A. Kamhoua, and K. A. Kwiat, Eds. J. Wiley, 2018, ch. 2, pp. 10–48.

[26] S. Chen, J. Meseguer, R. Sasse, H. J. Wang, and Y.-M. Wang, "A systematic approach to uncover security flaws in GUI logic," pp. 71–85.

[27] J. Meseguer and G. Roşu, "The rewriting logic semantics project," *Theoretical Computer Science*, vol. 373, pp. 213–237, 2007.

[28] K. Bae, J. Meseguer, and P. C. Ölveczky, "Formal patterns for multirate distributed real-time systems," *Science of Computer Programming*, vol. 91, pp. 3–44, 2014.

[29] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and K. Sonmez, "Pathway logic: Symbolic analysis of biological signaling," pp. 400–412.

[30] C. Talcott, S. Eker, M. Knapp, P. Lincoln, and K. Laderoute, "Pathway logic modeling of protein functional domains in signal transduction," pp. 568–580.

[31] J. Meseguer, "Conditional rewriting logic as a unified model of concurrency," *Theoretical Computer Science*, vol. 96, no. 1, pp. 73–155, 1992.

About the FrodoKEM lattice-based algorithm

Miguel Ángel González de la Torre
Luis Hernández Encinas
Institute of Physical and Information Technologies (ITEFI)
Spanish National Research Council (CSIC)
{ma.gonzalez, luis.h.encinas}@csic.es

Araceli Queiruga Dios
Department of Applied Mathematics
University of Salamanca
queirugadios@usal.es

***Abstract*—Lattice-based cryptography is one of the most promising areas in regards to public key cryptosystems and key encapsulation mechanisms in the post-quantum era. Once the National Institute of Standards and Technology establishes new post-quantum standards, it would be of interest to study if it is possible to define, propose, and implement new algorithms, based on such standards, to be applied in several specific environments. In this sense, we analyze lattice-based algorithms in order to modify some of them with this mentioned objective. In particular, in our ongoing research we pretend to study the FrodoKEM proposal as one of the most promising candidates from which to derive new lattice-based algorithms.**

***Index Terms*—Latticed-based cryptography, Post-quantum cryptography, FrodoKEM**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

The advance in quantum computation is every day closer to develop functional quantum computers, which represents a threat for current cryptography, both symmetric and asymmetric. Shor's algorithm [1] will allow quantum computers to break the public key encryption schemes whose security is based on the integer factorization or discrete logarithm problems. This threat also affects to symmetric encryption because the use of Grover's [2] algorithm supposes that the key length should be, at least, twice the length that is today to achieve the same security level. Moreover, a quantum algorithm which uses Simon's subroutines [3] in a novel way was presented in [4] and in [5], which seems to support the idea that symmetric cryptography is more threatened by quantum computing than it might initially seem.

To address this situation, the National Institute of Standards and Technology (NIST) started in 2016 an international call [6] to determine new standard algorithms for the so called post-quantum cryptography (PQC), i.e., to propose sufficiently secure algorithms to resist the attacks from quantum computers. This process is being developed in several rounds. The first round received 69 submissions, but only 15 of them have reached the current third round.

In fact, in July 2020, NIST published the list of algorithms that having passed the second round, now constitute the list of candidates that are part of the third round [7]. The report justifying the reasons why some candidates have not passed the second round has been made public in [8]. This report includes both the list of candidates that are considered the finalists and pass to the third round and a list of alternative algorithms.

The goal of the NIST PQC standardization call is to establish several algorithms as viable encryption schemes which be invulnerable to the power of quantum computing; however today nobody knows exactly the capabilities of the future quantum computers or what kind of algorithm might be developed for breaking the new cryptography.

It seems that the idea of NIST is to standardize efficient schemes in practice, belonging to different families of cryptosystems, in such a way that the significant advance in the cryptanalysis in one of these families could lead to the abandonment of one of the chosen standards and still having a viable secure option. This is the reason for which NIST have chosen one group of finalists and another of alternative candidates. In addition, NIST will not choose all these standards simultaneously but will prioritize the proposals that may be closer to standardization.

The algorithms presented to this call can be categorized into five areas, depending on the underlying mathematical problem that provides the security of the algorithm. These areas are:

- Lattice-based cryptography.
- Code-based cryptography.
- Hash-based cryptography.
- Multivariate-based cryptography.
- Elliptic curve isogeny-based cryptography.

The NIST PQC call affects to two categories of cryptographic algorithms: on one side, public key encryption schemes and key encapsulation mechanisms (PKE/KEM), and on the other, public key signature schemes. NIST mentioned in its call that it is possible that in both cases more than one scheme would be considered as standard.

In our research, we will focus our attention in the PKE/KEM category and, in particular, in the lattice-based proposals, for both finalists and alternatives. More specifically, in our ongoing research we pretend to study in deep the FrodoKEM proposal as one of the most promising candidates from which to derive new algorithms. In other words, we would like to determine the needed conditions in order to define and implement post-quantum algorithms, based on FrodoKEM, for devices with restricted computational power and other more general IoT devices.

The rest of this paper is organized as follows. In section II the notation and some of the main related works about the NIST call are mentioned. Section III presents a short description of the FrodoKEM proposal and other lattice-based algorithms, and finally, in section IV we comment the results that we expect to obtain in our research.

II. Related works and Notation

A. NIST PQC call

The criteria that NIST has followed during the elimination process (the three rounds) have been specified in [6], consider-

ing that security is the most important factor in the evaluation process.

In this sense, it is important to recall that INDistinguishability under Chosen Plaintext Attack (IND-CPA) security is defined as follows: given an asymmetric encryption of one of two attacker-chosen plaintexts and the public key, the attacker cannot determine which of the messages was encrypted. This definition applies to mechanisms when one-time public keys are used. Moreover, in the INDistinguishability under Adaptive Chosen Ciphertext Attack (IND-CCA2) security, it is considered that the attacker fails under the conditions for IND-CPA with also access to a decryption oracle. The oracle can be used on all ciphertexts but the challenge ciphertext.

The semantic security standard for PKE/KEM in the NIST call is set as IND-CCA2. Moreover, against brute force attacks there are 5 levels of security based on the difficulty of breaking certain parameters sets of AES (levels 1, 3 and 5) or find a collision for a 256 or 384-bit hash functions (levels 2 and 4 respectively).

The other criteria that are involved in the NIST selection process are cost and performance, and algorithm and implementation characteristics. In fact, in the second round, cost and performance was crucial, since the security of a considerable group of candidates was similar. The third round of the call is still ongoing but is expected to be resolved in 2022.

The NIST algorithms have been objective of a deep and meticulous study and there are many works and publications about them. Along the different rounds in the standardization process, the designers of each algorithm introduced improvements in their submissions. For example, one of the last additions in FrodoKEM is a new implementation of the matrix multiplication, based on the results of [9]. Another example is the submission for NTRU; initially two different submissions based on a similar background were submitted and now there is just one, which also improves the PKE/KEM transformation using results in [10].

Lattice-based cryptography seems like the best option for a PKE/KEM standard when it comes to post-quantum encryption. There are several reasons to choose lattice-based cryptography instead of one of the other areas. Most of the finalists and alternative proposals in the third round are based on lattices, hence, one can infer that in both security and performance lattice-based algorithms are among the best choices. Moreover, some lattice-based problems have been known for a long time which means that the hardness of these problems is well studied; however not all the lattice-based problems have this background, which encourages the study of all the lattice-based algorithms, not just the most efficient ones.

B. Problems in the lattice-based cryptography

In this work, a probabilistic public key encryption will be denoted as the set $\text{PKE} = \{\text{KeyGen}, \text{Enc}, \text{Dec}, M, C\}$, where KeyGen, Enc, and Dec are the key generation, the encryption, and the decryption algorithms, respectively. The set M is the set of possible messages and C is an optional randomness set. The set M can be omitted to reduce the notation, and when the PKE is deterministic, C is not considered. Moreover, we denote by $\text{KEM} = \{\text{KeyGen}, \text{Encaps}, \text{Decaps}\}$ a generic key encapsulation mechanism, where KeyGen, Encaps, and Decaps are the key generation, the encapsulation, and the decapsulation algorithms, respectively.

If we denote by $x \leftarrow_R X$ that the element x is chosen uniformly at random from the set X, the Learning With Errors (LWE) problem can be stated as follows: given pairs $(\mathbf{a}_i, b_i)$, such that $\mathbf{a}_i \leftarrow_R \mathbb{Z}_q^n$ and $b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i$, where $e_i \leftarrow_R \mathbb{Z}_q$ is an error, the goal is to find the secret vector $\mathbf{s} \in \mathbb{Z}_q^n$. The idea is to determine the vector $\mathbf{s}$ from several samples like the following ones:

$$\begin{aligned} \mathbf{a}_1 &\in \mathbb{Z}_q^n, \quad b_1 = \langle \mathbf{s}, \mathbf{a}_1 \rangle + e_1, \\ \mathbf{a}_2 &\in \mathbb{Z}_q^n, \quad b_2 = \langle \mathbf{s}, \mathbf{a}_2 \rangle + e_2, \\ &\vdots \\ \mathbf{a}_r &\in \mathbb{Z}_q^n, \quad b_r = \langle \mathbf{s}, \mathbf{a}_r \rangle + e_r. \end{aligned}$$

If the error e_i is not added to the inner product of $\mathbf{s}$ and $\mathbf{a}_i$, then $\mathbf{s}$ can be recovered efficiently by the Gaussian elimination method in the expression $\mathbf{b} = A\mathbf{s}$, where A is the matrix of vectors $\mathbf{a}_i$.

If we consider the ring defined as $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$, the Ring Learning With Errors (RLWE) problem is the same problem defined as before but now $(\mathbf{a}_i, \mathbf{b}) \in \mathcal{R}_q \times \mathcal{R}_q$. The ring structure of $\mathcal{R}_q$ allows the computation to be simpler and the public key to be smaller; however it also supposes a possible higher vulnerability. Moreover, the Module Learning With Errors (MLWE) problem is analogous to the previous one but considering a structure of module instead of that of a ring. Finally, the Module Learning With Rounding (MLWR) problem is a variant of the MLWE where the addition of small error terms is replaced by rounding from one modulus to a smaller second modulus.

C. Research rationale

As we have mentioned before, in our research we are interested in lattice-based finalists and, specially, in the FrodoKEM proposal [11], which is a lattice-based algorithm considered as alternative in the third round of the NIST call. In fact, FrodoKEM presents a more reliable security than other lattice-based algorithms. Moreover, in the consideration of FrodoKEM as an alternative proposal, the NIST evaluation stated that in case that the lattice-based finalists present irreparable weakness, FrodoKEM is the more suitable replacement as lattice-based post-quantum standard. Finally, we have also considered that the Bundesamt für Sicherheit in der Informationstechnik (BSI) maintains its recommendation of FrodoKEM as a PQC mechanism with a high security margin against future attacks. BSI considers that FrodoKEM has not been included among the finalists for the third round of the NIST PQC call due to considerations of the efficiency of the mechanism, but there are currently no doubts about its security [12].

The main reason why FrodoKEM is not a finalist is that compared with the other candidates it has a higher computational cost. However, this algorithm was deliberately designed from a conservative standpoint. The security of the public key encryption associated to Frodo, named FrodoPKE, relays in the LWE problem, which is a non structured classical lattice problem. Previously, in §II-B we have presented other versions of this problem that use an algebraic structure in

the lattice, providing a better performance (less computational cost and memory space is needed). In particular, in the last round of NIST, it seems like MLWE problem is the version of the problem that presents a better performance maintaining a high security. Today there are not known weaknesses of the MLWE problem that can not be also applied to LWE; however the study of these problems is still on an early stage. FrodoKEM design prioritizes the security of the system over its functionality and, while this might have been the main reason why it is now an alternative instead of a finalist, there is a strong argument in favor of the approach of our research.

III. FrodoKEM and other lattice-based algorithms

In this section, we will describe the main characteristics of the FrodoKEM algorithm and we will compare it with the finalists of the NIST call also based on lattices: Kyber, NTRU, and SABER.

A. FrodoKEM

First of all, we present a short scheme of how FrodoPKE works [11]. Tables I–III show the KeyGen, the Enc, and the Dec algorithms for FrodoPKE, respectively.

The public and private keys, (pk, sk), are generated by the FrodoPKE.KeyGen algorithm such that $pk = (seed_A, B)$, where $B = AS + E$, and $sk = S^T$ (^{T} denotes transposition). The matrix A is pseudorandomly generated by the Frodo.Gen algorithm which uses as input a seed, $seed_A$, and the dimension, n, of the matrix. In [11] two different instructions to generate A are recommended, one using AES128 and another using SHAKE128. Moreover, matrices S and E are obtained from the sample matrix algorithm, Frodo.SM, where r is a pseudorandom sequence of bits, $dims$ are the dimensions of the matrices, and T_χ is the distribution table for sampling (see Table I).

TABLE I
FrodoPKE key generation algorithm

FrodoPKE.KeyGen()
$seed_A \leftarrow_R \{0,1\}^{len_{seed_A}}$
$A \leftarrow$ Frodo.Gen$(seed_A, n), A \in \mathbb{Z}_q^{n \times n}$
$r \leftarrow_R C$
$S^T, E \leftarrow$ Frodo.SM$(r, dims, T_\chi)$
$B = AS + E$
return $(pk := (seed_A, B), sk := S^T)$

To encrypt a message $m \in M$, the algorithm FrodoPKE.Enc calculates $B' = S'A + E'$ and $V = S'B + E'' = S'(AS + E) + E'' = S'AS + S'E + E''$, where S'. E' and E'' are generated by the Frodo.SM algorithm. The ciphertext is defined as $c := (B', V +$ Frodo.Encode$(m))$, being Frodo.Encode a function for encoding bit sequences as matrices of integers (see Table II).

TABLE II
FrodoPKE encryption algorithm

FrodoPKE.Enc(pk, m, r)
$A \leftarrow$ Frodo.Gen$(seed_A)$
$S', E', E'' \leftarrow$ Frodo.SM $(r, dims, T_\chi)$
$B' = S'A + E'$
$V = S'B + E''$
return $c := (B', V +$ Frodo.Encode$(m))$

The decryption algorithm starts calculating $M = V +$ Frodo.Encode$(m) - B'S$ (see Table III). If we replace in this expression the definition of each term presented before, we have:

$$M = \text{Frodo.Encode}(m) + S'E - E'S + E''.$$

The last step consists in applying the function Frodo.Decode, which represents Frodo.Encode(m) as a bit string and cancels the error $S'E - E'S + E''$. The distribution of error is chosen in such a way that the decode function inverts encode and also withdraws the error of the final expression.

TABLE III
FrodoPKE decryption algorithm

FrodoPKE.Dec(sk, c)
$M = V +$ Frodo.Encode$(m) - B'S$
return $m' \leftarrow$ Frodo.Decode(M)

To construct FrodoKEM from FrodoPKE, a variant of the Fujisaky-Okamoto transformation $\left(\text{FO}^{\not\perp}\right)$, proposed in [13], is applied. This transformation assures that the resulting KEM is IND-CCA secure in a classical random oracle model (ROM) and moreover it has a security reduction in the quantum ROM (QROM).

A short description of the key generation, encapsulation and decapsulation algorithms of FrodoKEM is shown in Tables IV–VI, respectively, being G_1, G_2, and H hash functions.

TABLE IV
FrodoKEM key generation algorithm

FrodoKEM.KeyGen()
$(pk, sk) \leftarrow$ FrodoPKE.KeyGen()
$s \leftarrow_R \{0,1\}^{len_s}$
$pkh \leftarrow G_1(pk)$
$sk' := (sk, s, pk, pkh)$
return (pk, sk')

TABLE V
FrodoKEM encapsulation algorithm

FrodoKEM.Encaps(pk)
$m \leftarrow_R M$
$(r, k) \leftarrow G_2(G_1(pk) \| m)$
$c \leftarrow$ FrodoPKE.Enc$(pk, m; r)$
$K \leftarrow H(c \| k)$
return (c, K)

TABLE VI
FrodoKEM decapsulation algorithm

FrodoKEM.Decaps(sk', c)
Recall $sk' = (sk, s, pk, pkh)$
$m' \leftarrow$ FrodoPKE.Dec(sk, c)
$(r', k') \leftarrow G_2(pkh \| m')$
$K'_0 \leftarrow H(c \| k')$
$K'_1 \leftarrow H(c \| s)$
if $c =$ FrodoPKE.Enc$(pk, m'; r')$ then $K' \leftarrow K'_0$
else $K' \leftarrow K'_1$
return K'

B. Comparison with other lattice-based algorithms

The three finalists in the PKE/KEM area of the NIST PQC standardization call whose security is based in lattices are: CRYSTALS-Kyber [14], NTRU [15], and SABER [16]. Contrary to FrodoKEM, which is a non structured lattice-based

scheme, all finalists are structured lattice-based schemes, and moreover, there are important differences between them.

CRYSTALS-Kyber is a KEM whose security is based on the hardness of the MLWE problem. Among all the lattice-based algorithms, Kyber has one of the best performances and NIST considers that is ready for standardization. An advantage of Kyber is that it shares a common framework with CRYSTALS-Dilithium, a signature scheme that is also finalist in that category.

NTRU is another lattice-based KEM, based on the RLWE problem, that was known even before the standardization process started. The current submission of NTRU is the merger of NTRUEncrypt and NTRU-HRSS-KEM algorithms, it also incorporates the results published in [10]. These algorithms have a higher performance cost than the two other finalists, mostly in the key generation algorithm. Among the finalists, NTRU is the algorithm that has been developed for more time, this gives a high level of confidence in its security.

SABER differentiates from other lattice-based algorithms since its security is based in the MLWR problem. In this case, instead of adding a small error term, the session key is obtained by rounding to a smaller integer modulus. There is not a suitable security reduction for SABER right now; however the performance of the algorithm is very good and, together with Kyber, both are considered to become the post-quantum standards.

In Table VII the difference in size of pk, sk and c (in bytes) between the respective parameter sets (separated by /) in ascending order of security of FrodoKEM and the finalist lattice-based algorithms can be appreciated.

TABLE VII
PUBLIC AND SECRET KEYS AND CIPHERTEXT SIZE (BYTES) COMPARISON FOR SOME LATTICE-BASED ALGORITHMS

Bytes	FrodoKEM	Kyber	NTRU	SABER
pk	9616/15632 21520	800/1184 1568	699/930 1230/1138	672/992 1312
sk	19888/31296 43088	1632/2400 3168	935/1234 1590/1450	1568/2304 3040
c	9720/15744 21632	768/1088 1568	699/930 1230/1138	736/1088 1472

It can be appreciated that the keys and ciphertext sizes for FrodoKEM are bigger than the sizes of the other algorithms. The reason behind it is that FrodoKEM works with matrices while the others just with vectors.

IV. EXPECTED RESULTS

We are looking for several results in this ongoing research, but all of them are focused in designing and implementing different lattice-based algorithms adapted to fulfil a certain set of conditions. In a first instance, we are interested in obtaining new algorithms, based on the FrodoKEM, in order to obtain an equivalent level of security, but with a lower computational cost. To obtain a convenient performance it might be needed an adaptation of FrodoKEM, which is also an important part of the research. Ideally, we would like to establish the needed conditions in order to define and implement post-quantum algorithms for devices with restricted computational power and other more general IoT devices. With this purpose the first step is to deeply analyze, study, and compare FrodoKEM algorithm with the rest of the lattice-based proposals, including their implementations.

ACKNOWLEDGEMENTS

This work was supported in part by ORACLE Project, with reference PCI2020-120691-2, funded by MCIN/AEI/10.13039/501100011033 and European Union "NextGenerationEU/PRTR", in part by the Spanish State Research Agency (AEI) of the Ministry of Science and Innovation (MCIN), project P2QProMeTe (PID2020-112586RB-I00/AEI/10.13039/501100011033), and in part by the EU Horizon 2020 research and innovation programme, project SPIRS (Grant Agreement No. 952622).

REFERENCES

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999, https://doi.org/10.1137/S0036144598347011.

[2] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Physical Review Letters*, vol. 79, no. 2, pp. 325–328, 1997, https://doi.org/10.1103/PhysRevLett.79.325.

[3] D. Simon, "On the power of quantum computation," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1474–1483, 1997, https://doi.org/10.1137/S0097539796298637.

[4] X. Bonnetain, A. Hosoyamada, M. Naya-Plasencia, Y. Sasaki, and A. Schrottenloher, "Quantum attacks without superposition queries: the offline Simon's algorithm," in *Proc. International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology - ASIACRYPT 2019, Lecture Notes Comput. Sci.*, vol. 11291, 2019, pp. 552–583, https://doi.org/10.1007/978-3-030-34578-5_20.

[5] ——, "Quantum attacks without superposition queries: the offline Simon's algorithm," *arXiv*, 2020, https://arxiv.org/abs/2002.12439.

[6] NIST, "Post-quantum cryptography," On-line publication, 2017, https://csrc.nist.gov/Projects/Post-Quantum-Cryptography.

[7] ——, "PQC standardization process: Third round candidate announcement," Online publication, 2020, https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement.

[8] ——, "Status report on the second round of the NIST post-quantum cryptography standardization process," U.S. Department of Commerce, Report NISTIR 8309, Tech. Rep., 2020, https://doi.org/10.6028/NIST.IR.8309.

[9] J. W. Bos, M. Ofner, J. Renes, T. Schneider, and C. van Vredendaal, "The matrix reloaded: Multiplication strategies in FrodoKEM," Cryptology ePrint Archive, Report 2021/711, 2021, https://ia.cr/2021/711.

[10] T. Saito, K. Xagawa, and T. Yamakawa, "Tightly-secure key-encapsulation mechanism in the quantum random oracle model," in *Proc. Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT 2000, Lecture Notes Comput. Sci.*, vol. 10822, 2018, pp. 520–551, https://doi.org/10.1007/978-3-319-78372-7_17.

[11] E. Alkim, J. W. Bos, L. Ducas, P. Longa, I. Mironov, M. Naehrig, V. Nikolaenko, C. Peikert, A. Raghunathan, and D. Stebila, "FrodoKEM learning with errors key encapsulation (Round 3 Submission)," Online publication, 2021, https://frodokem.org/#spec.

[12] BSI, *Cryptographic Mechanisms: Recommendations and Key Lengths, version 2022-01*, Bundesamt für Sicherheit in der Informationstechnik, BSI TR-02102-1, 2022/01/28, 2022, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf.

[13] D. Hofheinz, K. Hövelmanns, and E. Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation," in *Proc. 15th International Conference Theory of Cryptography TCC'2017, Lecture Notes Comput. Sci.*, vol. 10677, 2017, pp. 341–371, https://doi.org/10.1007/978-3-319-70500-2_12.

[14] R. Avanzi, J. Bos, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-Kyber," Online publication, 2020, https://pq-crystals.org/.

[15] C. Chen, O. Danba, J. Hoffstein, A. Hulsing, J. Rijneveld, J. M. Schanck, P. Schwabe, W. Whyte, and Z. Zhang, "NTRU," Online publication, 2020, https://ntru.org/.

[16] A. Basso, J. M. B. Mera, J.-P. D'Anvers, A. Karmakar, S. Sinha, M. V. Beirendonck, and F. Vercauteren, "SABER: Mod-LWR based KEM (Round 3 Submission)," Online publication, 2020, https://www.esat.kuleuven.be/cosic/pqcrypto/saber/.

Posters

Sesión Poster 1: Investigación ya publicada I

Evaluation of Supervised Learning Models Using TCP Traffic for the Detection of Botnets

Felipe Castaño *†, Javier Velasco-Mata *†, Roberto A. Vasco-Carofilis *†,
Eduardo Fidalgo *†, Luis Fernandez †, George Azzopardi ‡

*Department of Electrical, Systems and Automation, Universidad de León, León, ES
†Researcher at INCIBE (Spanish National Cybersecurity Institute), León, ES
‡Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence, University of Groningen, Groningen, NL
Email:{felipe.castano, javier.velasco, andres.vasco, eduardo.fidalgo}@unileon.es,
luis.fernandez@incibe.es, g.azzopardi@rug.nl

***Abstract*—Botnets are used to accomplish different kinds of cybercrimes by attackers called *botmasters*, as large-scale credit card fraud, mass spam or DDoS, among others. In this work, we analyzed two popular methods to detect botnets: the internet traffic and the use of Sinkhole servers which try to deceive bots simulating a botnet controller. We have evaluated these methods with four machine learning algorithms, on two new datasets that we built, namely *TCP-Int*, containing internet traffic samples and *TCP-Sink*, containing network traffic samples from a sinkhole server. The results show that a decision tree (DT) based model achieves the highest performance with a 0.99 F1-score for the first dataset and 0.99 AUC for the second dataset proving that the best choice to build a botnet detection algorithm is DT.**

***Index Terms*—Cybersecurity, Botnets, Network Traffic, Sinkhole, TCP, Machine Learning**

Type of contribution: *Research already published [1]*

I. Introduction

Cybersecurity is, nowadays, a critical practice due to the massive growth of internet use. Protecting the users and their information is vital since they are vulnerable to attackers. For this reason, in recent years, several studies have been published that take cybersecurity as a central axis, and provide advances trying to make the internet a safer environment. As a result, important advances have been presented in several fields such as phishing detection, fraudulent e-commerce websites detection, spam detection, and bot detection [2], [3], [4].

A botnet is a network of previously infected computers by malware, called bots. Botnets are used to accomplish different kinds of cybercrimes as large-scale credit card fraud, mass spam, and Distributed Denial of Service (DDoS) by attackers called botmasters.

Two main approaches are the current state-of-the-art to detect botnets: traffic classifiers and sinkhole servers. For the former, botnets require to communicate with their botmaster to receive orders, and this process generates network traffic that can be used to identify them. Some solutions use a signature-based method [5] analyzing plain network communications and trying to detect malicious patterns. These solutions can, however, be bypassed using a mechanism like obfuscation and encryption. In order to solve this problem, machine learning algorithms have been investigated, classifying a network according to general traffic characteristics instead of specific signatures [6].

The latter method uses sinkhole servers, and it is designed to deceive botnets. A sinkhole simulates a real botnet controller and collects attempts of bots to communicate with its botmaster [7]. Given that sinkholes receive non-malicious connections from spiders and web crawlers too, it is necessary to use network traffic classifiers in this approach. This work explores TCP network classification of both regular and sinkhole communications.

II. Literature Review

Over the years, several machine learning (ML) algorithms have been proposed to solve network traffic classification problems without reaching a consensus on which is better. Sangkatsanee et al. [8] achieved an accuracy of 99% using a Shallow Decision Tree (DT) in a private dataset, outperforming other ML algorithms, like shallow neural networks and Bayesian networks. However, Kim et al. [9] proposed a method that uses a Support Vector Machine (SVM), and showed that it outperforms kNN, Naïve Bayes (NB), DT and shallow neural networks, reaching an accuracy of 97.8% in network anomaly detection.

III. Datasets

We built two new datasets using TCP network traffic. The first one integrates traces from TCP connections and is called TCP-Int. It contains information about the source and destination ports, mean and variance of payload, time intervals of sent packets, time intervals of response packets, packets exchanged per second, number of bytes exchanged, SYN flags, and exchanged packets. The samples are classified into four classes: normal, Kelihos, Miuref, and Sality. The dataset is balanced and contains 44231 samples per class. This data is publicly available thanks to Stratosphere Project [10].

A second balanced dataset, TCP-Sink, was built using network traffic from a sinkhole server. The samples were supplied by the Spanish National Cybersecurity Institute (INCIBE) and consists of 4027 samples divided into two classes, non-malicious, and malicious. TCP-Sink contains the same information per sample as TCP-Int.

IV. Classifiers and Experimentation

We implemented pipelines with four machine learning algorithms that are widely used in the state of the art of botnet detection [6], [8], [11]: DT, kNN, SVM, and NB. Figure 1 illustrates an overview of our methodology. As to evaluation we computed the mean F1-Score, receiver operating characteristic (ROC) curves, and area under the curve (AUC) across the 10-fold-cross-validation to compare the performance of the algorithms.

Fig. 1. Proposed methodology.

We designed the experiments with the following parameters. For DT, the maximum depth was set as unlimited. For k-NN, after an empirical evaluation, we decided to use the Euclidean distance in a 13-dimension space of the features and k=1 in the TCP-Int dataset and k=13 in the TCP-Sink dataset. For SVM, a linear kernel was used with a cost of C=1.0. Finally, for NB we used a Gaussian model as all independent variables are continuous.

As shown in Table I, DT achieves the best performance in the TCP-Int dataset with a 0.99 F1-score, followed by k-NN that also achieves high effectiveness. Moreover, SVM shows a poor performance especially for the Normal and Sality classes, indicating that the classes are not linearly separable.

TABLE I
Dataset TCP-Int: F1 scores by class

Class	DT	k-NN	SVM-linear	NB-Gauss
Normal	**0.99**	0.96	0.66	0.49
Kelihos	**1.00**	0.99	0.86	0.75
Miuref	**1.00**	0.98	0.83	0.51
Sality	**0.99**	0.97	0.64	0.51
Mean	**0.99**	0.97	0.74	0.56

For the TCP-Sink dataset, we used ROC curves and AUC scores since they provide more information for binary classification problems. Even in this case, DT and kNN achieve the best results with an AUC of 0.99 and 0.97, respectively, the ROC curves of which are shown in Figure 2.

V. Conclusions and Future Work

In this paper, we addressed the botnet detection problem. We built two datasets: TCP-Int containing regular internet communications of normal and botnets traffic, and TCP-Sink dataset containing traffic captured by a private sinkhole server gathered by INCIBE. Then, we tested four different ML algorithms, proving that DT and k-NN show the best performance in both datasets.

According to the results, the best choice for building a botnet detection on TCP traffic is DT. In the future, we will analyse the importance of all input features and consider feature selection techniques. Moreover, we aim to extend our data sets by more samples and more diversity and aim to experiment with more sophisticated techniques, including ensemble models like random forests, boosted trees. Finally, we will study the possibility of using the data in their text format as input using text comprehension of neural networks.

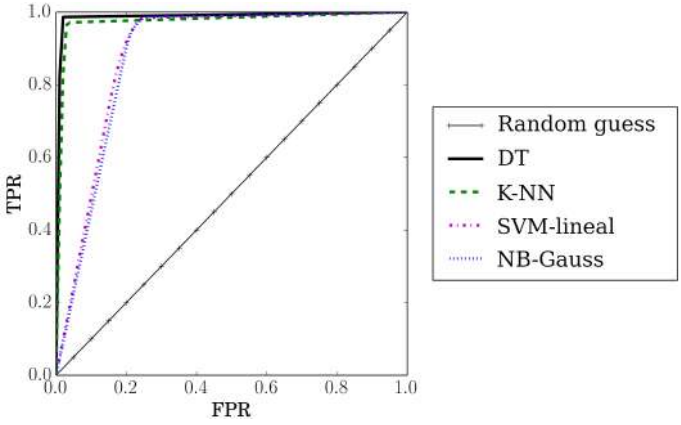


Fig. 2. ROC curves for the TCP-Sink dataset.

VI. Acknowledgement

This research was funded by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01.

References

[1] J. Velasco-Mata, E. Fidalgo, V. González-Castro, E. Alegre, and P. Blanco-Medina, "Botnet detection on tcp traffic using supervised machine learning," in *International Conference on Hybrid Artificial Intelligence Systems*. Springer, 2019, pp. 444–455.

[2] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, and F. Jáñez-Martino, "Fraudulent e-commerce websites detection through machine learning," in *Hybrid Artificial Intelligent Systems*, H. Sanjurjo González, I. Pastor López, P. García Bringas, H. Quintián, and E. Corchado, Eds. Cham: Springer International Publishing, 2021, pp. 267–279.

[3] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, and E. Fidalgo, "Trustworthiness of spam email addresses using machine learning," in *Proceedings of the 21st ACM Symposium on Document Engineering*, 2021, pp. 1–4.

[4] M. Sánchez-Paniagua, E. Fidalgo, V. González-Castro, and E. Alegre, "Impact of current phishing strategies in machine learning models for phishing detection," in *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*, Á. Herrero, C. Cambra, D. Urda, J. Sedano, H. Quintián, and E. Corchado, Eds. Cham: Springer International Publishing, 2021, pp. 87–96.

[5] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, "Independent comparison of popular dpi tools for traffic classification," *Computer Networks*, vol. 76, pp. 75–89, 2015.

[6] G. Kirubavathi and R. Anitha, "Botnet detection via mining of traffic flow characteristics," *Computers & Electrical Engineering*, vol. 50, pp. 91–101, 2016.

[7] H. Kim, S.-S. Choi, and J. Song, "A methodology for multipurpose dns sinkhole analyzing double bounce emails," in *International Conference on Neural Information Processing*. Springer, 2013, pp. 609–616.

[8] P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo, "Practical real-time intrusion detection using machine learning approaches," *Computer Communications*, vol. 34, no. 18, pp. 2227–2235, 2011.

[9] H. Kim, K. C. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: myths, caveats, and the best practices," in *Proceedings of the 2008 ACM CoNEXT conference*, 2008, pp. 1–12.

[10] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *computers & security*, vol. 45, pp. 100–123, 2014.

[11] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 29–35.

Extracting Composition and Social Engineering Features to Measure Spam Address Credibility

Felipe Castaño*†, Franciso Jáñez-Martino*†, Pablo Blanco-Medina*†,
Alexandra Bonnici‡, Santiago Gonzalez†, Eduardo Fidalgo*†

*Department of Electrical, Systems and Automation, Universidad de León, León, ES
†Researcher at INCIBE (Spanish National Cybersecurity Institute), León, ES
‡University of Malta, Msida, MLT
Email:{felipe.castano, francisco.janez, pablo.blanco, eduardo.fidalgo}@unileon.es,
{alexandra.bonnici@um.edu.mt, s.g@incibe.es}

***Abstract*—Spam emails are increasingly being used to send phishing, malware, and other types of scams. Cybercriminals constantly enhance spam emails making them more contextualized and sophisticated to deceive users and organizations. In this work, we present a methodology to detect spam emails focusing on the information provided by the sender address. We extracted 18 binary features using Natural Language Processing and social engineering techniques to train four machine learning classifiers on a novel dataset we named Email Addresses Quality (EAQ-6K). The proposed set of features achieved a performance of 88.17% accuracy and 0.808 F1-Score using a Naïve Bayes algorithm on the EAQ-6K dataset. This tool can be used by Cybersecurity Organizations and companies to enhance the accuracy of their spam filters or even detect spam campaigns.**

***Index Terms*—Cybersecurity, Spam Email Detection, Machine Learning**

Contribution: *Research already published [1]*

I. Introduction

Spam emails represent more than 50% of worldwide online correspondence, and this percentage continues to increase [2]. The Europol European Cybercrime Centre (EC3) warned that 78% of cyberattacks had their root in spam emails [3], which expose companies and citizens to scams, such as hacking extortion, malware, phishing, or identity theft [4].

As these kinds of attacks evolve constantly, classic detection methods like white and black lists have become outdated. Instead, Natural Language Processing (NLP) and machine learning based algorithms have been widely used in classification problems, showing good performance [5], [6].

Spam filters commonly used email body content, ignoring data from the email header which may speed up the process. Experts in Cybersecurity consider that email addresses contain relevant information to extract [7]. However, users have trouble identifying dangerous emails [8], since spam usually contains company names or domains that seem trustworthy.

In our work [1], we proposed an approach to measure the quality of a spam email that focuses on the information provided by email addresses. Our proposal might be useful for Cybersecurity Organizations or companies to enhance the accuracy of their spam filters or even detect spam campaigns.

II. Methodology

The proposed methodology can be summarized in three steps: (i) category and feature definition for high/low quality spam addresses; (ii) dataset creation and (iii) feature extraction for training machine learning classifiers.

TABLE I
Examples of low and high quality spam email addresses

Low Quality
pop@achieverecruit.xyz
notification+KHm9BEbYo6kGSC9cBaJRB@parliamena.cf
High Quality
johnhag224@gmail.com
mail@amazonsupport.info

A. Quality Definition

The quality of spam email addresses can be used by cybersecurity experts as an indicator to identify the most harmful spam campaigns that target companies and people. The more trustworthy a spam address seems, the higher the risk of interacting with the malicious spam email.

In our work, we have classified email addresses into two categories: low and high quality. High-quality spam email addresses imitate corporate, news, and company email addresses. They also take into account social engineering techniques, like using popular email services, domains, and brands that seem trustworthy. On the other hand, low-quality spam email addresses contain unrelated characters, numbers, or randomly written words. Table I presents samples for each class.

B. Dataset EAQ-6K

We retrieved a total of 6569 email addresses from the Bruce Guenter Project[1], which is the most up-to-date spam dataset. We focused on samples from 2019 to 2020.

[1]http://untroubled.org/spam/ Retrieved March 2022

Fig. 1. Email address components

TABLE II
FEATURES CAPTURED FROM EACH EMAIL ADDRESS.

ID	Features	Explanation	Affirmative Response
F1	ifusername_tooshort	contains less than 4 chars	LQ
F2	ifusername_random	contains random chars	LQ
F3	ifusername_toonums	contains more than 4 numbers	LQ
F4	ifusername_toochars	contains more than 4 special chars	LQ
F5	ifusername_singleword	is a single word	HQ
F6	ifusername_corporative	imitate a corporative address	HQ
F7	ifusername_onlyname	is only a name or surname	HQ
F8	ifusername_domainrelation	there is a relation among them	HQ
F9	ifdomain_tooshort	contains less than 4 chars	LQ
F10	ifdomain_random	contains random chars	LQ
F11	ifdomain_toonums	contains more than 2 numbers	LQ
F12	ifdomain_chars	contains more than 2 special chars	LQ
F13	ifdomain_popularemail	is a well-know email service provider	HQ
F14	ifdomain_incluidemail	contains the word "mail"	HQ
F15	ifdomain_popularcompany	includes a popular company	HQ
F16	ifdomain_SMEcompany	could be a SME company name	HQ
F17	iftld_manytlds	has more than one tlds	LQ
F18	iftld_unknown	is unknown or unpopular	LQ

Five people were recruited to classify these addresses following the questionnaire in Table II, with the final label being the mode of the five answers. We labelled 5181 email addresses as low-quality while 1388 were labelled as high quality. Our resulting dataset was named Email Addresses Quality (EAQ-6K) and it is publicly available[2].

C. Feature extraction

Our extraction feature process is inspired on the work proposed by Sahingoz et al. [9], which used 38 NLP features and evaluated seven machine learning algorithms to classify phishing, increasing the detection rate by 10.86%.

The URL features were adapted to email address structure, as shown in Figure 1. In addition, we defined 18 binary questions in collaboration with the Spanish National Cybersecurity Institute (INCIBE), capturing different features.

As shown in Table II, features F1 to F5, F9 to F12, and F17 extract information related to composition properties or extension from some part of the address using NLP techniques. Features F6 to F8, F13 to F16 and F18 are focused on capturing social engineering and subjective information.

III. EXPERIMENTATION

We implemented four machine learning algorithms widely used in similar classification problems [6], [9], [10]: Support Vectors Machine (SVM), Naïve Bayes (NB), Random Forest (RF), and Logistic Regression (LR). Using as input an 18-element binary vector where a binary value of '1' represents a positive answer while '0' represents a negative answer.

The parameter set for the experiments is explained in this section. We used a multinomial distribution for NB. In the case of the SVM model, we used a linear kernel and a cost C=0.1. For LR, we chose a cost C=0.1 and the Stochastic Average Gradient (SAG) solver. We used 50 estimators and five as maximum depth for the RF algorithm. Finally, we used the default value for any other algorithm parameters. The performance of each algorithm is reported using accuracy and F1-Score. As shown in Table III, NB achieved the best performance and the lowest execution time.

The features were checked using InterpretML [11]. This tool describes the relationship between features and classes and checks for interference between classes. We used LR in this process, as it is the only algorithm compatible with the InterpretML analysis tool.

[2]http://gvis.unileon.es/dataset/email-addresses-quality-eaq-6k/ Retrieved March 2022

TABLE III
EVALUATION OF FOUR TRADITIONAL MACHINE LEARNING ALGORITHMS.

Classifier	Accuracy(%)	F1-Score	Runtime (µs)
NB	**88.17**	0.808	0.04
SVM	80.70	0.762	6.90
LR	83.49	0.787	0.16
RF	85.40	**0.817**	11.10

The results show that F8 and F11 features are relevant to define if the sample is a low-quality one. Moreover, F6 and F13 features contribute the most to high-quality spam.

IV. CONCLUSIONS

A methodology to measure the quality of spam email addresses was proposed in this work. We also presented and published the novel dataset EAQ-6K. Our results showed that the proposed features provide enough information to achieve 88.17% accuracy and 0.808 F1-Score on our spam dataset.

Furthermore, after an InterpretML analysis, we discovered that the most relevant features are F8 and F11 for low quality email addresses, while F6 and F13 are more important for high-quality email addresses.

V. ACKNOWLEDGEMENT

This research was funded by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01, as well as the grant 'Ayudas para la realización de estudios de doctorado en el marco del programa propio de investigación de la Universidad de León Convocatoria 2018'.

REFERENCES

[1] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, and E. Fidalgo, "Trustworthiness of spam email addresses using machine learning," in *Proceedings of the 21st ACM Symposium on Document Engineering*, 2021, pp. 1–4.

[2] R. M. A. Mohammad, "A lifelong spam emails classification model," *Applied computing & informatics.*, vol. ahead-of-print, 2020.

[3] Europol, "Spear phishing, a law enforcement and cross-industry perspective," 2019. [Online]. Available: https://www.europol.europa.eu/media-press/newsroom/news/europol-publishes-law-enforcement-and-industry-report-spear-phishing

[4] L. Gallo, A. Maiello, A. Botta, and G. Ventre, "2 years in the anti-phishing group of a large company," *Computers & security*, vol. 105, p. 102259, 2021.

[5] F. Castaño, M. Sánchez-Paniagua, J. Delgado, J. Velasco-Mata, A. Sepúlveda, E. Fidalgo, and E. Alegre, "Evaluation of state-of-art phishing detection strategies based on machine learning," in *Actas de las VI Jornadas Nacionales*, 6 2021, pp. 47–48.

[6] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, and F. Jáñez-Martino, "Fraudulent e-commerce websites detection through machine learning," in *Hybrid Artificial Intelligent Systems*, H. Sanjurjo González, I. Pastor López, P. García Bringas, H. Quintián, and E. Corchado, Eds. Cham: Springer International Publishing, 2021, pp. 267–279.

[7] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, p. e01802, 2019.

[8] M. Alshaikh, S. B. Maynard, and A. Ahmad, "Applying social marketing to evaluate current security education training and awareness programs in organisations," *Computers & security*, vol. 100, p. 102090, 2021.

[9] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," *Expert Systems with Applications*, vol. 117, pp. 345–357, 3 2019.

[10] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Computing and Applications*, vol. 31, no. 8, pp. 3851–3873, 2019.

[11] H. Nori, S. Jenkins, P. Koch, and R. Caruana, "Interpretml: A unified framework for machine learning interpretability," *arXiv preprint arXiv:1909.09223*, 2019.

A review of Towards a data collection methodology for Responsible Artificial Intelligence in health: A prospective and qualitative study in pregnancy

Andreea M. Oprescu *, Gloria Miró-Amarante *, Lutgardo García-Díaz †, Ángel Chimenea Toscano †, Victoria E. Rey ‡, Ricard Martínez-Martínez § and MCarmen Romero-Ternero *

*Departamento de Tecnología Electrónica, Universidad de Sevilla
†Hospital Universitario Virgen del Rocío, Sevilla
‡Clínica Victoria Rey, Sevilla
§Departamento de Derecho Constitucional, Ciencia Política y de la Administración, Universitat de Valencia

***Resumen*—This study contextualizes and justifies why IT security and privacy play a relevant role in the design and implementation of healthcare systems. Specifically, it presents a new methodology called DMM-HERA (Data Management Methodology in the context of heterogeneous environment and Responsible artificial intelligence Applications in health). Several steps of the proposed methodology consider IT security and privacy criteria in order to implement them from the very beginning (by design and by default) in healthcare intelligent solutions. This is particularly relevant for regulatory compliance. The work presents a detailed qualitative study on intelligent healthcare systems with a specific population (150 pregnant women) and identifies various obstacles and barriers that determine the perception of IT security and privacy for these users.**

***Index Terms*—IT security by design, privacy by design, data governance, responsible artificial intelligence, healthcare**

Tipo de contribución: *Investigación ya publicada - Information Fusion, part of Special Issue "Advances in Explainable (XAI) and Responsible (RAI) Artificial Intelligence" Volumes 83–84, 2022, Pages 53-78, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2022.03.011.*

I. Introduction

Responsible Artificial Intelligence (RAI) rests on three pillars of equal importance: 1) society must be prepared to take responsibility for the impact of AI (researchers and developers should be trained to be aware of their own responsibilities when it concerns the development of AI systems with direct impact on society and RAI is an issue of regulation and legislation), 2) RAI implies the need for mechanisms that allow AI systems to reason about and act according to ethics and human values (it requires models and algorithms to represent and reason about, and take decisions based on human values, and justify their decisions according to their effect on those values), and 3) it is necessary to understand how different people work with and live with AI technologies across cultures to develop frameworks for RAI (AI must be understood as part of socio-technical relations) [1].

Under these principles, the EU Guidelines [2] presented a set of 7 key requirements AI systems must meet to be deemed trustworthy: 1) Human agency and oversight (human-in-the-loop), 2) Technical Robustness and safety, 3) Privacy and data governance, 4) Transparency (and traceability), 5) Diversity, nondiscrimination, and fairness (no bias), 6) Social and environmental wellbeing, and 7) Accountability (and algorithm auditability).

From this standpoint, our study on intelligent healthcare systems focuses on aspects such as Human-Centered Design, eXplainable AI, Privacy, and Information Security by Design [3].

II. Data collection methodology

Obtaining data is a crucial step in an AI worflow. Traditionally, data for AI applications were obtained from medical records and medical instruments. Data fusion can be used to improve quality of AI models, since it allows more data types, from wearable devices, environmental sensors, smartphones, among others. However, as the type of data and the number of interconnected devices increases, the privacy and information security of the solutions must be reinforced. In this context of multimodal data, we propose a preliminary methodology consisting of the following steps:

1. Study stakeholders needs and expectations, identify potential trust and adoption barriers from RAI perspective.
2. Identify all data sources.
3. Identify all implemented health services workflows.
4. Determine which data sources are needed for workflow.
5. Determine which intelligent function/process fits in each workflow.
6. Determine what data fusion techniques are appropriated.
7. Determine which AI techniques fit for each intelligent function/process.
8. Analyze the impact of using the data and the selected techniques from the point of view of each RAI aspect:
   a) Human agency and oversight
   b) Technical Robustness and safety
   c) Privacy and data governance
   d) Transparency (and traceability)
   e) Diversity, non-discrimination and fairness (no bias)
   f) Societal and environmental well-being
   g) Accountability (and algorithm auditability).
9. Analyze the gap between *as-is* and *to-be* to facilitate change management and reduce the possibility of AI technology adoption and trust barriers.

10. Deploy the RAI-based solution with information fusion and evaluation (Deming cycle).

In Step 8, the study shows two detailed tables with questions for the analysis of the impact of using the data and the selected techniques from the viewpoint of each RAI aspect, including IT security and data governance. Furthermore, the SATORI study [4] or the ALTAI High-Level Expert Group (Assessment List for Trustworthy Artificial Intelligence) could be used for self-assessment [5].

III. Pregnancy Case Study

The qualitative questionnaire included a section dedicated to the perceptions and concerns that the pregnant women have on Privacy and Information Security. 150 patients from *Hospital Universitario Virgen del Rocío* and *Clinic Victoria Rey* completed the questionnaire (52.6 % with low-risk pregnancy and 44 % with high-risk pregnancy; 2.6 % N/A). The participant's mean age in years was 34.04 (SD 5.7). All participants reported having a smartphone with Internet connection and 78 % reported using the smartphone several times a day. The results of the section are the following:

The strongest privacy concerns reported were protection of personal data, fear of being heard or seen without being aware, the use of data that they have not authorized, and their information being seen by people other than their physician.

Participants were asked how often they read the privacy policies of the services they used. 49.4 % of the participants reported reading the privacy policy only when they were interested in the service. Few participants, 7.3 % indicated that they always read the privacy policy, 14.6 % read it frequently, and 24.6 % never read it.

Social media privacy policy: 50 % of the participants were aware of the existence of a privacy policy and changed their configuration to more restrictive, 32.6 % reported not being aware of the privacy policy. 17.4 % N/A.

Participants were asked about their awareness of the permissions granted to applications installed on their phones. 56 % reported that they are generally familiar with the granted permissions. Half indicated that they would change the default permissions according to their personal privacy preferences.

Concerns about the information security of IT services: 74 % of the participants reported being concerned about the information security of the services they use, of which 20.6 % even felt discouraged from using certain services. 18 % reported not being concerned, of which 12 % answered that they trusted the service provider to ensure data security.

Participants were surveyed about smartphone security measures. The most popular was screen locking with a password, pattern, PIN, or biometric scan, uninstalling unwanted or unused applications, performing periodic backups, avoiding installing applications from unknown sources, allowing automatic operating system updates, and installing an antivirus. 94 % indicated having at least one of the said security measures. 36.8 % had one or two security measures in place, followed by 29.7 % who had three or four security measures. 32 % applied five or six security measures and 7.8 % applied seven or eight features. On average, three measures were implemented.

Two-factor authentication security measure: 77.3 % of the participants reported some level of agreement when asked about their thoughts on applications using the two-factor authentication factor. They thought that (a) the application was more trustworthy by implementing this additional layer of security: 35.3 %, (b) it was a reasonable measure: 34.5 %, and (c) it was adequate: 30.2 %. On another note, 17.3 % reported finding it annoying or unnecessary (23 %).

With respect to the use of a pregnancy medical application on their smartphone, pregnant women were surveyed about the trustworthiness of the application with respect to its provider. The potential providers would be the public health system, a private company that works for the public health system, and a private company specialized in this type of service that had no relationship with the public health system. In general, 86 % of the participants felt that the most reliable option was the public health system.

IV. Conclusion

In general, pregnant women showed interest in using an intelligent healthcare solution to assist them throughout their pregnancy. However, various obstacles to the adoption of technological solutions have been identified, mainly due to concerns about privacy and security, lack of digital skills, and skepticism of technology. Addressing privacy and information security requirements is vital to carry out successful implementations of AI solutions in the healthcare sector.

The development of regulatory compliance processes for a technology cannot be static. The very nature of AI makes this principle a rule. We must understand that we are in an early phase of this technology, in which we have learned that data analysis processes are very sensitive to bias. On the other hand, we begin to entrust basic decision-making processes to this technology, with legal or material consequences for its recipients. Proper compliance requires a state of ongoing monitoring and updating that is deployed at several levels: 1) learn from the operation of the technology itself. The results obtained, the operation errors, incidents, and ultimately any verified or verifiable element should also be indexed and studied by legal support. And not only to prevent potential conflicts and responsibilities, but above all to improve compliance conditions; 2) deepen the design of compliance by proposing improvements when necessary; 3) accompany each phase or evolution of the product.

Referencias

[1] V. Dignum, Responsible Artificial Intelligence: Designing AI for Human Values, ITU Journal: ICT Discoveries, vol. Speccial Issue, no. 1, 2017.

[2] European Commission, Ethics Guidelines for Trustworthy AI. 2020. [Online]. Available: https://www.aepd.es/sites/default/files/2019-12/ai-ethics-guidelines.pdf

[3] Oprescu, A., Miró-Amarante, G., García-Díaz, L., Rey, V., Chimenea-Toscano, A., Martínez-Martínez, R. & Romero-Ternero, M.C. Towards a data collection methodology for Responsible Artificial Intelligence in health: A prospective and qualitative study in pregnancy. *Information Fusion*. 83-84 pp. 53-78 (2022)

[4] Nielsen, R., Gurzawska, A. & Brey, P. Principles and Approaches in Ethics Assessment. Ethical Impact Assessment and Conventional Impact Assessment. Deliverable 1.1. Annex 1.a. IN: SATORI. Ethical Assessment of Research and Innovation: A Comparative Analysis of Practices and Institutions in the EU and selected other countries. , https://satoriproject.eu/media/1.a-Ethical-impact-assessmt-CIA.pdf

[5] High-Level Expert Group on Artificial Intelligence Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment. (2020), https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment

Intelligent detection and recovery from cyberattacks for small and medium-sized enterprises

Carmen María Alba
Fundación I+D del Sotware Libre (FIDESOL)
cmjimenez@fidesol.org

Resumen- **El crecimiento de los ciberataques se ha incrementado a niveles alarmantes en los últimos años. El caso de las pymes es particularmente preocupante, ya que los delincuentes aprovechan los sistemas más vulnerables para ejecutar sus ataques, pero también pueden ser víctimas de ataques no dirigidos. Además, el coste de sufrir un ciberataque es mucho mayor (tanto desde el punto de vista económico como reputacional) para las pymes. En este trabajo proponemos una plataforma para detectar ataques de tipo DDoS (del inglés, *Distributed Denial of Service*), SQLI (del ingles, *SQL Injection*) y DGA (del inglés, *Domain Generation Algorithm*), que son algunos de los ataques que pueden resultar más perjudiciales para las pymes.**

***Index Terms*- pymes, ciberataques, detección de ataques, recuperación ataque, blockchain, machine learning.**

Tipo de contribución: *Investigación ya publicada "Intelligent detection and recovery from cyberattacks for small and medium-sized enterprises" in International Journal of Interactive Multimedia and Artificial Intelligence.* DOI:10.9781/ijimai.2020.08.003

INTRODUCCIÓN

Los ciberataques se encuentran en continua evolución, siendo cada vez más sigilosos, inteligentes y frecuentes. Los atacantes han dirigido su foco a explotar vulnerabilidades de los sistemas de información de infraestructuras críticas y grandes organizaciones. Sin embargo, y especialmente en los últimos años, también se dirigen hacia las pymes, pues son la base de la actividad y economía de España y Europa. Los ataques a las pymes tienen como objetivo perturbar o inutilizar los servicios básicos de estas empresas. Algunos de los ataques más comunes en 2018 fueron la suplantación de identidad (*phishing*), la ingeniería social y el secuestro de datos [1]. Otro ejemplo de robo de datos son los troyanos bancarios, que persiguen la apropiación de cuentas bancarias [2]. En cualquier caso, las consecuencias derivadas de sufrir un ciberataque pueden ser nefastas para una pyme. No sólo se ven obligadas a interrumpir sus sistemas y accionar todos los protocolos de seguridad previstos para riesgos severos, sino que se enfrentan a multas económicas que pueden suponer su ruina. Muchos de estos ataques usan otras técnicas para entrar en los sistemas como el DGA, la SQLi, o el DDoS.

Desde IASEC se propone una herramienta capaz de detectar estos tipos de ataque, y de una solución a los problemas de ciberseguridad en las pymes.

IASEC

El proyecto IASEC tiene como finalidad la creación de una unidad de innovación conjunta de investigadores entre Fidesol y Vector (ahora Softtek). El principal objetivo de IASEC es llevar a cabo actividades de investigación y desarrollo para la construcción y optimización de algoritmos y herramientas que permitan crear soluciones que mejoren la ciberseguridad en empresas e instituciones.

IASEC está formado por tres hitos claramente diferenciados. En el **Hito 1**, se estudian las amenazas más importantes que repercuten en la empresa y se analizan las principales etapas que conforman un ataque informático. Además, se proporcionan los medios para optimizar la detección y recuperación de sistemas. El **Hito 2** se centra en la gestión de identidad de usuarios y dispositivos en internet, para la certificación de la identidad digital. Por último, el **Hito** 3 se fundamenta en la obtención del conocimiento necesario sobre la detección y gestión de publicación de información falsa en internet. Para abordar los objetivos del proyecto, se hace uso de diferentes algoritmos de Inteligencia Artificial en combinación con Blockchain. Este artículo (enmarcado dentro del **Hito 1**) se centra en la detección y recuperación automática de sistemas en una pyme tras sufrir un ataque.

PROPUESTA IASEC PARA DETECCIÓN Y RESPUESTA FRENTE A ATAQUES EN PYMES

En este artículo proponemos una plataforma inteligente de ciberseguridad, diseñada para ayudar a las pymes a asegurar sus sistemas. El objetivo es proporcionar una solución que de respuesta frente a ataques de tipo DDoS, SQLI y DGA. Para crear esta solución en primer lugar se ha estudiado el plan de acción que sufre un ciberataque. Dicho plan está divdido en 4 fases, que se describen a continuación, desde el punto de vista de IASEC:

a) **Prevención.** Consiste en evitar que un sistema sea atacado o comprometido, adoptando para ello las medidas pertinentes para dificultar estas acciones.

b) **Detección.** Esta fase tiene como objetivo la localización de irregularidades dentro del funcionamiento normal del sistema. En IASEC hemos estudiado diferentes algoritmos de aprendizaje automático para detectar los principales ataques en pymes.

c) **Contención.** El objetivo de esta fase es minimizar el impacto de un incidente, aislando las partes vulnerables del sistema para protegerlo.

d) **Recuperación.** Tras eliminar las amenazas, es necesario llevar a cabo acciones para devolver los sistemas a un estado anterior al ataque. Algunos ejemplos son los planes de respuesta a incidentes o medidas de recuperación automática.

En IASEC proponemos una plataforma basada en una arquitectura de microservicios (Fig.1) para la detección

inteligente de los ataques anteriormente mencionados (DDoS, SQLI, y DGA).

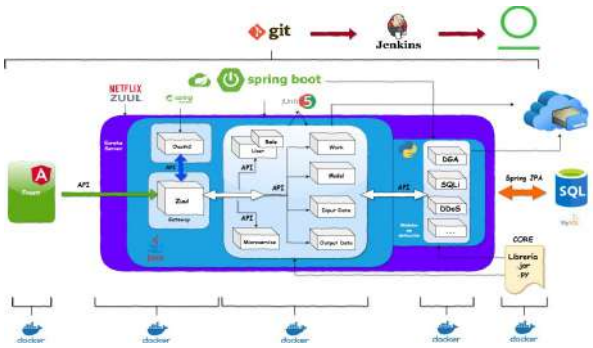


Fig. 1. Arquitectura de microservicios de la plataforma IASEC

La plataforma consta de una interfaz de usuario, que permite la interacción con los microservicios, además de añadir conjuntos de datos para el entrenamiento de los modelos de detección. Los datos a analizar son procesados y se almacenan en un sistema de almacenamiento estático.

Posteriormente se envían a los microservicios para realizar una detección en base a los modelos generados. Estos modelos se pueden adaptar a las necesidades de cada organización.. Los microservicios de detección son sistemas escalables, permitiendo la posibilidad de añadir nuevos módulos, como el de recuperación. Los microservicios son:

a) **DDoS:** se han tomado como referencia dos algoritmos, *RandomForest* [3] y Árboles de decisión [4]. Para el entrenamiento y la validación se ha utilizado el conjunto *KDD'99 cup* [5]. Tras la evaluación, seleccionamos *RandomForest*, ya que proporciona mejores resultados en términos de precisión (99,96%).

b) **SQLI:** se ha tomado como referencia [6]. En este caso, se han comparado los algoritmos Naïve Bayes y Desicion Stump. Para ejecutar estos algoritmos hemos utilizado el conjunto libre de datos XML proporcionado por la ECML-PKDD (del inglés, *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*) [7]. A la vista de los resultados obtenidos, hemos decidido utilizar Decision Stump por presentar mejores resultados (96,95% frente a 93% de Naïve Bayes).

c) **DGA:** se ha planteado una detección de dominios anómalos clasificando las familias de DGA según [8]. Para esto, se ha implementado un red neuronal de tipo LSTM, ya que por sus características, presenta una alta precisión en detección de dominios DGA y clasificación por familias. Para entrenar esta red neuronal se ha usado la base de datos de dominio Alexa[9].

Gracias a la arquitectura de microservicios, el sistema IASEC es una plataforma flexible y escalable, que admite cualquier conjunto de datos, siempre que tengan la estructura adecuada. Además, la plataforma IASEC admite la implementación de otros microservicios para detectar otro tipo de ataques, o incluso la implementación de un sistema no supervisado para la detección de anomalías. Otra de las ventajas de este sistema reside en su sencillez, tanto en el despliegue como en su uso, por lo que lo hacen especialmente atractivo desde el punto de vista de la seguridad en las pymes.

Conclusiones

En este artículo analizamos los problemas relativos a ciberseguridad que sufren las pymes en la actualidad. Prestando especial interés a la detección y recuperación de sistemas, una de las principales conclusiones obtenidas, es que no solo es importante establecer medidas preventivas frente a ataques, sino es aún más importante realizar una detección temprana de ataques. Así, detectando un ataque en tiempo real, se conseguirá minimizar los riesgos y establecer las medidas de recuperación oportunas.

El mayor problema identificado en las pymes con respecto a ciberseguridad, es la falta de recursos para poder invertir en sistemas eficientes. La solución propuesta en este artículo, está enmarcada en el primer Hito 1 del proyecto IASEC. Esta solución ofrece a las pymes una plataforma de seguridad que implementa varios microservicios para la detección de ataques DDoS, SQLI y DGA. La arquitectura propuesta es escalable, permitiendo la posibilidad de adaptarse a las necesidades de cada pyme.

Los resultados obtenidos han sido óptimos en la primera versión implementada de estos microservicios, por lo se espera obtener una versión mejorada de la plataforma, capaz de ejecutar la detección de ciberataques en tiempo real. El objetivo es que la nueva versión de la plataforma disponga de microservicios de recuperación tras sufrir un ataque. Estos microservicios utilizarán blockchain para dotar de integridad al sistema.

Agradecimientos

El proyecto IASEC ha sido financiado por la Agencia IDEA (Junta de Andalucía). Código del proyecto: 402C1800002.

Referencias

[1] C.M. Arce, "Ciberseguridad y crímenes informáticos: el lado oscuro de la red", *Revista Académica Arjé,* vol. 2, n.2 , pp. 14-19, 2019.

[2] D,Kiwia, A. Dehghantanha, K.-K. R.Choo and J. Slaughter, "A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence", *Journal of Computational Science*, vol. 27, pp. 394-409, 2018.

[3] I.Moles, "Ancert: aplicación de técnicas de machine learning a la seguridad", *Repositorio institucional (O2)*, 2018.

[4] J. M. Rodriguez, "Aplicación de técnicas de Machine Learning a la detección de ataques", *Repositorio institucional (O2)*, 2018. [Online]. Available: http://hdl.handle.net/10609/81126 [Accessed: Apr 17, 2020].

[5] Irvine, "KDD Cup 1999 Data", 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[6] P. Aaby, "Evaluating Web App Datasets towards Detection of SQL Injection Attacks with Machine Learning Techniques", 2016. [Online]. Available: https://cutt.ly/fyxosIm [Accessed: Apr 17, 2020]

[7] C.R. Raïsi, J.Brissaud, G.Dray, P.Poncelet, M. Roche and M. Teisseire, "Web Analyzing Traffic Challenge: Description and Results", en *The 18th european conference on machine learning and The 11th european conference on principles and practice of knowledge discovery in databases*, pp. 47-52, 2007.

[8] OSINT, "Feeds from Bambenek Consulting", 2019. [Online]. Available:https://osint.bambenekconsulting.com/feeds/.[Accessed: Apr 17. 2020].

[9] Alexa Internet, "Alexa - Top sites", 2020. [Online]. Available: https://www.alexa.com/topsites. [Accessed: Apr 17, 2020].

Dataset creation and feature extraction for the detection of fraudulent websites

Juan José Delgado Sotés
Dpto. IESA, Universidad de León
Researcher at INCIBE, León, Spain
jdelgs01@estudiantes.unileon.es

Manuel Sánchez-Paniagua
Dpto. IESA, Universidad de León
León, Spain
manuel.sanchez@unileon.es

Javier Velasco-Mata
Dpto. IESA, Universidad de León
Researcher at INCIBE, León, Spain
javier.velasco@unileon.es

*Eduardo Fidalgo
Dpto. IESA, Universidad de León
Researcher at INCIBE, León, Spain
eduardo.fidalgo@unileon.es

Juan Prieto Carballal
Researcher at INCIBE
León, Spain
juan.prieto@incibe.es

George Azzopardi
University of Groningen
Groningen, Netherlands
g.azzopardi@rug.nl

***Abstract*—Fraudulent websites offer a medium to collect personal data from users. These websites allow the perpetrator to steal both the user's money and their identity or credit card information. The current way of detecting such sites is based upon reporting, blacklisting and filtering; all valid passive prevention methods that also suffer from out-dates and false positives. In this work, we present Features for Fraudulent Websites (FFW-282), a dataset which comprises 282 samples from real and fraudulent websites which are described with a set of 11 features. Using a Random Forest to detect fraudulent websites on this dataset we achieved a mean F1-score of 75% over a 5-fold cross-validation.**

***Index Terms*—E-commerce · Fraud detection · Machine Learning · Cyber-security · Fraudulent websites**

Contribution: Summary of *Fraudulent E-Commerce Websites Detection Through Machine Learning* [1]

I. Introduction

In the last few years, the Internet has become one of the leading distribution channels for retail trade. According to Statista, a global business data platform, in 2025, the penetration of e-commerce will reach $63,1\%$ [1]. To improve user experience, e-commerce websites usually share a similar structure. Undesirably, this feature is exploited by fraudsters in creating fake websites, with the aim of stealing personal data including financial information as well as money [2].

The rule-based method is the most used approach to automate the detection of online web scams. The fast growth of these sites and their similarity with the original ones, however, make their effectiveness weaker and weaker. We propose a new approach for detecting fraudulent e-commerce websites using features extracted from them and also from third-party services, combined with Machine Learning (ML) algorithms, as shown in the schematic diagram of Fig. 1.

In order to demonstrate the effectiveness of our proposed solution we created a new data set that we call Features for Fraudulent Websites or FFW-282 for short, which contains 181 legitimate and 101 fraudulent websites, thus a total of 282 samples.

[1]Report available at https://www.statista.com/outlook/dmo/ecommerce/worldwide. Last accessed in March 2022

II. Related work

Literature presents different approaches for detecting fraudulent websites using feature extraction. Wadleigh et al. [3] worked with URL, web content and WHOIS properties, while Mostard et al. [4] used HTML code features. Furthermore, Woo et al. [5] used a larger set of features by combining those based on content structure with the types mentioned above.

Moreover, there are third-party, legit online tools like ScamAdviser, that comprise the analysis of WHOIS, Alexa ranking and the possible SSL certificate with the aim to help the user spotting fraudulent web sites. Other similar tools are ScamFoo, Scamner or Web of Trust, which also provide a confidence score using information from external services like MozRank, social media and community ratings. However, these services require an active participation from the end user, who could benefit from an automatic system that detects fraudulent websites.

III. Methodology

A. Feature descriptor

A novel feature descriptor is developed to characterize websites. It is inspired by the works in [3], [5], and includes the following features: *High discounts*, *Domain age*, *Months registered* and *Social media*, in addition to seven new features from third party sources, namely policies pages, Trustpilot[2], e-commerce technologies and SSL certificates. The complete list of features extracted for each website is:

1) *High discounts* (Positive integer): Number of banners offering high discounts detected on the webpage.
2) *Domain age* (Positive integer): Number of months the webpage server has been registered in WHOIS up to the current date.
3) *Months registered* (Positive integer): Total number of months purchased by the webpage on WHOIS, i.e., from registration to future expiration date.
4) *Social media* (Positive integer from 0 to 3): Count of how many of the three most popular social networks for e-commerce (Twitter, Facebook and Instagram) are linked in the webpage.

[2]https://www.trustpilot.com/

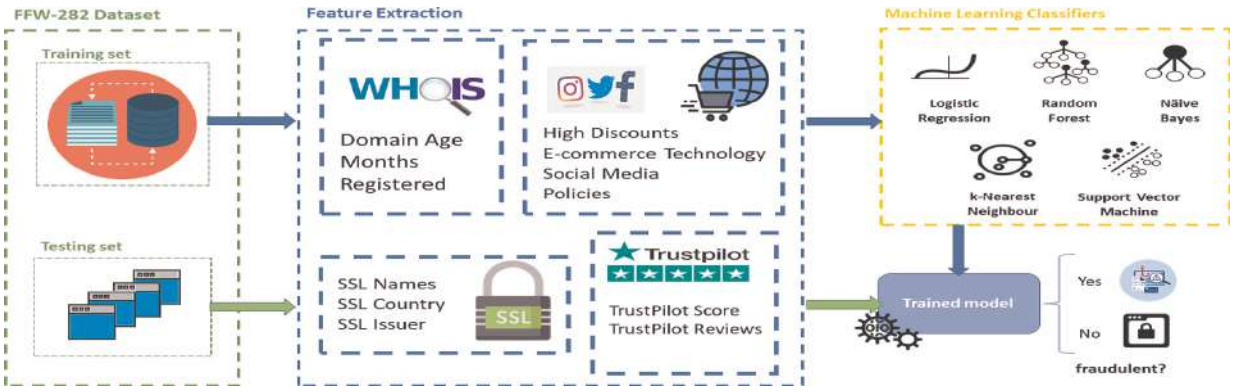


Fig. 1. A schematic overview of the proposed pipeline.

5) *SSL names* (Positive integer): Number of names registered in the SSL certificate.
6) *SSL country* (Binary): Whether the SSL certificate was issued in a risky country, i.e., restricted from legitimate SSL issuers.
7) *SSL issuer* (Positive integer): We found 16 companies that issues SLL certificates. Scammers usually prefer companies that offers certificates cheap or for free.
8) *Trustpilot score* (Float from 1 to 5): A mean of scores reported by other users.
9) *Trustpilot reviews* (Positive integer): Total number of reviews on Trustpilot.
10) *E-commerce technologies* (Positive integer): Total number of legitimate e-commerce frameworks detected on the webpage.
11) *Policies* (Positive integer): Number of links that redirect to policy pages work correctly, i.e., are not blank or redirect to empty pages.

We evaluate our novel feature descriptor by comparing the following five ML classification models that are widely used in the literature [6]: Random Forest (RF), Support Vector Machine (SVM), k-Nearest Neighbor (kNN), Logistic Regression (LR) and Naive Bayes (NB).

B. Data set FFW-282

The new data set FFW-282 was compiled as follows. First, we collected samples using ScamAdviser reports to gather 197 suspicious domains from February to June 2020. Then, we manually analyzed the samples and labeled the domains with a suspicious score of less than 75% as legitimate, adding to this category the top 50 worldwide e-commerce domains and 34 more well-known sites. Finally, the resulting dataset comprises 181 legitimate and 101 fraudulent websites[3].

IV. Experiments and results

We used 5-fold cross-validation to compare the five ML models with the novel 11-element feature descriptor on the new FFW-282 data set. The experiments were implemented in Python 3. Table I shows the results that we obtain. The RF classification model achieves the most superior performance that results in an F1-score of 75%.

[3]Dataset is publicly available at https://gvis.unileon.es/dataset/features-from-fraudulent-websites-282/

TABLE I
AVERAGE RESULTS OF THE FIVE SELECTED ML CLASSIFICATION MODELS OVER THE FFW-282 DATA SET USING 5-FOLD CROSS VALIDATION

Algorithm	Precision	Recall	F1-Score	Accuracy
RF	80.0	70.59	**75.00**	**85.96**
kNN	60.87	82,35	70.00	78.95
SVM	61.90	76.47	68.42	78.95
LR	55.56	88.24	68.18	75.44
NB	53.33	94.12	68.09	73.68

V. Conclusions and future work

This work introduces the new dataset FFW-282, which allows the benchmarking of fraudulent website detection algorithms. Besides, we proposed a feature set of 11 elements, four of which borrowed from existing studies and seven novel ones based on third party services. Random Forest achieved the best performance over FFW-282, with an F1-Score of 75%. In the future, we aim to explore additional features to form a richer descriptor that can improve the performance.

VI. Acknowledgement

This work was supported by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01, and by the FPU (Formación de Profesorado Universitario) grant of the Spanish Government with reference FPU18/05804.

References

[1] M. S. Paniagua, F. Jáñez-Martino, M. Sánchez-Paniagua, E. Fidalgo, and E. Alegre, "Fraudulent E-Commerce Websites Detection Through Machine Learning," in *HAIS*. Springer, 2021, pp. 267–279.
[2] F. Castaño, M. Sánchez-Paniagua, J. Delgado, J. Velasco-Mata, A. Sepúlveda, E. Fidalgo, and E. Alegre, "Evaluation of state-of-art phishing detection strategies based on machine learning," *Jornadas Nacionales de Investigación en Ciberseguridad*, 2021.
[3] J. Wadleigh, J. Drew, and T. Moore, "The e-commerce market for "lemons": Identification and analysis of websites selling counterfeit goods," *WWW 2015 - Proceedings of the 24th International Conference on World Wide Web*, pp. 1188–1197, may 2015.
[4] W. . Mostard, B. . Zijlema, and M. Wiering, "Combining visual and contextual information for fraudulent online store classification," in *IEEE/WIC/ACM International Conference on Web Intelligence*. ACM, 2019, pp. 84–90.
[5] K. T. Wu, S. H. Chou, S. W. Chen, C. T. Tsai, and S. M. Yuan, "Application of machine learning to identify Counterfeit Website," *ACM International Conference Proceeding Series*, pp. 321–324, nov 2018.
[6] M. Das, S. Saraswathi, R. Panda, A. K. Mishra, and A. K. Tripathy, "Exquisite Analysis of Popular Machine Learning–Based Phishing Detection Techniques for Cyber Systems," *Journal of Applied Security Research*, vol. 16, no. 4, pp. 538–562, 2021.

Auto-encoder framework for extractive text summarization

Daniel Diaz
Dpto. IESA, Univesidad de León
Researcher at INCIBE
León, Spain
ddiao@unileon.es

Akanksha Joshi
Dpto. IESA, Univesidad de León
León, Spain
ajos@unileon.es

*Javier Velasco-Mata
Dpto. IESA, Univesidad de León
Researcher at INCIBE
León, Spain
javier.velasco@unileon.es

Juan José Delgado Sotes
Dpto. IESA, Univesidad de León
Researcher at INCIBE
León, Spain
jdelgs01@estudiantes.unileon.es

Antonio Sepúlveda
Researcher at INCIBE
León, Spain
antonio.sepulveda@incibe.es

Guru Swaroop Bennabhaktula
Dpto. IESA, Univesidad de León
Researcher at INCIBE, León, Spain
University of Groningen, The Netherlands
g.s.bennabhaktula@rug.nl

***Abstract*—Illicit content produced on the dark web is increasing and the number of documents that must be analyzed by the police forces grows outside human capability. Summarizing these documents allows the authorities to speed up the analysis work. The approach used in this work is based on auto-encoders. It establishes a qualification and a ranking of the most important sentences of the document to carry out an extractive summary of these documents. Besides, this work proposes a new dataset, TIDSumm, that allows the evaluation of texts taken from the Tor deep network about illegal activities.**

***Index Terms*—Cybersecurity, Auto-encoders, Text summarizing**

Contribution: Extended summary of the work *SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders* [1]

I. Introduction

Due to the amount of data available, the Internet has become a great channel to obtain information on any topic. However, the growing data can also be a problem for the public forces in cases when people use the Internet to deal with illegal content. In that case, authorities must manually review and analyze documents to find crime evidence. Most of the time, those documents do not contain relevant information, and their large sizes make this a time-consuming task. That makes it necessary to implement a tool to perform an extractive (concatenate the most relevant sentences) or abstractive (rewrite the text) [2] summary of the text. This shorter version could provide a first overview of the content to check if it contains illegal content.

This work introduces a novel approach to text summarization based on an auto-encoder network with three metrics: Sentence novelty, Sentence position relevance metric, and Sentence Content Relevance. These three metrics help select the sentences that will be part of the summarised text. Besides, this work presents a novel dataset based on Tor documents.

II. Background

The problem of text summarization addressed using a supervised learning approach requires a gold summary. On the other hand, unsupervised learning can also be used for text summarization. In this case, heuristics are used to rank the sentences of the summary, making it more adaptable to a domain change. Within the unsupervised approaches, deep auto-encoders [3] and graph-based ranking approaches [4], with vectorization methods such as skip-thought vectors [5] or paragraph vectors [6] have shown potential.

III. Methodology

A. Dataset

This paper introduces a novel dataset for text summarization called TIDSumm (Tor Illegal Documents SUMMarization), focused on illicit activities texts taken from the DUTA dataset [7]. This dataset contains 100 documents from DUTA and two extractive summaries of around 100 words per document. The summaries of each document were written by two different people.

B. Preprocessing and Sentence encoder

The text preprocessing starts by removing the metadata and XML/HTML tags from documents, which are cleaned by omitting sentences shorter than five words, lowercasing letters, and removing the redundant characters and white spaces, numbers, and emails. Then, texts are mapped into fixed-length vectors of real numbers called *skip-thought vectors* [5]. In this work, that representation was obtained by using the unidirectional skip-thoughts model [5] trained on an unlabelled Book-Corpus dataset [8].

C. Summary generator

In the first step, each sentence is processed sequentially to obtain its corresponding skip-thoughts vector of dimension 2400. Once the embedding of each sentence is obtained, this representation of the text is passed through a trained auto-encoder network that receives a concatenated sequence of M embedding phrases (textual-unit embeddings) as input to make a low dimension representation (Latent Representation).

The Content Relevance Metric ($score^{ContR}(D, S_i)$) measures the importance of a given sentence by comparing the

TABLE I
ROUGE SCORES ON TIDSUMM DATASET OF SUMMCODER AND FIVE STATE-OF-THE-ART APPROACHES

Method	ROUGE-1	ROUGE-2	ROUGE-SU4	ROUGE-L
Luhn [11]	40.5	23.8	21.2	32.4
TextRank [4]	46.3	29.4	31.8	39.6
LexRank [12]	35.3	17.4	22.9	29.7
LSA [13]	40.0	21.4	24.5	33.5
KLsum [14]	44.0	28.8	27.2	36.0
SumBasic [15]	38.4	19.4	22.9	28.6
SummCoder	58.8	48.9	45.9	49.3

latent representation of a document and a modified latent representation product of substituting the sentence embeddings with zeros before the use of the auto-encoder. This comparison is made with the cosine similarity between the two latent representations: the more similar, the less important is the sentence.

The Sentence novelty ($score^{Nov}(D, S_i)$) and Sentence position relevance ($score^{PosR}(D, S_i)$) metrics help to avoid redundant or repetitive sentences in the summary, giving a higher score to the novel ones. The final score is computed following Equation 1 where α, β and γ assign relative weights to the different scores and must meet that $\alpha + \beta + \gamma = 1$. Empirically, we obtained that $\alpha = 0.45$, $\beta = 0.20$ and $\gamma = 0.35$ generate a good summary.

$$score^{f}(D, S_i) = \alpha \cdot score^{ContR}(D, S_i) + \beta \cdot score^{Nov}(D, S_i) + \gamma \cdot score^{PosR}(D, S_i) \quad (1)$$

Finally, the sentences are sorted and ranked based on their scores and the target summary is constructed by concatenating the L first sentences in the ranking in the order of appearance on the original document.

We used four metrics to measure the accuracy of the model: ROUGE-N [9] (ROUGE-1, ROUGE-2) which measures the recall between the generated abstract and the gold abstract, ROUGE-L which evaluates the fluency and the maximum string length of the digest using the Longest Common Subsequence (LCS), and ROUGE-SU4 that counts skipped bigrams along with unigrams and allows skipping at most four unigrams within the bigram components.

IV. EXPERIMENTS AND RESULTS

To determine the best network parameters for the auto-encoder, we tested seven different architectures over more than 14,000 documents taken from the CNN and dailymail dataset [10]. The best architecture consisted of an input layer, five hidden layers, and an output layer with 24000, 4096, 1024, 512, 1024, 4096, and 24000 neurons respectively. After that, we compared our model SummCoder with five state-of-the-art approaches over the TIDSumm dataset, as shown in Table I.

V. CONCLUSIONS AND FUTURE WORK

This work presents the framework SummCoder for single-document extractive text summarization using deep auto-encoders, sentence embedding, and three proposed metrics that generate the final sentence selection score. The use of auto-encoders and the Content Relevance Metric helps to learn a latent representation of a document, which is used to determine the relevance of its sentences. Additionally, the Sentence Novelty and the Sentence Position scores help to reduce the redundancy of sentences.

Besides, the approach is tested on our novel TIDSumm dataset and compared with other well-known models achieving highly competitive performance. In future work, this approach is planned to be extended for query-based text summarization and multi-document text summarization. Moreover, we plan to explore other variations of the auto-encoder network.

VI. ACKNOWLEDGEMENT

This work was supported by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01, and by the FPU (Formación de Profesorado Universitario) grant of the Spanish Government with reference FPU18/05804. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] A. Joshi, E. Fidalgo, E. Alegre, and L. Fernández-Robles, "Summcoder: An unsupervised framework for extractive text summarization based on deep auto-encoders," *Expert Systems with Applications*, vol. 129, pp. 200–215, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417419302192

[2] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 1–66, 2017.

[3] G. Kumar and L. F. D'Haro, "Deep autoencoder topic model for short texts," *Proc. of IWES*, 2015.

[4] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[6] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*. PMLR, 2014, pp. 1188–1196.

[7] M. W. Al Nabki, E. Fidalgo, E. Alegre, and I. De Paz, "Classifying illegal activities on tor network based on web textual contents," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 35–43.

[8] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27.

[9] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[10] M. Hu, A. Sun, and E.-P. Lim, "Comments-oriented document summarization: understanding documents with readers' feedback," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 291–298.

[11] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.

[12] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[13] J. Steinberger, K. Jezek *et al.*, "Using latent semantic analysis in text summarization and summary evaluation," *Proc. ISIM*, vol. 4, no. 93-100, p. 8, 2004.

[14] A. Haghighi and L. Vanderwende, "Exploring content models for multi-document summarization," in *Proceedings of human language technologies: The 2009 annual conference of the North American Chapter of the Association for Computational Linguistics*, 2009, pp. 362–370.

[15] A. Nenkova and L. Vanderwende, "The impact of frequency on summarization," *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, vol. 101, 2005.

Review of Breaking Trivium Stream Cipher Implemented in ASIC Using Experimental Attacks and DFA

F. E. Potestad-Ordóñez
IMSE-CSIC/U. Sevilla
potestad@imse-cnm.csic.es

E. Tena-Sánchez
IMSE-CSIC/U. Sevilla
erica@imse-cnm.csic.es

C. Fernández-García
IMSE-CNM-CSIC
carlos@imse-cnm.csic.es

V. Zuñiga-González
IMSE-CNM-CSIC
virginia@imse-cnm.csic.es

J. M. Mora-Gutiérrez
IMSE-CNM-CSIC
jmiguel@imse-cnm.csic.es

C. Baena-Oliva
IMSE-CSIC/U. Sevilla
cbaena@imse-cnm.csic.es

P. Parra-Fernández
IMSE-CSIC/U. Sevilla
pparra@imse-cnm.csic.es

A. J. Acosta-Jiménez
IMSE-CSIC/U. Sevilla
acojim@imse-cnm.csic.es

C. J. Jiménez-Fernández
IMSE-CSIC/U. Sevilla
cjesus@imse-cnm.csic.es

***Abstract*—In this paper, we present a review of the work [1]. In this work a complete setup to break ASIC implementations of standard Trivium stream cipher was presented. The setup allows to recover the secret keys combining the use of the active non-invasive technique attack of clock manipulation and Differential Fault Analysis (DFA) cryptanalysis. The attack system is able to inject transient faults into the Trivium in a clock cycle and sample the faulty output. Then, the internal state of the Trivium is recovered using the DFA cryptanalysis through the comparison between the correct and the faulty outputs. The secret key of the Trivium were recovered experimentally in 100% of the attempts, considering a real scenario and minimum assumptions.**

***Index Terms*—fault attack, Trivium, ASIC, DFA, key recovery.**

Tipo de contribución: Investigación ya publicada

I. Introduction

The Trivium stream cipher [2] was one of the eSTREAM project finalists and is part of the ISO/IEC 29192-3 [3] standard for lightweight stream ciphers. From an 80-bit secret key denoted as KEY and an 80-bit initialization vector denoted as IV, this cipher is able to generate in a synchronous way up to 2^{64} bits of key stream. Fig. 1 shows a schematic representation of Trivium internal structure. As it can be seen, its internal structure is performed by three shift registers comprising 288 bits in total and ten XOR gates and three AND gates for the feedbacks. Each of these three shift registers are composed by 93, 84 and 111 bits respectively. The KEY and the IV are loaded in the internal register, along with some prefixed zeros and ones. After the first 1152 clock cycles, the cipher generates a valid pseudorandom bit sequence. The key stream is the result of the XOR operations.

II. Cipher Vulnerability against DFA

A. Theoretical vulnerability

DFA is essentially a theoretical attack where if any attacker is able to inject transient faults into the operation of a device (either in its encryption or decryption processes) through the use of mathematical formulation, can obtain secret information contained in the device and thus endanger its security. Of the different assumptions necessary to carry out DFA on the Trivium stream cipher, the most important one is that the attacker is able to inject a single effective fault into the ciphers internal state and capture both the correct key stream

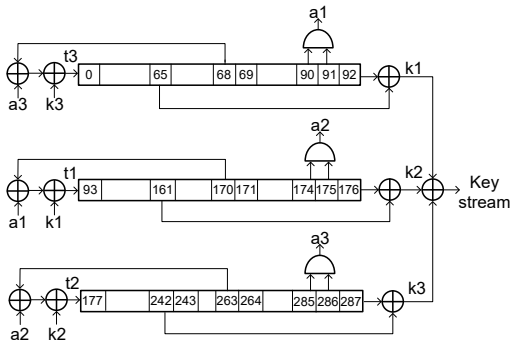


Fig. 1. Schematic representation of the Trivium stream cipher.

and the one originated by that fault. For our system it is necessary to capture 800 bits. A more complete description of the mathematical aspects of the DFA system can be found in the references [4], [5]. In summary, and taking into account DFA nomenclature, the attacker is able to obtain both the key stream of the Trivium cipher $\{z_i\}_{i=1}^{\infty}$ produced by its internal state $IS t_0$ and the key stream produced from a transient random fault $\{z'_i\}_{i=1}^{\infty}$, introduced into the internal state which is now called $IS' t_0$ because it contains the fault. The same attack and the capture of the faulty key stream must be carried out repeatedly under the same conditions: i.e., using the same key and IV and always attacking in the same clock cycle t_0.

B. Experimental vulnerability

The results presented in [6] for attacks carried out by manipulating the clock signal in FPGA implementations of the Trivium showed that it is possible to inject faults only in flip-flops with feedback inputs: namely, position bits 0, 93 and 177 of the internal register or its neighbouring flip-flops. In addition, even with these flip-flops, faults can only be introduced in those that change their value. It is therefore only possible to obtain an average of three faulty key streams. Tests carried out on ASIC implementations showed the same behaviour: faults are injected into flip-flops whose inputs come from feedback or their neighbours. This is a serious problem because, in order to recover the internal state of the Trivium using the developed DFA, it is necessary to have more than three faulty key streams. Therefore, the developed

DFA system requires faulty key streams generated by faults injected in different positions for the same clock cycle. The fewer faulty key streams we have, the greater the brute force effort needed to recover the internal state of the cipher.

III. Experimental attack and results

The experimental attacks were carried out in a Trivium cipher implemented in a 90 nm ASIC technology. The key and IV to be used are loaded serially in the ASIC, and the clock and control signals of the Trivium are connected to the ASIC input pads. The key stream of the Trivium is connected to an output pad of the ASIC.

A. Attack Using Clock Glitches and how to achieve Multiple Faults in the Same Clock Cycle

To inject the faults into the cipher an active non-invasive fault injection system has been designed. It is based on inserting short pulses in the clock signal. This technique allows violating the setup times of the flip-flops making the sampled value on its output be erroneous which represents a fault injection in the Trivium cipher. The clock signal with the short pulses is externally generated and has to pass through the input pads of the integrated circuit to reach the circuit. This is a great challenge because low frequencies will not inject faults in the circuit, but very high frequencies can be filtered by the circuit pads.

It is possible to inject faults into more internal flip-flops because stream ciphers and Trivium in particular are built with shift registers. The shift registers make the fault injected into a flip-flop in one clock cycle appear as a fault injected in the next position of the shift register in the next clock cycle. If a fault is injected into the first bit of any of the three shift register (0, 93, 177), the faulty bit will not contribute to the key stream generation until it reaches one of the bits used for key stream generation (65, 161, and 242). During these clock cycles, the fault is only shifted through the shift register. Inject a fault in position 0, is therefore equivalent to introduce one fault in the position $0+n$, n clock cycles later. This increases the number of positions in which faults can be injected.

B. Results

To carry out the attack on the Trivium, we have used random keys and IVs. The attack cycle was set to $t_0 = 1332$, and the succession of attacks started in cycle $t_{20} = 1312$. Table I shows the results of the attack for Trivium. The table includes the number of the fault injection attempt, the fault injection cycle, the relative position of the injected fault (as if it was a fault inserted in cycle $t_0 = 1332$) and the number of bits of internal state retrieved by the DFA. The results show that with an average of 22 to 32 effective faults, it is possible to obtain the 288 bits of the internal state and therefore recover the secret key of the cipher. It should be noted that, in the case presented in Table I, after 25 attacks it would be feasible to break the cipher by brute force since most of the bits of the internal state are known.

TABLE I
Results obtained from the attack on Trivium.

F.I.A.[1]	C.C.[2]	R.P.[3]	B.R.[4]	F.I.A.	C.C.	R.P.	B.R.
1	1312	115	26	17	1296	131	–
2	1311	24	49	18	1295	132	229
3	1310	24	–	19	1294	134	234
4	1309	119	77	20	1293	134	–
5	1308	119	–	21	1292	135	250
6	1307	121	111	22	1291	220	257
7	1306	121	–	23	1290	222	274
8	1305	123	128	24	1289	222	–
9	1304	123	–	25	1288	139	279
10	1303	208	155	26	1287	147	281
11	1302	33	181	27	1286	249	282
12	1301	33	–	28	1285	143	285
13	1300	211	211	29	1284	143	–
14	1299	129	217	30	1283	228	285
15	1298	129	–	31	1282	145	287
16	1297	38	223	32	1281	230	288

[1] Fault Injection Attempt; [2] Clock Cycle of the attack; [3] Relative Position of the fault; [4] Number of bits Retrieved.

IV. Conclusions

This work describes the complete experimental breaking of Trivium ciphers implemented in ASIC technology. In 100% of the attempts, the secret key and IV were retrieved with minimal assumptions and in a real scenario. Firstly, experimental attacks were performed injecting a single fault into the internal register of the Trivium ciphers changing the external clock signal. Secondly, to inject faults in many positions of the internal register, we took advantage of the fact that the Trivium is built on the basis of shift registers. Thirdly, an inverse-operation Trivium was designed to get the secret key from a known internal state. The achievement of these three steps, together with the developed setup has allowed to obtain the secret key of the Trivium implemented in the ASIC. The work we have presented demonstrates that it is possible to experimentally break the security of ASIC implementations of the Trivium cipher using fault attacks and Differential Fault Analysis, in a short time and in a real scenario.

Acknowledgments

This work has been funded by project SCAROT 1380823-US/JUNTA/FEDER, UE. Thanks to SPIRS Project with Grant Agreement No. 952622 under the European Union's Horizon 2020 programme and Grant PID2020-116664RB-I00 funded by MCIN/AEI/10.13039/501100011033.

References

[1] F.E. Potestad-Ordóñez, M. Valencia-Barrero, C. Baena-Oliva, P. Parra-Fernández, C.J. Jiménez-Fernández, "Breaking Trivium Stream Cipher Implemented in ASIC Using Experimental Attacks and DFA". in *Sensors*, vol. 20, num. 6909, pp. 1-19, 2020.

[2] C.D. Cannière, "Trivium: A stream cipher construction inspired by block cipher design principles". in *Proceedings of the 9th International Conference on Information Security (ISC'06)*, pp. 171–186, 2006.

[3] International Organization for Standardization: ISO/IEC 29192-3:2018. in *Information Security—Lightweight Cryptography—Part 3: Stream Ciphers*, International Organization for Standardization: Geneva, Switzerland, 2018.

[4] M. Hojsík, B. Rudolf, "Differential Fault Analysis of Trivium." in *Proceedings of the International Workshop on Fast Software Encryption (FSE'08)*, pp. 158–172, 2008.

[5] Y. Hu, J. Gao, Q. Liu, Y. Zhang, "Fault analysis of Trivium." in *Des. Codes Cryptogr*, vol. 62, pp. 289–311, 2012.

[6] F.E. Potestad-Ordóñez, C.J. Jiménez-Fernández and M. Valencia-Barrero, "Vulnerability Analysis of Trivium FPGA Implementations." in *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, pp. 3380–3389, 2017.

A Review of Noise-based Cyberattacks Generating Fake P300 Waves in Brain-Computer Interfaces

Enrique Tomás Martínez Beltrán
Department of Information and Communications Engineering
University of Murcia, 30100 Murcia, Spain
enriquetomas@um.es

Mario Quiles Pérez
Department of Information and Communications Engineering
University of Murcia, 30100 Murcia, Spain
mqp@um.es

Sergio López Bernal
Department of Information and Communications Engineering
University of Murcia, 30100 Murcia, Spain
slopez@um.es

Alberto Huertas Celdrán
Communication Systems Group CSG
Department of Informatics IfI
University of Zurich UZH
CH—8050 Zürich, Switzerland
huertas@ifi.uzh.ch

Gregorio Martínez Pérez
Department of Information and Communications Engineering
University of Murcia, 30100 Murcia, Spain
gregorio@um.es

***Abstract*—Brain-Computer Interfaces are devices that enable two-way communication between an individual's brain and external devices, allowing the acquisition of neural activity and neurostimulation. Considering the first one, electroencephalographic signals are widely used for the acquisition of subjects' information. Therefore, a manipulation of the data acquired by a vulnerable BCI framework may cause a malfunction of the deployed applications. In this regard, this paper defines four noise-based cyberattacks attempting to generate fake P300 waves in two different phases of a BCI framework. A set of experiments show that the greater the attacker's knowledge regarding the P300 waves, processes, and data of the BCI framework, the higher the attack impact. In this sense, the attacker with less knowledge impacts 1% in the acquisition phase and 4% in the processing phase, while the attacker with the most knowledge impacts 22% and 74%, respectively.**

***Index Terms*—Brain-Computer Interfaces, Cybersecurity, Data Integrity, Electroencephalographic signal, P300**

Tipo de contribución: *Investigación ya publicada*

I. Introduction

Brain-Computer Interfaces (BCIs) allow the monitoring of neural signals or the stimulation of a set of neurons. These devices have been extensively used in medicine, offering diagnostic capabilities to detect abnormal behaviors in the brain or to treat neurological diseases such as Parkinson's. The advance of BCI technologies in the last decade has led to their use in other areas such as military, or driving assistance.

Electroencephalography (EEG) is the most common technique for neural data acquisition, consisting in acquiring brain waves from the scalp using non-invasive electrodes. Most of the current BCI applications are based on evoked potentials (ERPs), representing the users' neural response to particular external stimuli, like auditory or visual. These EEG signals are then processed and used as inputs to computational systems to make decisions. In this direction, one of the most used ERPs is the P300, which represents a positive voltage peak in the EEG when the user identifies a known visual or auditory stimuli from a set of unfamiliar stimuli.

Despite the relevance and possibilities of BCIs, these systems generate significant concerns in terms of cybersecurity. In recent years, numerous articles in the literature have focused on the lack of security measures in both BCI software and hardware. The relevance and value of the neural data obtained increase the criticality of BCI devices, which are vulnerable to the impact of cyberattacks that compromise their integrity, availability or confidentiality.

This work summarizes the research published in [1], which proposed a study of the impact of cyberattacks focused on maliciously generating P300 in the EEG signals to determine the impact on BCI devices. In particular, the study defines and implements four noise-based attack profiles that produce misclassification and thus anomalous operation in the interaction with BCI devices. The variation between profiles depends on the existing knowledge about the BCI device, aspects of the EEG signals and the framework.

II. Use Case and Experimental Setup

Since the knowledge of cyberattacks affecting BCIs is an open challenge, this work determined the impact of noise-based attacks in a real use case. Based on that, this work deploys a scenario consisting of three components: (1) a monitor where visual stimuli are presented to the subject, (2) a non-invasive BCI headset, and (3) a BCI framework that synchronizes the EEG signals with the visual stimuli and processes the data to detect P300 waves. These visual stimuli generate a reaction in the subject's brain waves based on the Oddball paradigm, whereby familiar visual stimuli (target) are presented within a set of unfamiliar ones (non-target).

The study defines four attacker profiles according to their knowledge of the BCI framework and the scenario. The selection is determined by the number of phases of the implemented BCI cycle: neural activity generation, EEG signals acquisition, processing, and P300 detection. Figure 1 shows the knowledge of the fourth attacker in the study. Each attacker generates two types of noise-based cyberattacks: (1) physical noise, in which noise is applied during neural acquisition, and (2) malware-based noise, in which noise is applied once the data have been processed.

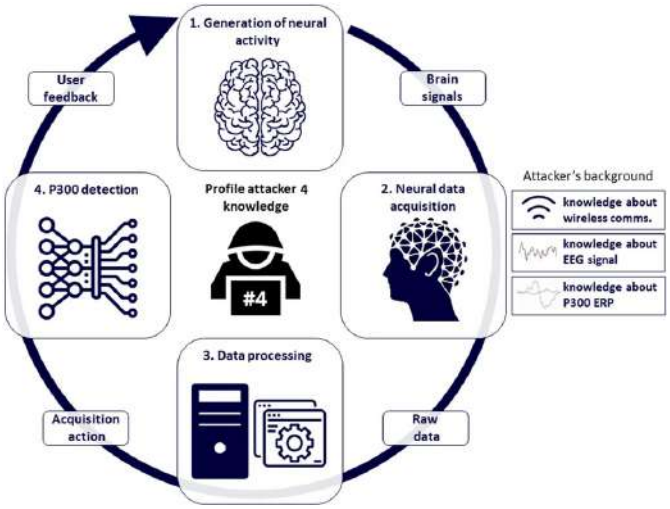


Figure 1. Four attacker: knowledge the P300 detection model details and outputs.

The noise generation is defined through Additive White Gaussian Noise (AWGN). Moreover, noise distributions are generated and applied based on the attacker's knowledge. On the one hand, static noise distributions are generated with a different power level for each of them. On the other hand, dynamic distributions are created by varying the power level during a specified period (see Table I). Figure 2 shows a fragment of dynamic noise generation performed by the fourth attacker. The objective is to adapt the signal to the generation of P300 potentials, which the P300 detector will identify.

Table I
TYPES OF NOISE GENERATED IN THE STUDY.

Type of noise	Power level	RMS noise level (dB)
Gaussian with static range	Low	≈ 0.8
Gaussian with static range	High	≈ 5
Gaussian with dynamic range	Adaptive	Between 0.8 to 5

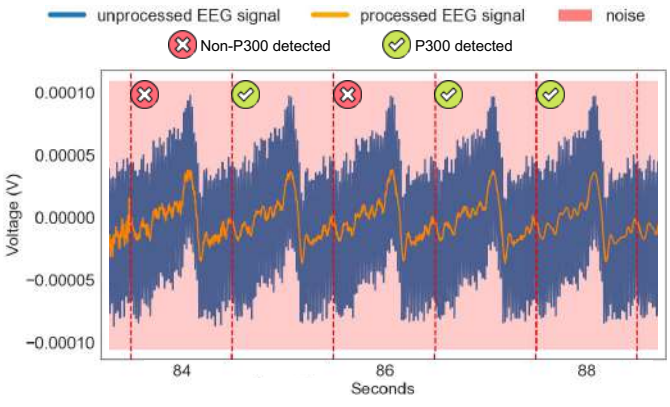


Figure 2. Noise-based cyberattack performed by the fourth attacker.

III. Result Analysis

The impact generated by each attack profile is measured from the BCI framework. In particular, it uses a set of Machine Learning (ML)-based models to provide an aggregated metric of performance attack using the Area Under the Curve (AUC) metric. A total of five classifiers for detecting P300 are implemented: (1) *C1* employs standardization algorithms and regressions, (2) *C2* uses Bayes' rule, (3) *C3* adds xDAWN spatial filter, (4) *C4* estimates based on a covariance matrix, and (5) *C5* employs the Minimum Distance to Mean. Since the goal of the attacks is to generate P300 waves in the EEG signals that does not contain them, the AUC value is obtained by evaluating only with EEG segments without a P300 wave.

Table II compares the AUC values obtained by each classifier, according to the noise behavior and the attacker profile. Also, the AUC values of the unaltered signal (legitimate signal) are included to estimate the impact of the attacker.

Table II
AUC VALUES OF CLASSIFIERS (CX) BY ATTACKER PROFILE AND NOISE BEHAVIOR. PN PHYSICAL NOISE, AND MN MALWARE-BASED NOISE.

	Attacker 1		Attacker 2		Attacker 3		Attacker 4		Legitimate signal
	PN	MN	PN	MN	PN	MN	PN	MN	
C1	0.74	0.72	0.74	0.70	0.68	0.45	0.60	0.20	0.75
C2	0.59	0.58	0.59	0.58	0.49	0.41	0.44	0.11	0.60
C3	0.54	0.53	0.52	0.52	0.50	0.31	0.47	0.10	0.54
C4	0.72	0.70	0.72	0.69	0.68	0.39	0.62	0.16	0.73
C5	0.78	0.79	0.77	0.71	0.70	0.47	0.62	0.21	0.79

The values demonstrate the slight progressive decrease of the AUC with physical noise in the different profile attacks. The decrease is between 1 and 22% concerning the legitimate signal, being 1% for the first profile and 22% for the fourth profile. On the other hand, the AUC values of the malware-based noise decrease between the second and third profiles, being 34% less, and between third and fourth profiles, being 55% less. Similarly, malware-based noise in the fourth profile has an impact of 74% in the legitimate signal. Therefore, generating noise in the processing phase by the fourth attacker profile has the most significant impact on the AUC, which translates into a high identification of P300 potentials.

IV. Conclusion

This paper presents four attacker profiles that generate noise-based cyberattacks affecting BCI frameworks. Two types of noise are generated for each attacker: (1) physical, affecting the acquisition phase of EEG signals, and (2) malware-based, impacting the processing phase. To test them, this work presents a scenario based on visual stimuli with the aim of generating P300 waves and acquiring them with a non-invasive BCI headset. The experimentation indicates that the proposed cyberattacks allow affecting EEG signals, where the attacker with the greatest knowledge of the BCI cycle has the greatest impact. Likewise, cyberattacks in the processing phase have a greater impact on the generation of the P300, making it a point of great interest for potential attackers.

Acknowledgements

This work has been partially supported by *(a)* the Swiss Federal Office for Defense Procurement (armasuisse) with the RESERVE project (CYD-C-2020003), and *(b)* 21629/FPI/21, Fundación Séneca, Región de Murcia (Spain).

References

[1] Martínez Beltrán, E. T., Quiles Pérez, M., López Bernal, S., Huertas Celdrán, A., and Martínez Pérez, G.: "Noise-based cyberattacks generating fake P300 waves in brain–computer interfaces", *Cluster Computing*, vol. 25, pp. 33-48, 2021.

A Review of "Evaluation of the Executional Power in Windows using Return Oriented Programming"

Daniel Uroz, Ricardo J. Rodríguez
Dpto. de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Spain
duroz@unizar.es, rjrodriguez@unizar.es

***Abstract*—Code-reuse techniques have emerged as a way to defeat the control-flow defenses that prevent the injection and execution of new code, as they allow an adversary to hijack the control flow of a victim program without injected code. A well-known code-reuse attack technique is Return-Oriented Programming (ROP), which considers and links together (relatively short) code snippets, named ROP gadgets, already present in the victim's memory address space through a controlled use of the stack values of the victim program. Although ROP attacks are known to be Turing-complete, there are still open question such as the quantification of the executional power of an adversary, which is determined by whatever code exists in the memory of a victim program, and whether an adversary can build a ROP chain, made up of ROP gadgets, for any kind of algorithm. To fill these gaps, in this paper we first define a virtual language, dubbed ROPLANG, that defines a set of operations (specifically, arithmetic, assignment, dereference, logical, and branching operations) which are mapped to ROP gadgets. We then use it to evaluate the executional power of an adversary in Windows 7 and Windows 10, in both 32- and 64-bit versions. In addition, we have developed ROP3, a tool that accepts a set of program files and a ROP chain described with our language and returns the code snippets that make up the ROP chain. Our results show that there are enough ROP gadgets to simulate any virtual operation and that branching operations are the less frequent ones. As expected, our results also indicate that the larger a program file is, the more likely to find ROP gadgets within it for every virtual operation.**

***Index Terms*—ROP chain, evaluation, Turing-completeness, Windows, automatic exploit**

Tipo de contribución: *Investigación ya publicada en* "Evaluation of the Executional Power in Windows using Return Oriented Programming," 2021 IEEE Security and Privacy Workshops (SPW), 2021, pp. 361-372. [1]

I. Extended Abstract

Software systems have increased in complexity and in size (measured as lines of code) during the last years. Nowadays, large software development teams are involved in several software projects at the same time, having a fixed time-to-market that urges them to end the development cycle as fast as possible, regardless of the software quality. Although automatic methods exist to improve the software quality, software vulnerabilities have dramatically increased, opening a window of opportunity to malicious exploitation [2].

Many of these vulnerabilities lead to control-hijacking attacks, which are the most popular category of memory exploits nowadays [3]. These attacks use code injection or its evolution, code-reuse attacks, to hijack the legitimate control flow of a victim program and execute malicious code. As a consequence, several defense approaches for control-hijacking attacks have been proposed, aiming to guarantee that the control flow of a program legitimately prevails. Examples of these include the use of stack cookies, inline software guards, runtime elimination of memory errors, control-flow integrity (CFI), protection of data and code pointers, address space layout randomization (ASLR), and write-xor-execute (W⊕X; also known as data-execution prevention, DEP).

Code-reuse techniques have emerged as a trend of advanced threats to mitigate the effects of the control-flow defenses that prevent the injection and execution of new code. These techniques allow an adversary to hijack the control flow of a victim program to perform malicious activities without injected code.

Return-Oriented Programming (ROP) is a code-reuse attack technique presented in 2007 for the x86 architecture (as an evolution of the *return-to-libc* attacks) [4]. ROP attacks have been demonstrated feasible in numerous architectures, such as RISC, Linux/86 and Solaris/SPARC architectures, and recently in RISC-V. In particular, ROP considers and links together (relatively short) code snippets already present in the process's memory address space, named as *ROP gadgets*. Each code snippet ends with an instruction that changes the program control flow (e.g., a ret instruction in Intel architectures), thus allowing an attacker who controls the stack to *chain* them together, controlling the order of code execution through the stack values. A chain of ROP gadgets is normally termed as a *ROP chain*. As the ROP chain links together these code snippets stored in memory pages marked as executable, ROP is able to evade control-flow defenses such as W⊕X.

ROP attacks are defeated with other control-flow defenses such as CFI. The security analysis of CFI solutions was first carried out in [5], raising questions on the true effectiveness of these solutions. The recent work in [6] presents a solution to precisely measure and verify the effectiveness of existing CFI solutions.

Modern operating systems such as Windows 10 incorporate native defense techniques based on CFI, as Control-Flow Guard (CFG) [7]. CFG prevents the exploitation of memory corruption vulnerabilities, ensuring that the control-flow of the program remains legitimate. This defense is implemented in kernel-space and at program execution the targets of indirect branches are checked to verify whether they are valid targets. CFG, though, is not system-wide as it only works with "CFG-aware" programs, i.e., programs that are compiled with this feature enabled. Hence, CFG requires support from the compiler and the operating system to fully implement it. Unfortunately, not many Windows programs incorporate this feature at the moment of this writing.

A natural question that arises in this context is to analyze the capabilities of an adversary, regardless of the control-flow defenses put in place. ROP attacks are known to be Turing-

complete [8], i.e., *ROP attacks are capable of any arbitrary computation*. Likewise, enough ROP gadgets to make up a Turing-complete set of operations have been found in Linux environments [9]. However, a question that still remains open is the quantification of the *executional power* of an adversary, which is determined by whatever code exists in the memory of a victim program. Noting that the ROP chain built by an adversary can be seen as the implementation of an algorithm designed to perform a desired task, if an adversary can find enough ROP gadgets for any arbitrary operation thus any algorithm can be implemented with a ROP attack. The more existing code is in a victim program, the more likelihood there is for finding useful gadgets [4]. Likewise, if the adversary is unable to find a ROP gadget for a specific operation needed in the ROP chain, the attack will likely fail.

Formally speaking, any real world computation can be translated into an equivalent computation that involves a Turing machine under the Church-Turing thesis [10]. Assuming this thesis holds, we can build a Turing machine that performs equivalent computations to the operations performed by a ROP chain.

In this paper, we define a Turing-complete set of operations that make up a virtual language. This virtual language, dubbed ROPLANG, defines a set of operations that are later mapped to specific ROP gadgets, thus representing a ROP chain in an abstract way. Specifically, we categorize operations in arithmetic, assignment, dereference, logical, and branching operations.

We have developed a tool, dubbed `ROP3`, that accepts as input a set of program files and a ROP chain described with ROPLANG, and returns the ROP gadgets that make up the ROP chain. Our tool is built on top of the Capstone disassembly framework [11] and is designed to work with the Windows binary file format. In addition, our tool was designed in a modular way to facilitate the adoption of other file formats and to extend the virtual language. For the sake of open science, we have released our software under the GNU/GPLv3 license [12]. In addition, `ROP3` is also a Python3 library, which facilitates the integration with other analysis pipelines.

We then use `ROP3` to find the ROPLANG operations to quantify the executional power of an adversary in a given environment. In particular, we evaluate the executional power of an attacker in Windows 7 and Windows 10, in its x86 and x86-64 versions, as Windows is still the predominant platform targeted by attackers [13]. For each Windows flavor, we have considered only the subset of system default dynamic link libraries (DLLs) contained in `KnownDlls` that are common across all the versions of Windows considered for the experimentation. This subset of DLLs is composed of a total of 20 DLLs, and we have also considered other DLLs such as `msvcrt.dll`, `psapi.dll`, `ws2_32.dll`, and `ntdll.dll`.

Our experiments show that the branching virtual operations are the less frequent operations, regardless of the architecture. Moreover, in 32 bits there are no results for any of the comparison operations as currently defined in ROPLANG, regardless of the Windows version. In fact, these operations only appear in one DLL of Windows 7 SP1 64-bit. The case of the unconditional branch operation is very interesting, as we have (few) results in 32-bit versions, while none in 64-bit. For the rest of virtual operations, the results are diverse, although we can find at least one result for each virtual operation. The results tend to be higher in bigger DLLs. As claimed by other authors [12], the longer the binary code is, the more likely to find ROP gadgets to perform any arbitrary operation. Our experimental results show us that an adversary can very likely find a ROP gadget for any arbitrary operation, just doing sophisticated linking of other operations when the operation needed is not directly found.

The full version of this paper (with a full description of the experiments) was published in [1].

Acknowledgments. This work was supported in part by the Aragonese Government under *Programa de Proyectos Estrategicos de Grupos de Investigación* (DisCo research group, ref. T21-17R). The research of D. Uroz was also supported by the Government of Aragon under a DGA predoctoral grant (period 2019-2023).

REFERENCES

[1] D. Uroz and R. J. Rodríguez, "Evaluation of the executional power in windows using return oriented programming," in *Proceedings of the 15th IEEE Workshop on Offensive Technologies (WOOT)*. IEEE, 2021, pp. 361–372.

[2] P. Johnson, D. Gorton, R. Lagerström, and M. Ekstedt, "Time between vulnerability disclosures: A measure of software product vulnerability," *Computers & Security*, vol. 62, pp. 278–295, 2016.

[3] V. van der Veen, N. dutt Sharma, L. Cavallaro, and H. Bos, "Memory Errors: The Past, the Present, and the Future," in *Proceedings of the 15th International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)*, D. Balzarotti, S. J. Stolfo, and M. Cova, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 86–106.

[4] H. Shacham, "The Geometry of Innocent Flesh on the Bone: Return-into-libc Without Function Calls (on the x86)," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*. New York, NY, USA: ACM, 2007, pp. 552–561.

[5] L. Davi, A.-R. Sadeghi, D. Lehmann, and F. Monrose, "Stitching the Gadgets: On the Ineffectiveness of Coarse-Grained Control-Flow Integrity Protection," in *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, 2014, pp. 401–416.

[6] Y. Li, M. Wang, C. Zhang, X. Chen, S. Yang, and Y. Liu, "Finding Cracks in Shields: On the Security of Control Flow Integrity Mechanisms," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1821–1835.

[7] Microsoft, "Control Flow Guard," Online; https://docs.microsoft.com/en-us/windows/win32/secbp/control-flow-guard., May 2018, accessed on April 19, 2022.

[8] M. Tran, M. Etheridge, T. Bletsch, X. Jiang, V. Freeh, and P. Ning, "On the Expressiveness of Return-into-libc Attacks," in *Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection (RAID)*, R. Sommer, D. Balzarotti, and G. Maier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 121–141.

[9] A. Homescu, M. Stewart, P. Larsen, S. Brunthaler, and M. Franz, "Microgadgets: Size Does Matter in Turing-Complete Return-Oriented Programming," in *Proceedings of the 6th USENIX Workshop on Offensive Technologies*. Berkeley, CA: USENIX, 2012.

[10] B. J. Copeland, "The Church-Turing Thesis," in *The Stanford Encyclopedia of Philosophy*, 2020th ed., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2020, accessed on April 19, 2022.

[11] Capstone, "Capstone – The Ultimate Disassembler," Online, http://www.capstone-engine.org/, accessed on April 19, 2022.

[12] D. Uroz and R. J. Rodríguez, "rop3 version 1.0," [Online; https://github.com/reverseame/rop3], Mar. 2021, accessed on April 19, 2022.

[13] R. Benzmüller, "Malware trends 2017," [Online; https://www.gdatasoftware.com/blog/2017/04/29666-malware-trends-2017], Oct. 2017, accessed on April 19, 2022.

A review of: Optimal Feature Configuration for Dynamic Malware Detection

David Escudero García
Research Institute of Applied science in Cybersecurity
Campus de Vegazana s/n, 24071, León, Spain
descg@unileon.es

Noemí DeCastro-García
Universidad de León
Campus de Vegazana s/n, 24071, León, Spain
ncasg@unileon.es

***Abstract*—Applying machine learning techniques to malware detection is a common approach to try to overcome the limitations of signature-based methods. However, it is difficult to engineer a set of features that characterizes the samples properly, especially when various file types may be a vector of infection. In this work, we configure several feature sets for dynamic malware detection extracted from API calls, network activity, signatures from the Cuckoo sandbox report, and some interactions with the file system and registry. We test combinations of these feature sets to ascertain whether they are good enough to distinguish between benign and malicious samples from a dataset containing several file types, obtained from public sources. The datasets present class imbalance to evaluate the model performance on more realistic scenarios in which not many malware samples are available.**

***Index Terms*—Machine Learning, Malware detection, Feature engineering**

Contribution Type: *Published Research*

I. Introduction

In this paper we present a work previously published in the *Computers & Security* journal in 2021 [1].

The threat presented by malware affects both, individual users and organizations. Since signature based approaches require the file to have been previously identified as malware in order to protect the user, research efforts have been directed towards achieving more intelligent detection, usually by applying machine learning.

Most of the research is focused on the detection of malware in executable files from either Windows [2] or Android [3]. However, according to a Verizon report, 45% of malware distribution is carried out through Office documents, with executables at 26% [4]. In addition, datasets with malicious samples usually present class imbalance that may bias the classifiers [5].

Therefore, in this work, we aim to provide an optimal feature configuration for malware detection on different file types (.pdf, .docx, .exe, .html, .xlsx), using machine learning techniques in the presence of class imbalance. The optimal configuration is obtained by analyzing statistically meaningful differences among the results of the models that are constructed with different combinations of feature sets.

We employ dynamic analysis for the construction of the feature sets. We extract different feature sets derived from API calls, network traffic, and the signatures provided by Cuckoo, combined with interaction with the file system and registry, which are commonly used in the literature. We decide to focus on dynamic analysis since, when successful, it provides a more accurate, file-agnostic characterization of sample behavior. This fact allows us to uniformly extract features from several file formats so that a greater number of related problems may be tackled. Otherwise, a different set of static features should be engineered for each file type.

II. Datasets

We have collected 19994 file samples, both benign and malicious, from different public sources. A total of 9999 malicious samples were downloaded from VirusShare [6] and 9995 benign samples were obtained from Digital Corpora [7] and files extracted from local computers. The distribution of file types is described in Table I.

Table I
Distribution of File Types in our File Set

	Word	Excel	HTML	PDF	Executable
Benign	2999	1999	2000	2499	498
Malicious	2769	233	3492	1006	2499

The analysis of files is carried out using the Cuckoo sandbox v2.0.7, using a Windows 7 VM for analysis. The machine was made more vulnerable by disabling several security options such as the firewall and user account control. The JSON report is used to extract API and signatures features and the pcap trace is used for the extraction of network features.

The feature sets used are the following[1]:

- API. Frequencies of categories of calls (file, network, etc.) and of its 2-grams. 342 features in total.
- Network. Summary statistics of the network traffic such as mean and standard deviation of certain quantities such as the bytes sent and received per second and duration of flows. 199 features in total.
- Signatures. Boolean features corresponding to the signatures provided by Cuckoo. In addition, some statistics regarding the interaction with the file system and registry such as the number of files written or types of keys created. 594 features in total.

We have carried out experiments with different combinations of the feature sets to determine which is the most favorable. In addition, we also introduce different degrees of imbalance by modifying the proportion of malware in the dataset. We use proportions of malware of 30%, 20%, 10% and 5%. The combinations of feature sets used are shown in Table II.

[1]Available at https://drive.google.com/drive/folders/173SO6RmkdmWa-5fM7xOz5ZL20eniPMQ-?usp=sharing.

Table II
DESCRIPTION OF FEATURE SETS USED IN EXPERIMENTS

Code	Feature set
$\mathcal{F}_4$	Network + signatures
$\mathcal{F}_5$	API + network + signatures
$\mathcal{F}_6$	API + network
$\mathcal{F}_7$	API + signatures
$\mathcal{F}_i^j$	unbalanced set of features $\mathcal{F}_i$ with a $j\%$ proportion of malicious samples

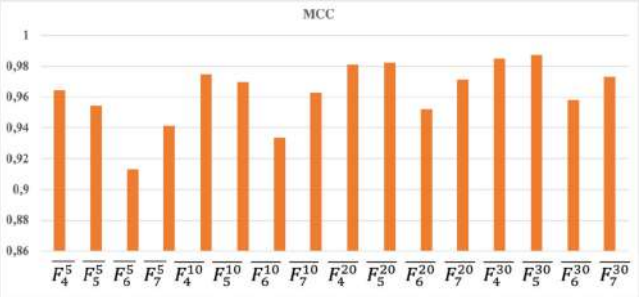


Figure 1. Median MCC for each dataset

III. EXPERIMENTS

We have used the Auto-sklearn library [8] in its 0.7.1 version to construct the models. Auto-sklearn requires that the user sets a time limit for the search of configurations. The maximum time for model construction is set to 15 minutes. We use 10-fold cross-validation as evaluation strategy and use the Matthews correlation coefficient (MCC) as metric, since it is more informative than accuracy in unbalanced datasets. It is defined in Eq. 1.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (1)$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively.

This process is repeated 50 times for each dataset. We apply Friedman's test followed by tests for two-paired samples to determine whether there are significant differences between the performances of different combinations of feature sets.

IV. RESULTS

The results of the statistical tests show that there are statistically significant differences between the MCC achieved for different feature sets, so we proceed to compare their prediction performance.

The MCC achieved in each dataset is shown in Fig. 1. Usually, we can consider a model good enough if it achieves a MCC greater than 0.95, without taking into account specific application requirements regarding false positives and false negatives. In our experiments, datasets $\mathcal{F}_4$ and $\mathcal{F}_5$ exceed that threshold for all degrees of imbalance and are in general the best performant. $\mathcal{F}_4$ achieves better results than $\mathcal{F}_5$ in greater degrees of imbalance despite both sets sharing network and signatures features, which signals that the API set may be redundant with others given its low-level nature.

In order to give a more accurate vision of model performance, Fig. 2 shows the true positive rate (sensitivity) and true negative rate (specificity) achieved for each dataset. A model

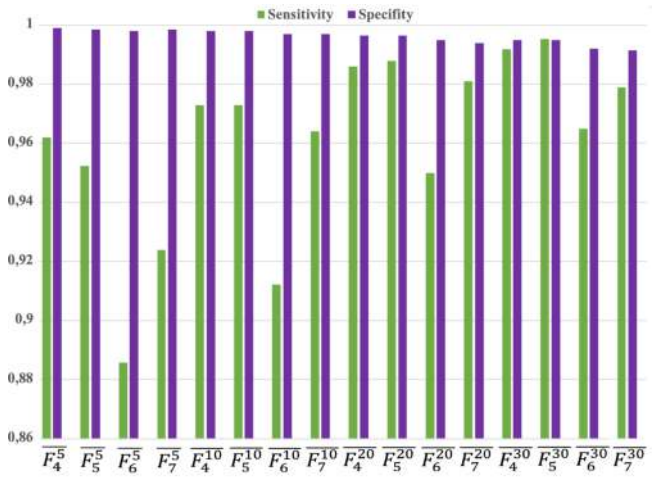


Figure 2. Median sensitivity and specificity for each dataset

with high sensitivity will detect most malicious samples and will not have many false negatives. A high-specificity model will correctly categorize benign samples and will not generate many false positives. As can be seen, the obtained models achieve higher specificity, nearing 1, which could be expected since benign samples are the majority class. In addition, sensitivity is also over 0.95 even at the highest degree of imbalance with $\mathcal{F}_4$ and $\mathcal{F}_5$.

V. CONCLUSIONS

In the original work we extracted several feature sets from dynamic analysis and tested their effectiveness in the problem of detecting malware across several file types in the present of class imbalance. The results show that the selected features are effective at detecting malware in our setting. In particular, the combination of network and signatures set achieves the highest results, with an MCC higher than 0.95 for all degrees of imbalance.

ACKNOWLEDGEMENTS

This work was partially supported by the Spanish National Cybersecurity Institute (INCIBE) under contract Art.83, key: X54. Also, we thank Ángel Luis Muñoz Castañeda for his advice regarding the manuscript.

REFERENCES

[1] D. Escudero-García and N. deCastro García, "Optimal feature configuration for dynamic malware detection," *Computers & Security*, 2021.
[2] W. Han, J. Xue, Y. Wang, Z. Liu, and Z. Kong, "Malinsight: A systematic profiling based malware detection framework," *Journal of Network and Computer Applications*, vol. 125, pp. 236–250, Jan. 2019.
[3] Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: Android malware characterization and detection using deep learning," *Tsinghua Science and Technology*, vol. 21, no. 1, pp. 114–123, Feb. 2016.
[4] Verizon, "2019 data breach investigations report," 2020. [Online]. Available: https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf
[5] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," *Journal of Network and Computer Applications*, vol. 153, Mar. 2020.
[6] Corvus Forensics, "Virusshare," 2011, accessed: Dec. 2019. [Online]. Available: https://virusshare.com/
[7] Digital Corpora, "Govdocs1 — (nearly) 1 million freely-redistributable files," 2018. [Online]. Available: https://digitalcorpora.org/corpora/files
[8] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proc. of 28 Conf. in Adv. in Neural Inf. Process. Syst.*, 2015, pp. 2962–2970.

Classifying suspicious Pastebin content using Machine Learning

Daniel Diaz
Dpto. IESA, Univesidad de León
Researcher at INCIBE
León, Spain
ddiao@unileon.es

Francisco Jañez-Martino
Dpto. IESA, Univesidad de León
Researcher at INCIBE
León, Spain
francisco.janez@unileon.es

Javier Velasco-Mata
Dpto. IESA, Univesidad de León
Researcher at INCIBE
León, Spain
javier.velasco@unileon.es

*Eduardo Fidalgo
Dpto. IESA, Univesidad de León
Researcher at INCIBE
León, Spain
eduardo.fidalgo@unileon.es

Oscar Garcia
Researcher at INCIBE
León, Spain
oscar.garcia@incibe.es

Enrique Alegre
Dpto. IESA, Univesidad de León
Researcher at INCIBE
León, Spain
enrique.alegre@unileon.es

***Abstract*—Pastebin is a notepad service where users can share text online, but cyber-criminals also used it to advertise hacking activities or distribute stolen information, among other activities. The automatic classification of these contents could help authorities to detect text related to suspicious or even illegal activities. However, due to the lack of labelled data, models cannot be properly trained to recognise these texts. To overcome this problem, we created PasteCC_17K, a new and publicly available dataset focused on suspicious texts from Pastebin. We trained three Machine Learning algorithms with the created dataset and achieved a F1-score of 98.70% and an accuracy of 98.63% combining a TF-IDF encoder with a Logistic Regression classifier.**

***Index Terms*—Pastebin, Text Classification, Bag of Words**

Contribution: Summary of *Classifying Pastebin content through the generation of PasteCC_17K labeled dataset.* [1]

I. Introduction

Notepad websites like Codepad[1] or Pastebin[2] allow users to share information among them, such as programming code or personal notes. However, these services also allow sharing content anonymously and thus can be used to share links to access illegal activities like child pornography, drug sales, or leak stolen information as presented in Fig. 1. Furthermore, it is difficult to manually detect such activities among the large volume of data posted on those sites. For example, in 2016, the last date it was updated, Pastebin registered a monthly average of 14 million of *pastes* (shared texts) [2]. For this reason, the automatic classification of such content might help authorities to find pastes related to suspicious or even illegal activities. Our contributions on this matter are the following:

- We created a novel dataset with labelled data from Pastebin samples of suspicious content.
- We evaluated the performance of six machine learning pipelines using our dataset to make an initial recommendation for the task of automatically classify suspicious pastes.

[1]http://codepad.org/
[2]https://pastebin.com/

Fig. 1. Example of leaked personal information on Pastebin

II. Background

Paul Dixon created Pastebin in 2002 as an unauthenticated web application for fast and comfortable post sharing. Since then, the use of this platform for suspicious or even illicit activities has grown. Hackers have recently used this network to publish confidential information on both people and institutions such as universities [3]. To monitor Pastebin, traditional methods use a predefined list of keywords and regular expressions, but those methods are costly and inefficient.

Using Natural Language Processing, the task of Text classification allows to categorise texts based on their content. Authors like Zhu et al.[4] and Lochter et al.[5] presented and evaluated several machine learning classifiers such as Logistic Regression (LR), Support Vector Machine (SVM), Naive Bayes (NB), C4.5 and K-Nearest Neighbours (KNN) for the task of Text Classification.

Although there are Deep Learning models that obtained higher results than traditional approaches [6] in some text classification tasks, they require large amounts of labelled data to be trained, which are not available for this problem.

III. Methodology

A. Dataset creation

The creation of the dataset was motivated by the lack of labelled data on suspicious Pastebin content. Initially, we crawled more than 500K pastes from Pastebin, and we

TABLE I
CATEGORY REPRESENTATION IN THE PASTECC_17K DATASET.

Category type	Category	Samples	Percent(%)
Suspicious 1.23%; 217 Samples			
	Hacking	86	0,49
	Counterfeit Credit Card	67	0,38
	Child-pornography	29	0,16
	Leaked-Data	19	0,11
	Drugs	14	0,08
	Counterfeit Personal Information	2	0,01
Normal 98.70%; 17423 Samples			
	Source Code	12396	70,27
	Multimedia	4719	26,75
	Forum	90	0,51
	Others	81	0,46
	Logs	65	0,37
	Encoded Text	25	0,14
	Crytocurrency	20	0,11
	General-pornography	17	0,10
	Personal	10	0,06
Total		**17640**	

TABLE II
RESULTS OF TRAINING CLASSIFIER WITH PASTECC_17K (%)

		TF-IDF LR	TF-IDF SVM	TF-IDF NB	BOW LR	BOW SVM	BOW NB
Accuracy		**98.63**	97.80	97.52	98.37	97.62	98.05
Precision	Macro	71.60	72.50	21.10	58.80	62.90	41.70
	Micro	98.60	97.80	97.50	98.40	97.60	98.10
	Weighted	98.70	98.80	95.40	98.70	98.70	97.50
Recall	Macro	72.10	63.20	21.00	63.20	59.90	38.10
	Micro	98.60	97.80	97.50	98.40	97.60	98.10
	Weighted	98.60	97.80	97.50	98.40	97.60	98.10
F1 Score	Macro	**72.10**	62.20	21.10	58.70	58.30	35.64
	Micro	**98.70**	97.80	97.50	98.40	97.60	98.10
	Weighted	**98.70**	98.10	97.50	98.50	98.00	97.70

manually labelled 804 samples. Then, we added the Pastebin samples from the DUTA-10K dataset [7] to the labelled data. We labelled the rest of the samples automatically using a LR model to predict the class of unlabelled samples when its confidence was higher than a certain threshold. In particular, we tested three thresholds over 100 samples each, 85%, 90% and 95%. We selected the 90% to label the rest of the data since it achieved a similar accuracy than the 95% one, but provided significantly more samples. This left us with a total of 17K labelled samples. PasteCC_17K features are summarised in Table I.

B. Text preprocessing, encoding and classification

First, we cleaned the extracted texts by removing the following elements: the stop-words, the format files, URL links, currency units, single letters, long words, repeated text, numbers, words with numbers and special characters.

After the cleaning, we encoded the texts into feature vectors using two well-known methods focused on representing the text by the frequencies of appearance of its words: Term Frequency-Inverse Document Frequency (TF-IDF) [8] and Bag of Words (BoW) [9].

Finally, we evaluated the PasteCC_17K dataset with three Machine Learning classifiers: SVM, LR, and NB [10].

IV. EXPERIMENTS AND RESULTS

To classify the pastes, we tested two encoding techniques, TF-IDF and BoW, and three classifiers, SVM, LR, and NB, whose results appear in Table II. In all cases, we trained the models with 70% of the samples and tested them with the remaining 30%. To cope with the heavy unbalance of the dataset –the categories Source Code and Multimedia represent 97.02% of the total dataset– we reported the precision, recall and F1-score using three averages: macro, micro and weighted. The best pipeline was the combination of TF-IDF as feature extractor combined with the LR classifier, with an weighted F1-Score of 98.70%.

V. CONCLUSIONS AND FUTURE WORK

This work addressed the problem of finding content that could be illegal in pastes of notepad websites such as Pastebin, classifying their content into different categories. Those categories were divided into two blocks: legal or suspected of being illegal. Due to the lack of publicly available datasets on this problem, we collected a dataset and semi-automatically labelled it, grouping $17,640$ examples into 15 categories where the minority of the examples correspond to suspicious activities (1.23%). In those samples were found content with advertising of hacking offers, and even Child Sexual Exploitation Material links.

In addition, we evaluate the performance of two encoding techniques combined with three classifiers, finding that all the classifiers obtained an accuracy higher than 97%. The best results were obtained with the TF-IDF encoding combined with the LR classifier and the worst performance was given by the BOW combined with LR.

For future work, we aim to use Active Learning techniques to ease the task of labelling unknown samples.

VI. ACKNOWLEDGEMENT

This work was supported by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01, and by the FPU (Formación de Profesorado Universitario) grant of the Spanish Government with reference FPU18/05804.

REFERENCES

[1] A. Riesco, E. Fidalgo, M. W. Al-Nabki, F. Jáñez-Martino, and E. Alegre, "Classifying pastebin content through the generation of pastecc labeled dataset," in *HAIS*. Springer, 2019, pp. 456–467.
[2] H. Herath, "Web information extraction system to sense information leakage," Ph.D. dissertation, University of Moratuwa Sri Lanka, 2017.
[3] N. Perlroth, "Hackers breach 53 universities and dump thousands of personal records online," *New York Times, New York*, 2012.
[4] D. Zhu and K. W. Wong, "An evaluation study on text categorization using automatically generated labeled dataset," *Neurocomputing*, vol. 249, pp. 321–336, 2017.
[5] J. V. Lochter, R. F. Zanetti, D. Reller, and T. A. Almeida, "Short text opinion detection using ensemble of classifiers and semantic indexing," *Expert Systems with Applications*, vol. 62, pp. 243–249, 2016.
[6] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, "Starspace: Embed all the things!" in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.
[7] M. W. Al Nabki, E. Fidalgo, E. Alegre, and I. De Paz, "Classifying illegal activities on tor network based on web textual contents," in *European Chapter of the Association for Computational Linguistics*, 2017, pp. 35–43.
[8] A. Aizawa, "An information-theoretic perspective of TF-IDF measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003.
[9] Z. Harris, "Distributional structure," *Word*, vol. 10, pp. 146–162, 1954.
[10] R. M. Silva, T. A. Almeida, and A. Yamakami, "Mdltext: An efficient and lightweight text classifier," *Knowledge-Based Systems*, vol. 118, pp. 152–164, 2017.

A Review of Neuronal Jamming Cyberattack over Invasive BCIs Affecting the Resolution of Tasks Requiring Visual Capabilities

Sergio López Bernal
Department of Information
and Communications Engineering
University of Murcia
30100 Murcia, Spain
slopez@um.es

Alberto Huertas Celdrán
Communication Systems Group CSG
Department of Informatics IfI
University of Zurich UZH
CH–8050 Zürich, Switzerland
huertas@ifi.uzh.ch

Gregorio Martínez Pérez
Department of Information
and Communications Engineering
University of Murcia
30100 Murcia, Spain
gregorio@um.es

***Abstract*—Invasive Brain-Computer Interfaces (BCIs) are used in medical scenarios to record, stimulate, or inhibit neural activity. Despite their advances, BCIs present vulnerabilities that attackers can exploit to affect neuronal activity. In this direction, this work designs and implements a novel neuronal cyberattack, Neuronal Jamming (JAM), that prevents neuronal spiking from occurring. Based on a lack of realistic neuronal topologies, we have trained a Convolutional Neural Network (CNN) to simulate a mouse exiting a particular maze. This model is translated to a biological neural topology and simulated to represent a portion of a mouse's visual cortex. The JAM impact has been evaluated over both biological and artificial networks, verifying that it can affect both neuronal activity and the ability of the mouse to exit the maze. Moreover, this work compares the impact of JAM and Neuronal Flooding (FLO), an existing Neural Cyberattack, highlighting that JAM presents a higher impact on neuronal spike rate.**

***Index Terms*—Cybersecurity, Safety, Neural Cyberattacks, Brain-Computer Interfaces**

Tipo de contribución: *Investigación ya publicada*

I. Introduction

BCIs provide bidirectional communications between the brain and external devices. Although they are being used in various scenarios, one of the most relevant is in health care, where invasive neurostimulation BCIs help treat neurological conditions such as Parkinson's disease or epilepsy by stimulating or inhibiting neuronal activity.

Current advance in BCIs also introduces cybersecurity concerns since modern neurostimulation systems present vulnerabilities. Taking advance of them, previous work has presented the concept of Neural Cyberattacks, novel cyberattacks that aim to disrupt spontaneous neuronal activity. Previous work from us presented, for the first time, two of these cyberattacks: FLO, aiming to overstimulate a set of neurons in a particular instant simultaneously; and Neuronal Scanning (SCA), sequentially stimulating a set of neurons, demonstrating that these attacks can indeed alter neural activity.

This work summarizes the research published in [1], whose main contribution is the definition and implementation of JAM cyberattacks, which focus on the inhibition of neural activity. We simulate a portion of the mice's visual cortex, defining a use case of a mouse that has to exit a determined maze. Since there is an absence of realistic neuronal topologies, the connections of the neurons in the network and their synaptic weights are extracted from training a CNN as their structure and function presents similarities with the visual cortex. The second contribution is evaluating the impact of JAM cyberattacks on both biological and artificial simulations. Moreover, the third contribution is comparing the impact between JAM and FLO from both biological and artificial perspectives. The final contribution is a comparison between biological and artificial scenarios based on linear correlation analysis between variables [1].

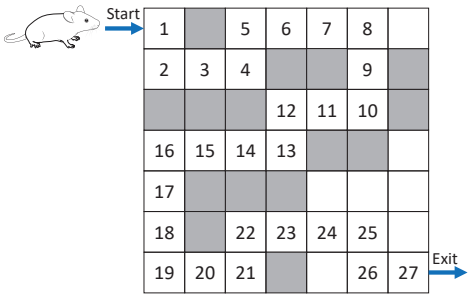


Figure 1. Maze used to model the movement of the mouse, including the optimal path between the starting and final cells.

II. Experimental Setup

Due to the absence of detailed neocortical topologies in mammalians, this work trains a CNN to solve the problem of a mouse trying to exit a determined maze (see Figure 1). This CNN is composed of two convolutional layers of 200 and 72 nodes, respectively, and a third dense layer of four nodes. The topological connections are then translated to synapses, and the weights of the model are used as synaptic weights. Based on that, a small section of the visual cortex of a mouse is simulated for 27 seconds, where the mouse stays one second per position of the optimal path of the maze. Moreover, the behavior of the neurons is represented by the Izhikevich neuronal model, receiving inputs from the surrounding cells of the maze. After implementing both scenarios, different metrics have been used to measure the impact of the attacks: the number of spikes and the temporal dispersion metrics in the biological approach, while the number of the steps to complete the maze and the success rate are used to evaluate the CNN.

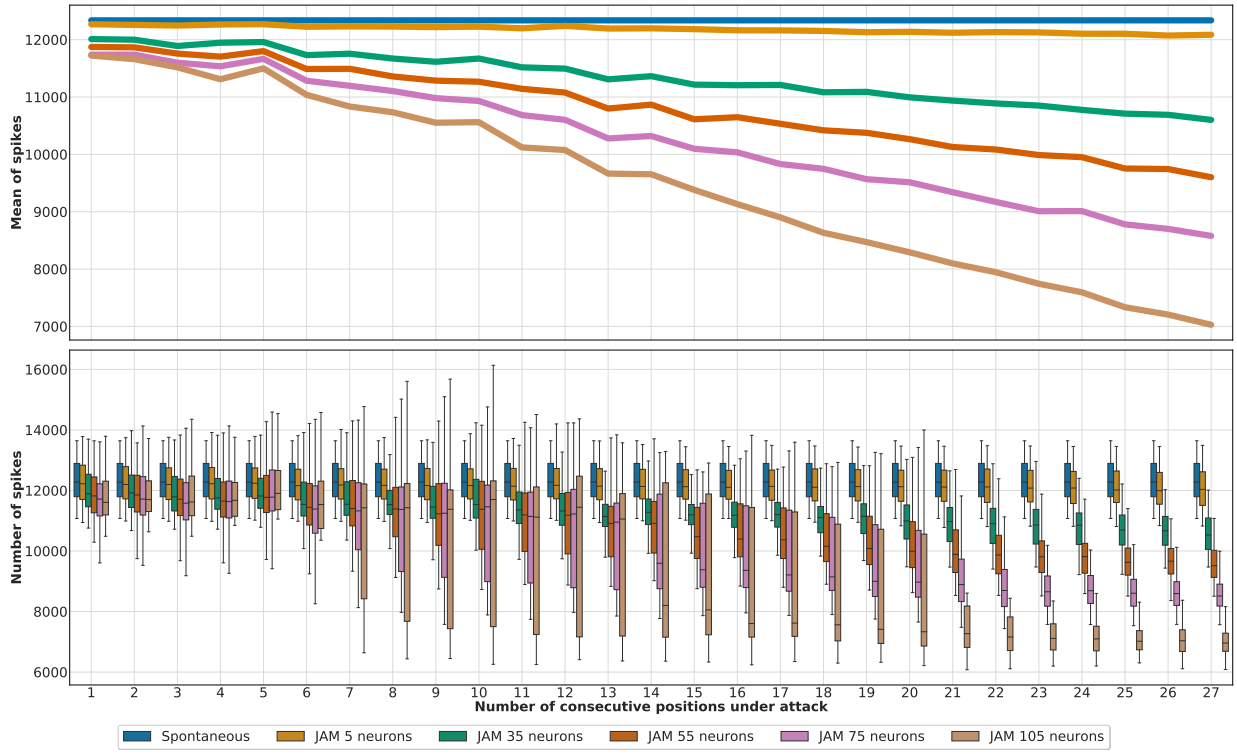


Figure 2. Distribution of the number of spikes based on the consecutive number of positions attacked for JAM cyberattacks.

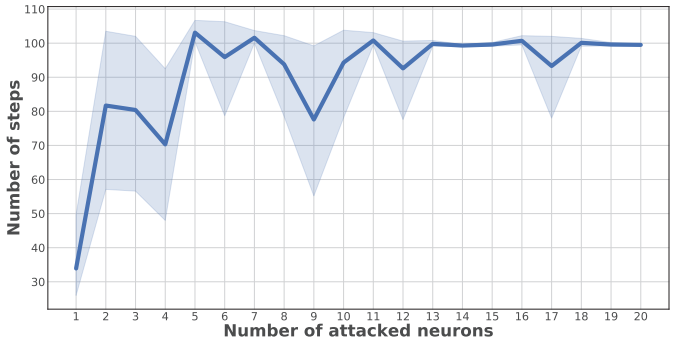


Figure 3. Number of steps for a range between one and 20 attacked neurons, with ten executions, for JAM cyberattacks.

III. Impact Caused by Neural Cyberattacks

The impact of JAM cyberattacks on the biological simulation considering the number of spikes is presented by Figure 2, which highlights that increasing the number of consecutive positions reduces the neural activity. Moreover, increasing the number of attacked neurons produces a higher impact. This is also observed for the temporal dispersion metric, although the particular results are not presented in this summary for conciseness. In the CNN, we opted to perform JAM cyberattacks over the whole simulation (27 positions), attacking from one to 20 nodes, since attacking more did not generate further impact (see Figure 3). Additionally, the comparison between biological and artificial scenarios offers a Pearson's correlation of 60% between the number of steps and the number of spikes.

Moving to FLO, delaying the instant of the attack to later positions reduced the impact from both biological metrics. From the CNN perspective, delaying the attack until position 21 increases the number of steps required to solve the maze. After that, the mouse can find the exit by probability. Comparing both scenarios, there is a Pearson's correlation of around 80% between the number of steps and both number of spikes and dispersion.

IV. Conclusion

The present summary highlights the main contributions of [1], first presenting JAM cyberattacks based on the inhibition of neuronal activity. For that, a CNN has been trained to solve the particular use case of a mouse trying to exit a maze, being the resulting topology translated and simulated in a neuronal simulator. This is justified by a lack of realistic neuronal representations and the similarities between CNNs and the visual cortex in terms of structure and behavior. Having both topologies, we analyze the impact of JAM on both biological and artificial scenarios, as well as the impact caused by JAM and FLO, comparing their impact over both scenarios. In summary, both cyberattacks can affect spontaneous neural activity and the ability of the mouse to exit the maze, although JAM presents a higher impact on neuronal spike rate.

V. Acknowledgment

This work has been partially supported by the Swiss Federal Office for Defense Procurement (armasuisse) with the RESERVE project (CYD-C-2020003).

References

[1] S. López Bernal, A. Huertas Celdrán, and G. Martínez Pérez, "Neuronal jamming cyberattack over invasive bcis affecting the resolution of tasks requiring visual capabilities," *Computers & Security*, vol. 112, p. 102534, 2022.

Sesión Poster 2: Investigación ya publicada II

Extended abstract: Hardware-Software Contracts for Secure Speculation

Marco Guarnieri*, Boris Köpf†, Jan Reineke‡, and Pepe Vila§

*IMDEA Software Institute †Microsoft Research ‡Saarland University §ARM

***Abstract*—Since the discovery of Spectre, a large number of hardware mechanisms for secure speculation has been proposed. Intuitively, more defensive mechanisms are less efficient but can securely execute a larger class of programs, while more permissive mechanisms may offer more performance but require more defensive programming. Unfortunately, there are no hardware-software contracts that would turn this intuition into a basis for principled co-design.**

In this paper, we put forward a framework for specifying such contracts, and we demonstrate its expressiveness and flexibility.

On the hardware side, we use the framework to provide the first formalization and comparison of the security guarantees provided by a representative class of mechanisms for secure speculation.

On the software side, we use the framework to characterize program properties that guarantee secure co-design in two scenarios traditionally investigated in isolation: (1) ensuring that a benign program does not leak information while computing on confidential data, and (2) ensuring that a potentially malicious program cannot read outside of its designated sandbox. Finally, we show how the properties corresponding to both scenarios can be checked based on existing tools for software verification, and we use them to validate our findings on executable code.

Tipo de contribución: *Investigación ya publicada*

Full paper: This is an extended abstract of "Hardware-software contracts for secure speculation" by M. Guarnieri, B. Köpf, J. Reineke and P. Vila, which appeared at the 42nd IEEE Symposium on Security and Privacy (S&P 2021). The paper is available at https://arxiv.org/abs/2006.03841.

I. Extended abstract

Speculative execution avoids expensive pipeline stalls by predicting the outcome of branching (and other) decisions, and by continuing the execution based on these predictions. When a prediction turns out to be incorrect, the processor rolls back the effects of speculatively executed instructions on the architectural state consisting of registers, flags, and main memory.

However, the microarchitectural state, which includes the content of various caches and buffers, is not (or only partially) rolled back. This side effect can leak information about the speculatively accessed data and thus violate confidentiality, see Figure 1a. Spectre attacks [1], [2] demonstrate that this vulnerability affects all modern general-purpose processors and poses a serious threat for platforms with multiple tenants.

A multitude of hardware mechanisms for secure speculation have been proposed. They are based on a number of basic ideas, such as delaying load operations until they cannot be squashed [3], delaying operations that depend on speculatively loaded data [4], [5], limiting the effect of speculatively executed instructions [6], [7], [8], [9], or rolling back the microarchitectural state when a misprediction is detected [10].

Intuitively, more defensive mechanisms are less efficient but can securely execute a larger class of programs, while more permissive mechanisms offer more performance but require more defensive programming.

```
1 if (y < size_A)
2   x = A[y];
3   temp &= B[x * 64];
```

(a) Program P_1

```
1 x = A[y];
2 if (y < size_A)
3   temp &= B[x * 64];
```

(b) Program P_2

Fig. 1: Program P_1 is the vanilla Spectre v1 example, where `A[y]` can be speculatively read and leaked into the data cache via an access to array `B`, for `y >= size_A`. Program P_2, is a variant where `A[y]` is accessed non-speculatively before the bounds check but the leak occurs during speculative execution.

For example, consider the variant of Spectre v1 shown in Figure 1b, where array `A` is accessed before the bounds check. Mechanisms delaying loads until they cannot be squashed [3] prevent speculatively leaking `A[y]`, for $y \geq$ `size_A`. In contrast, more permissive mechanisms that delay only loads depending on speculatively accessed data [4], [5] do *not* prevent the leak, because `A[y]` is accessed non-speculatively.

While the performance characteristics of secure speculation mechanisms are well-studied, there has been little work on (1) characterizing the security guarantees they provide, and in particular on (2) investigating how these guarantees can be effectively leveraged by software to achieve global security guarantees.[1] That is, we lack hardware-software contracts that support principled co-design for secure speculation, and that would formalize the intuition described above.

Contracts: In the full version of this paper [11], we propose a framework for specifying such contracts, based on three basic building blocks: an ISA language, a model of the microarchitecture, and an adversary model specifying which microarchitectural components (such as caches or branch predictor state) are observable via side-channels.

Contracts specify which program executions a side-channel adversary can distinguish. A contract in our framework is defined in terms of *executions* and *observations* made on these executions, and it is formalized in terms of a labelled ISA semantics. A CPU satisfies a contract if, whenever two program executions agree on all observations, they are guaranteed to be indistinguishable by the adversary at the microarchitectural level. The contract semantics can mandate exploration of mispredicted paths, effectively requiring agree-

[1] A notable exception to (1) is STT [5], which is backed by a security property that guarantees the confidentiality of speculatively loaded data. However, this property alone does not provide an actionable basis for (2), as preventing leakage of non-speculatively accessed data (as in Figure 1b) is declared out of scope [5, Section 4].

ment on observations corresponding to transient instructions.

Secrets at the program level must not affect contract observations, because then they can become visible to the adversary. Hence, contracts exposing more observations correspond to hardware with weaker security guarantees, whereas contracts exposing fewer observations correspond to hardware with stronger guarantees. The extreme case is a contract with no observations, which is satisfied by an ideal side-channel resilient platform that can securely execute every program.

Software Side: Our framework provides a basis for deriving requirements that *software* needs to satisfy to run securely on a specific platform. For deriving such requirements, we consider two scenarios typically considered in the literature:

- In the first scenario, called "constant-time programming", the goal is to ensure that a benign program, such as a cryptographic algorithm, does not leak information while computing on confidential data.
- In the second scenario, which we call "sandboxing", the goal is to restrict the memory region that a potentially malicious program, such as a Web application, can read from.[2]

For each scenario, we identify program-level properties that guarantee security on hardware that satisfies a given contract. We stress that secure speculation approaches usually *either* consider constant-time programming [12], [13], [14], [15] *or* sandboxing [16], [17]. In contrast, our framework supports *both* goals through program-level properties.

We provide tool support for automatically checking if programs are secure in both scenarios. For this, we extend a static analysis tool for detecting speculative leaks [12] to cater for different contracts, and we use it to validate all examples used in the paper on x86 executable code.

Hardware Side: We use our framework to define contracts for a comprehensive set of recent hardware mechanisms for secure speculation: disabling speculation, delaying speculative load operations [3], and speculative taint tracking [4], [5].

To this end, we formalize each mechanism in the context of a variant of the simple speculative out-of-order processor from [14] and we prove that it satisfies specific contracts against an adversary that observes caches, predictors, and (part of) the reorder buffer during execution. We show that the contracts we define form a lattice, and we use this to give, for the first time, a rigorous comparison of the security guarantees offered by different secure speculation mechanisms.

Our analysis highlights that the studied mechanisms [3], [4], [5] prevent leaks of speculatively accessed data, and confirms the results of [5]. For software, this means that "sandboxing" is supported out-of-the-box, in the sense that programs only need to place appropriate bounds checks, but no speculation barriers.

Our analysis also shows that the mechanisms offer no support for "constant-time programming". This means that programs that are constant-time in the traditional sense [18] still require additional checks [12], [14] or insertion of speculation barriers [19], even if hardware mechanisms for secure speculation are deployed.

Summary of contributions: Motivated by a lack of hardware-software contracts that support principled co-design for secure speculation, we propose a novel framework for expressing security contracts between hardware and software.

Our framework is expressive enough to (1) characterize the security guarantees provided by recent proposals for secure speculation, and (2) provide program-level properties formalizing how to leverage these hardware guarantees to achieve global, end-to-end security for different scenarios. From a theoretical perspective, we provide the first characterization of security for a comprehensive class of hardware mechanisms for secure speculation. From a practical perspective, we show how to automate checks for programs to run securely on top of these mechanisms.

Acknowledgments: This work was supported by a grant from Intel Corporation, Juan de la Cierva grant FJC2018-036513-I, Spanish project RTI2018-102043-B-I00 SCUM, and Madrid regional project S2018/TCS-4339 BLOQUES.

[2]In the terminology of [2], sandboxing aims to block disclosure gadgets.

REFERENCES

[1] P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, and Y. Yarom, "Spectre Attacks: Exploiting Speculative Execution," in *S&P 2019*.

[2] C. Canella, J. Van Bulck, M. Schwarz, M. Lipp, B. von Berg, P. Ortner, F. Piessens, D. Evtyushkin, and D. Gruss, "A Systematic Evaluation of Transient Execution Attacks and Defenses," in *USENIX Security 2019*.

[3] C. Sakalis, S. Kaxiras, A. Ros, A. Jimborean, and M. Själander, "Efficient invisible speculative execution through selective delay and value prediction," in *ISCA 2019*.

[4] O. Weisse, I. Neal, K. Loughlin, T. F. Wenisch, and B. Kasikci, "NDA: Preventing speculative execution attacks at their source," in *MICRO 2019*.

[5] J. Yu, M. Yan, A. Khyzha, A. Morrison, J. Torrellas, and C. W. Fletcher, "Speculative Taint Tracking (STT): A Comprehensive Protection for Speculatively Accessed Data," in *MICRO 2019*.

[6] M. Yan, J. Choi, D. Skarlatos, A. Morrison, C. Fletcher, and J. Torrellas, "InvisiSpec: Making speculative execution invisible in the cache hierarchy," in *MICRO 2018*.

[7] K. N. Khasawneh, E. M. Koruyeh, C. Song, D. Evtyushkin, D. Ponomarev, and N. Abu-Ghazaleh, "SafeSpec: Banishing the spectre of a meltdown with leakage-free speculation," in *DAC 2019*.

[8] V. Kiriansky, I. A. Lebedev, S. P. Amarasinghe, S. Devadas, and J. S. Emer, "DAWG: A defense against cache timing attacks in speculative execution processors," in *ISCA 2018*.

[9] S. Ainsworth and T. M. Jones, "Muontrap: Preventing cross-domain spectre-like attacks by capturing speculative state," in *ISCA 2020*.

[10] G. Saileshwar and M. K. Qureshi, "Cleanupspec: An "Undo" approach to safe speculation," in *MICRO 2019*.

[11] M. Guarnieri, B. Köpf, J. Reineke, and P. Vila, "Hardware-software Contracts for Secure Speculation," in *S&P 2021*.

[12] M. Guarnieri, B. Köpf, J. F. Morales, J. Reineke, and A. Sánchez, "SPECTECTOR: Principled detection of speculative information flows," in *S&P 2020*.

[13] G. Barthe, G. Betarte, J. Campo, C. Luna, and D. Pichardie, "System-level non-interference for constant-time cryptography," in *CCS 2014*.

[14] S. Cauligi, C. Disselkoen, K. v. Gleissenthall, D. Stefan, T. Rezk, and G. Barthe, "Towards constant-time foundations for the new spectre era," in *PLDI 2020*.

[15] M. Balliu, M. Dam, and R. Guanciale, "InSpectre: Breaking and fixing microarchitectural vulnerabilities by formal analysis," in *CCS 2020*.

[16] C. Carruth, "Speculative load hardening," http://releases.llvm.org/8.0.0/docs/SpeculativeLoadHardening.html, 2018.

[17] M. Miller, "Mitigating speculative execution side channel hardware vulnerabilities," https://blogs.technet.microsoft.com/srd/2018/03/15/mitigating-speculative-execution-side-channel-hardware-vulnerabilities/, 2018.

[18] J. B. Almeida, M. Barbosa, G. Barthe, F. Dupressoir, and M. Emmi, "Verifying constant-time implementations," in *USENIX Security 2016*.

[19] M. Vassena, C. Disselkoen, K. von Gleissenthall, S. Cauligi, R. G. Kici, R. Jhala, D. M. Tullsen, and D. Stefan, "Automatically eliminating speculative leaks from cryptographic code with blade," in *POPL 2021*.

A Review of Federated Learning for Malware Detection in IoT Devices

Valerian Rey [1], Pedro Miguel Sánchez[2,*], Alberto Huertas[3], Gérôme Bovet[4],
Gregorio Martínez Pérez[2], Burkhard Stiller[3]
[1]*École Polytechnique Fédérale de Lausanne (EPFL)*, 1015 Lausanne, Switzerland
[2]*Department of Information and Communications Engineering, University of Murcia*,
30100–Murcia, Spain. [pedromiguel.sanchez@um.es]
[3] *Communication Systems Group (CSG), Department of Informatics (IfI), University of Zurich UZH*,
CH–8050 Zürich, Switzerland
[4] *Cyber-Defence Campus, armasuisse Science & Technology*, CH–3602 Thun, Switzerland.

***Abstract*—This work investigates the possibilities enabled by federated learning concerning IoT malware detection and studies security issues inherent to this new learning paradigm. In this context, a framework that uses federated learning to detect malware affecting IoT devices is presented. N-BaIoT, a dataset modeling network traffic of several real IoT devices while affected by malware, has been used to evaluate the proposed framework. Both supervised and unsupervised federated models able to detect malware affecting seen and unseen IoT devices of N-BaIoT have been trained and evaluated. Besides, an adversarial setup with several malicious participants poisoning the federated model has been considered. Different model aggregation functions acting as countermeasures are thus evaluated, providing a significant improvement against malicious participants.**

***Index Terms*—IoT, Federated Learning, Adversarial Attack**

Tipo de contribución: *Investigación ya publicada*

I. Introduction

Behavioral fingerprint is a promising strategy to detect malware affecting Internet of Things (IoT) devices. Network communications, resource consumption, or system calls can be monitored to identify deviations on normal behaviors. Once the behavior sources are monitored, Machine Learning (ML) and Deep Learning (DL), have gained enormous relevance to achieve successful malware detection. Nowadays, most solutions using ML/DL rely on a central entity in charge of collecting data from different devices and training models. However, this approach is not suitable for scenarios where behaviors contain sensitive or confidential data. Federated Learning (FL) is gaining huge relevance in the last years as a collaborative and privacy-preserving ML paradigm.

However, despite the novelty and benefits of FL approaches, their application in real-world scenarios still presents several open questions that must be analyzed and solved (or at least improved). In this sense, some of the most relevant open challenges can be summarized as: 1) how can FL be used in the IoT context to build joint models without sharing sensitive data?; 2) how do FL approaches affect the performance of traditional anomaly detectors and classifiers in IoT scenarios?; 3) what is the impact of different adversarial attacks affecting federated models designed to detect cyberattacks on IoT scenarios?; and 4) are existing countermeasure mechanisms able to mitigate the effects of adversarial attacks?; and if so, 5) what are the most suitable countermeasures for IoT scenarios?.

This paper presents a summary of [1], whose main contributions are:

- A security framework that uses FL to detect, in a privacy preserving fashion, cyberattacks affecting IoT devices. The proposed framework covers both anomaly detection and classification approaches.
- A pool of experiments measuring the framework performance when detecting malware in IoT devices. The next scenarios have been compared: i) a centralized approach where all the data is shared, ii) a distributed approach where each entity trains an independent local model, and iii) a federated approach where a joint model is generated sharing the local model updates.
- The evaluation of the impact of several adversarial attacks. Besides, different aggregation functions are applied as countermeasure mechanisms to improve the framework resilience.

II. Proposed Federated Learning System

This work is a summary of the paper presented in [1]. It details the design of a FL-based framework, describing its components and how they interact with each other during the model training and evaluation. Besides, it also depicts how the framework is deployed for our validation use case, which leverages the N-BaIoT dataset. This dataset includes traffic traces generated by different botnets and divided in 9 different clients (8 for training and 1 for validation in unseen devices).

Figure 1 details the architecture of a client after data acquisition, as well as its interactions with the server. For model update, two approaches are implemented: MINI-BATCH AVG, where aggregation is made in each training batch; and MULTI-EPOCH AVG, where aggregation is made per epoch, reducing the communication costs.

Both supervised and unsupervised approaches are tested using a Multi-Layer Perceptron and an Autoencoder, respectively. Besides, in the supervised setting, three different configurations are tested: 7.87% benign traffic and 92.13% attack traffic, 50% benign and 50% attack, and 95% benign and 5% attack.

Table I shows the results of both approaches when detecting botnet activities. It can be appreciated how no performance loss is present in the two federated configurations (MINI-BATCH AVG and MULTI-EPOCH AVG) compared to the centralized one,

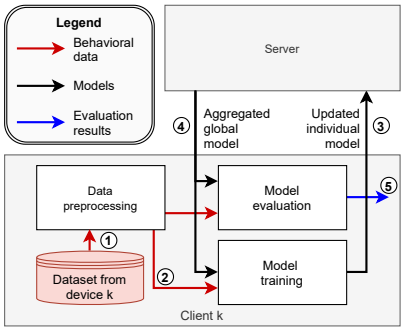


Figure 1: Federated client architecture during training and evaluation.

with +99% in all metrics. Besides, the naive (local) model results are improved in all cases.

		Naive	MULTI-EPOCH AVG	MINI-BATCH AVG	Central.
Known devices (7.87%)	Acc.	99.78	99.92	99.96	99.96
	TPR	99.98	99.98	99.98	99.98
	TNR	97.49	99.26	99.69	99.69
New device (7.87%)	Acc.	98.94	99.89	99.89	99.88
	TPR	99.58	99.97	99.98	99.97
	TNR	91.44	99.00	98.93	98.83
Known devices (50%)	Acc.	99.92	99.82	99.93	99.91
	TPR	99.97	99.97	99.97	99.97
	TNR	99.88	99.67	99.88	99.85
New device (50%)	Acc.	98.36	99.63	99.58	99.52
	TPR	98.79	99.90	99.95	99.93
	TNR	97.93	99.35	99.21	99.10
Known devices (95%)	Acc.	99.92	99.79	99.93	99.93
	TPR	99.89	99.93	99.98	99.98
	TNR	99.92	99.78	99.92	99.93
New device (95%)	Acc.	98.59	99.43	99.38	99.42
	TPR	97.79	99.55	99.82	99.81
	TNR	98.63	99.42	99.36	99.40

Known devices	TPR	88.00	99.98	99.98	99.98
	TNR	97.38	94.84	95.12	95.56
New device	TPR	87.77	99.98	99.98	99.98
	TNR	59.66	92.61	91.78	92.76

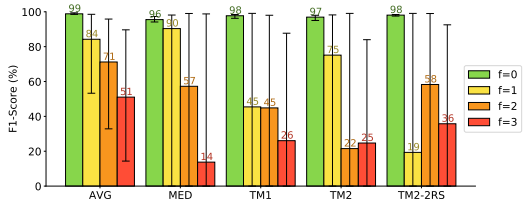
Table I: Supervised (up) and Unsupervised (down) results.

III. ADVERSARIAL ATTACKS

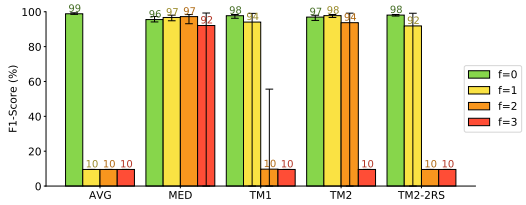
Once the performance of the federated approach is verified, it is evaluated how different adversarial attacks affect the federated approach. Three attacks are applied for the supervised approach: *i) All labels flipping*, where all labels are flipped; *ii) Gradient factor attack*, where the malicious clients multiply their gradients by a negative factor before updating their local model; and *iii) Model cancelling attack*, where malicious clients try to bring all global model parameters to 0.

To secure the federation against malicious clients, model aggregation and update processing solutions are applied. Additionally to averaging (AVG), the selected robust aggregation methods are Coordinate-wise median, Coordinate-wise trimmed mean and s-Resampling.

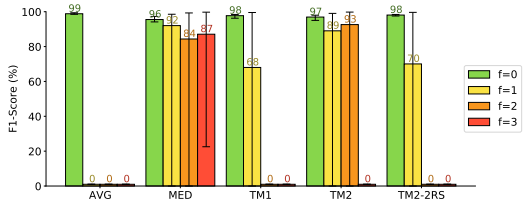
Figure 2 shows how the F1-Score of the model tested on the devices owned by the clients (the known devices) varies in the different implemented attacks according to the aggregation function. *Averaging* (AVG) is the best aggregation function when all clients are honest. *Coordinate-wise median* aggregation (MED) presents more resilience against most attack scenarios considered. Overall it has the best results among the tested aggregation functions in the adversarial setup. Unsurprisingly, *Coordinate-wise trimmed mean* (TM(c)) fails when used against more than c malicious clients, as clearly demonstrated in the *model cancelling* attack results.



(a) All labels flipping attack.



(b) Gradient factor attack.



(c) Model cancelling attack.

Figure 2: F1-Scores under the different tested attacks for each aggregation function, with $f = 0$, 1, 2 or 3 malicious clients.

IV. CONCLUSIONS

This work proposes a privacy-preserving framework for IoT malware detection that leverages FL to train and evaluate both supervised and unsupervised models. It has been demonstrated that the privacy of the data can be preserved without losing model performance by following the federated approach, using the N-BaIoT dataset has been used due to its realistic setup. The resilience of the federated models against malicious clients using robust aggregation has been tested through the following adversarial attacks: i) all labels flipping attack, ii) gradient factor attack, and iii) model cancelling attack.

ACKNOWLEDGEMENTS

This work has been supported by *(a)* the Swiss Federal Office for Defense Procurement (armasuisse) with the RESERVE and CyberTracer (CYD-C-2020003) projects and *(b)* the University of Zürich UZH.

REFERENCES

[1] V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, "Federated learning for malware detection in iot devices," *Computer Networks*, p. 108693, 2021.

An Optimizing Protocol Transformation for Constructor Finite Variant Theories in Maude-NPA

Damián Aparicio-Sánchez, Santiago Escobar, Julia Sapiña
VRAIN, Universitat Politècnica de València, Valencia, Spain
{daapsnc,sescobar,jsapina}@upv.es
Raúl Gutiérrez
Universidad Politécnica de Madrid, Madrid, Spain
r.gutierrez@upm.es

***Resumen*—Maude-NPA is an analysis tool for cryptographic security protocols that takes into account the algebraic properties of the cryptosystem. Maude-NPA can reason about a wide range of cryptographic properties. However, some algebraic properties, and protocols using them, have been beyond Maude-NPA capabilities, either because the cryptographic properties cannot be expressed using its equational unification features or because the state space is unmanageable. In this paper, we provide a protocol transformation that can *safely* get rid of cryptographic properties under some conditions. The time and space difference between verifying the protocol with all the crypto properties and verifying the protocol with a minimal set of the crypto properties is remarkable. We also provide, for the first time, an encoding of the theory of bilinear pairing into Maude-NPA.**

***Index Terms*—crypto protocol analysis, Diffie-Hellman, exponentiation, bilinear pairing, protocol transformation**

Tipo de contribución: *Investigación ya publicada en Computer Security – ESORICS 2020, GGS Rating A+, CORE:A*
D.O.I: https://doi.org/10.1007/978-3-030-59013-0_12

I. Introduction

Maude-NPA [9] is an analysis tool for cryptographic security protocols that takes into account the algebraic properties of the cryptosystem. Sometimes algebraic properties can uncover weaknesses of cryptosystems and, in other cases, they are part of the protocol security assumptions. Maude-NPA is based on unification and performs backwards search from an attack state pattern to determine whether or not it is reachable. Maude-NPA can be used to reason about a wide range of cryptographic properties [1], [10], including cancellation of encryption and decryption, Diffie-Hellman exponentiation [8], exclusive-or [17], and some approximations of homomorphic encryption [11], [19].

However, some algebraic properties and protocols using them have been beyond Maude-NPA capabilities, either because the cryptographic properties cannot be expressed using its equational unification features or because the state space is unmanageable. We provide a protocol transformation that can substantially reduce the search space, i.e., given some cryptographic properties, expressed using the equational unification features of Maude-NPA, and a protocol, we are able to transform the protocol in such a way that some cryptographic properties are no longer necessary, and thus can be safely removed. The time and space difference between verifying the protocol with all the crypto properties and verifying the protocol with a minimal set of the crypto properties is remarkable. We also provide, for the first time, an encoding of the theory of bilinear pairing into Maude-NPA that goes beyond the encoding of bilinear pairing available in Tamarin [2], the only crypto tool with such an equational theory.

Our protocol transformation relies on a program transformation from [15] for rewrite theories in Maude. This program transformation relies on *constructor term variants* [14], which is an extension of *term variants* [5], [12]. Nowadays, several crypto analysis tools rely on the variant-based equational unification capabilities of Maude, such as Maude-NPA, Tamarin [7] and AKISS [3]. These tools may be benefited from our protocol transformation and, furthermore, from our encoding of the theory of bilinear pairing. We present several increasingly complex case studies: Diffie-Hellman protocol, STR (Skinny TRee) protocol, Joux protocol and suit of TAKs (Tripartite Authenticated Key group) protocols. We have demonstrated in Table I that the time and space difference between verifying the protocol with all the crypto properties and verifying the protocol with a minimal set of the crypto properties is remarkable (an average speedup of $2{,}54$).

II. Conclusions

The main contributions of this work are: (i) we provide a non-trivial protocol transformation based on [15]; (ii) since the protocols of our experiments do not satisfy the conditions of [15], we provide a more powerful protocol transformation that we implemented[1] and pays off in practice; (iii) we provide an encoding of bilinear pairing that can handle all the protocols of our experiments and goes beyond the encoding of bilinear pairing available in Tamarin, the only crypto tool with such an equational theory; (iv) we implemented the algorithm of [18] for the computation of constructor variants [14] from scratch; and (v) there was no implementation of the program transformation of [15] and we implemented it.

Since Tamarin [7] and AKISS [3] use term variants, they could be adapted to use both our protocol transformation and our encoding of the theory of bilinear pairing. They may even be useful for other crypto tools with more limited crypto properties such as ProVerif [4], OFMC [16], Scyther [6] or Scyther-proof [13]. Specially, since Proverif [4] already incorporated the notion of destructors and constructors time ago. As future work, we plan to study how the protocol transformation applies to other families of protocols and crypto properties such as homomorphisms [19].

[1]The tool is available online at http://safe-tools.dsic.upv.es/cvtool.

		Before Transformation		*After Transformation*			
Protocol	*Property*	*States*	*Time (ms)*	*States*	*Time (ms)*	*States (%)*	*Speedup*
DH	auth	137	308,066	111	132,756	81.02	2.32
	secrecy	138	322,731	104	142,015	75.36	2.27
STR	auth	34	43,144	31	16,010	91.18	2.69
	secrecy	250	1,016,469	117	408,960	46.80	2.49
Joux	auth	38	85,579	37	30,012	97.37	2.85
	secrecy	55	247,712	58	78,384	105.45	3.16
TAK1	secrecy	25	259,619	20	126,998	80.00	2.04
TAK2	secrecy	67	365,797	46	152,842	68.66	2.39
TAK3	secrecy	117	670,775	67	216,350	57.26	3.10
TAK4	secrecy	57	371,770	48	181,850	84.21	2.04

Tabla I
EXPERIMENTAL RESULTS FOR THE TRANSFORMED PROTOCOLS.

ACKNOWLEDGEMENT

Partially supported by the EU (FEDER) and the Spanish MCIU under grant RTI2018-094403-B-C32, by the Spanish Generalitat Valenciana under grant PROMETEO/2019/098, and by the US Air Force Office of Scientific Research under award number FA9550-17-1-0286. Julia Sapiña has been supported by the Generalitat Valenciana APOSTD/2019/127 grant

REFERENCIAS

[1] Maude-NPA manual v3.1, http://maude.cs.illinois.edu/w/index.php/Maude_Tools:_Maude-NPA
[2] The Tamarin-Prover Manual June 4, 2019). Available on: https://tamarin-prover.github.io/manual/tex/tamarin-manual.pdf
[3] Baelde, D., Delaune, S., Gazeau, I., Kremer, S.: Symbolic verification of privacy-type properties for security protocols with XOR. In: 30th IEEE Computer Security Foundations Symposium, CSF 2017, pp. 234–248. IEEE Computer Society (2017).
[4] Blanchet, B.: Modeling and verifying security protocols with the applied pi calculus and ProVerif. Found. and Trends in Privacy and Security **1**(1–2), 1–135 (2016)
[5] Comon-Lundh, H., Delaune, S.: The Finite Variant Property: How to Get Rid of Some Algebraic Properties. In: Proc. of the 16th Int. Conference on Rewriting Techniques and Applications (RTA 2005). LNCS, vol. 3467, pp. 294–307. Springer.
[6] Cremers, C.J.F.: The scyther tool: Verification, falsification, and analysis of security protocols. In Computer Aided Verification, 20th Int. Conference, CAV 2008. LNCS, vol. 5123, pp. 414–418. Springer (2008).
[7] Dreier, J., Duménil, C., Kremer, S., Sasse, R.: Beyond subterm-convergent equational theories in automated verification of stateful protocols. In Principles of Security and Trust, 2017. LNCS, vol. 10204, pp. 117–140. Springer (2017).
[8] Escobar, S., Hendrix, J., Meadows, C., Meseguer, J.: Diffie-Hellman Cryptographic reasoning in the Maude-NRL protocol analyzer. In: Proc. 2nd Int. Workshop on Security and Rewriting Techniques (SecReT 2007)
[9] Escobar, S., Meadows, C., Meseguer, J.: Maude-NPA: Cryptographic Protocol Analysis Modulo Equational Properties. In: FOSAD 2007/2008/2009 Tutorial Lectures. LNCS, vol. 5705, pp. 1–50. Springer (2009).
[10] Escobar, S., Meadows, C., Meseguer, J.: Maude-NPA: Cryptographic protocol analysis modulo equational properties. In: Aldini, A., Barthe, G., Gorrieri, R. (eds.) FOSAD 2008/2009 Tutorial Lectures. LNCS, vol. 5705, pp. 1–50. Springer (2009)
[11] Escobar, S., Kapur, D., Lynch, C., Meadows, C.A., Meseguer, J., Narendran, P., Sasse, R.: Protocol analysis in Maude-NPA using unification modulo homomorphic encryption. In Proc. of PPDP 2011, pp. 65–76. ACM (2011).
[12] Escobar, S., Sasse, R., Meseguer, J.: Folding variant narrowing and optimal variant termination. J. Log. Algebr. Program. **81**(7-8), 898–928 (2012).
[13] Meier, S., Cremers, C., Basin, D.: Strong invariants for the efficient construction of machine-checked protocol security proofs. In: 2010 23rd IEEE Computer Security Foundations Symposium. pp. 231–245 (2010)
[14] Meseguer, J.: Variant-based satisfiability in initial algebras. Sci. Comput. Program. **154**, 3–41 (2018).
[15] Meseguer, J.: Generalized rewrite theories, coherence completion, and symbolic methods. J. Log. Algebr. Meth. Program. **110** (2020).
[16] Mödersheim, S., Viganò, L.: The open-source fixed-point model checker for symbolic analysis of security protocols. In: Foundations of Security Analysis and Design V, FOSAD 2007/2008/2009. LNCS, vol. 5705, pp. 166–194. Springer (2009).
[17] Sasse, R., Escobar, S., Meadows, C., Meseguer, J.: Protocol analysis modulo combination of theories: A case study in maude-npa. In: Security and Trust Management. pp. 163–178. Springer (2011)
[18] Skeirik, S., Meseguer, J.: Metalevel algorithms for variant satisfiability. J. Log. Algebraic Methods Program. 96, 81-110 (2018)
[19] Yang, F., Escobar, S., Meadows, C.A., Meseguer, J., Narendran, P.: Theories of homomorphic encryption, unification, and the finite variant property. In: Proc. of PPDP 2014. pp. 123–133. ACM (2014).

Review of Gate-Level Hardware Countermeasure Comparison Against Power Analysis Attacks

E. Tena-Sánchez
IMSE-CSIC/U. Sevilla
erica@imse-cnm.csic.es

F. E. Potestad-Ordóñez
IMSE-CSIC/U. Sevilla
potestad@imse-cnm.csic.es

V. Zuñiga-González
IMSE-CNM-CSIC
virginia@imse-cnm.csic.es

C. Fernández-García
IMSE-CNM-CSIC
carlos@imse-cnm.csic.es

J. M. Mora-Gutiérrez
IMSE-CNM-CSIC
jmiguel@imse-cnm.csic.es

C. J. Jiménez-Fernández
IMSE-CSIC/U. Sevilla
cjesus@imse-cnm.csic.es

A. J. Acosta-Jiménez
IMSE-CSIC/U. Sevilla
acojim@imse-cnm.csic.es

***Abstract*—In this paper, we present a review of the work [1]. The fast settlement of Privacy and Secure operations in the Internet of Things (IoT) is appealing the selection of mechanisms to achieve a higher level of security at the minimum cost and with reasonable performances. In recent years, dozens of proposals have been presented to design circuits resistant to Power Analysis attacks. In this paper a deep review of the state of the art of gate-level countermeasures against Power Analysis attacks has been done, performing a comparison between hiding approaches (the power consumption is intended to be the same for all the data processed) and the ones considering a masking procedure (the data are masked and behave as random). The most relevant proposals in the literature, 35 for hiding and 6 for masking, have been analyzed, not only by using data provided by proposers, but also those included in other references for comparison.**

***Index Terms*—hardware countermeasures; gate level; VLSI design of cryptographic circuits; side-channel attacks (SCAs); information security; logic design; Internet of things (IoT).**

Tipo de contribución: Investigación ya publicada.

I. Introduction

The high growth that the Internet of Things (IoT) is experiencing has brought with it an increase in the exchange of sensitive information from interconnected users. Traditionally, the mathematical algorithm and the length of the key defined the security of crypto-systems. However, the physical implementation of cryptographic algorithms leads to information leakages that can be exploited by third parties to reveal critical data [2], [3]. Among the different types of attacks, the so-called Side-Channel Attacks (SCAs) belong to the group of passive noninvasive attacks and are those where the cryptographic device is not manipulated, e.g. there is no trace that a malicious agent has had access to the device and there is no damage to the circuit [2], [3]. Among SCAs, those based on analysis of the power consumption (Power Analysis, PA) produced by the circuit have attracted significant attention from the research community [3].

Since the emergence of power analysis attacks in the late 1990s, numerous countermeasures have been proposed by the scientific community to search for alternatives to minimize the weak points of crypto-circuits [4], [5], [6]. There are several countermeasure strategies at the hardware level. These countermeasures range from the layout up to algorithm level and go from attack detection to adding redundant blocks to obfuscate possible information leakage. They can be classifyed depending on the technique used to break the data correlation with the power consumption: hiding (the power consumption is intended to be the same for all the data processed) or masking (the data are masked and behave as random). This review focus on gate-level hiding and masking countermeasures.

II. State of the art: Gate-level countermeasures against power analysis attacks

PA attacks exploit the correlation between power consumption and the data that are processed by the cryptographic device during encryption, following several strategies, to reveal the critical data. Hardware countermeasures are oriented towards breaking the relationship between data being processed and consumed power. To break this rate at the gate level, two different mechanisms are widely used: hiding and masking techniques. The hiding attempts to have the same power consumption at the gate, circuit, or algorithm level, independently of the data being processed. In masking, the critical data are masked with a random data sequence during encryption such that operations on the masked data are indistinguishable from random data.

A. Gate-level masking

Gate-level masking consists of computing both the inputs and the mask inside the gate itself. In these implementations, each masked signal a_m is propagated along with its mask m_a, being the unmasked signal $a = a_m \oplus m_a$. The simplest way to perform masking is through boolean masking, where an input word gets masked by being XOR-ed by a random value. Arithmetic masking involves more complex arithmetic operations within specific algorithms. Boolean masking is preferably used at the gate level, while at the algorithm or circuit level, the use of dedicated arithmetic masking techniques that best suit the algorithm are recommended.

B. Gate-level hiding

Hiding tries to achieve exactly the same power consumption in operations, regardless of the data being processed. Since the first PA attacks were presented, there have been numerous logic style proposals that seek to be resistant to these attacks by having data-independent power consumption. In a first approach, this identical consumption can be achieved using dual-rail signals and differential gates, where the true and

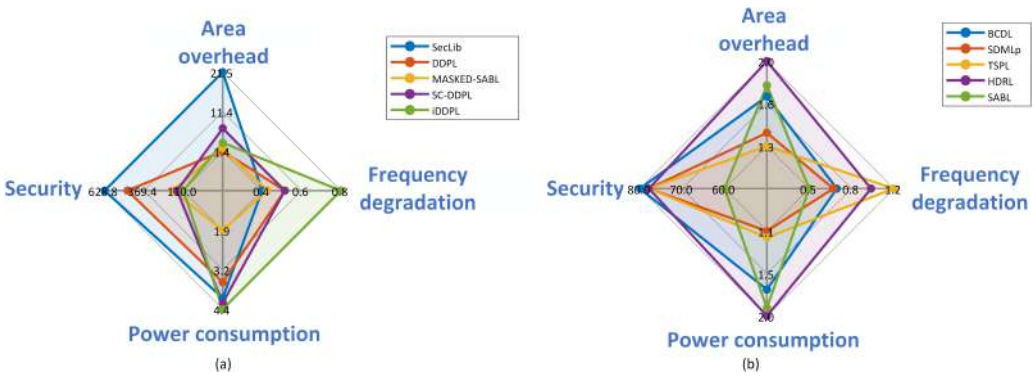


Fig. 1. Top 5 countermeasures in security levels (a) and top 5 countermeasures with best trade-off between performance and security levels (b).

complemented outputs are simultaneously generated: in every clock cycle, one of the differential branches performs the gate function and the other one its complement at the same time.

Since hiding means exact power consumption independently of the data processed, it implies full symmetry. However, most of these techniques suffer from the difficulty of tailoring the place and route operation so that the capacitive load of two wires is equal. This is particularly difficult in nanometric technologies, where the transistor sizes and wiring widths continuously shrink. Placing and routing a circuit manually, i.e. doing a full-custom (FC) design, significantly increases the design costs. An additional drawback is the so-called early evaluation, also called data-dependent time-of-evaluation, referring to the cases where a gate evaluates its output at different time instances depending on the value of its input. It becomes more problematic when several of such gates are cascaded to realize a combinational circuit, causing the power consumption pattern of the circuit to have a clear dependency on its input value.

III. Comparative analysis of gate-level countermeasures

The presented analysis considers the most relevant solutions in the literature, 35 hiding proposals, and 6 based on masking, not only by using the data provided by proposing authors, but also those included in the other references for comparison. For a complete analysis, please refer to [1]. Advantages and drawbacks of the proposals are analysed, showing quantified data for cost, performance (delay and power), and estimated security level, when available. The comparison between performances, features, and security levels of these proposals is not easy to carry out, given the variety of approaches and considered technologies. However, a summary of the comparative analysis is presented using the normalized values presented by the reference authors of each countermeasure.

Fig. 1-a provides a visual comparison of the top 5 countermeasures with the best security levels. Fig. 1-b depicts the top 5 countermeasures with the best trade-off between security values and area-delay-product performance and area overhead. From these figures, it can be seen that, typically and as expected, the higher the security the higher the cost. However, this is not always the case. For example, in Fig. 1-b it can be seen that the SABL approach has approximately the same power and area costs as BCDL but provides significantly less protection against PA. Nevertheless, in addition to performance degradation and security levels, it is also important to consider the inherent design difficulties of each proposal, as well as the feasibility of including the countermeasure in the design.

IV. Conclusions

In this paper a deep review of the state-of-the-art of gate-level countermeasures against power analysis attacks has been done. This work also visually depicts the performance, cost, and security level relation of the several solutions to better assist cryptodesigners in the selection of the best solution, style according to their constraints. Overall, these results suggest that RSL and DRSL solutions are the best approaches when considering masking, while BCDL, SDMLp, TSPL, HDRL and SABL are those with best security-performance figures. It can also be concluded that hiding proposals reach higher security levels, but with more difficult design constraints, which, if not met, can result in security weaknesses. Finally, this review also suggests that the combination of masking and hiding, as in Masked_SABL, can provide the most secure solution, but at the cost of more complexity.

Acknowledgments

This work has been funded by project SCAROT 1380823-US/JUNTA/FEDER, UE. Thanks to SPIRS Project with Grant Agreement No. 952622 under the European Union's Horizon 2020 programme and Grant PID2020-116664RB-I00 funded by MCIN/AEI/10.13039/501100011033.

References

[1] E. Tena-Sánchez, F. E. Potestad-Ordóñez, C. J. Jiménez-Fernández, A. J. Acosta and R. Chaves, "Gate-Level Hardware Countermeasure Comparison against Power Analysis Attacks," *Applied Sciences*, 12(5), 2390, 2022.

[2] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Springer, 2007.

[3] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis", *in Proc. of International Cryptology Conference (CRYPTO'99)*, pp. 388-397, 1999.

[4] K. Tiri, M. Akmal, and I. Verbauwhede, "A Dynamic and Differential CMOS Logic With Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards," *in Proc. of the 28th European Solid-State Circuits Conference (ESSCIRC'02)*, pp. 403-406, 2002.

[5] M. Nassar, S. Bhasin, J. Danger, G. Duc, S. Guilley, and A. E. P. Effect, "BCDL : A High Speed Balanced DPL for FPGA with Global Precharge and no Early Evaluation," *in Proc, of the Conference on Design, Automation and Test in Europe (DATE'10)*, pp. 849–854, 2010.

[6] B. Fadaeinia, M. T. Hasan Anik, N. Karimi and A. Moradi, "Masked SABL: A Long Lasting Side-Channel Protection Design Methodology," *in IEEE Access*, vol. 9, pp. 90455-90464, 2021.

Semantic Attention Keypoint Filtering for Darknet Content Classification

Aitor Del Río*†, Eduardo Fidalgo*†, Pablo Blanco-Medina*†, Deisy Chaves*
Alejandro Prieto Castro†, Enrique Alegre Gutierrez*†

*Department of Electrical, Systems and Automation, Universidad de León, León, ES
†Researcher at INCIBE (Spanish National Cybersecurity Institute), León, ES
Email:{aitor.rio, eduardo.fidalgo, pablo.blanco, deisy.chaves, enrique.alegre}@unileon.es,
alejandro.prieto@incibe.es

***Abstract*—The Tor Darknet hosts several domains containing images with potentially illicit content. Automatic image classification can help Law Enforcement Agencies (LEAs) in finding evidence of illegal activity. The background of the image usually presents an additional difficulty for this task. In our work, we introduce Semantic Attention Keypoint Filtering (SAKF), a pixel-level filtering strategy to remove the features that do not belong to the object foreground. Our proposal obtains higher precision than the baseline method, Bag of Visual Words (BoVW) using dense Scale-Invariant Feature Transform (SIFT). We compared the performance of SAKF against the extracted features from MobileNet and ResNet50 convolutional neural networks. Our proposal obtained an accuracy of 87.98% in a custom-based Tor image dataset, highlighting its potential to support LEAs investigations on Tor darknet.**

***Index Terms*—Tor Darknet, Image Classification, Bag of Visual Words.**

Type of contribution: *Research already published [1]*

I. INTRODUCTION

The main characteristic of Tor [2] is its layer of anonymity, which is used frequently to hide different types of suspicious activities. As Al Nabki et al. studied in [3], 20% of the crawled domains contained criminal activities such as illegal drugs selling [4], documentation counterfeit [5] or child sexual abuse images and videos [6].

To classify the categories found within the Darknet based on their images, we improved our previous work [7] [5] [8] by proposing a new method called Semantic Attention Keypoint Filtering (SAKF), which removes the background information of an image by excluding keypoints based on the distance to foreground and background dictionaries. Hence, SAKF enhances the foreground information of the images to improve the classification of illicit activities. Our approach can help Law Enforcement Agencies (LEAs) on finding evidence of illicit activities inside the images in an automated way.

II. SEMANTIC ATTENTION KEYPOINT FILTERING

The pipeline of our contribution, as seen in Fig. 1, is threefold: (i) extraction of signatures for the image foreground and background based on saliency maps; (ii) computation of dictionaries using the extracted descriptors; and (iii) generation of Bag of Visual Words (BoVW) [9] for classifying the image into illicit activities.

First, we use the image signature proposed by [10], a binary saliency map algorithm, which considers the image as a sum of foreground and background parts.

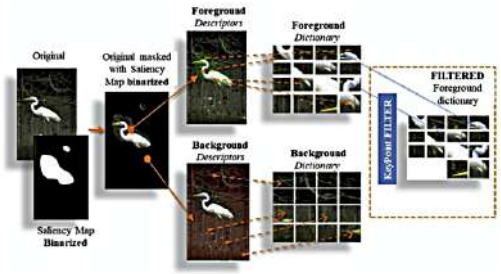


Figure 1. Proposed SAKF method

Let SM_σ be the image signature computed using the [10] algorithm. The image signature is binarised using the Otsu threshold, obtaining SM_{bin} where the attention zone corresponds to white pixels and the foreground to black pixels.

Dense Scale-Invariant Feature Transform (SIFT) descriptors are obtained from the original image in two groups, foreground, d_F, and background, d_B. If a descriptor is computed from a keypoint in the SM_{bin} zone, it is labelled as a foreground descriptor. Otherwise, it is labelled as a background descriptor.

Afterwards, using d_F and d_B, we compute two dictionaries called VD_F and VD_B. We use SAKF to select the foreground descriptors whose semantic meaning is closer to the white area of SM_{bin}. We perform a semantic attention selection for the foreground descriptors and filter them using their Euclidean distance to the VD_F and VD_B. The result can be seen in the Fig. 2, where the black points symbolize the foreground and the red dots corresponds to the SAKF descriptors.

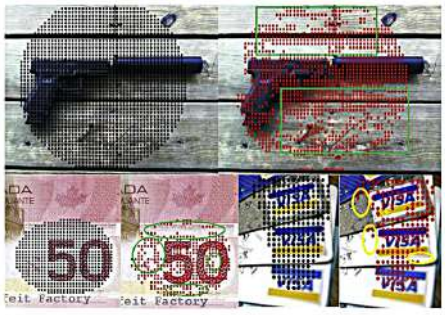


Figure 2. Samples of TOIC where the foreground descriptors are represented as black dots and SAKF descriptors as red dots

Lastly, we generate a BoVW descriptors matrix only with the foreground descriptors whose measured distance is lower to VD_F than VD_B. BoVW counts the number of appearances of a specific element in an image. Each image is represented as the frequency of the elements appearing in the image. When the foreground and background dictionaries are constructed, the BoVW descriptor matrix contains the keypoints of the interest object.

III. EXPERIMENTAL RESULTS

Our proposal uses the same configuration as [11], [8], [7], [12]; dense SIFT descriptors with step and size of seven, K-means to obtain 2048 visual words dictionary, BoVW features vectors built through a hard assignment approach and a Support Vector Machine (SVM) with a linear kernel for classification.

We evaluate the proposed approach on Tor images by using our new dataset, TOr Image Categories (TOIC) [5], which is composed by 698 real images crawled from Tor Hidden Services and DUTA (Darknet Usage Text Addresses) [3] domains. Images are classified into five categories of illegal content: Drugs, Violence Weapons, Counterfeit Money, Counterfeit Personal Identification and Counterfeit Credit Cards. We split TOIC in 75% for training and 25% for testing.

In our study [1], we include a comparison with two Convolutional Neural Networks (CNN), MobileNet [13], and ResNet50 [14], both pre-trained with the ImageNet dataset. We extracted features from the last layer of each network and feed a SVM with a linear kernel, as we did with dense SIFT together with BoVW encoding.

The experimental results can be seen in the Table I. In the baseline approach [5], we obtained an accuracy of 85.78% on TOIC. Using SAKF, we obtain an accuracy of 87.98%, corresponding to an increase of 2.20% over the baseline. Our proposal also outperforms both of the selected CNN architectures.

We conclude that a proper selection of the features that are used to create the visual dictionaries for the foreground and background descriptors, where only salient features of the interest object are considered, can increase the results of the classification task.

Table I
IMAGE CLASSIFICATION ACCURACY ON TOIC DATASET

Methodology	Accuracy(%)
Hand dSIFT + BoVW (Baseline) [5]	85.78
MobileNet	74.06
ResNet50	81.37
Hand dSIFT + BoVW (SAKF)	87.98

IV. CONCLUSIONS

In this paper, we proposed SAKF, a new method which improves the image classification on Darknet images, based on a BoVW model that filters background information from the image and extracts descriptors only from the foreground. This approach can be used by LEAs to automatically classify illicit content within Tor domains.

We compared our results with previous work [5] and two CNNs (MobileNet and ResNet50). Our experiments highlight that SAKF achieves better results with an accuracy of 87.98% in our TOIC dataset

ACKNOWLEDGEMENTS

This research was supported by the framework agreement between the University of León and INCIBE by the Addendum 22 and 01, as well as the grant 'Ayudas para la realización de estudios de doctorado en el marco del programa propio de investigación de la Universidad de León Convocatoria 2018'. Also, this research was supported by the INCIBE grant 'INCIBEI-2015-27359' which belongs to 'Ayudas para la Excelencia de los Equipos de Investigación avanzada en ciberseguridad'.

REFERENCES

[1] E. Fidalgo, E. Alegre, L. Fernández-Robles, and V. González-Castro, "Classifying suspicious content in tor darknet through semantic attention keypoint filtering," *Digit. Investig.*, vol. 30, pp. 12–22, 2019.

[2] M. W. Al Nabki, E. Fidalgo, E. Alegre, and V. González-Castro, "Detecting emerging products in tor network based on k-shell graph decomposition," *III Jornadas Nacionales de Investigación en Ciberseguridad (JNIC)*, 2017.

[3] M. W. A. Nabki, E. Fidalgo, E. Alegre, and L. Fernández-Robles, "Torank: Identifying the most influential suspicious domains in the tor network," *Expert Syst. Appl.*, vol. 123, pp. 212–226, 2019.

[4] A. Joshi, E. Fidalgo, E. Alegre, and L. Fernández-Robles, "Summcoder: An unsupervised framework for extractive text summarization based on deep auto-encoders," *Expert Syst. Appl.*, vol. 129, pp. 200–215, 2019.

[5] E. Fidalgo, E. Alegre, V. González-Castro, and L. Fernández-Robles, "Illegal activity categorisation in darknet based on image classification using CREIC method," in *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17, León, Spain, September 6-8, 2017, Proceedings*, ser. Advances in Intelligent Systems and Computing, H. P. García, J. Alfonso-Cendón, L. Sánchez-González, H. Quintián, and E. Corchado, Eds., vol. 649. Springer, 2017, pp. 600–609.

[6] A. Gangwar, E. Fidalgo, E. Alegre, and V. González-Castro, "Pornography and child sexual abuse detection in image and video: A comparative evaluation," in *8th International Conference on Imaging for Crime Detection and Prevention, ICDP 2017, Madrid, Spain, December 13-15, 2017*. IET / IEEE, 2017, pp. 37–42.

[7] E. Fidalgo, E. Alegre, V. González-Castro, and L. Fernández-Robles, "Compass radius estimation for improved image classification using edge-sift," *Neurocomputing*, vol. 197, pp. 119–135, 2016.

[8] E. Fidalgo, E. Alegre, V. González-Castro, and L. Fernández-Robles, "Boosting image classification through semantic attention filtering strategies," *Pattern Recognit. Lett.*, vol. 112, pp. 176–183, 2018.

[9] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," *European Conference on Computer Vision (ECCV) International Workshop on Statistical Learning in Computer Vision*, p. 59–74, 2004.

[10] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 194–201, 2012.

[11] E. Fidalgo, E. Alegre, V. González-Castro, and L. Fernández-Robles, "Illegal activity categorisation in darknet based on image classification using CREIC method," in *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17, León, Spain, September 6-8, 2017, Proceedings*, ser. Advances in Intelligent Systems and Computing, H. P. García, J. Alfonso-Cendón, L. Sánchez-González, H. Quintián, and E. Corchado, Eds., vol. 649, 2017, pp. 600–609.

[12] E. Fidalgo, E. Alegre, L. Fernández-Robles, and V. González-Castro, "Fusión temprana de descriptores extraídos de mapas de prominencia multinivel para clasificar imágenes," *Revista Iberoamericana de Automática e Informática*, vol. 16, no. 3, pp. 358–368, 2019.

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 770–778.

Image hashing based on frequency dominant neighborhood structure

Rubel Biswas*†, Aitor Del Río*†, Roberto A. Vasco-Carofilis*†, Guru Swaroop Bennabhaktula*†‡, Verónica De Mata†, Enrique Alegre*†

*Department of Electrical, Systems and Automation, Universidad de León, León, ES
†Researcher at INCIBE (Spanish National Cybersecurity Institute), León, ES
‡Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence, University of Groningen, Groningen, NL
Email:{rubel.biswas, aitor.rio, andres.vasco, enrique.alegre}@unileon.es,
veronica.demata@incibe.es, g.s.bennabhaktula@rug.nl

***Abstract*—Numerous services are provided in Tor darknet. However, due to the anonymity provided, some of them can be considered suspicious or even illegal in certain countries. In this paper, we propose to detect such services using a new perceptual hashing approach we named Frequency-Dominant Neighborhood Structure (F-DNS). After evaluating the robustness of F-DNS under common attacks, we introduce Darknet Usage Services Image-2K (DUSI-2K), a new dataset containing screenshots of active domains in Tor. Using DUSI-2K dataset, we demonstrate that perceptual hash can be used for detecting Tor services using only a screenshot of the service itself. The experimental results demonstrate that F-DNS obtains an accuracy of 98.75%, surpassing other state-of-the-art methods applied to DUSI-2K.**

***Index Terms*—Perceptual hashing, Tor, DCT, Image classification.**

Type of contribution: *Research already published [1]*

I. INTRODUCTION

The Onion Router (Tor) [2] is one of the most famous anonymous networks on the Deep Web that provides a wide range of legal and illegal hidden services. Al Nabki et al. [3] conclude that 48% of the content is legal and, at least, 20% of domains contain content considered as illegal in many countries. To monitor these services manually by the Law Enforcement Agencies (LEAs) is challenging. Therefore, there is a need to monitor such hidden services automatically [2], [4], [5].

By taking into account the LEAs interest, we present a new perceptual hashing method, called Frequency-Dominant Neighborhood Structure (F-DNS) [1], for the task of detecting Tor domains using only their screenshots.

II. DARKNET USAGE SERVICE IMAGES-2K (DUSI-2K)

DUSI [6] dataset[1] contained snapshots of six categories of Tor domains. We extended it in DUSI-2K[1], with 16 categories of Tor domains using a semi-supervised process. We can see an example of images in the Figure 1.

III. F-DNS HASH CONSTRUCTION

F-DNS hashing approach consists of three steps: (I) pre-processing, (II) feature extraction, and (III) hash generation. In the pre-processing phase, the original image is first converted to gray-scale, and then a Gaussian filter is applied.

Figure 1. DUSI-2K dataset examples in the following order: Blog - Casino - Cryptocurrency - Cryptolocker - Down - Drugs

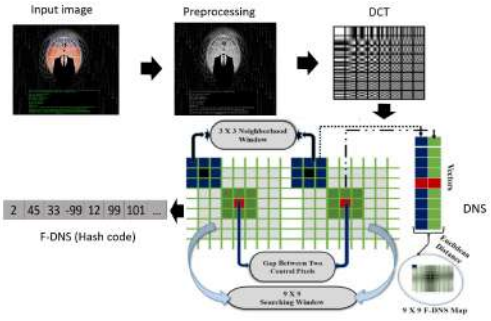


Figure 2. Pipeline of F-DNS hashing method.

In the feature extraction phase, the Discrete Cosine Transform (DCT) is applied over the gray-scale image looking for the texture energy of each pixel. After that, the output of the DCT is used as an input to the Dominant Neighborhood Structure (DNS) [7] to extract the features.

Given a central pixel x, the DNS D is obtained by determining the intensity similarity of all pixels x', which are in the searching window of size $N \times N$ pixels. To compute the similarity of each pixel x', the Euclidean distance is measured using the intensities in the flattened matrices $M \times M$ around x and x'. The area of $M \times M$ pixels is called the neighborhood window.

As the DNS is computed from the DCT of an image, we call the extracted map Frequency-Dominant Neighborhood Structure (F-DNS). After obtaining N F-DNS maps from the whole image, we compute the Frequency-Global Neighborhood Structure (F-GNS) by summing up the obtained F-DNS maps.

In the hash generation phase, the final hash of the image is obtained using the DCT coefficients but without taking into

[1]https://gvis.unileon.es/dataset/darknet-usage-service-images/

account the first row and column of the DCT coefficients matrix. This is done to exclude the information from the average pixel values. Finally, the hash is composed of 64 real values from the top-left 8×8 DCT coefficients matrix.

In this work, we used $N = 9$, i.e, 9×9 pixels, and $M = 3$, i.e, 3x3 pixels. Figure 2 depicts the complete process.

IV. RESULTS OF THE EXPERIMENTS

To evaluate the performance of F-DNS [1], we select 35 images from USC-SIPI [8] dataset for generating their attack versions through 10 different content-preserving operations. Then, we computed the correlation coefficient scores between the hash codes of the attacked images and the hash code of each original one by using F-DNS method.

We have compared the performance of F-DNS against RP-IVD [9], SS-Salient-SF [10], and pHash [11] by following the procedure mentioned in F-DNS. The average score of the correlation coefficients is reported in Table I where F-DNS obtain the highest results for all operations except for JPEG compression and Watermark embedding.

Table I
MEAN CORRELATION COEFFICIENT SCORES

Operation	SS-Salient-SF	RP-IVD	pHash	F-DNS (ours)
Brightness adjustment	0.9448	0.9583	0.9884	**0.9985**
Contrast adjustment	0.8942	0.9920	0.9967	**0.9993**
Gamma correction	0.9719	0.9957	0.9990	**0.9995**
Salt and pepper noise	0.9963	0.9872	0.9612	**0.9999**
Multiplicative noise	0.9947	0.9939	0.9754	**0.9999**
Gaussian filter	0.9927	0.9973	0.9988	**0.9999**
JPEG compression	**0.9997**	0.9986	0.9979	0.9993
Scaling	0.9723	0.9773	0.9704	**0.9875**
Rotation	0.0438	0.2959	0.2773	**0.9365**
Watermark embedding	**0.9989**	0.9601	0.9894	**0.9989**

Our proposal performs best against most content-preserving operations, and stands out for its performance in rotation, which is one of its major advantages over similar proposals.

Subsequently, we have tested our proposal using DUSI-2K. We randomly selected 79 templates from each Tor domain category and computed their hash codes using F-DNS. Later, we computed the hash code of each of the remaining 1545 images and determined their similarity with each class's 79 reference hash codes. A prediction was made based on the class with the highest similarity score and the performance was measured using accuracy.

The experiment was repeated 20 times and an average accuracy was computed. We also compared the DUSI-2K classification performance of F-DNS with RP-IVD [9], SS-Salient-SF [10] and pHash [11]. Since this task can be considered as image classification, we also considered the results from using Inception-ResNet-v2 [12] descriptors and ResNet50 [13] descriptors. 75% of the dataset was used for training a Support Vector Machine (SVM) with a lineal kernel and 25% for testing. The results are reported in Table II. Here, we can see that our approach, with 98.75% of accuracy, improves the rest of the methods.

Table II
TOR DOMAIN CLASSIFICATION ACCURACIES IN DUSI-2K

Method	Overall Accuracy	Method	Overall Accuracy
RP-IVD	95.84%	Inception-ResNet-v2	85.19%
SS-Salient-SF	95.45%	ResNet50	82.07%
pHash	89.91%	F-DNS	**98.75%**

V. CONCLUSIONS

We presented an image-based dataset named DUSI-2K that comprised 16 categories with 2500 snapshots of Tor domains and a new hashing approach F-DNS. We compared the performance of F-DNS with other hashing and image classification methods, and showed that F-DNS obtains the highest average correlation coefficient scores as well as the best DUSI-2K classification accuracy. We can put this work in the context of cybersecurity as INICBE did with this work, used by them to detect manipulated images in disk with child sexual abuse took by the police.

ACKNOWLEDGEMENTS

The work was supported by the Addendum 01, under the framework agreement between the University of León with INCIBE. Also, the support Of Nvidia Corporation for the donation of GeForce GTX Titan X and K-40.

REFERENCES

[1] R. Biswas, V. González-Castro, E. Fidalgo, and E. Alegre, "Perceptual image hashing based on frequency dominant neighborhood structure applied to tor domains recognition," *Neurocomputing*, vol. 383, pp. 24–38, 2020.

[2] M. W. Al Nabki, E. Fidalgo, E. Alegre, and V. González-Castro, "Detecting emerging products in tor network based on k-shell graph decomposition," *Jornadas Nacionales de Investigación en Ciberseguridad (JNIC)*, 2017.

[3] M. W. A. Nabki, E. Fidalgo, E. Alegre, and L. Fernández-Robles, "Torank: Identifying the most influential suspicious domains in the tor network," *Expert Syst. Appl.*, vol. 123, pp. 212–226, 2019. [Online]. Available: https://doi.org/10.1016/j.eswa.2019.01.029

[4] E. Fidalgo, E. Alegre, L. Fernández-Robles, and V. González-Castro, "Classifying suspicious content in tor darknet through semantic attention keypoint filtering," *Digit. Investig.*, vol. 30, pp. 12–22, 2019. [Online]. Available: https://doi.org/10.1016/j.diin.2019.05.004

[5] E. Fidalgo, E. Alegre, L. Fernández-Robles, and V. González-Castro, "Fusión temprana de descriptores extraídos de mapas de prominencia multinivel para clasificar imágenes," *Revista Iberoamericana de Automática e Informática*, vol. 16, no. 3, pp. 358–368, 2019.

[6] R. Biswas, E. Fidalgo, and E. Alegre, "Recognition of service domains on TOR dark net using perceptual hashing and image classification techniques," in *8th International Conference on Imaging for Crime Detection and Prevention, ICDP 2017, Madrid, Spain, December 13-15, 2017*. IET / IEEE, 2017, pp. 7–12. [Online]. Available: https://doi.org/10.1049/ic.2017.0041

[7] F. M. Khellah, "Texture classification using dominant neighborhood structure," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3270–3279, 2011. [Online]. Available: https://doi.org/10.1109/TIP.2011.2143422

[8] The usc-sipi image database. [Online]. Available: http://sipi.usc.edu/database

[9] Z. Tang, X. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 1, pp. 200–214, 2016. [Online]. Available: https://doi.org/10.1109/TIFS.2015.2485163

[10] C. Qin, X. Chen, J. Dong, and X. Zhang, "Perceptual image hashing with selective sampling for salient structure features," *Displays*, vol. 45, pp. 26–37, 2016. [Online]. Available: https://doi.org/10.1016/j.displa.2016.09.003

[11] C. Zauner, "Implementation and benchmarking of perceptual image hash functions," *Master thesis*, 2010.

[12] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, S. P. Singh and S. Markovitch, Eds. AAAI Press, 2017, pp. 4278–4284. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

Hardware dedicado para la optimización temporal del algoritmo NTRU

Eros Camacho-Ruiz, Macarena C. Martínez-Rodríguez, Santiago Sánchez-Solano, Piedad Brox
Instituto de Microelectrónica de Sevilla (CSIC / Universidad de Sevilla), España
C/Américo Vespucio, 28, 41219, Sevilla, España
camacho@imse-cnm.csic.es

***Resumen*—Los actuales algoritmos criptográficos se encuentran amenazados por la inminente llegada de la computación cuántica, por lo que los organismos internacionales, especialmente aquellos relacionados con la ciberseguridad, están potenciando el estudio e implementación de algoritmos que permitan volver a establecer entornos seguros de comunicación. En concreto, se plantean los algoritmos criptográficos post-cuánticos. Dentro de los algoritmos propuestos se encuentra el NTRU. Su principal inconveniente es el excesivo tiempo que requiere la multiplicación de polinomios usada en el proceso de cifrado. Por ello, este trabajo tiene como principal objetivo estudiar la posibilidad de utilizar hardware dedicado para acelerar la multiplicación. El uso de técnicas de codiseño hardware/software permite una implementación eficiente del criptosistema, donde las partes más costosas se ejecutan a nivel hardware. Este breve resumen recoge las últimas aportaciones que el grupo de investigación ha realizado en esta línea.**

***Index Terms*—Criptografía Post-Cuántica, Sistemas Empotrados, Codiseño HW/SW, NtruEncrypt**

Tipo de contribución: *Investigación ya publicada*

I. Introducción

Actualmente se requieren métodos de intercambio de información seguros y confidenciales. Uno de los métodos más utilizados es la criptografía asimétrica, que usa un par de claves público-privada para el envío de mensajes. Su principal ventaja es que evita el problema del intercambio de claves inherente a los sistemas de cifrado simétricos. Sin embargo, se encuentra especialmente amenazada por la inminente llegada de la computación cuántica [1]. Un ejemplo claro es la vulnerabilidad que presenta el algoritmo RSA, utilizado en las comunicaciones a través de Internet. La seguridad de dicho algoritmo [2] se encuentra comprometida por el algoritmo de Shor [3], el cual demuestra que el uso de ordenadores cuánticos permitiría llevar a cabo ataques de fuerza bruta en tiempos polinómicos. Por otro lado, este algoritmo también puede calcular en tiempo polinómico el conocido como Problema del Logaritmo Discreto (DLP) que es la base de otros sistemas criptográficos asimétricos [4].

Existe por tanto, la necesidad del desarrollo de nuevos algoritmos criptográficos que sean seguros frente a ataques de ordenadores clásicos y cuánticos, y que permitan trabajar con los protocolos de comunicación ya existentes. Así, emergen los denominados algoritmos criptográficos post-cuánticos. El NIST (*National Institute of Standards and Technology*) [5] lanzó un concurso para evaluar cuáles de estos son los mejores para su futura estandarización. Varias de las propuestas presentadas a esta competición están basadas en el algoritmo NTRU (*Nth Degree Truncated Polynomial Ring Unit*) [6] el cual utiliza tamaños de clave pequeños y opera a mayor velocidad que otros criptosistemas con el mismo nivel de seguridad. Una versión de dicho algoritmo, NTRUEncrypt, se encuentra estandarizada por el IEEE (*Institute of Electrical and Electronics Engineers*) en la norma IEEE 1363.1 [7]. Durante la ejecución del NTRUEncrypt, la multiplicación de polinomios es lo que más tiempo consume. Una posible solución para acelerar este proceso es el uso de técnicas de codiseño *hardware/software* (HW/SW), donde el criptosistema se ejecuta en un procesador a nivel software, mientras que el multiplicador se encuentra implementado a nivel hardware.

II. Entorno matemático

El algoritmo NTRU está basado en una estructura algebraica denominada anillo de polinomios truncado descrito por:

$$R_{N,t} = \mathbb{Z}_t[x]/(x^N - 1) \tag{1}$$

donde N indica el grado del polinomio y $\mathbb{Z}_t[x]$ todos los posibles polinomios cuyos coeficientes son enteros módulo t. Cualquier polinomio truncado dentro de este anillo puede ser descrito como:

$$a(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{N-2} x^{N-2} + a_{N-1} x^{N-1} \tag{2}$$

La multiplicación entre dos polinomios (por ejemplo, $c(x) = a(x) \times b(x)$) se define como:

$$c_k = \sum_{\substack{i,j=0,1,\ldots,N-1 \\ i+j=k \bmod N}} ((a_i \cdot b_j) \bmod t), k = 0, \ldots, N-1 \tag{3}$$

NTRUEncrypt requiere la definición de algunos parámetros para establecer la estructura matemática del criptosistema: N es el grado del polinomio; p y q son dos enteros que representan los módulos de diferentes tipos de multiplicaciones; y por último, d_f, d_g y d_r son enteros que representan el número de coeficientes no nulos de algunos polinomios. A partir de estos parámetros se pueden definir distintos niveles de seguridad recogidos en [7].

Durante el proceso de cifrado del NTRUEncrypt se requiere el cálculo de una multiplicación entre dos polinomios truncados. Específicamente los polinomios que intervienen en la multiplicación son el polinomio de ofuscamiento, $r(x)$ y la clave pública, $h(x)$. El resultado de la multiplicación es sumado con el mensaje, $m(x)$, para terminar el proceso de cifrado sobre $e(x)$. Matemáticamente puede expresarse como $e(x) = (h(x) \times r(x) + m(x)) \bmod q$. Utilizando la Ec. 3 es posible llevar a cabo la multiplicación $h(x) \times r(x)$. Es fácil comprobar que se requerirán N^2 ciclos de reloj para completar la operación si esta se lleva a cabo de forma secuencial y cada uno de los productos parciales se ejecuta en un ciclo de reloj.

III. Desarrollo del trabajo

III-A. Arquitectura propuesta

Con la idea de paralelizar al máximo posible la operación de multiplicación y reducir los ciclos de operación se plantea la posibilidad de utilizar estructuras tipo *Linear Feedback Shift Register* (LFSR). Así, en [8] se describe un módulo hardware que lleva a cabo la multiplicación de polinomios en el cifrado del NTRUEncrypt utilizando LFSRs. En [9] introducimos el módulo en un sistema empotrado añadiendo la posibilidad de interoperar con el procesador donde se encuentra el criptosistema ejecutándose a nivel software.

Por otro lado, explotando que el polinomio $r(x)$ es disperso (cuenta con un gran número de ceros) se plantea en [10] la posibilidad de acelerar el sistema eliminando un ciclo de operación cuando se encuentren dos ceros consecutivos. En [11] se demuestra que existe una degradación del nivel de seguridad debida a esta aceleración. Sin embargo, esta brecha en la seguridad es solventada en el trabajo que presentamos en [12] que también utiliza estructuras LFSRs pero permite acelerar cuando en $r(x)$ se encuentren dos, tres y cuatro ceros consecutivos. Dicha arquitectura se muestra en la Figura IV. Esta solución reduce en aproximadamente un 50 % el tiempo de operación de la multiplicación a nivel hardware con respecto a trabajos anteriores.

III-B. Implementación

Para las implementaciones hardware descritas en [9] y [12] se ha utilizado el entorno PYNQ (*Python productivity for ZYNQ*). Este entorno ha permitido desarrollar de una manera rápida y eficiente la implementación del criptosistema NTRUEncrypt tanto a nivel software como a nivel de codiseño HW/SW. PYNQ cuenta con un marco de trabajo basado en Python sobre un sistema operativo Linux, proporcionado por un dispositivo Zynq-7000 SoC (*System-on-Chip*) del tipo XC7Z020-1CLG400C. Este SoC cuenta con una FPGA (*Field Programmable Gate Arrays*), donde se implementan los módulos hardware, y un procesador (ARM Cortex-A9) donde se ejecuta el criptosistema a nivel software. Utilizando buses de interconexión AXI4-Stream es posible acelerar el ancho de banda de las transferencias de los coeficientes de los polinomios. Dicho esquema es mostrado en la Figura IV.

A nivel software, es posible realizar aceleraciones incluyendo en el código Python fragmentos de código en C cuya ejecución es más rápida. En [12] se demuestra que es posible conseguir acelerar hasta 5 veces el tiempo de ejecución del cifrado completo para una solución híbrida de Python+C, mientras que hasta más de 600 veces para una solución exclusivamente en Python. En términos de recursos, según recoge [12], el sistema embebido que incluye el módulo multiplicador ocupa un total de la FPGA de aproximadamente el 62 %, 71 % y 92 % para los conjuntos de parámetros *ees541ep1*, *ees613ep1* y *ees887ep1* respectivamente.

IV. Conclusiones

En este resumen se han puesto en contexto los avances, recogidos en [9] y [12], que el grupo de investigación ha desarrollado recientemente en la aceleración de algoritmos criptográficos post-cuánticos, incluidos en el criptosistema NTRUEncrypt. Los resultados muestran que utilizando arquitecturas del tipo LFSR se consiguen aceleraciones muy ventajosas en el cifrado, requiriendo el consiguiente incremento en recursos.

Las líneas de avance del grupo van en el sentido de abordar el problema desde la perspectiva de dispositivos *wearable* donde los recursos son muchos más limitados. El primer avance se realizó recientemente en [13], estando actualmente centrados en la implementación eficiente de la versión de NTRU presentada a la tercera ronda de la competición del NIST.

Agradecimientos

Este trabajo ha recibido financiación del proyecto SPIRS (Secure Platform for ICT Systems Rooted at the Silicon Manfacturing Process) con el *Grant Agreement* nº 952622 *in the European Union's Horizon 2020 Research and Innovation programme*. M.C.M.R. disfruta de una beca Postdoc de la Junta de Andalucía financiada por PO FSE de la UE y Eros Camacho-Ruiz disfruta de una ayuda del tipo Formación del Profesorado Universitario (FPU20/03008).

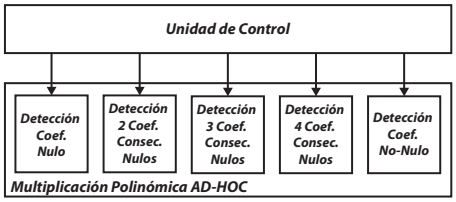


Figura 1. La arquitectura requiere de una Unidad de Control para implementar una multiplicación polinómica AD-HOC.

Figura 2. Esquema simplificado del sistema empotrado para el criptosistema NTRUEncrypt.

Referencias

[1] V. Mavroeidis, *et al.*, "The impact of quantum computing on present cryptography," *International Journal of Advanced Computer Science and Applications*, 2018.
[2] C. Paar and J. Pelzl, *Understanding Cryptography*, 2010.
[3] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *in Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, IEEE Comput. Soc. Press, 1994.
[4] U. Vazirani, "On the power of quantum computation," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 1998.
[5] NIST, "Post-Quantum Cryptography," 2020.
[6] J. Hoffstein, *et al.*, "NTRU: A ring-based public key cryptosystem," *in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1998.
[7] IEEE, "IEEE Standard Specification for Public Key Cryptographic Techniques Based on Hard Problems over Lattices," tech. rep., IEEE, 2009.
[8] B. Liu and H. Wu, "Efficient architecture and implementation for NTRUEncrypt system," *in Midwest Symposium on Circuits and Systems*, 2015.
[9] E. Camacho-Ruiz, *et al.*, "Accelerating the Development of NTRU Algorithm on Embedded Systems," *2020 XXXV Conference on Design of Circuits and Integrated Systems (DCIS)*, 2020, pp. 1-6
[10] B. Liu and H. Wu, "Efficient multiplication architecture over truncated polynomial ring for NTRUEncrypt system," *in Proceedings - IEEE International Symposium on Circuits and Systems*, 2016.
[11] K. Braun, *et al.*, "Secure and Compact Full NTRU Hardware Implementation," *IEEE/IFIP International Conference on VLSI and System-on-Chip, VLSI-SoC*, vol. 2018-Octob, pp. 89–94, 2019.
[12] Eros Camacho-Ruiz, *et al.*, "Timing-Optimized Hardware Implementation to Accelerate Polynomial Multiplication in the NTRU Algorithm,"*J. Emerg. Technol. Comput. Syst.* 17, 3, Article 35 (July 2021), 16 pages.
[13] S. Sánchez-Solano, *et al.* "Multi-Unit Serial Polynomial Multiplier to Accelerate NTRU-Based Cryptographic Schemes in IoT Embedded Systems," *Sensors*, vol. 22, no. 5, p. 2057, Mar. 2022

Diseño y evaluación de las prestaciones de funciones físicas no clonables basadas en osciladores en anillo sobre FPGAs

Macarena C. Martínez-Rodríguez, E. Camacho-Ruiz, P. Brox and S. Sánchez-Solano
Instituto de Microelectrónica de Sevilla, CSIC / Universidad de Sevilla, España
C/Américo Vespucio, 28, 41092, Sevilla, España
{macarena,camacho,brox,santiago}@imse-cnm.csic.es

***Resumen*—Los esquemas de seguridad basados en funciones físicas no clonables aprovechan las características intrínsecas del hardware para mejorar la seguridad de los dispositivos electrónicos. Este resumen presenta dos trabajos para diseñar y caracterizar funciones físicas no clonables basados en osciladores en anillo propuestas por nuestro grupo de investigación. El primero se centra en el flujo de diseño y caracterización basado en una herramienta incluida en el entorno de Matlab, mientras que el segundo presenta y caracteriza una función física no clonable basada en osciladores en anillo muy compacta y altamente configurable usando un flujo de diseño para sistemas empotrados basado en el entorno PYNQ.**

***Index Terms*—PUFs, RO-PUFs, FPGA**

Tipo de contribución: *Investigación ya publicada*

I. Introducción

Aumentar el nivel de seguridad de los sistemas empotrados es un desafío ambicioso. Los esquemas de seguridad basados en funciones físicas no clonables (Physical Unclonable Functions, PUFs) aprovechan las características intrínsecas del hardware para la generación de identificadores digitales y números aleatorios que permiten asegurar la protección de dispositivos electrónicos donde se integran. Un PUF implementa un mecanismo reto-respuesta que asigna una salida (respuesta) a una entrada (reto) [1]. La respuesta del PUF debe cumplir los siguientes requisitos: (i) unicidad: distintas instancias de un PUF devuelven respuestas distintas ante el mismo reto, (ii) repetibilidad: el PUF debe devolver una respuesta sin cambios cuando se aplica el mismo reto, e (iii) imprevisibilidad: la respuesta debe ser difícil de predecir incluso para el propio diseñador. Estas propiedades permiten poder usar la respuesta del PUF para reconstruir una clave criptográfica tantas veces como se requiera mediante algoritmos de datos auxiliares [2].

Los PUFs basados en silicio explotan la variabilidad en el proceso de fabricación de un circuito integrado en tecnologías CMOS. Entre este tipo de PUFs se encuentran los basados en memorias y los basados en retrasos de las señales electrónicas. En el primer grupo se encuadran los PUFs basados en SRAM, que aprovechan para obtener la respuesta del PUF el hecho de que las diferencias entre las tensiones de umbral de los transistores de cada celda de memoria hacen que estas se inicialicen con un valor u otro al alimentar el circuito [3]. Sin embargo, frecuentemente no se pueden implementar sobre FPGAs, ya que en este tipo de dispositivos los valores iniciales están fijados a un valor predeterminado eliminando por completo la impredicibilidad. Los PUFs basados en retrasos miden las diferencias temporales entre señales electrónicas generadas y/o transmitidas por circuitos a priori idénticos. Entre ellos están los PUFs basados en arbitradores y los basados en osciladores en anillo (Ring Oscillators, ROs). Un PUF basado en arbitrador evalúa la diferencia de retraso relativo entre dos trayectos con la misma longitud de diseño. Este tipo de PUF se construye utilizando un conjunto de multiplexores concatenados y un arbitrador [4], y son difíciles de implementar en FPGAs, ya que es complicado asegurar las mismas rutas de retraso en estos dispositivos. Por otra parte, el funcionamiento de los RO-PUFs consiste en comparar la diferencia de frecuencia de oscilación proporcionadas por dos ROs. Comparando ambos PUFs basados en retrasos, los RO-PUFs suelen presentar un mejor rendimiento en términos de unicidad, repetiblidad e imprevisibilidad frente a los PUFs basados en arbitradores [5].

II. Desarrollo del trabajo

Un oscilador en anillo es una cadena realimentada de un número impar de etapas inversoras que genera a la salida una señal oscilante. La frecuencia de oscilación es inversamente proporcional a la suma de los retrasos que se producen en cada una de las etapas inversoras. En la literatura aparecen diversos trabajos que describen la construcción de RO-PUFs usando bancos de ROs. Para obtener la respuesta asociada a un determinado reto, dos de los ROs son seleccionados y sus salidas son utilizadas como señal de reloj de dos contadores. La respuesta del PUF puede ser o bien el bit de signo que indica cuál de los contadores ha alcanzado un valor de cuenta mayor después de un tiempo determinado [2], o bien pueden ser varios bits del contador asociado al RO más lento cuando el RO con mayor frecuencia haya hecho saturar a su contador [6]. La salida del PUF se obtiene concatenando el o los bits seleccionados tras la aplicación de la secuencia de retos.

III. Aportaciones

La caracterización de los RO-PUFs es crucial para evaluar sus prestaciones e incluso para determinar qué bits del contador asociado al RO más lento cumplen los requisitos para ser seleccionados como respuesta del PUF. Debido a que la caracterización no se puede realizar por simulación, ya que el comportamiento del PUFs es intrínseco a su proceso de fabricación, hemos desarrollado dos trabajos que se centran en establecer metodologías de diseño y caracterización de los RO-PUFs. En uno de ellos, además, se propone un nuevo diseño de RO-PUF altamente compacto y configurable.

III-A. Caracterización de RO-PUFs usando Matlab en dispositivos FPGAs

En [7] presentamos un marco unificado para diseñar, implementar y evaluar el rendimiento de los RO-PUFs en FPGAs utilizando el entorno Matlab. El flujo de diseño utiliza la herramienta de procesamiento digital de señales (DSP) Xilinx System Generator integrada en el entorno Matlab/Simulink. El uso de esta metodología basada en modelos facilita la descripción de los diseños y la evaluación de las prestaciones de los PUFs, proporcionando un entorno eficiente para aplicar los retos al RO-PUF, adquirir las respuestas mediante el uso de co-simulación hardware y calcular un conjunto de métricas para cuantificar la estabilidad, la probabilidad y la entropía para determinar qué bits pueden conformar la respuesta del PUF, y las métricas de intra e inter distancia de Hamming que determinan la unicidad y repetibilidad de los PUFs. Adicionalmente, se prueba la robustez de los PUFs ofuscando y recuperando secretos.

El flujo de diseño se aplicó para evaluar el rendimiento de RO-PUFs implementados en 17 placas Basys 3 con FPGAs de la familia Artix-7. Se desarrollaron varios scripts y funciones para evaluar las propiedades de los PUFs. Los resultados experimentales ponen de manifiesto que se pueden usar 2 bits por cada reto para generar la respuesta del PUF obteniendo un buen rendimiento en términos de repetibilidad y unicidad. La robustez del PUF en la ofuscación y recuperación de un secreto se corrobora con resultados experimentales. Ningún dispositivo impostor es capaz de recuperar un secreto en ninguno de los escenarios estudiados.

III-B. Caracterización de RO-PUFs en sistemas empotrados

En [8] abordamos el diseño de un RO-PUF configurable que emplea varias estrategias para proporcionar una solución eficiente en términos de área, respuesta temporal y rendimiento. La implementación de RO-PUFs en dispositivos lógicos programables fue concebida para minimizar el uso de los recursos disponibles, combinando adecuadamente algunas de las técnicas reportadas previamente en la literatura, mientras que la velocidad de operación puede optimizarse seleccionando adecuadamente el tamaño de los elementos utilizados para obtener la respuesta del PUF. El banco de ROs fue diseñado de forma compacta de tal manera que con los mismos elementos lógicos se pueden implementar dos ROs distintos, aumentando el tamaño efectivo del PUF. La longitud de la respuesta del PUF también se puede aumentar seleccionando más de un bit en cada comparación de pares de ROs. El diseño proporciona dos opciones para este propósito, utilizar el bit de signo o bien seleccionar uno o más bits del contador asociado al RO más lento.

Propusimos una metodología para caracterizar RO-PUFs basada en el entorno PYNQ [9] que facilita la comunicación entre hardware y software en un sistema empotrado de la familia Zynq-7000. Sobre el sistema de procesado se ejecuta un sistema operativo Linux y el hardware se integra mediante una serie de rutinas en lenguaje C. El RO-PUF propuesto se concibió como un módulo IP parametrizable que incorpora una interfaz estándar AXI4-Lite para facilitar su integración con el sistema de procesado. El tamaño del banco de ROs y su ubicación en la FPGA se pueden elegir antes del proceso de síntesis e implementación. Otros aspectos de la funcionalidad del PUF (como el tamaño de los contadores, la generación de señales de habilitación o la estrategia a aplicar en la secuencia de retos) se pueden elegir dinámicamente utilizando los registros de E/S mapeados en el espacio de memoria del sistema de procesado. Mediante una función C se establecen los valores de configuración, la secuencia de retos y se capturan los datos de salida del módulo-IP. Los datos se usan para calcular el conjunto de métricas necesarias para la caracterización del RO-PUF.

Se llevó a cabo una exhaustiva batería de pruebas (diseñando un conjunto de programas en C que recorren diferentes configuraciones) con el fin de analizar la influencia en los índices de calidad del PUF de diferentes parámetros y opciones. Estas se ejecutaron en 15 placas de desarrollo Pynq-Z2 que implementan un diseño que incorpora 10 PUFs diferentes. Los resultados incluidos en el artículo ilustran el procedimiento para la selección de bits de la respuesta del PUF (bits de signo y/o bits de contador) que permite establecer compromisos adecuados entre repetibilidad y unicidad.

IV. Conclusiones

Este trabajo resume brevemente las líneas de nuestro grupo de investigación sobre el diseño y las metodologías de caracterización de RO-PUFs sobre FPGAs. En [7] presentamos una metodología usando Matlab y verificamos además la robustez del PUF ofuscando y recuperando un secreto. En [8] presentamos una metodología de caracterización basada en el entorno PYNQ para sistemas empotrados. Propusimos también un diseño compacto y configurable, caracterizando y analizando sus prestaciones.

Agradecimientos

Este trabajo ha recibido financiación del proyecto SPIRS con el Grant Agreement nº 952622 in the European Union's Horizon 2020 Research and Innovation programme. M.C.M.R. disfruta de una beca Postdoc de la Junta de Andalucía financiada por PO FSE de la UE y Eros Camacho-Ruiz disfruta de una ayuda del tipo Formación del Profesorado Universitario (FPU20/03008).

Referencias

[1] R. Pappu, *et al.*, "Physical One-Way Functions,"*Science*, vol. 297, no. 5589, pp. 2026-2030, 2002.

[2] G. Edward Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of Design Automation Conference*, pp. 9-14, 2007.

[3] S. S. Riya and V. Lalu, "Stable cryptographic key generation using SRAM based Physical Unclonable Function," in *Proc. IEEE Int. Conf. on Smart Electronics and Communication*, pp. 653-657, 2020.

[4] A. Shamsoshoara, *et al.*, "A survey on physical unclonable function (PUF)-based security solutions for Internet of Things," in *Computer Networks*, vol. 183, 2020.

[5] G. Komurcu, *et al.*, "Analysis of Ring Oscillator structures to develop a design methodology for RO-PUF circuits," in *IFIP/IEEE 21st Int. Conf. on Very Large Scale Integration (VLSI-SoC)*, 2013.

[6] F. Kodytek and R. Lorencz, "Improved ring oscillator PUF on FPGA and its properties," in *Microprocessors and Microsystems*, vol. 47, pp. 55 - 63, 2016.

[7] M.C. Martínez-Rodríguez,*et al.*, "Design Flow to Evaluate the Performance of Ring Oscillator PUFs on FPGAs," in *Conf. Design of Circuits and Integrated Systems (DCIS)*, pp. 1 - 6, 2021

[8] M.C. Martínez-Rodríguez, *et al.*, "A Configurable RO-PUF for Securing Embedded Systems Implemented on Programmable Devices," in *Electronics*, 2021

[9] PYNQ—Python productivity for Zynq. Online: http://www.pynq.io/, Accedido el 03/06/2022

Optimal botnet detection on network data

Daniel Diaz
Dpto. IESA,
Univesidad de León
Researcher at INCIBE
León, Spain
ddiao@unileon.es

Javier Velasco-Mata
Dpto. IESA,
Univesidad de León
Researcher at INCIBE
León, Spain
javier.velasco@unileon.es

Aitor Del Río
Dpto. IESA,
Univesidad de León
Researcher at INCIBE
León, Spain
ariof@unileon.es

*Eduardo Fidalgo
Dpto. IESA,
Univesidad de León
Researcher at INCIBE
León, Spain
eduardo.fidalgo@unileon.es

***Abstract*—Along with the rising number of devices connected to the Internet, botnets also grow in size, supposing billionaire losses to global economies. Therefore, it is necessary to develop efficient detector systems that can be easily updated in light of new types of bots. In this work, we present two datasets based on the well known CTU-13: one that balances the classes, QB-CTU13, and a second one that adds data from three challenging botnets, EQB-CTU13. Moreover, we use the information Gain and Gini Importance to conduct a study on an initial set of eleven well-tested features to obtain an optimized subset, and we tested the performance of three Machine Learning classifiers in the task of botnet detection. In our experimentation, Decision Tree shows the best trade-off between detection rate with an F1-score of 85% and an average of 0.78 microseconds for classifying each sample.**

***Index Terms*—Botnet, Decision Tree, Optimization**

Contribution: Summary of *Efficient Detection of Botnet Traffic by features selection and Decision Trees* [1]

I. Introducción

Botnets are one of the main threats on the Internet, whose main purpose is to amplify other cyber-attacks such as phishing [2], [3]. It is estimated that in 2021 the economic losses related to cybercrime ascended to $6.9 billion, as reported by the FBI's Internet Crime Complaint Center (IC3) [4]. Furthermore, according to the Spamhaus report Botnet Threat Update[1], the last quarter of 2021 experimented a 23% rise in the number of botnet Command and Control servers. This circumstance explains the need to develop botnet detectors capable of being quickly updated, such as the ones based on Machine Learning (ML).

Although there is research on botnet detection using ML [5], [6], we noticed a lack of research on an efficient detection, that is, one also considers the time required by the algorithm to classify the samples. In our paper, [1], we proposed the following contributions:

- We developed two new datasets based on the widely used CTU-13 dataset [5] that solves the problem of the imbalances between classes of the CTU-13 dataset.
- After assessing the importance of eleven network features, we proposed three optimal feature sets for a fast botnet detection and test them with three ML algorithms: Decision Tree (DT), Random Forest (RF) and k-Nearest Neighbors (k-NN).

[1]Accessible at https://www.spamhaus.org/news/article/817/spamhaus-botnet-threat-update-q4-2021, last checked April 2022

II. Background

Recent works show that k-NN and DT based models like RF usually obtains the best performance in the task of botnet detection on network flows. Gradelrab et al. [7] reported that DT achieved an F1-score of 95% detecting traces of six different botnets. This result matches with our previous research [8] where DT produced an F1-score of 99% on the TCP-Int dataset which included three botnets different than the ones used in the work of Gradelrab et al. and for the case of the k-NN model, it achieved an F1-score of 97% in the TCP-Int dataset.

III. Methodology

First, we selected an initial set of eleven features used in our previous work [8]:

- The source (sPort) and destination (dPort) ports.
- The mean (mLen) and variance (vLen) of the payload lengths of the packets.
- The mean (mTime) and variance (vTime) of the interval times between consecutive packets in the TCP flow, independently of the machine that sent them.
- The mean (mResp) and variance (vResp) of the interval times between a packet reaches a machine, and this machine responds to with another packet.
- The total number of bytes exchanged in the TCP flow (nbytes).
- The total number of SYN flags used during the TCP flow (nSYN).
- The total number of packet exchanged in the TCP flow (nPackets).

Then, we proposed three candidates of optimized feature subsets using the information provided by the Gini Importance (GI) and Information Gain (IG) measures from the initial feature set.

Finally, we used the QB-CTU13 dataset to compare the candidate sets of features using the ratio between the F1-score over the mean time required to classify a sample, expressed in microseconds.

IV. Experimentation and Results

A. Datasets

We balanced the samples among classes of the CTU-13 dataset by balancing the sample quantity of the major classes and including all the samples of the minor classes, in a dataset we called Quasi-Balanced CTU-13 (QB-CTU13). Then, we

further improved the dataset by adding samples generated by three modern and reportedly challenging botnets [9], [10], as detailed in Table I. We called this dataset Extended QB-CTU (EQB-CTU13).

TABLE I
SAMPLES PER CLASS IN QB-CTU13 AND EQB-CTU13 DATASETS.

Class	QB-CTU13	EQB-CTU13
Normal	3890	3890
Neris	3890	3890
Rbot	3890	3890
Virut	3890	3890
Murlo	2036	2036
NSIS	355	355
Donbot	233	233
Sogou	36	36
Bunitu	-	3890
Miuref	-	3890
NotPetya	-	111

Based on the higher results of GI and IG shown in Table II, we proposed the following features sets:

- 5-feature set: [dPort, nPackets, nBytes, vLen, mLen]
- 6-feature set: [dPort, nPackets, nBytes, vLen, mLen, mTime]
- 7-feature set: [dPort, nPackets, nBytes, vLen, mLen, mTime, vTime]

TABLE II
SAMPLES PER CLASS IN QB-CTU13 AND EQB-CTU13 DATASETS

Feature	Gini Imp.	Info. Gain
dPort	0.669	0.611
nPackets	0.167	0.291
vLen	0.086	0.614
mTime	0.026	0.463
vTime	0.009	0.519
mResp	0.005	0.456
mLen	0.005	0.612
vResp	0.002	0.436
sPort	0.002	0.053
nBytes	0.000	0.617
nSYN	0.000	0.045

B. Performance evaluation

We selected DT, RF and k-NN as the candidates to build an efficient botnet detector following the outcomes of the literature review from Section II. In Fig. 1, we compared the three models, along with the three proposals of optimized feature sets, as well as the full feature set of eleven features. We used the EQB-CTU13 dataset for this experiment. Both the 5-feature and the 6-feature sets achieved the best performance when using a DT, with an F1-Score of 0.85 and average of $0.78\mu s$ for classifying a sample. The 6-feature set on the same classifier achieved a 0.87 F1-score while classifying samples in a meantime of $0.8\mu s$.

V. DISCUSSION AND CONCLUSION

K-NN models achieved lower performance than DT based models due to the time cost of searching the nearest neighbors. Besides, comparing RF with a single DT, it can be seen that adding more DTs for classification supposed a bigger computational expense than the benefit of a better F1-score. Thus, in the perspective of performance, a single DT is preferable.

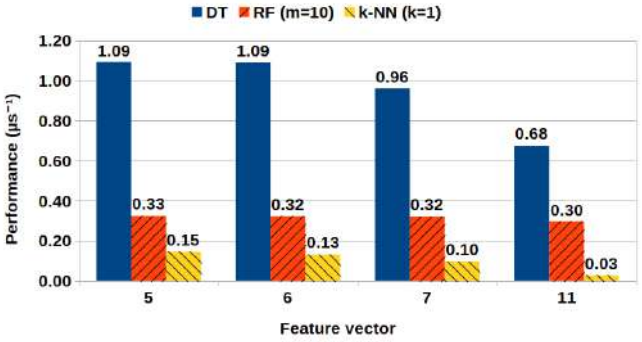


Fig. 1. Performance, as the ratio of F1-score over the mean time to classify a sample, using the four feature sets and the classifiers DT, RF and k-NN

The 5-feature and the 6-feature sets achieved a similar performance, but if an extra feature is added, it should be computed too, penalizing the time performance. For that reason, we recommend the 5-feature set and the DT as the best configuration among the tested ones.

For future works, we would study botnet detectors on high bandwidths networks, where the large number of packets transmitted by second makes it necessary to use high performative classifiers.

ACKNOWLEDGEMENTS

This work was supported by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01, and by the FPU (Formación de Profesorado Universitario) grant of the Spanish Government with reference FPU18/05804.

REFERENCES

[1] J. Velasco-Mata, V. González-Castro, E. F. Fernández, and E. Alegre, "Efficient detection of botnet traffic by features selection and decision trees," *IEEE Access*, vol. 9, pp. 120 567–120 579, 2021.

[2] F. Castaño, M. Sánchez-Paniagua, J. Delgado, J. Velasco-Mata, A. Sepúlveda, E. Fidalgo, and E. Alegre, "Evaluation of state-of-art phishing detection strategies based on machine learning," in *JNIC Live*. Ediciones de la Universidad de Castilla-La Mancha, 2021.

[3] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, A.-N. M. Wesam, and V. González-Castro, "Phishing url detection: A real-case scenario through login urls," *IEEE Access*, vol. Early Access, pp. 1–1, 2022.

[4] FBI, "2021 Internet Crime Report," FBI's Internet Crime Complaint Center (IC3), February 2022. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2021_IC3Report.pdf

[5] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100–123, 2014.

[6] L. F. B. Silva, L. N. Utimura, K. A. P. da Costa, M. A. Z. M. e Silva, and S. d. G. D. Prado, "Study on machine learning techniques for botnet detection," *IEEE Latin America Transactions*, vol. 18, no. 05, pp. 881–888, 2020.

[7] M. S. Gadelrab, M. ElSheikh, M. A. Ghoneim, and M. Rashwan, "BotCap: Machine learning approach for botnet detection based on statistical features," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 10, no. 3, pp. 563–579, 2018.

[8] J. Velasco-Mata, E. Fidalgo, V. González-Castro, E. Alegre, and P. Blanco-Medina, "Botnet detection on TCP traffic using supervised machine learning," in *International Conference on HAIS*. Springer, 2019, pp. 444–455.

[9] D. C. Le, N. Zincir-Heywood, and M. I. Heywood, "Unsupervised monitoring of network and service behaviour using self organizing maps," *Journal of Cyber Security and Mobility*, pp. 15–52, 2019.

[10] B. Abraham, A. Mandya, R. Bapat, F. Alali, D. E. Brown, and M. Veeraraghavan, "A comparison of machine learning approaches to detect botnet traffic," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

A summary of: "Defending Industry 4.0: An enhanced authentication scheme for IoT devices"

Nasour Bagheri
Nbagheri@sru.ac.ir

Saru Kumari
Saryusiirohi@gmail.com

Carmen Camara
macamara@pa.uc3m.es

Pedro Peris-Lopez
pperis@inf.uc3m.es

***Abstract*—To address the security concerns of Industry 4.0, recently, Garg *et al.* proposed a lightweight authentication protocol, and Akram *et al.* showed some of its security drawbacks. We continue this line by exposing how Garg *et al.*'s protocol suffers from non-invasive and invasive attacks. First, we explain a passive attacker can trace any two communicating nodes to compromise their location privacy. Next, we show an active though non-invasive adversary can compromise the integrity of the exchanged messages without being detected and run a de-synchronization attack. Besides, the adversary can extract any shared session key from any pair of nodes in the protocol. We named this attack a pandemic session key disclosure attack, and its consequences are more harmful than the impersonation of a compromised node. Finally, we disclose how the proposed scheme does not guarantee privacy protection for the keys when we assume an honest but curious server. To overcome those existing security flaws, we finally propose a revised protocol called `TARDIGRADE`. First, our informal analysis and then our formal security analysis using the Real-or-Random model (ROR) shows that `TARDIGRADE` provides the desired security, and likewise, our performance analysis confirms a reasonable cost compared to Garg *et al.*'s protocol.**

***Index Terms*—Internet of Things, Industry 4.0, Security, Privacy**

Type of contribution: *Published article (max. 2 pages)*

I. Introduction

There are several challenges to deploy IoT technology in the Industry 4.0 setting, and one vital challenge is the different cyber threats, physical attacks, or both, that are targeting the IoT devices. To dealing with these concerns, as an example, Esfahani *et al.* [1] proposed a web authentication mechanism to prevent man-in-the-middle attacks in industry 4.0 supply chains. Radanliev *et al.* [2] presented a systematic synthesis of the literature related to the impact of IoT-based supply chains and their related cyber risks. Sengupta *et al.* [3] proposed a fog-based architecture to provide the desired security for Industrial IoT (IIoT) and Industry 4.0. Chamikara *et al.* [4] devoted their study to introduce a framework for reaching privacy and reliability in the IIoT. Zhang *et al.* [5] proposed an anonymous batch authentication scheme for smart vehicular networks, as a type of IIoT, and Zhao and Dong [6] proposed an entropy based feature selection method for IIOT. Some researches, e.g. [7], also suggested the use of Software Defined Networks (SDN) networks in Industries 4.0. However, SDN has its drawbacks when it comes to security [8]. To address the security concerns of employing IoT based smart devices in industry 4.0, Garg *et al.* [9] recently proposed a lightweight mutual authentication and key agreement protocol, which is more efficient compared to the previous related works. Unfortunately, later Akram *et al.* [10] have shown that the scheme does not provide the desired security under a (semi-)invasive adversarial model.

II. Security analysis of Garg *et al.*

We first throw a little light on Akram *et al.*'s comment on the security of Garg et al.'s protocol. Akram *et al.* [10] showed that in a (semi-)invasive adversarial model in which the adversary can access the N_i's memory, the attacker could disclose the secret value d_i and also its CRPs. Given that information, then it is straightforward to impersonate N_i at any time. They also suggested a remedy to fix this security hole, which could also be a solution for any other similar protocol in which the secret values are directly stored in the node's memory. The solution consists of holding a randomised version of the private data in the memory of the node. These values, when necessary, can be reordered to their correct form by a dedicated assembly code. Unfortunately, if the adversary can access the program data of the N_i's processor (e.g., a micro-controller), the adversary could compromise the assembly instructions and consequently discloses the secret values. Besides, we want to highlight that Akram *et al.* [10] do not claim a full-proof solution and only gave some indications.

To continue in this vein, we highlight other security pitfalls of Garg *et al.*'s protocol as below: 1) Similarly than in [10], we assume the adversary can compromise N_i and, consequently, achieve its memory records (i.e., $(< \mathcal{C}_{i1}, \mathcal{R}_{i1} >, < \mathcal{C}_{i2}, \mathcal{R}_{i2} >, d_i)$). We also assume that the server S uses constant and preshared CRPs for each node; otherwise, the protocol does not work as was already mentioned in [10]. Then, we extend the Akram *et al.*'s (semi-)invasive attack to what we name as a pandemic session key disclosure attack. Under this pandemic approach, the adversary can disclose the session key even if the protocol runs between non-compromised nodes; 2) We show how the protocol presents security pitfalls even under a non-invasive adversarial model. The adversary succeeds in a traceability attack and can compromise the integrity of the messages exchanged in the protocol.

The proposed attacks are mainly based on the observations described below: **Observation-1.** In a key agreement

TABLE I
SECURITY COMPARISON, WHERE P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9 AND P_{10} RESPECTIVELY DENOTE SECURITY AGAINST REPLAY ATTACK,IMPERSONATION ATTACK, TRACEABILITY AND ANONYMITY, SECRET DISCLOSURE ATTACK, SESSION KEY SECURITY, DE-SYNCHRONIZATION ATTACK, MAN-IN-THE-MIDDLE ATTACK, INSIDER ADVERSARY, FORWARD SECRECY AND PANDEMIC ATTACK.

Protocol	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
[11]	✓	✓	×	✓	×	×	×	×	×	✓
[12]	✓	×	×	✓	✓	✓	×	✓	×	✓
[13]	✓	×	×	✓	-	×	✓	×	-	✓
[9]	✓	×	×	✓	×	✓	✓	×	✓	×
Our	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

session between N_i and N_j, the S server sends $SK^i_{info} = \mathcal{C}_{i2} \oplus H(\mathcal{C}_{i2}\|\mathcal{C}_{j2}\|r_s\|d_s)$ to N_i and $SK^j_{info} = \mathcal{C}_{j2} \oplus H(\mathcal{C}_{i2}\|\mathcal{C}_{j2}\|r_s\|d_s)$ to N_j. Assuming that the adversary has compromised N_i, he can extract $H(\mathcal{C}_{i2}\|\mathcal{C}_{j2}\|r_s\|d_s)$ from SK^i_{info} and therefore $\mathcal{C}_{j2}$ from SK^j_{info}; **Observation-2.** In step 1 of the protocol, the N_i node sends its identifier and the identifier of N_j in plain-text over a public and insecure channel; **Observation-3.** In steps 2 and 3 of the protocol, the S server sends the messages related to the shared key to the N_i and N_j nodes, respectively. However, the integrity of these messages is not guaranteed.

III. TARDIGRADE, THE REVISED PROTOCOL

In this section, we propose TARDIGRADE as a revised version of the Garg *et al.*'s protocol to remedy its security flaws. In a nutshell and as a significant difference with the original protocol, in our proposed solution, we require that nodes can generate CRPs. We also assume that each node is equipped with a reliable $PUF(.)$. Similar to Garg *et al.*'s protocol, TARDIGRADE includes three phases, i.e. initialization, registration, and mutual authentication & key agreement phases, respectively.

To show the security soundness of TARDIGRADE against various attacks, we provide our formal and informal security reasoning in this section. The formal security proof is conducted on the real or random model, and informal security analysis against various attacks, including pandemic, replay, impersonation, and de-synchronization attacks, is also provided. Table I represents a security comparison between TARDIGRADE and the most relevant PUF-based works, including the Garg *et al.*'s scheme.

Garg *et al.* compared their proposal to the state of the art of related works and showed how their scheme outperforms the existing solutions in terms of security and efficiency. Therefore, for the sake of avoiding repetition, we only compare TARDIGRADE with Garg *et al.*'s protocol in terms of performance. We present the comparison between the required primitives of Garg *et al.*'s protocol and TARDIGRADE scheme in Table II.

Summarising, in this paper, we present several new powerfull attacks (including some non-invasive) against Garg *et al.* protocol. Then we propose an enhanced version, called TARDIGRADE, to remedy the known and harmful attacks against the original protocol.

TABLE II
REQUIRED PRIMITIVES

Protocol	N_i	N_j	S
Garg *et al.*'s	ECC, H, RNG,PUF	ECC, H, RNG,PUF	ECC, H, RNG
TARDIGRADE	ECC, H, PUF	ECC, H, PUF	ECC, H, RNG

ACKNOWLEDGEMENT

This work was supported by Leonardo Grant for Researchers and Cultural Creators, BBVA Foundation (P2019-CARDIOSEC); by the Spanish Ministry of Science, Innovation and Universities grant PID2019-111429RBC21(ODIO); and by the Comunidad de Madrid (Spain) under the project CYNAMON (P2018/TCS-4566), co-financed by European Structural Funds (ESF and FEDER).

REFERENCES

[1] A. Esfahani, G. Mantas, J. C. Ribeiro, J. Bastos, S. Mumtaz, M. A. Violas, A. M. D. O. Duarte, and J. Rodriguez, "An efficient web authentication mechanism preventing man-in-the-middle attacks in industry 4.0 supply chain," *IEEE Access*, vol. 7, pp. 58 981–58 989, 2019.

[2] P. Radanliev, D. D. Roure, K. R. Page, J. R. C. Nurse, R. M. Montalvo, O. Santos, L. T. Maddox, and P. Burnap, "Cyber risk at the edge: current and future trends on cyber risk analytics and artificial intelligence in the industrial internet of things and industry 4.0 supply chains," *Cybersecur.*, vol. 3, no. 1, p. 13, 2020.

[3] J. Sengupta, S. Ruj, and S. D. Bit, "A secure fog-based architecture for industrial internet of things and industry 4.0," *IEEE Trans. Ind. Informatics*, vol. 17, no. 4, pp. 2316–2324, 2021.

[4] M. A. P. Chamikara, P. Bertók, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "A trustworthy privacy preserving framework for machine learning in industrial iot systems," *IEEE Trans. Ind. Informatics*, vol. 16, no. 9, pp. 6092–6102, 2020.

[5] J. Zhang, H. Zhong, J. Cui, Y. Xu, and L. Liu, "An extensible and effective anonymous batch authentication scheme for smart vehicular networks," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3462–3473, 2020.

[6] L. Zhao and X. Dong, "An industrial internet of things feature selection method based on potential entropy evaluation criteria," *IEEE Access*, vol. 6, pp. 4608–4617, 2018.

[7] T. Lins and R. A. R. Oliveira, "Cyber-physical production systems retrofitting in context of industry 4.0," *Comput. Ind. Eng.*, vol. 139, p. 106193, 2020.

[8] K. Benzekki, A. E. Fergougui, and A. E. Elalaoui, "Software-defined networking (SDN): a survey," *Security and Communication Networks*, vol. 9, no. 18, pp. 5803–5833, 2016.

[9] S. Garg, K. Kaur, G. Kaddoum, and K.-K. R. Choo, "Towards secure and provable authentication for internet of things: Realizing industry 4.0," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4598–4606, 2019.

[10] M. A. Akram, K. Mahmood, S. Kumari, and H. Xiong, "Comment on "towards secure and provable authentication for internet of things: Realizing industry 4.0"," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4676–4681, 2020.

[11] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A puf-based secure communication protocol for iot," *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 3, pp. 67:1–67:25, 2017.

[12] U. Chatterjee, V. Govindan, R. Sadhukhan, D. Mukhopadhyay, R. S. Chakraborty, D. Mahata, and M. M. Prabhu, "Building PUF based authentication and key exchange protocol for iot without explicit crps in verifier database," *IEEE Trans. Dependable Secur. Comput.*, vol. 16, no. 3, pp. 424–437, 2019.

[13] U. Chatterjee, D. Mukhopadhyay, and R. S. Chakraborty, "3paa: A private PUF protocol for anonymous authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 756–769, 2021.

Zephyrus: An information hiding mechanism leveraging Ethereum data fields

Mar Gimenez-Aguilar
Computer Security Lab, Universidad Carlos III de Madrid
Av. Universidad, 30, ES-28911 Leganes (Spain)
mgimenez@inf.uc3m.es

Jose M. de Fuentes
Computer Security Lab, Universidad Carlos III de Madrid
Av. Universidad, 30, ES-28911 Leganes (Spain)
jfuentes@inf.uc3m.es

Lorena González-Manzano
Computer Security Lab, Universidad Carlos III de Madrid
Av. Universidad, 30, ES-28911 Leganes (Spain)
lgmanzan@inf.uc3m.es

Carmen Camara
Computer Security Lab, Universidad Carlos III de Madrid
Av. Universidad, 30, ES-28911 Leganes (Spain)
macamara@pa.uc3m.es

***Abstract*—Permanent availability makes blockchain technologies a suitable alternative for building a covert channel. Previous works have analysed its feasibility in a particular blockchain technology called Bitcoin. However, Ethereum cryptocurrency is gaining momentum as a means to build distributed apps. The novelty of this paper relies on the use of Ethereum to establish a covert channel considering all transaction fields and smart contracts. No previous work has explored this issue. Thus, a mechanism called $Zephyrus$, an information hiding mechanism based on steganography, is developed. Moreover, its capacity, cost and stealthiness are assessed both theoretically, and empirically through a prototype implementation that is publicly released. Disregarding the time taken to send the transaction to the blockchain, its retrieval and the mining time, experimental results show that, in the best case, 40 Kbits can be embedded in 0.57 s. for US$ 1.64, and retrieved in 2.8 s.**

***Index Terms*—Ethereum, Information hiding, Steganography, Blockchain**

Tipo de contribución: *Investigación ya publicada*

I. Introduction

Steganography is the art of concealing messages within a non-secret piece of data called cover [1].

The extensive adoption of cryptocurrencies make them interesting to build covert channels over a publicly available medium.

Motivating use cases. There are three situations in which such a covert communication is interesting. On one hand, in the *panic button case* a threatened individual is willing to leave some secret material to be released in case of emergency and thus, without immediacy in mind. On the other hand, in a *sabotage case* a malicious insider aims to immediately exfiltrate sensitive data without being detected. Also looking for this feature, in a *censorship case* an individual is willing to share information in a controlled and censored environment.

In this paper the following contributions are achieved:

- Development of a steganographic system in Ethereum, called $Zephyrus$. It considers all Ethereum transaction and smart contracts fields to hide a secret. The design of $Zephyrus$ leverages a large amount of real-world Ethereum blockchain data to ensure the stealthiness of the secret. Indeed, the mechanism is assessed in terms of capacity, stealthiness and cost.
- An open-source proof of concept is released to foster further research. Moreover, it is used to assess the time taken for embedding and revealing a secret in a real-world Ethereum network.

The structure of the paper is the following. The proposed mechanism is introduced in Section II, whereas Section III focuses on its evaluation. Finally, Section IV concludes the paper and points out future research directions.

II. Proposed mechanism

A. Embedding strategies

On the one hand, crypto-related fields (such as *Receiver addresses*) and those without patterns (bytes32 and address type) have been shown to have high entropies. Therefore, these fields are used in full for embedding purposes (strategy S1).

On the other hand, strategy S2 is applied over those fields which count on acceptable variability, but in which a subset of $numval$ values are prominently common (such as *Gas* and *Gas Price*) . This leads to a capacity given by Equation 1.

$$Capacity_{S2}(bits) = \lfloor \log_2 numval \rfloor \quad (1)$$

Strategy S3 is applied in fields with acceptable variability and exhibiting some patterns in their values (e.g *Value* or uint data type). Therefore, for a value of total length l, the capacity of this strategy is given by Equation 2.

$$Capacity_{S3}(bits) = \lfloor \log_2(81 \times 10^{l-z-2}) \rfloor \quad (2)$$

Last but not least, a bytecode-specific strategy S4 is also proposed. As opposed to the previous ones, S4 does not consider the values of the data fields, but the set of instructions contained in the bytecode. Therefore, it provides with variable capacity.

B. Embedding procedure

The embedding process starts by preparing the secret to make it suitable for Ethereum transactions. Afterwards, data is hidden in fields according to their size and type.

1) Secret preparation: A key generation function is used to generate keys for the encryption processes. Firstly, the secret is symmetrically encrypted. Secondly, control data is also encrypted but with a stream cipher to keep the resulting size at a minimum. Finally, the secret is split if it exceeds the capacity of the transaction fields at stake.

2) Data hiding: For the sake of clarity, the description of the hiding process is divided into three main blocks, namely addresses, transaction information and smart contract data.

a) In addresses: The three types of addresses (namely *Sender, Receiver* and *Contract* ones) can be modified in all cases, thus S1 strategy is applied.

b) In transaction information: Capacity and effort to do the embedding varies greatly among fields.

The *Value* field can be used considering its underlying patterns (strategy S3).

By contrast, the most prominent *Gas limit* and *Gas price* values (strategy S2) are considered for representing the secret. In this case, their use is bounded by LBB, and also by $LBGL$ in *Gas limit*.LBB means that the sender's balance should be bigger than the cost of sending the transaction. By contrast, LBGL refers to the maximum block gas limit.

c) In smart contracts: Depending on the field, a different embedding strategy is used, specially when bytecode is at stake. *Swarm hash* field can be used in full (strategy S1) and with no limitations.In the case of*Function arguments* appear within function calls on the smart-contracts or in a *Contract constructor*. In practice, the capacity is limited by LBB, $LBGL$ and the technical limit for each argument type (called ArL). Respecting bytecode, on the other hand, two alternatives can be chosen – including instructions to represent the secret in an unreachable part of the code (called *Non-executable bytecode*) or in a reachable one (called *Executable bytecode*).

C. Revealing mechanism

This process is analogous to the embedding one but in reverse order. Firstly, hidden data is extracted considering the field at stake. Secondly, control information is decrypted to delimit the message appropriately. Finally, the decryption is enforced.

III. Evaluation

The evaluation of the proposed mechanism is performed from a theoretical and a practical point of view. The compliance of established goals is analysed (Section III-A).

A proof of concept has been implemented to measure the time taken for the proposed mechanism, as well as its associated costs, considering a previous analysis of 16,942,215 transactions and 65,346 contracts.

A. Goals compliance

1) Stealthiness: Since the secret has been tailored to be disguised as normal values for each field, almost all fields pass unnoticed to attackers, as there are no hints they might leverage on.

2) Simplicity: The proposed mechanism achieves simplicity as long as there is no special requirement to embed secret information in any of the fields. However, the computational effort varies among fields. Most of them, only require one operation to hide information.

3) Efficiency: Efficiency in terms of the amount of sent information is studied herein. For this purpose, the size of the secret has to be higher than the data to be privately shared with the receiver beforehand – otherwise, the mechanism would not be needed.

Efficiency of the amount of sent information, called Information Efficiency (IE), depends on the secret size $\|S\|$.

The system is efficient as long as IE > 1 (see Table I).

TABLE I
MAXIMUM SECRET SIZE, COST AND IE PER FIELD IN OUR EXPERIMENTS

Field	Max secret size (bits)	Additional cost (ether)	Cost \ Fee (ether)	Cost \ Fee (USD)	IE
Receiver address	40,760	-	0.005355	$ 1.64	105.05
Swarm hash	65,240	-	0.2631	$ 80.44	168.14
Gas Price	1,000	0.09	-	$ 27.51	2.58
Value	5,560	8.3137	-	$ 2,542	14.33
Gas limit	736	-	0.2575	$ 78.73	1.90
Function arguments	43,824	-	0.01073	$ 3.28	112.95
Constructor arguments	122,240	-	0.4490	$ 137.28	315.05
Non-executable bytecode	4,496	-	0.2215	$ 67.72	11.59

4) Cost: Embedding information in each of the fields has an associated cost. It is related to the fees required for sending information to Ethereum's blockchain.

5) Secret integrity: The immutability property of Ethereum ensures that the secret embedded in most fields can always be recovered.

6) Practical results: The most efficient field, regarding stealthiness and cost is to embed a message in *Function arguments* allowing up to 43,824 bits for $ 3.28. However, inserting data in the *Receiver address* also provides great results. The most expensive one, *Value*, allows 5,560 bits for around $ 2,542, as real Ether is transferred.

IV. Conclusions and future work

Ethereum permanent availability is an appealling feature to build covert communications on top of it. We consider all of its fields to develop a mechanism and a open source tool to hide information in it. Our results, published in [2], shows that some information can be concealed in most transaction fields while remaining stealthy in real-world conditions.

Future work will go towards the identification of the optimal fields to embed information considering time, cost and stealthiness; multi-field support, interactive channels and adaptive steganographic techniques.

Acknowledgment

This work has been partially supported by grants CAVTIONS-CM-UC3M and DEPROFAKE-CM-UC3M funded by UC3M and the Gov. of Madrid (CAM); by CAM by grant P2018/TCS-4566-CM, co-funded with ERDF; and by Min. of Science and Innovation of Spain by grant PID2019-111429RB-C21.

References

[1] S. Katzenbeisser and F. Petitcolas, *Information hiding techniques for steganography and digital watermarking*. Artech house, 2000.

[2] M. Gimenez-Aguilar, J. M. De Fuentes, L. González-Manzano, and C. Camara, "Zephyrus: An information hiding mechanism leveraging ethereum data fields," *IEEE Access*, vol. 9, pp. 118 553–118 570, 2021.

Sesión Poster 3

Un nuevo enfoque DevSecOps al modelo Viewnext-UEx

Javier Alonso Díaz
Cátedra ViewNext-UEx
Escuela Politécnica
javieralonso@unex.es

[1]José Carlos Sancho Núñez
[2]Óscar Mogollón Gutiérrez
Universidad de Extremadura
Escuela Politécnica
[1,2]{jcsanchon, oscarmg}@unex.es

Mohammad Hossein Homaei
Universidad de Extremadura
Escuela Politécnica
mhomaein@alumnos.unex.es

Resumen- **El incremento en la demanda de aplicaciones ha generado que las organizaciones aceleren el desarrollo software. Entre las diversas metodologías creación de software se identifica que los modelos DevOps son los marcos de trabajo más óptimos para las organizaciones, al integrar las operaciones en el desarrollo. Independientemente, de la metodología aplicada es importante mantener el enfoque de seguridad. Por ello, en esta contribución se analizan diferentes modelos que incorporan actividades de seguridad en el ciclo de vida de desarrollo de software y, en base a ellos, se propone un nuevo modelo con enfoque DevSecOps. Esta propuesta transforma el modelo Viewnext-UEx específico de seguridad en un marco de trabajo que incluye los procesos de negocio y operaciones. El nuevo modelo presenta un conjunto de actividades categorizadas en función de las dimensiones y el grado de madurez de su cumplimiento para la implementación en las empresas.**

***Index Terms*- Software Seguro, Ingeniería del Software, DevSecOps.**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

La demanda de software y la emergente externalización de los servicios software a la nube ha generado que las organizaciones necesiten acelerar el desarrollo de software.

Los equipos desarrolladores hacen uso de las metodologías de desarrollo de software tradicionales, ágiles y DevOps para mejorar la productividad de los procesos de creación de software. En los modelos de software tradicionales (cascada, espiral, etc.), los equipos de desarrollo trabajaban de forma secuencial y aislada, de manera que antes de comenzar una fase es necesario que la anterior haya finalizado. Al ser realizadas por distintos equipos de trabajo provoca que, por lo general, se produzca una falta de coordinación en las transiciones de una fase a otra. Con la aparición de los modelos ágiles (SCRUM, XP, etc.) se mejora considerablemente en rapidez y eficacia, ya que el proceso de creación de software se trabaja de manera iterativa e incremental (*sprints*), obteniendo una retroalimentación común a todos los implicados en el proyecto, lo que permite abordar cambios y ajustes de una manera óptima. Sin embargo, la falta de procesos en estas metodologías genera que los equipos se desvíen y que la colaboración entre las fases sea compleja de mantener en el tiempo. Para solventar estas cuestiones, se propone la metodología DevOps (desarrollo + operaciones) que avanza en una integración entre los perfiles de desarrollo y de sistemas, permitiendo automatizar operaciones habituales como el *testing*, el despliegue y los reportes al cliente, lo que hace que se acelere la productividad de los desarrollos.

Con la optimización en los procesos dedicados a la creación de software resuelta, aparece la imponente necesidad de abordar que el software generado también sea seguro. El número de ataques gestionados por el Centro Criptológico Nacional (CCN-CERT) [1]–[5] en los últimos años 2016 (20.940), 2017 (26.500), 2018 (38.029), 2019 (42.997) y 2020 (82.532) ha generado una alta preocupación en la empresas, que necesitan ideas para abordar este problema.

En este sentido, una de las posibles soluciones es la filosofía DevSecOps que integra las prácticas de seguridad en un modelo DevOps. La metodología DevSecOps implica desarrollo, seguridad y operaciones. Estos modelos, permiten utilizar de manera óptima la agilidad y la rapidez de reacción que ofrece el enfoque DevOps, integrando los mecanismos de seguridad desde el inicio del desarrollo. La diferencia fundamental entre el sistema DevSecOps y los enfoques convencionales, es que el equipo de seguridad aplica las medidas correspondientes una vez el producto ha finalizado. DevSecOps incorpora y hace cumplir controles de seguridad significativos sin ralentizar la velocidad de despliegue.

En la línea de asegurar los procesos de desarrollo de software de manera segura, se ha trabajado en proponer un marco de trabajo seguro denominado ViewNext-UEx [6] que muestra resultados prometedores en la aplicación de una metodología preventiva, con respecto al desarrollo reactivo. Sin embargo, en esa propuesta no se integran las actividades de operación en el desarrollo. En este trabajo se presentan los avances realizados para evolucionar modelos SSDLC a modelos DevSecOps. Para ello, se propone un nuevo modelo DevSecOps con 143 actividades de seguridad, categorizadas en función 6 grados de madurez.

II. Estado del Arte

Realizado un análisis de metodologías específicas que integran la seguridad por defecto [7] se identifican una serie de modelos esenciales:

- *Software Assurance Maturity Model* (SAMM) [8].
- *Building Security In Maturity Model* (BSIMM) [9].
- *Microsoft Security Development Lifecycle* (MSDL) [10].
- *Comprehensive, Lightweight Application Security Process* (CLASP) [11].
- Modelo de Desarrollo de Software Seguro Viewnext-UEx [6].
- *OWASP DevSecOps Maturity Model* (DSOMM) [12].

La Tabla 1 indica para cada modelo la compañía, el número de actividades, el enfoque (madurez o prescriptivo) y si aplica un enfoque DevOps.

Tabla I
MODELOS Y CARACTERÍSTICAS

Modelo	Compañía	Nº Actividades	Tipo modelo	DevOps
SAMM [8]	OWASP	15	Madurez	✓
BSIMM [9]	BSIMM	12	Prescriptivo	X
MSDL [10]	Microsoft	17	Madurez	X
CLASP [11]	OWASP	24	Prescriptivo	X
Viewnext-UEx [6]	Viewnext	14	Madurez	X
DSOMM [12]	OWASP	137	Madurez	✓

Los modelos descritos anteriormente se centran en el desarrollo de software siguiendo un ciclo de vida con seguridad por defecto.

Así, Muhammad y otros identifican en [13] los retos más prioritarios del paradigma DevSecOps en proyectos de software. Estos retos se agrupan en la falta de estándares de codificación segura, la falta de herramientas de pruebas automatizadas para la seguridad y la falta de conocimiento en las pruebas estáticas para la seguridad son los retos más prioritarios para el paradigma DevSecOps.

Por tanto, se evidencia que la aplicación del enfoque DevOps supone un cambio de paradigma, en el que se considera responsables de la seguridad a todos los implicados en el desarrollo, incluidas las operaciones. Para facilitar la integración, Nisha y Khandebharad proponen en [14] un marco completo de migración de DevOps a DevSecOps.

Y de manera más específica Kumar y Goyal, presentan en [15] un modelo conceptual con filosofía DevSecOps compuesto por herramientas basado software libre en la nube..

En esta línea, el modelo de madurez OWASP DevSecOps (DSOMM) [12], creado por *The Open Web Application Security Project®,* implementa un pipeline seguro, que fortalece la estrategia DevOps priorizando actividades a implementar, utilizando mejores prácticas e introduciendo herramientas útiles en esta materia. Sin embargo, aunque en la teoría este modelo puede resultar una buena opción, para una compañía de desarrollo de software su implementación resulta ciertamente costosa. Principalmente, se proponen una gran cantidad de actividades por nivel. Por lo que en este trabajo se realiza un estudio que propone particularizar este modelo hacia versiones más sencillas, rediseñando actividades y niveles y proponiendo diferentes modelos que permitan la implantación en cualquier proyecto de desarrollo de software, de manera gradual y ordenada.

III. METODOLOGÍA

El modelo de madurez Viewnext-UEx DevSecOps propuesto en este trabajo, al igual que el modelo de OWASP, está formado por un conjunto de dimensiones (niveles), subdimensiones y actividades a implementar. El modelo propuesto permite categorizar el nivel de un proyecto o de una organización en función de las actividades involucradas en la realización de ese proyecto. Esto es, el grado de madurez Viewnext-UEx DevSecOps se determina según las actividades que se implementan en el desarrollo de software. Esto supone que la mejora en el grado de madurez está condicionada por la incorporación de nuevas actividades. Sin embargo, la inclusión de una nueva actividad supone un incremento en los costes para la organización, económicos, de recursos, de tiempo y de conocimiento necesario para llevar a cabo la nueva actividad. A cambio de estos costes, la implementación de nuevas actividades proporciona un beneficio o utilidad.

Teniendo en cuenta estos factores, el modelo Viewnext-UEx DevSecOps se categorizan en dimensiones (grados de madurez) y subdimensiones y se ordenan un total de 143 actividades, que permiten a una organización mejorar significativamente y de manera escalonada el desarrollo de software. Para ello, se realiza un complejo estudio para determinar la dependencia de actividades, ya que existen actividades que dependen de otras. Las actividades también se clasifican en función del valor que aportan, el tiempo, los conocimientos y el coste necesario para su implementación, así como las dependencias respecto a otras actividades.

A continuación, se presentan agrupadas las dimensiones y subdimensiones del modelo de madurez Viewnext-UEx DevSecOps.

Construcción y Despliegue (1)
- Construcción (1.1)
- Despliegue (1.2)
- Gestión de parches (1.3)

Cultura y Organización (2)
- Diseño (2.1)
- Educación y Orientación (2.2)
- Proceso (2.3)

Implementación (3)
- Fortalecimiento de la aplicación (3.1)
- Desarrollo y Control del Código Fuente (3.2)
- Fortalecimiento de la infraestructura (3.3)
- Control de versiones (3.4)

Recopilación de información (4)
- Recopilación de log (4.1)
- Vigilancia (4.2)

Pruebas y Verificación (5)
- Prueba de Aplicación (5.1)
- Consolidación (5.2)
- Profundidad dinámica para aplicaciones (5.3)
- Profundidad dinámica para infraestructura (5.4)
- Profundidad estática para aplicaciones (5.5)
- Profundidad estática para infraestructura (5.6)

IV. RESULTADOS

El estudio del modelo DevSecOps OWASP permite considerar en la propuesta Viewnext- UEx DevSecOps seis grados de madurez como son: iniciación, base, intermedio, avanzado, experto y completo (que sería la totalidad del modelo DevSecOps OWASP).

A. Grado de madurez 1. Iniciación.

En el grado de madurez de iniciación se implementan 26 actividades, las cuales corresponden con el 21.75% de implementación del modelo completo.

Este nivel, se centra en el desarrollo de actividades fundamentales. Entre ellas destaca un software de integración y entrega continua, actividades de la fase de diseño y monitorización, un registro de aplicaciones centralizado, la realización de modelos simples de amenazas, ejecución de pruebas en busca de secretos en el código y escaneos periódicos en busca de vulnerabilidades.

B. Grado de madurez 2. Base.

Este grado de madurez base, incorpora las actividades definidas en el nivel anterior y las definidas por Sancho y otros en [6]. En total, se implementan un total de 23 actividades, que corresponde a un 35.96% de implementación completa del modelo. En este nivel se centra en la construcción, diseño y se observa la importancia de la realización de pruebas unitarias de manera habitual, la monitorización de los eventos, en minimizar la superficie de ataque, establecer un tiempo de vida para las imágenes y establecer el principio de seguridad por defecto.

C. Grado de madurez 3. Intermedio.

Este grado de madurez, propone 39 actividades, que corresponden con un 69.03% de implementación respecto al modelo completo. En este grado de madurez, se da importancia a la formación en ámbito de la ciberseguridad al personal involucrado en el desarrollo del software. Las actividades se focalizan en la realización y verificación de pruebas en los entornos virtualizados, los componentes en la nube, la red y tratamiento de defectos.

D. Grado de madurez 4. Avanzado.

Este grado de madurez, se forma con 18 actividades, que han permitido a las organizaciones controlar el análisis de vulnerabilidades estático y dinámico, así como la ejecución de pruebas sobre las aplicaciones y controlan en gran medida la detección de vulnerabilidades en sus infraestructuras. En este grado de madurez se define un plan de continuidad empresarial y plan de emergencia, conocido por sus siglas en inglés BCDR (*Business Continuity and Disaster Recovery*). Una vez implementado se cubre un total del 80.18% de implementación del modelo completo.

E. Grado de madurez 5. Experto.

En el grado experto se aplican 10 actividades dedicadas a la definición de procesos para realizar la entrega de una nueva versión de software de manera gradual. En cuanto a la dimensión de Construcción, las imágenes deben construirse o diariamente o en el tiempo de ejecución. Para evitar problemas de integración se debe realizar la entrega como *Blue/Green* que consiste en tener dos versiones de la aplicación en producción, con el objetivo de observar cómo afecta a los usuarios la nueva modificación. Implementado este grado de madurez se completa un 86.72% del modelo final.

F. Grado de madurez 6. Completo.

Las 27 actividades que completan el modelo se centran en reforzar las aplicaciones. Se comprueba de manera aleatoria instancias en producción con el fin de asegurar que los desarrolladores implementan los servicios para resistir a fallas de las instancias. Además, para reforzar la comprensión en seguridad, todo el equipo debe participar en un evento de hackeo. En este tipo de eventos se presenta una aplicación vulnerable y el equipo debe analizar las posibles vulnerabilidades.

G. Niveles de cumplimiento de la propuesta.

La Tabla 2 recoge el porcentaje de cumplimiento del modelo DevSecOps definido por OWASP respecto a este modelo (Viewnext-UEx DevSecOps) para cada una de las subdimensiones definidas anteriormente en los diferentes grados de madurez que componen el modelo. Por otro lado, en la Fig. 1, podemos encontrar el nivel de cumplimiento para cada uno de los grados de madurez respecto al modelo original OWASP DevSecOps.

Tabla II
CUMPLIMIENTO DEL MODELO FRENTE A OWASP MATURITY MODEL

Dim.	Sub.	Ini.	Base	Int.	Avan.	Exp.	Compl
1	1.1	33,33	33,33	66,67	83,33	100,00	100
	1.2	11,11	11,11	55,56	66,67	77,78	100
	1.3	0,00	33,33	66,67	66,67	100,00	100
2	2.1	66,67	66,67	66,67	83,33	83,33	100
	2.2	0,00	20,00	46,67	66,67	73,33	100
	2.3	0,00	0,00	50,00	75,00	75,00	100
3	3.1	0,00	0,00	0,00	0,00	0,00	100
	3.2	5,26	21,05	68,42	84,21	89,47	100
4	4.1	50,00	50,00	66,67	66,67	83,33	100
	4.2	61,54	69,23	76,92	76,92	76,92	100
5	5.1	0,00	50,00	50,00	100,00	100,00	100
	5.2	0,00	10,00	60,00	60,00	70,00	100
	5.3	11,11	33,33	66,67	100,00	100,00	100
	5.4	0,00	33,33	66,67	83,33	83,33	100
	5.5	44,44	100,00	100,00	100,00	100,00	100
	5.6	16,67	33,33	66,67	75,00	83,33	100
	5.7	0,00	0,00	0,00	20,00	40,00	100

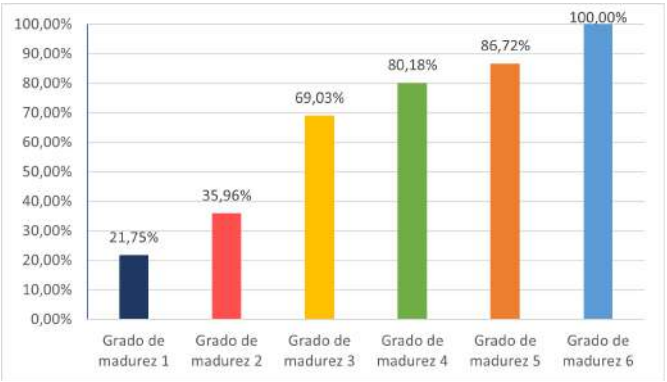


Fig. 1 Grado de implementación del modelo Viewnext-UEx DevSecOps respecto al modelo OWASP DevSecOps.

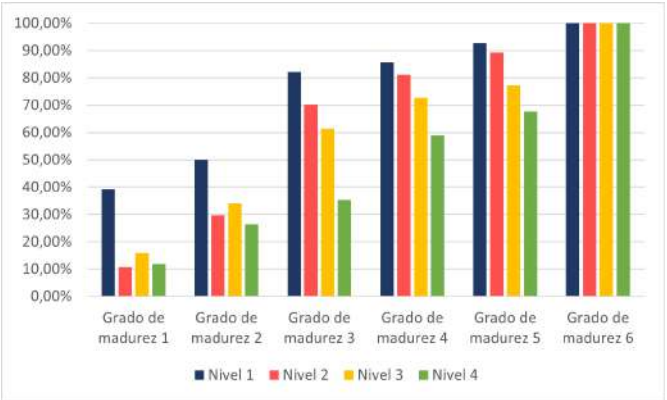


Fig. 2 Cumplimiento de los niveles OWASP DevSecOps en cada grado de madurez.

La Fig. 2, muestra el cumplimiento del modelo original (DevSecOps de OWASP) frente al nuevo modelo Viewnext-UEx DevSecOps. Se produce un rediseño del modelo en cada grado de madurez y se implementan actividades de cada nivel.

H. Análisis del estado de la madurez de los proyectos.

Definido el modelo Viewnext-UEx DevSecOps es precisa la implantación e implementación por las diferentes organizaciones. Sin embargo, cada organización parte desde un estado inicial totalmente distinto al resto, e incluso, los objetivos entre las diferentes organizaciones van a ser son diferentes. Por ello, primeramente, se debe realizar un cuestionario basado en un conjunto de preguntas agrupadas en función de las dimensiones que componen el modelo. En base a los resultados, se determina en qué punto exacto del modelo se encuentra cada organización y se indican las actividades que primeramente se deben implementar para establecer un grado de madurez concreto. Cuando la organización haya completado con éxito un grado de madurez del modelo, en función de los objetivos de esta, se analiza un conjunto de actividades ordenadas en función de los objetivos que la organización debe implementar para llegar al siguiente grado de madurez.

V. Conclusiones

Evidenciada la necesidad añadir la cultura DevOps a los procesos de desarrollo del software para mejorar la productividad y tener que complementarla con el enfoque de seguridad, se propone el modelo Viewnext-UEx DevSecOps compuesto por 143 actividades. Este modelo incorpora la seguridad para garantizar la confidencialidad, integridad y disponibilidad de las aplicaciones, a la vez que tienen en cuenta las operaciones y diferentes niveles de madurez en su cumplimiento para facilitar el uso y la implementación en cualquier empresa generadora de software.

Agradecimientos

Los autores agradecen la financiación recibida por parte de la Junta de Extremadura (Fondo Europeo de Desarrollo Regional), Consejería de Economía, Ciencia y Agenda Digital, bajo el proyecto GR21099 y a ViewNext, empresa de Servicios de Tecnologías del grupo IBM España.

Referencias

[1] Centro Criptológico Nacional, “Ciberamenazas y Tendencias Edición 2017,” 2017. [Online]. Available: https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/2221-ccn-cert-ia-16-17-ciberamenzas-y-tendencias-edicion-2017-resumen-ejecutivo-1/file.html.

[2] Centro Criptológico Nacional, “Informe de Ciberamenazas y Tendencias 2018,” 2018. [Online]. Available: https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/2835-ccn-cert-ia-09-18-ciberamenzas-y-tendencias-edicion-2018-1/file.html.

[3] Centro Criptológico Nacional, “Informe de Ciberamenazas y Tendencias 2019,” 2019. [Online]. Available: https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/3776-ccn-cert-ia-13-19-ciberamenazas-y-tendencias-edicion-2019-1/file.html.

[4] Centro Criptológico Nacional, “Ciberamenazas y Tendencias Edición 2020,” 2020. [Online]. Available: https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/5377-ccn-cert-ia-13-20-ciberamenazas-y-tendencias-edicion-2020/file.html.

[5] Centro Criptológico Nacional, “Ciberamenazas y Tendencias Edición 2021,” 2021.

[6] J. Carlos, S. Núñez, C. Lindo, and P. G. Rodríguez, “SPECIAL SECTION ON EMERGING APPROACHES TO CYBER SECURITY A Preventive Secure Software Development Model for a Software Factory: A Case Study,” doi: 10.1109/ACCESS.2020.2989113.

[7] J. C. Sancho Núñez, A. Caro Lindo, and P. García Rodríguez, “Análisis de metodologías de Desarrollo de Software Seguro,” in Jornadas Nacionales de investigación en Ciberseguridad (JNIC), 2016, pp. 42–47.

[8] “SAMM. The Model.” .

[9] “Software Security Assessment Report | BSIMM.” .

[10] “The Security Development LifeCycle - TechNet Articles - United States (English) - TechNet Wiki.” .

[11] V. Figueroa, U. Blas, P. A. Córdoba, I. Superior, J. X. Bahía, and B. A. Baires, “OWASP CLASP.” .

[12] “OWASP DevSecOps Maturity Model - Activities Overview.” .

[13] M. A. Akbar, K. Smolander, S. Mahmood, and A. Alsanad, “Toward successful DevSecOps in software development organizations: A decision-making framework,” Inf. Softw. Technol., vol. 147, no. October 2021, p. 106894, 2022, doi: 10.1016/j.infsof.2022.106894.

[14] T. N. Nisha and A. Khandebharad, “Migration from DevOps to DevSecOps: A complete migration framework, challenges, and evaluation,” Int. J. Cloud Appl. Comput., vol. 12, no. 1, 2022, doi: 10.4018/IJCAC.2022010102.

[15] R. Kumar and R. Goyal, “Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC),” Comput. Secur., vol. 97, p. 101967, 2020, doi: 10.1016/j.cose.2020.101967.

Explorando el tráfico IBR mediante una darknet basada en conexiones domésticas

Rodolfo García-Peñas
Universidad Internacional de La Rioja
Avda. de La Paz 137, 26006
Logroño, La Rioja, España
kix@kix.es
rodolfo.garcia@unir.net

Rafael A. Rodríguez-Gómez
Universidad de Granada
Dpt. de Teoría de la Señal
Telemática y Comunicaciones,
CITIC-UGR, España
rodgom@ugr.es

Carlos Enrique Montenegro-Marin
Universidad Distrital Francisco José de Caldas /
Fundación Universitaria Internacional de La Rioja.
Carrera 7 40B-58. Bogotá, Colombia
cemontenegrom@udistrital.edu.co
carlos.montenegro@unir.net

***Resumen*—El tráfico de fondo de Internet (*Internet Background Radiation*, IBR) se caracteriza por ser tráfico no productivo, enviado incluso a direcciones que no existen, servidores que no responden o servidores que no esperan recibir tráfico [1]. Dentro de este tráfico se esconden escaneos de nodos y puertos, búsqueda de vulnerabilidades, expansión de virus o la adquisición de nodos *zombie* para *botnets*. La identificación y el análisis de este tráfico se ha realizado habitualmente mediante *darknets* y telescopios de red (*Network Telescopes*), basados en redes que no están en uso en Internet. En el presente artículo se analiza la viabilidad de adquisición y análisis de tráfico IBR utilizando para ello las conexiones domésticas que tienen los clientes de los proveedores de acceso a Internet (ISPs) y equipamiento de bajo coste. Por último, se validan los resultados obtenidos con estudios previos basados en *darknets* tradicionales.**

***Index Terms*—IBR, Darknet, Network Telescope, Internet Security**

Tipo de contribución: *Investigación original*

I. Introducción

El aumento del número de ataques a través de Internet aumenta constantemente, así como la complejidad de los mismos. Detrás del tráfico de las comunicaciones habituales que utilizan los usuarios para visitar páginas web, leer el correo o realizar compras, existe un tráfico de fondo que trata de realizar acciones completamente distintas, buscando nodos para añadir a una *botnet*, analizando los servicios de un sistema, explotando vulnerabilidades, etc. Desde hace muchos años se han utilizado las mismas herramientas para poder descubrir este tráfico de fondo, identificar los tipos de ataque que van apareciendo, así como los comportamientos de los atacantes. Sin embargo, Internet ha cambiado mucho en los últimos años, con crecimiento en el número de personas conectadas, el agotamiento de las direcciones IPv4, la aparición de nuevas tecnologías como IoT, etc.

Mediante el presente artículo se ofrece una nueva alternativa para la adquisición y el análisis del tráfico de fondo de Internet, con ventajas sobre las herramientas actualmente utilizadas, como son un menor coste, una mejor ocultación a su detección, siendo más dinámica, sencilla y más distribuida.

En los siguientes apartados se detalla que es el tráfico de fondo de Internet, se hace un análisis del mismo, así como una visión cronológica de los estudios más relevantes sobre este tipo de tráfico, mostrando su evolución a lo largo del tiempo. En el siguiente apartado se describen las herramientas habituales para la adquisición del tráfico. Posteriormente se muestra la propuesta realizada, describiendo su arquitectura física y lógica. Mediante la propuesta se ha obtenido una situación actualizada del tráfico de fondo, que en el apartado de análisis y resultados se valida con los estudios más relevantes indicados anteriormente. Finalmente, se detallan las conclusiones del estudio.

II. Internet Background Radiation

El tráfico de fondo de Internet (*Internet Background Radiation*, IBR[1]) es aquel tráfico que tiene como destino direcciones IP o puertos donde no hay un dispositivo o en caso de existir, no espera esa conexión. El tráfico IBR es parte del tráfico de Internet y está directamente relacionado con gran multitud de factores, como son como son el número de equipos conectados a Internet, la seguridad de los sistemas, las vulnerabilidades, la evolución de los tipos de ataque, la evolución de las herramientas y un largo etcétera. La siguiente visión cronológica muestra el impacto de estos factores en el tráfico IBR.

Unos de los primeros análisis realizados sobre el tráfico IBR fue realizado en 2004 por Pang [1], lo describía y mostraba las características del mismo, indicando que podía ser de tipo malicioso (escáneres de vulnerabilidades, gusanos, etc.) o benigno (errores de configuración, por ejemplo, indicando una máquina equivocada en una comunicación). En este análisis se realizaba una clasificación de los ataques por el protocolo, la aplicación e incluso por el tipo de *exploit* específico que se estaba utilizando (cuando era pertinente). En esa época los ataques se orientaban principalmente a servicios mal configurados o bien con vulnerabilidades, especialmente aquellos que permitía obtener acceso al sistema, como serían Telnet y SSH.

En el año 2006, con aproximadamente un mil millones de direcciones IP asignadas, mostradas como las clases A asignadas a cada RIR (*Regional Internet Registry*, Registro Regional de Internet) en la *Fig. 1*, se estimó [2] que el número de nodos conectados era aproximadamente de 187 millones. Internet llevaba ya muchos años en funcionamiento antes del artículo de Pang [1], así en el año 2007 Allman [3] escribió un artículo donde mostraba el análisis de 12,5 años de registros de red obtenidos en Lawrence Berkeley National Laboratory (LBNL) en Berkeley, CA, USA. El estudio realiza

[1]También conocido como *Internet Background Noise* (IBN)

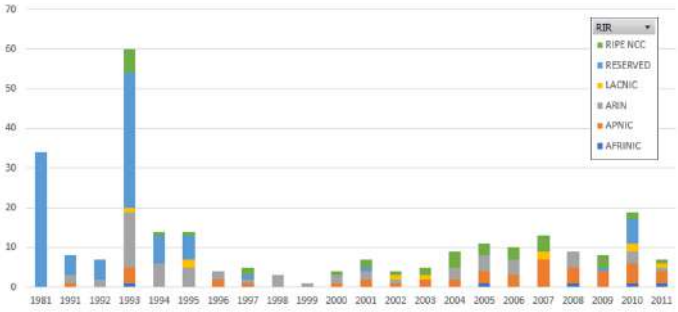


Figura 1. Bloques de clase A por RIR 1981 - 2011

una interesante separación entre qué son escaneos y qué no lo son, identificando los escaneos con aquellas conexiones que se realizan desde un único host o equipo. Por otro lado, hace un análisis de los puertos más escaneados a lo largo del tiempo, siendo los principales HTTP, SMB, NetBIOS, RPC, SQL, FTP, SSH, Telnet, con menos intensidad SMTP y DNS y únicamente a durante un periodo que comienza en 2004 con mucha intensidad y posteriormente decrece hasta final del estudio, el puerto 9898/TCP utilizado por la puerta trasera (*backdoor*) Sasser [4].

En ese periodo de tiempo ocurrieron eventos, que, si actualmente pueden ser habituales, en aquel momento fueron algo novedoso. En octubre del año 2008 apareció el gusano Conficker [5] , que fue un caso conocido porque aproximadamente el 6 % de los computadores de Internet de la época se infectaron por este gusano. En su funcionamiento realizaba técnicas de escaneo para poder infectar a otros equipos, dado que se intentaba conectar al servicio *Server* de *Windows* y explotar una vulnerabilidad en el mismo, generando para ello tráfico IBR.

En el año 2010 se realizó un informe sobre el estado de IBR, como continuación al análisis realizado previamente por Pang [1]. Este informe, llamado *Internet Background Radiation Revisited* [6] se presentan algunos conceptos y puntos de vista novedosos. El primer concepto está relacionado por el agotamiento de direcciones IPv4 que venía ocurriendo en esos años. El histórico de la asignación de las direcciones IPv4 que IANA realizó a los RIR se puede observar de forma visual en *Fig. 1*. En la misma se han agregado los datos de forma que aparezcan únicamente los cinco RIR y un conjunto de direcciones reservadas, si bien dentro de los bloques de reservadas se encuentran las primeras asignaciones que se hicieron a compañías, como Apple, HP, DoD (*Department of Defense*, Departamento de Defensa de EEUU), etc. El eje Y muestra el número de clases A asignadas a cada RIR. Se puede observar cómo hay un elevado conjunto de datos en los años 1981 a 1995 relacionados con estas asignaciones reservadas.

Si se centra el análisis en los últimos años, eliminando la parte de las direcciones reservadas, se observa un crecimiento elevado desde el año 2004 en asignaciones de direccionamiento, ocasionado principalmente por la expansión del número de usuarios de Internet. APNIC (RIR para Asia y Pacífico), que realiza la asignación a países con alto nivel de población, como Japón, China e India obtiene en los últimos siete bloques justo antes del agotamiento (aproximadamente 64 millones de direcciones / año). Este dato es importante porque desde

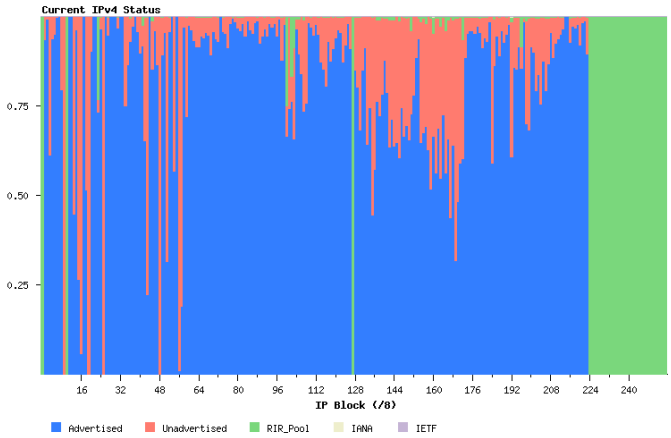


Figura 2. Uso actual del direccionamiento IPv4. Fuente potaroo.net

el artículo de Pang de 2004, en 7 años se habían asignado aproximadamente 79 clases A, añadiendo un volumen de aproximadamente el 36 % de las direcciones totales de Internet, permitiendo una superficie de ataque mucho mayor.

En el año 2012 llegó el agotamiento del direccionamiento IP, en el que los RIR no recibirían más direcciones de IANA y, dependiendo de su política, asignarían el direccionamiento restante mediante ciertas restricciones, principalmente en bloques de menor tamaño. Esto no significa que todo el direccionamiento estuviera en uso, dado que los LIR (*Local Internet Registry*, Registro Local de Internet) habrían recibido reservas de direcciones, para ponerlas en uso en un corto plazo de tiempo, pero sí que estaban en su mayoría asignadas. En la *Fig. 2* se puede observar que de todo el espacio de direccionamiento (Clases A de Internet en el eje de abscisas, uso de cada una de ellas en el eje de ordenadas), la mayoría está utilizado, pero queda un elevado porcentaje que no está anunciado (en color naranja), y, por lo tanto, no está en uso (color azul).

En el artículo de Wustrow [6] se incluye como novedad una referencia a las *botnets*, como el gusano Conficker indicado previamente. Si bien en ese momento seguía existiendo tráfico relacionado con escaneo y reconocimiento (algo que será continuo a lo largo del tiempo), aparecen estos nuevos tipos de ataque y con ellos tráfico de búsqueda de objetivos a infectar. En relación al uso del direccionamiento y las *botnets*, a principio de 2013 se publicó un análisis sobre el número de nodos de Internet [7], donde se mostraba que había unos 1.300 millones de dispositivos (de las aproximadamente 3.700 millones de direcciones posibles). Este análisis, que se realizó mediante una *botnet* compuesta por 420.000 dispositivos [8] y muestra el gran crecimiento del número de dispositivos conectados, un 700 % desde 2006.

Relacionado con el aumento de direcciones usadas y por lo tanto con la superficie de ataque, en el año 2013 aparecieron nuevas técnicas de escaneo masivo. Hasta ese momento era habitual la realización de escaneo de redes y de puertos utilizando herramientas como Nmap [9]. Sin embargo, Nmap no está diseñado para realizar un escaneo masivo. En este año se desarrollaron dos herramientas con este propósito. La primera es ZMap [10], fue presentada en USENIX y permitía

en ese momento realizar un escaneo completo de todo el espacio de direcciones de Internet en aproximadamente 45 minutos, utilizando para ello un ordenador y una conexión Gigabit Ethernet. Este estudio está realizado en la Universidad de Míchigan y en ese momento esta entidad podría disponer de esa conexión a Internet, pero no era lo habitual para el conjunto de los usuarios de Internet. ZMap indica en su página web que actualmente es capaz de realizar un análisis completo de Internet en 5 minutos con una conexión 10 Gigabit Ethernet, lo que requiere actualmente una conexión corporativa o de tipo universitario para ese volumen de tráfico. La segunda herramienta que apareció ese año 2013 fue MASSCAN [11], que tiene un funcionamiento similar y un ratio de 10 millones de paquetes por segundo para una conexión 10 Gigabit Ethernet, tardando también 5 minutos en analizar todo Internet.

A partir de 2014 el tráfico de IBR sigue un patrón mucho más continuo. El estudio "An Internet-Wide View of Internet-Wide Scanning" [12] realiza una nueva revisión del tráfico IBR, en la que aparecen nuevos patrones que se convierte en algo común. Según el artículo el escaneo de redes está ampliamente extendido en Internet, con amplio crecimiento del uso de las herramientas ZMap y Masscan, y el crecimiento de las *botnets*. En este momento el origen de los escaneos maliciosos no proviene principalmente de las *botnets*, como ocurría en los estudios de los años anteriores, sino de proveedores de hosting, entrando como un nuevo actor. Siguen produciéndose los ataques previos, como por ejemplo, ataques de fuerza bruta en conexiones SSH o Telnet, pero también aparecen ataques para las vulnerabilidades que van apareciendo, como eran en ese momento un fallo en los *routers* Linksys, OpenSSL Heartbleed o vulnerabilidades en los servicios de NTP. En procedimiento de los atacantes consiste en, al identificarse una nueva vulnerabilidad, se utilizan las herramientas de análisis masivo para recorrer todo el espacio de direcciones y encontrar los objetivos. Posteriormente se realiza el ataque con las herramientas correspondientes. Todo esto ocurre en las 24 horas siguiente a la aparición de la vulnerabilidad. Esta tendencia ha continuado durante los siguientes años, con el mismo modo de proceder y únicamente variando los tipos de vulnerabilidades. Durante 2018 y 2019 [13] hubo un fuerte impacto con botnets Mirai [14], orientadas a realizar ataques de denegación de servicio distribuido (DDoS). Durante los últimos años se han incrementado los ataques relacionados con la obtención de recompensas económicas, como *ramsonware*, venta de servicios a través de la red oscura (*Dark Web*) relacionados con ataques en Internet (obtención o robo de datos, pero también contratación de servicios de DDoS), ataques del CEO y en general ataques de ciberdelincuencia organizada. La mayoría de estos ataques no están relacionados con la tipología de ataque que se está analizando en este estudio, pero algunos de ellos pueden propagarse a través de Internet, generando y usando tráfico de tipo IBR.

III. Adquisición de tráfico IBR

En el apartado anterior se han observado diferentes tipos de tráfico IBR, desde escaneos masivos de nodos y puertos, hasta escaneos más específicos en busca de vulnerabilidades o puertas traseras. Para la adquisición de tráfico IBR es por tanto suficiente con estar conectado a Internet esperando la recepción de este tipo de paquetes. Muchos de los estudios anteriormente indicados han utilizado la información que obtenían a través de sus redes en uso, recolectando la información a través de cortafuegos o IDS [3] o más recientemente mediante redes que no están en uso y, por lo tanto, no emiten ningún tipo de tráfico, conocidas como *darknets*. Cuando estas redes *darknet* se agrupan, formando conjuntos de miles de equipos [12] suelen llamarse *network telescopes* (telescopios de red). En la *Tabla I* se pueden observar algunos de los usados en algunos de los artículos referenciados, siendo el de *UCSD Network Telescope*, conocido habitualmente como *CAIDA*, uno de los más utilizados en multitud de artículos de investigación y siendo de los más importantes.

Tabla I
Telescopios de red conocidos

Red	Direcciones	Nombre	Fecha
1/8	16M	APNIC	23/03/2010 - 30/03/2010
44/8	16M	UCSD N.T.	01/01/2001 - 04/06/2019
44/8	12M	UCSD N.T.	05/06/2019 - Actualidad
35/8	11M	Merit Network	05/10/2005 - Desconocido
50/8	16M	ARIN	12/03/2010 - 19/03/2010
107/8	16M	ARIN	25/03/2010 - 31/03/2010
Varias	1300 redes	Akamai	En 2009 y 2019
/16	65K	HEAnet	03/2019 (1 semana)
/15	131K	SURFNet	Desconocido

Otro de los telescopios de red con una gran relevancia es el de Akamai [15]. A diferencia de los otros telescopios se creó utilizando la propia infraestructura que Akamai utiliza para prestar servicio a sus clientes. Por lo tanto, no se utiliza una única red agregada, sino múltiples y dispersas por diferentes continentes y no es una red completamente oscura, dado que existen nodos activos en la misma. Este concepto se conoce como *greynet* [16].

La utilización de *greynets* es interesante porque, a diferencia de las *darknets* algunos nodos pueden interactuar con el atacante [16]. Cuando un atacante está analizando una *darknet* puede estar enviando paquetes TCP con el *flag* de SYN activado, intentando establecer una comunicación con un servicio. Al no existir respuesta, la *darknet* no podrá conocer el objeto de dicha comunicación. En una *greynet*, el destino puede responder (en el caso de estar el puerto abierto) al establecimiento de la conexión, y de ahí poder obtener más información, como por ejemplo el tipo de ataque realizado, el virus que ocasiona la comunicación, etc.

Existen múltiples artículos relativos al análisis de tráfico IBR, algunos de ellos de hace muchos años como los ya citados [1] [6] [12], se escriben prácticamente todos los años varios artículos sobre el tráfico IBR o sobre análisis del mismo (*botnets* [8], *crawlers* [17] etc.) y existen varios telescopios de red activos (conocidos y privados) que muestran la importancia del tráfico IBR. Por ello es interesante explorar nuevas opciones, más dinámicas y sencillas para obtención y análisis de este tipo de tráfico.

IV. Darknet doméstica

Según la evolución de los telescopios de red mostrados en la *Tabla I*, y teniendo en cuenta el agotamiento de direcciones IPv4, cada vez es más complicado disponer de grandes bloques de direcciones sin utilización para poder crear *darknets*. Otro de los problemas principales de las *darknets* es que los

atacantes pueden conocerlas y eliminarlas de sus búsquedas de objetivos. La información de utilización de las direcciones está registrada en la información de los RIR, por lo que con poco esfuerzo es posible obtener una superficie de ataque real que evite los espacios de direcciones usadas por las *darknets*. La *greynet* creada por Akamai evita en parte estos problemas.

Es complicado disponer de grandes bloques de direcciones para realizar estos análisis, por ejemplo, por investigadores. Además, no es viable disponer de grandes equipos de comunicaciones (enrutadores, cortafuegos, etc.) para realizar el análisis y normalmente terminan utilizando algunas de las redes existentes, por ejemplo, CAIDA.

Estas *darknets* o *greynets* están pensadas para destinos de las comunicaciones, pero no para tránsito de las mismas. Por ejemplo, un proveedor de servicios de Internet (ISP) no puede analizar el tráfico de o hacia sus clientes para ver si es tráfico legítimo, primero porque sería algo que podría ir en contra de la neutralidad de la red, pero también porque el volumen de tráfico a analizar sería imposible. En el caso de los proveedores de *hosting* o *housing* no es posible conocer qué servicio están ofreciendo sus clientes y por lo tanto, no pueden analizar el tráfico IBR.

En base a lo anterior, la propuesta consiste en la creación de una red extensa de sensores que permita detectar los ataques que se están realizando en Internet mediante la adquisición del tráfico IBR, utilizando equipamiento doméstico. Para ello se recopilará todo el tráfico que llegue a los *router* que permiten conectar a los usuarios a Internet y que no haya sido solicitado por dicho usuario, siendo por lo tanto tráfico de tipo IBR. Dado que la implantación de IPv6 es todavía pequeña en España y que la mayoría de los ataques a gran escala se realizan sobre IPv4, se ha decidido realizar este análisis inicialmente en IPv4. Es posible realizar un análisis similar mediante IPv6, pero requiere de mecanismos adicionales a las soluciones planteadas.

La adquisición del tráfico se basa en el funcionamiento de las tablas de NAT de los *routers* domésticos. Este es el funcionamiento habitual, donde el interface externo ADSL, FTTH, HFC, etc.) tiene configurada la IP pública del usuario y se realiza NAT para poder disponer de múltiples dispositivos por el usuario. En el caso de existir diferentes tráficos (televisión, teléfono, alarmas, etc.), habitualmente separados por VLANes, este estudio hace referencia únicamente a la VLAN de Internet, que es desde donde se realizarán los ataques que se están analizando.

En el funcionamiento normal de una comunicación de un usuario doméstico, cuando un equipo de dentro de la LAN quiere comunicarse con un equipo de Internet, envía el tráfico IP a través del router. El *router* identificará la comunicación mediante una entrada en la tabla de NAT, registrando las direcciones IP y los puertos origen y destino de la comunicación enviados por el equipo de la LAN y cambiando las dirección y el puerto origen de la comunicación para indicar la IP pública del *router* y un puerto no usado previamente por otro equipo. De esta forma, cuando el equipo destino responda al usuario, el *router* puede identificar mediante el puerto destino (puerto origen indicado por el router) a que dispositivo de dentro de la LAN corresponde el tráfico. Esto se puede observar en la *Fig. 3* como tráfico establecido.

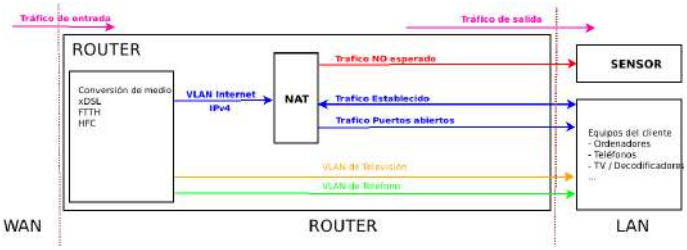


Figura 3. Tráfico de usuarios domésticos

El usuario también puede ofrecer servicios a Internet, por ejemplo, un servidor web. En este caso se introduce una entrada fija en la tabla de NAT en la que se registra que para el puerto (o puertos) seleccionados, el tráfico se debe entregar al nodo indicado por el usuario. En la *Fig. 3* aparece como tráfico de puertos abiertos.

En el caso del tráfico IBR, cuando el *router* recibe el tráfico lo busca en la tabla de NAT, como cualquier tráfico legítimo, pero no encontrará ninguna entrada que corresponda a la dirección IP y puerto origen y, por lo tanto, descartará el tráfico. Esto se puede observar en la *Fig. 3* como tráfico no esperado, solo que en vez de descartar el tráfico, se enviará a un sensor que realizará el registro del mismo.

IV-A. Arquitectura

Para este análisis se ha realizado una pequeña arquitectura, mostrada en *Fig. 4*, basada en el *router* que tiene el usuario. El *router* enviará el tráfico IBR al sensor del usuario (indicado como una flecha roja en la figura), de forma que cada conexión a Internet tendrá un sensor. Todos los sensores formarán la red de sensores, que es la parte principal de esta red. El tráfico recogido por los sensores será enviado a una o varias colectoras (flujos indicados con flechas verdes), utilizando para ello una comunicación HTTPS. La tercera parte consiste en el sistema de análisis y visualización de la información, que, si bien es la parte menos importante del análisis, ofrece una muestra de la viabilidad de la solución propuesta.

IV-B. Configuración del router

La entrada del tráfico, tanto legítimo como IBR, se realiza a través del *router* del usuario. Para este análisis se han utilizado los equipos proporcionados por los proveedores de acceso a Internet (ISP), no siendo necesario realizar ningún desembolso. Para hacer que el tráfico IBR (tráfico IPv4 con origen Internet, que no sea de una comunicación previamente

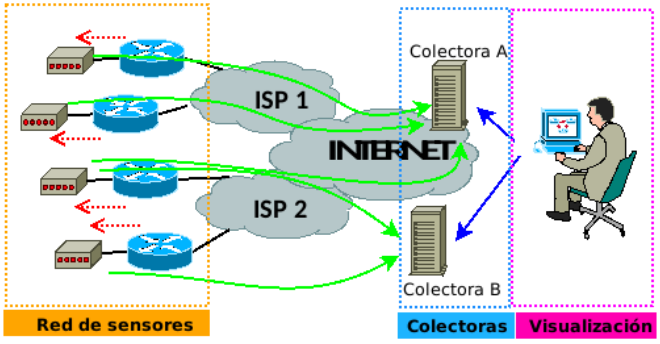


Figura 4. Arquitectura física

Figura 5. Configuración del router

establecida desde la LAN y que no esté incluido dentro de los puertos abiertos por el usuario) se envíe del *router* al sensor en vez de descartarlo, se debe realizar una pequeña configuración, que se puede implementar de dos formas.

La opción más sencilla y rápida para realizar esta función es utilizar la opción de "equipo de DMZ" que incluyen los *routers* de los ISPs. Esta opción le indica al *router* que todas las comunicaciones entrantes (que no correspondan a sesiones previamente establecidas desde la red LAN) se envíen a un equipo específico de la LAN, en este caso el sensor. La otra opción es redireccionar todos los puertos que no estén ya abiertos mediante la apertura de puertos (port forwarding). En ambos casos, el tráfico entrante sin entradas en la tabla de NAT serán enviados al sensor. Es por lo tanto necesario en ambos casos, indicar la IP que tendrá el sensor en la LAN y que dependerá de cada uno de los usuarios.

En la *Fig. 5* se muestra un ejemplo de port forwarding a la IP 192.168.1.166 de todos los puertos TCP/UDP y además del tráfico ICMP.

IV-C. Implementación de los sensores

El sensor es la parte principal de la red y para su implementación se han buscado varios objetivos. El primero es utilizar hardware de bajo coste, de forma que pueda ser instalado por cualquier usuario. El sistema de configuración debe ser sencillo y confiable, de forma que ofrezca al usuario tranquilidad en relación con el sistema.

Como hardware, tras varios análisis, se han utilizado diferentes modelos de Raspberry Pi (desde versión 1, Nano, etc.), que disponen de conexión de red a un precio muy económico. Se ha utilizado una tarjeta SD de al menos 1GB para la instalación del software.

Debido a que la configuración de cada uno de los sensores depende de la red doméstica del usuario, se ha buscado un método sencillo de instalación y configuración, que además ofrezca transparencia en el proceso. Para ello se han utilizado dos partes diferenciadas, una imagen original de *Raspberry Pi OS* versión *Lite*, descargada directamente desde el sitio oficial[2] y un conjunto de scripts que leen un fichero de configuración y establecen una configuración única para el usuario. El usuario debe bajar la imagen y grabarla en la tarjeta SD, como realizaría con cualquier imagen de Raspberry Pi, copiará en la unidad de arranque de la tarjeta los scripts y editará el fichero de configuración *config.sh*.

Un ejemplo de la configuración se puede ver en *Fig. 6*, donde se indica la dirección IP que tendrá el sensor, que coincidirá con la dirección indicada en el *router* para el forwarding, la zona horaria, la IP del gateway, configuración DNS, la colectora (o colectoras) donde reportará la información y el usuario y contraseña utilizadas en en el reporte a la colectora.

Los scripts son los mostrados en la *Tabla II*, donde se indica de forma resumida lo que hace el script. Al tratarse de ficheros independientes (y de tamaño pequeño, habitualmente menos de 10 líneas), el usuario puede auditar y comprender rápidamente lo que está realizando. Los scripts instalan el software necesario y establecen la configuración (principalmente la dirección IP del sensor dentro de la LAN y el usuario y la contraseña con la que enviarán los datos a las colectoras).

El usuario únicamente deberá modificar el fichero *config.sh* y, tras arrancar el dispositivo la primera vez, ejecutará el fichero *runme.sh*, que sustituirá la configuración en los ficheros de plantilla y los copiará en sus correspondientes destinos, haciendo una copia de seguridad de la información previa en caso de existir.

Tabla II
SCRIPTS DE CONFIGURACIÓN

Script	Descripción
config.sh	Fichero de configuración
runme.sh	Fichero a ejecutar por el usuario
01_create_dhcpcd.sh	Configuración de la red
02_disable_ipv6.sh	Desactivación de IPv6
03_timezone.sh	Configuración horaria
04_create_hnet_files.sh	Configuración
05_packages_install_1.sh	Instalación de software
06_syslog-ng_config.sh	Configuración de syslog
07_firewall.sh	Configuración del cortafuegos
08_regenerate_ssh_keys.sh	Generación de claves SSH
files/hnet.ini	Plantilla configuración general
files/hnet.py	Script de ejecución
files/iptables-rules.sh	Plantilla de cortafuegos
files/syslog-ng.conf	Configuración de syslog

El funcionamiento del conjunto consiste en que el tráfico llega al *router* del usuario. Dado que el tráfico es de tipo IBR y que el usuario no lo espera, el *router* lo enviará al sensor usando para ello la dirección IP establecida. El tráfico será recibido por el interface de red del sensor y se enviará al firewall, implementado mediante *netfilter*. El firewall no aceptará ninguna conexión entrante y por lo tanto descartará el paquete, informando mediante *syslog* de la acción y añadiendo la cadena de texto *INPUT:DROP:* . El software de *syslog* utilizado enviará el log a un script de ejecución permanente (llamado *hnet.py*) que, al recibir la entrada de log, la procesará y si corresponde, la enviará a las colectoras.

Este funcionamiento se puede ver en la *Fig. 7*

El script *hnet.py* realiza una serie de pasos previos al envío,

```
1  #!/usr/bin/env bash
2  LOCALNET="192.168.1.0/24"
3  IPADDRESS="192.168.1.166/24"
4  GATEWAY="192.168.1.1"
5  DNS="8.8.8.8"
6  TIMEZONE="Europe/Madrid"
7  SIEM="hnet.example.com"
8  HNET_COLLECTORS="https://example.com:443/cgi-bin/up.py"
9  HNET_USER="username"
10 HNET_PASS="password"
11 # Regenerar claves SSH: yes or no
12 REGENERATE_SSH_KEYS="yes"
```

Figura 6. Configuración del sensor

[2]Raspberry Pi: https://www.raspberrypi.com/software/

como son eliminar todo el tráfico que sea de carácter local a la red del usuario y que no venga de Internet, eliminación de información que no es relevante (como direcciones MAC o interfaces de red), incluir una marca horaria (*timestamp*) en la entrada de log, añadir el usuario y contraseña y por último realizar la conexión HTTPS con las colectoras y realizar el envío de la información.

IV-D. Recolección de la información

Las colectoras están provistas de un script de tipo CGI (*Common Gateway Interface*) con el que atienden las conexiones de los diferentes sensores y reciben la información. La información enviada tiene formato JSON (*JavaScript Object Notation*) formado por tres campos: Usuario, contraseña y log. Al recibir la información, comprueba el usuario y contraseña y, si son válidos, procesa la información recibida.

El procesado de la información consiste en analizar la dirección IP origen del tráfico IBR y, mediante una consulta a un servidor web auxiliar, añadir información adicional a la información recibida. Este proceso de enriquecimiento de la información añade el número de AS (sistema autónomo) correspondiente a la dirección IP, la red padre a la que pertenece la red según los registros de los diferentes RIR, el país en el que se encuentra registrada la dirección IP y la descripción incluida por el propietario de la dirección IP.

Adicionalmente al enriquecimiento de la información y debido a que el log generado por el sensor incluye la dirección IP de la LAN y no la IP pública, se añade la dirección IP pública del sensor. También se incluye un timestamp de la hora de recepción en la colectora.

Con el objetivo de facilitar el procesado de la información en el sistema de visualización, el script de la colectora separa en diferentes ficheros, dependiendo del protocolo, el tráfico recibido. Se utilizan cuatro ficheros diferentes, uno para TCP, otro para UDP, otro para ICMP y otro para el resto del tráfico. La razón es que los campos de cada uno de los protocolos son diferentes y es más sencillo separar la información antes de introducirla en el sistema de visualización.

IV-E. Visualización de la información

Para el análisis de los ficheros se ha optado por una solución basada en Elasticsearch[3], un motor de análisis de datos distribuido. La *Fig. 8* muestra el flujo completo, donde la información es enviada por los diferentes sensores utilizando el script *hnet.py*. Las entradas de log son recibidas por la colectora, mediante el script CGI *up.py*, y que separa la información en los cuatro ficheros anteriormente indicados. La información se introduce en Elasticsearch utilizando para ello Logstash[4], que lee los cuatro ficheros generados y los introduce en el motor de análisis. Finalmente, para la visualización de la información se utiliza la herramienta Kibana[5], que es accedida por los usuarios mediante un navegador web. La función de recolección (colectora) y el procesado con Elasticsearch podrían ser separados en diferentes equipos, pero para esta prueba se ha incluido todo en un mismo sistema. Un ejemplo de visualización de la información se puede ver en la *Fig. 9*, donde se pueden ver el número de ataques por país.

[3]ElasticSearch: https://www.elastic.co/es/what-is/elasticsearch
[4]LogStash: https://www.elastic.co/es/logstash/

Figura 7. Flujo del tráfico en el sensor

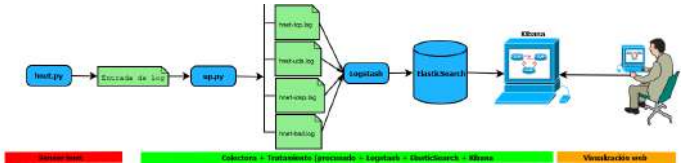


Figura 8. Flujo completo del log

V. Análisis y resultados

La *darknet* se ha llevado a cabo con cinco sensores, y durante un periodo de dos meses (5 de junio a 5 de agosto de 2021). De media ha sido capaz de recopilar aproximadamente 8.000 registros por sensor al día. Con esta información ha sido posible identificar diferentes situaciones previamente explicadas en diferentes artículos referentes a IBR, que se han utilizado para validar la información obtenida. Algunos de los casos son los siguientes:

Fuentes de origen: La gran mayoría de los ataques se originan desde sistemas autónomos dedicados a *hosting*. Por ejemplo, importantes empresas de Rusia, Estados Unidos o Francia dedicadas a este mercado. En este sentido, la mayoría del tráfico generado por algunos países se corresponde al tráfico generado desde estos AS. Por ejemplo, de los 442K paquetes recibidos desde Rusia, 394K provienen de un único AS. En Francia, de los 27K paquetes recibidos, 22.500 vienen de un hosting, 2.600 de de otro y cerca de 1.000 de un tercero, representando sobre el 90 % del tráfico generado. Esta situación fue previamente identificada por Durumeric en

[5]Kibana: https://www.elastic.co/es/kibana/

Figura 9. Visualización de la información en Kibana

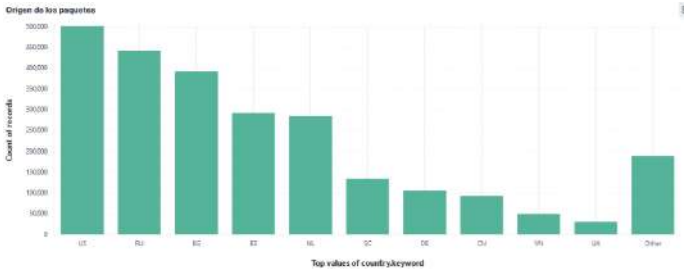


Figura 10. Países de origen del tráfico IBR

2014 [12]. Existe un cambio sobre los países que más tráfico generan, pero al igual que se indica en el artículo indicado, las fuentes están asociadas no a países, sino a proveedores de hosting o ISPs. Mientras que Durumeric indicaba como principales fuentes de origen China (31 %), Estados Unidos (22 %), Alemania (9.5 %), Holanda (8.8 %) y Rusia (4.8 %), las fuentes actuales con mayor número de ataques serían Estados Unidos (20 %, 3 proveedores de servicios), Rusia (17 %, dos proveedores), Bulgaria (15 %), España (11 %) y Holanda (11 %). Se puede observar la información recogida en *Fig. 10*. En este caso, China aparece como el octavo, con un 3.7 % y Alemania sería el séptimo (4,2 %), mostrando un elevado crecimiento Rusia. Este crecimiento se podía observar en el análisis realizado por Akamai en 2019 [15], donde Rusia ya mostraba un aumento hasta el 8 %. En el estudio de Akamai se muestra un volumen muy elevado del tráfico de Ukrania (41 %), si bien representa el décimo lugar del análisis realizado. Es interesante el caso de Bulgaria, dado que dos AS (con únicamente 256 direcciones cada uno) generan todo el tráfico obtenido.

Protocolos: Observando los protocolos, el 80 % del tráfico corresponde al protocolo TCP (2 millones de paquetes), un 19 % de UDP (488k paquetes) y aproximadamente 12.000 paquetes de tipo ICMP, todos ellos de tipo *echo*. Se han recibido unos 2.000 paquetes de tipo GRE, probablemente buscando túneles basados en este protocolo. Residualmente se ha recibido tráfico SCTP (6 paquetes), DCCP (2 paquetes) y IPv6 (IPv6 over IPv4). 5 paquetes tenían como protocolo el cero. Analizando los protocolos de transporte TCP y UDP, en el caso de TCP la mayoría del tráfico tiene como destino el puerto SSH (59.500 paquetes) y el puerto Telnet (32.200 paquetes). El puerto TCP/1 tiene 28.000 paquetes y el puerto TCP/7680 (WUDO, Windows Update Delivery Optimization) unos 20.500 paquetes. Otros puertos con más de 5.000 paquetes serían TCP/445 (NetBIOS), TCP/1433 (SQL Server), TCP/2375 y TCP/2376 (Docker), TCP/81, TCP/80, TCP/443, TCP/8080 (varios servicios Web), TCP/3389 (RDP para administración remota de Windows). Los puertos TCP/5555 y TCP/1026, usdos por múltiples servicios y también por malware, se encuentran también entre los más utilizados. Estos puertos son similares a los obtenidos en el análisis realizado por Akamai en 2019 [15] y Durumeric en 2014 [12]. En el caso de UDP, si observamos toda la información recogida, los puertos más consultados son el UDP/22000, UDP/53431, UDP/29123 y UDP/6602. En este caso, llama la atención por la diferencia el estudio realizado por Akamai [15], con el de Durumeric [12] y también con el de Wustrow en 2010 [6]. Tras analizar la información, se observan dos factores diferentes. Por un lado, al realizar un filtro por los puertos indicados anteriormente se observa que son volúmenes de tráfico puntuales, que se realizan durante un periodo de varios días (por ejemplo, se ha observado 5 días con un pico de 24.400 paquetes para el puerto UDP/53431 y de 30 días para el puerto UDP/22000, con un ratio de 2500 paquetes al día). Al ser un volumen tan elevado de tráfico en estos puertos, modifica sustancialmente los puertos usados y es necesario tenerlos en cuenta para el análisis. Por otro lado se ha observado una diferencia de comportamiento entre los sensores instalados en las redes de unos ISPs y otros, mientras que en unos casos sí se muestran los mismos puertos que en los estudios previos (*Fig. 11*), en otros no se observan algunos de estos puertos, por ejemplo, el UDP/53 (DNS). La razón es que el ISP está filtrando algunos de estos puertos, porque son utilizados para realizar ataques de denegación de servicio distribuido (DDoS) mediante preamplificación. Teniendo en cuenta ambos factores, la información es similar a los estudios previamente realizados.

Otros factores: Se han realizado análisis de otros factores adicionales mencionados en artículos previos. Se ha observado que para aproximadamente 2.5 millones de paquetes TCP, 1.95 millones tenían únicamente el flag de SYN activado, mientras que 73.000 tenían los flags de ACK y PSH. En los análisis de Wustrow [6] y Akamai [15] se indicaba el elevado porcentaje de tráfico con el flag de SYN activado, llegando a ser del 98 % en este último. Mediante la visualización de la gráfica de puertos TCP destino, donde en la *Fig. 12*, en el eje X se han incluido los primeros 1.300 puertos más usados, se puede ver como muy pocos escaneos de puertos (en torno a 300) se realizan contra todos los puertos posibles haciendo un eje X plano en torno a ese valor, mientras que la mayoría de los ataques se realizan sobre un conjunto de puertos muy pequeño (menos de 50).

El sistema recoge únicamente información de la cabecera, no recogiéndose información relativa al payload. Se ha analizado la longitud de los paquetes recibidos, siendo en TCP la

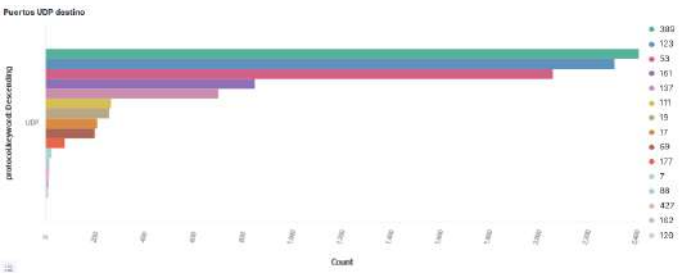


Figura 11. Puertos UDP destino más usados

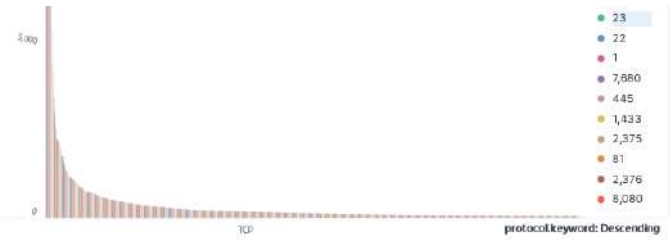


Figura 12. Puertos TCP destino más usados

mayoría de ellos de tamaño 44 (86 %), de tamaño 52 (7.5 %), tamaño 60 (3 %) y el resto de tamaños un porcentaje menor del 1 %. En el caso de UDP, los tamaños más usados son 44 (44 %), 96 (21 %), 132 (18 %) y el resto de tamaños son porcentajes inferiores. En el caso de TCP, al igual que en otros estudios [6] [15] los paquetes no llevan payload, mientras que en UDP podría ser interesante poder obtener el payload para ver la información enviada y de esta forma poder obtener más información del ataque. Sin embargo, la realización de este cambio puede complicar la sencillez del modelo utilizado y debe evaluarse.

VI. Conclusiones

Mediante los resultados recogidos a lo largo de este análisis y tras comparar con los resultados obtenidos de otros estudios previos, se puede concluir que es viable la realización de *darknets* mediante la utilización de conexiones domésticas.

Las *darknets* mediante este tipo de conexiones tienen ventajas sobre las *darknets* tradicionales, como un coste muy bajo de creación, configuración y puesta en marcha en minutos, no requieren mantenimiento y son transparentes para el usuario.

Al basarse en conexiones domésticas son mucho más flexibles y dinámicas que las *darknets* tradicionales, no siendo tan importante ocultar la existencia de los sensores, dado que el direccionamiento IP del usuario puede cambiar al apagar el *router*. En el caso de las *darknets* tradicionales es muy importante ocultar las redes usadas para la adquisición de tráfico, debido a que al ser identificadas los generadores de IBR podrían no enviarles tráfico. Adicionalmente, las *darknets* mediante conexiones domésticas pueden ayudar a proteger a todos los usuarios, porque en caso de ser identificadas, el generador de tráfico IBR debe elegir si analizar esa red y ser detectado o no hacerlo para ningún usuario.

Tras el agotamiento de las direcciones IPv4, es cada vez más necesario el uso de todo el espacio de direccionamiento IP, haciendo que los telescopios de red con grandes volúmenes de direcciones puedan ver reducido su espacio de direccionamiento. Las *darknet* creadas con equipamiento doméstico se encuentran dentro de rangos de direcciones en uso (convirtiendo la *darknet* en una *greynet*) y por lo tanto son un objetivo habitual del tráfico IBR, mientras que las *darknets* tradicionales se encuentran en muchos casos en espacios de direccionamiento de Internet no utilizado, pudiendo ser identificadas y no analizadas por los generadores de tráfico IBR, ahorrando tiempo y recursos.

Es recomendable distribuir la *darknet* en varios ISPs, evitando posibles problemas de filtrado de puertos, como los mostrados en el estudio. En caso de que el usuario tenga puertos abiertos en el *router*, por defecto, su tráfico no podrá será analizado por el sensor.

El sistema es escalable, pudiendo añadir los nodos según las necesidades. Según se ha visto en el estudio, con muy pocos nodos se puede obtener un gran volumen de información, si bien un elevado volumen de tráfico, con pocos nodos, puede distorsionar la realidad de la información recogida. Sería interesante evaluar, como un estudio posterior, el número de nodos mínimo recomendable para la realización de una *darknet* doméstica y a partir de qué número de nodos la información obtenida es redundante y no aporta valor adicional.

Otros trabajos futuros podrían analizar la viabilidad de implementar el sensor dentro del *router* del ISP o incluirlo en otro tipo de dispositivos. Se podrían estudiar métodos para la adquisición del *payload*, con el objetivo de obtener información relevante adicional. También puede ser interesante la introducción conjunta de sensores pasivos y sensores con ciertos servicios dentro de la *darknet*.

Agradecimientos

A las PErsonas que han participado aportando sus sensores, conexiones y tiempo.

Este trabajo ha sido parcialmente financiado por los proyectos SICRAC (PID2020-114495RB-I00) y ANIMaLICoS (PID2020-113462RB-I00) del Ministerio de Ciencia, Innovación y Universidades.

Referencias

[1] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, y L. Peterson, «Characteristics of internet background radiation», en Proceedings of the 4th ACM SIGCOMM conference on Internet measurement - IMC '04, Taormina, Sicily, Italy, 2004, p. 27. doi: 10.1145/1028788.1028794.

[2] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, y J. Bannister, «Exploring Visible Internet Hosts through Census and Survey», p. 16. 2006.

[3] M. Allman, V. Paxson, y J. Terrell, «A brief history of scanning», en Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, 2007, pp. 77-82. doi: https://doi.org/10.1145/1298306.1298316.

[4] Hutcheson, L. «MS04-011 LSASRV Exploit; Sasser Worm Update: Sasser.b» https://isc.sans.edu/forums/diary/Updated+MS04011+LSASRV+Exploit+Sasser+Worm+Update+Sasserb/182/. SANS ISC InfoSec Forums. 2004.

[5] Microsoft. «Microsoft Security Bulletin MS08-067 Critical. Vulnerability in Server Service Could Allow Remote Code Execution (958644)». https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2008/ms08-067 23 de octubre de 2008.

[6] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, y G. Huston, «Internet background radiation revisited», en Proceedings of the 10th annual conference on Internet measurement - IMC '10, Melbourne, Australia, 2010, p. 62. doi: 10.1145/1879141.1879149.

[7] Anónimo. «Port scanning /0 using insecure embedded devices» https://seclists.org/fulldisclosure/2013/Mar/166 2013.

[8] E. Le Malécot y D. Inoue, «The Carna Botnet Through the Lens of a Network Telescope», en Foundations and Practice of Security, vol. 8352, J. L. Danger, M. Debbabi, J.-Y. Marion, J. Garcia-Alfaro, y N. Zincir Heywood, Eds. Cham: Springer International Publishing, 2014, pp. 426-441. doi: 10.1007/978-3-319-05302-8_26.

[9] Fyodor. «The Art of Scanning» Phrack Magazine Volume 7, Issue 51 ,pág. 11 de 17. http://phrack.org/issues/51/11.html. Septiembre 1997.

[10] Z. Durumeric, E. Wustrow, y J. A. Halderman, «ZMap: Fast Internet-Wide Scanning and its Security Applications», p. 16.

[11] Graham, R. (2013). «MASSCAN: Mass IP port scanner» https://github.com/robertdavidgraham/masscan

[12] Z. Durumeric, M. Bailey, y J. A. Halderman, «An Internet-Wide View of Internet-Wide Scanning», p. 15. 2014.

[13] Pearson, D. T. «An exploration of the overlap between open source threat intelligence and active internet background radiation» http://vital.seals.ac.za:8080/vital/access/manager/Repository/vital:32299?site_name=GlobalView:RhodesUniversity-FacultyofScience, ComputerScience]. 2020.

[14] Cell, N. J. «Mirai. NJCCIC Threat Profile» https://www.cyber.nj.gov/threat-center/threat-profiles/botnet-variants/mirai-botnet. 2016.

[15] P. Richter y A. Berger, «Scanning the Scanners: Sensing the Internet from a Massively Distributed Network Telescope», en Proceedings of the Internet Measurement Conference, New York, NY, USA, oct. 2019, pp. 144-157. doi: 10.1145/3355369.3355595.

[16] W. Harrop y G. Armitage, «Defining and Evaluating Greynets (Sparse Darknets)», en The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)l, Sydney, NSW, Australia, 2005, pp. 344-350. doi: 10.1109/LCN.2005.46.

[17] M. Zolotykh, «Study of Crawlers of Search Engine 'Shodan. io'», en 2021 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), 2021, pp. 0419-0422. doi: 10.1109/USBEREIT51232.2021.9455018.

Aplicación de técnicas de reducción de dimensionalidad y balanceo en ciberseguridad

Óscar Mogollón Gutiérrez
José Carlos Sancho Núñez
Universidad de Extremadura
Escuela Politécnica
{oscarmg,jcsanchon}@unex.es

MohammadHossein Homaei
Universidad de Extremadura
Escuela Politécnica
mhomaein@alumnos.unex.es

Javier Alonso Díaz
Universidad de Extremadura
Escuela Politécnica
javieralonso@unex.es

***Resumen*—El número de ataques en el ámbito IoT ha incrementado significativamente en los últimos años. Para hacer frente a posibles actuaciones fraudulentas, numerosos estudios proponen la construcción de sistemas basados en Machine Learning y conjuntos de datos especializados. Con el objetivo de reducir la complejidad de los modelos y hacer frente a la problemática del desbalanceo, en este artículo se propone un análisis del impacto de la reducción de dimensionalidad mediante diferentes técnicas: ANOVA, correlación de Kendall y PCA. Tras la reducción de dimensionalidad, se evalúa como influye el balanceo del conjunto de datos con la técnica de sobremuestreo SMOTE. Los resultados de la investigación demuestran que la reducción de dimensionalidad mediante la correlación de Kendall supone una mejora en la complejidad a la vez que se mantiene la capacidad de clasificación del modelo. La aplicación de SMOTE al conjunto de menor dimensión no supone una mejora en la capacidad de detección del modelo.**

***Index Terms*—reducción de dimensionalidad, sobremuestreo, detección de intrusiones, UNSW-NB15**

Tipo de contribución: *Investigación original*

I. Introducción

El auge del Internet de las Cosas (IoT) es cada vez mayor. Se estima que el número de dispositivos conectados a Internet en 2022 sea de 29 billones, de los cuales, unos 19 billones correspondan a dispositivos IoT [1]. Este elevado número de dispositivos interconectados implica que las formas en que la tecnología puede ser vulnerada con fines maliciosos aumenten considerablemente. Según el informe realizado por Kaspersky, una empresa internacional de seguridad informática, 1.5 billones de dispositivos IoT fueron vulnerados en el primer semestre de 2021. Esta cifra es superior al doble de los ciberataques registrados en los primeros seis meses del 2020, que se sitúa en torno a seiscientos millones [2].

Con motivo de las deficiencias identificadas en los dispositivos IoT [3], se hace necesaria la construcción de soluciones que busquen mitigar el impacto de los ciberataques en un entorno interconectado. De ahí que muchos investigadores centren sus esfuerzos en investigaciones que giren en torno a la ciberseguridad con un enfoque reactivo, donde se identificará el problema en tiempo real y se establecerán medidas para mitigar el impacto de los ataques mediante la construcción de Sistemas de Detección de Intrusiones (IDS) [4], [5].

Según NIST [6], existen principalmente dos tipos de IDS según la naturaleza de los datos recopilados. Por un lado, los IDS basados en host (HIDS) monitorizan los eventos que ocurren en un único sistema en busca de actividad sospechosa. Por este motivo, este tipo de sistemas solo pueden proteger el dispositivo donde se instala. Por otro, los IDS basados en red (NIDS) se encargan de analizar el tráfico de una red y, en caso de identificar tráfico inesperado procedente de alguno de los dispositivos conectados, llevar a cabo acciones protectoras.

Para construir un IDS una de las técnicas más efectivas es el uso de la Inteligencia Artificial (IA) [7]. Existen conjuntos de datos creados para este fin que recogen actividad legítima y maliciosa. En la literatura, los investigadores emplean estos conjuntos de datos para probar la efectividad de los modelos de IA [8], [9]. Sin embargo, se han encontrado dos problemas recurrentes que no se suelen tenerse en cuenta de manera conjunta. El primero de ellos es que muchos de estos trabajos se centran en maximizar el rendimiento del IDS sin tener en cuenta el alto coste computacional que conlleva su implantación en dispositivos más simples orientados a IoT [10], [11]. Otra cuestión que debe considerarse a la hora de trabajar con un conjunto de datos es la distribución de categorías. Generalmente, suelen presentar un desbalanceo, siendo mayoritarias las muestras relativas a tráfico legítimo. La presentación de resultados a partir de un conjunto no balanceado influye en la evaluación final de los experimentos, como en [8] donde la propuesta presenta una baja capacidad para clasificar muestras de varias categorías.

Para resolver ambos problemas en este trabajo se presenta un estudio que muestra cómo influye la reducción/selección de características en el rendimiento de un IDS. El conjunto de características resultantes permitirá la construcción de modelos más eficientes. Para comprobarlo, se tiene en cuenta la complejidad computacional de los modelos construidos usando el tiempo de entrenamiento. Una vez seleccionado el conjunto óptimo de características, se balancea el conjunto para demostrar cómo se ve afectada la detección de ataques. Las métricas de evaluación más comunes en este tipo de problemas: Accuracy, Precision, Recall, F1-Score.

La organización del artículo es la siguiente: primero, en la Sección II se describen los trabajos relacionados en el ámbito de la detección de intrusiones aplicando técnicas de reducción de características y/o balanceo de datos con mayor repercusión en el ámbito científico, en la Sección III se especifican los algoritmos de balanceo y clasificación aplicados en los experimentos, en la Sección IV se detalla la metodología aplicada para la construcción del modelo y en la Sección V se muestran y discuten los resultados obtenidos. Por último, en la Sección VI se exponen las conclusiones de la investigación junto a posibles líneas futuras de investigación que mejoren

el funcionamiento la solución propuesta.

II. Trabajos relacionados

En los últimos años, el número de estudios relativos a los sistemas de detección de intrusiones ha crecido considerablemente. Una de las principales causas de este este aumento de publicaciones está directamente relacionado con el auge de los dispositivos inteligentes: teléfonos, relojes, altavoces, dispositivos IoT, entre otros. Como consecuencia de este aumento de dispositivos, la variedad de ataques informáticos crece y, por ello, se hace necesaria la construcción de sistemas de detección de intrusiones eficaces. Para ello, en los últimos años se han publicado diferentes datasets con el propósito de simular tráfico de red y cómo este se afectado al sufrir los algunos de los ataques más comunes.

El dataset KDD99 fue creado en 1998 por DARPA a partir de tráfico de red generado. Los ataques recogidos pertenecen a cinco categorías: Normal, denegación de servicios (DoS), user-to-root (U2R), remote-to-local (R2L) y Probing Attack (Probe). Respecto a la distribución de categorías, el tamaño del conjunto de entrenamiento es de 4898431 tuplas y el conjunto de prueba se compone de 2984154 tuplas [12].

Basándose en el dataset anterior, surgió en 2009 NSL-KDD. El nuevo conjunto fue construido tras un estudio de las muestras más representativas realizado por [13]. El número de características se mantiene, pero la distribución de categorías pasa a convertirse en 125973 y 22544 de entrenamiento y prueba, respectivamente.

Un dataset más reciente es UNSW-NB15. Este conjunto fue creado en 2015 por el Cyber Range Lab del UNSW de Canberra con el objetivo de simular un entorno heterogéneo de tráfico legítimo y de ataque real [14]. El tráfico recopilado se enmarca en una de las siguientes 10 categorías: Analysis, Backdoor, DoS, Exploits, Fuzzers, Reconnaissance, Shellcode, Worms o Normal, correspondiéndose esta última categoría al tráfico legítimo. El conjunto de datos original consta de 2540044 tuplas, 49 características y presenta un alto desbalanceo de clases, ya que solo la distribución de tuplas correspondiente al tráfico legítimo representa más de un 87 % del total de muestras.

Dada la complejidad de los conjuntos de datos, derivada de la variedad de sistemas y los avances tecnológicos, se hace necesaria la aplicación de técnicas de ingeniería de características que mejoren la eficiencia de los algoritmos de inteligencia artificial. Dos de las tareas más importantes son la reducción de dimensiones y el remuestreo. La primera de ellas hace referencia a la eliminación de características que empeoran el rendimiento de un clasificador, o aumentan su complejidad sin variar su comportamiento. La segunda comprende el conjunto de técnicas que tiene como objetivo tratar con conjuntos de datos con un desequilibrio en el número de muestras de determinadas categorías en un dataset.

En la literatura, existen estudios que aplican diferentes métodos de reducción de dimensiones para disminuir la complejidad de los datos y mejorar la eficiencia de los sistemas. Las técnicas de reducción de dimensiones se dividen en dos: selección de características y extracción de características. La selección de características consiste en seleccionar un subconjunto de características del conjunto original, mientras que la extracción consiste en el proceso de generación de un nuevo conjunto de características, de menor dimensión, a partir de las originales. De acuerdo con la revisión realizada por Dhal sobre técnicas de selección de características, estas pueden clasificarse en cinco grupos principales: statistical, probability, similarity, sparse learning y algoritmos evolutivos [15]. En el trabajo de Anouncia [16], los métodos de extracción de características se agrupan en dos: lineales o no lineales dependiendo de si las características resultantes se han obtenido como combinación lineal de la originales o no. Entre los estudios más influyentes que aplican técnicas de reducción de dimensiones se encuentra: Zhang que aplica selección de características mediante Welch's-test para la detección de Alzheimer [17], van der Maaten aplica un método no lineal de extracción de características t-SNE para demostrar el rendimiento de esta técnica sobre múltiples conjuntos de datos [18].

Siguiendo con la revisión de trabajos relacionados, las técnicas de remuestreo se clasifican en dos tipos [19]. Las técnicas del primer grupo, data-level, consisten en balancear el número de muestras de cada categoría mediante técnicas de preprocesado como Synthetic Minority Oversampling Technique SMOTE [20] o cluster-based sampling [21]. El segundo grupo, algorithm-level, lo componen un conjunto de algoritmos capaces de aprender la distribución de un conjunto de datos desequilibrado como one-class learning, improved algorithm, cost sensitive learning, ensemble and hybrid technique. En [22] se emplea SMOTE para entrenar un modelo para la detección de accidentes de tráfico en tiempo real o en la contribución presentada por Khan [23] se construye una red neuronal profunda cost-sensitive para mejorar el aprendizaje en datasets de imágenes desbalanceados.

Para la construcción de IDS, una de las técnicas más aplicadas es la construcción de modelos de inteligencia artificial basadas en Machine Learning [24]. Para mejorar la eficacia y eficiencia de los modelos encargados de clasificar el tráfico recogido por un IDS, algunos estudios aplican técnicas de reducción de dimensiones, de remuestreo o ambas. El artículo publicado por Manimurugan [25] lleva a cabo una reducción de dimensiones aplicando PCA y consiguen obtener un accuracy del 92.48 %. Diferentes técnicas de remuestreo se llevan a cabo el ya mencionado trabajo de Bagui [26]. Respecto a investigaciones que aplican ambas técnicas se encuentra el estudio llevado a cabo en [27] donde se aplica una extracción de características basada en PCA junto Uniform Distribution Based Balancing para balancear los datos.

III. Método

En este trabajo se presenta un estudio de la influencia de la reducción de dimensionalidad y balanceo para la clasificación de ciberataques. Para llevar a cabo la investigación se ha empleado un conjunto de datos que recoge el tráfico de red en periodos de ataques y de tráfico legítimo. En esta sección se ofrece una descripción del conjunto de datos, UNSW-NB15, características, distribución de muestras y ciberataques recogidos. Posteriormente, se detallan las técnicas de reducción de dimensiones y balanceo que se han llevado a cabo para la construcción de modelos. Finalmente, se expone una explicación concisa de los algoritmos de clasificación aplicados, junto al software y hardware que ha permitido llevar a cabo la investigación.

III-A. Conjunto de datos

El conjunto de datos UNSW-NB15 fue creado por el Cyber Range Lab del UNSW de Canberra con el objetivo de simular un entorno heterogéneo de tráfico legítimo y de ataque real [14]. El conjunto de datos original consta de 2540044 tuplas y presenta un alto desbalanceo de clases, ya que solo la distribución de tuplas de tipo Normal representa más de un 87 % del total de muestras. No obstante, los autores han publicado un conjunto de datos reducido tanto en el número de filas como en el número de características [28]. Los experimentos de esta investigación se han realizado con el dataset reducido. Consta de 44 características siendo 40 numéricas y 4 categóricas.

III-B. Algoritmos

Para la construcción del clasificador ensemble propuesto en este paper trabajo se han combinado los algoritmos de Machine Learning y Deep Learning que mejores resultados obtienen, según resultados previos de los experimentos que hemos realizado, y además son los más utilizados en la literatura científica [29].

- Naïve Bayes (NB). El algoritmo de Naïve Bayes es un algoritmo basado en el teorema de Bayes en el que se asume la independencia entre características de muestras pertenecientes a una misma clase [30].
- K Nearest Neighbors Classifier (KNN). El algoritmo KNN es un algoritmo de tipo IBL (Instance-Based Learning). Esto implica que el proceso de clasificación del modelo se base en el conocimiento adquirido durante la fase de entrenamiento. La clasificación de una muestra consiste en aplicar un algoritmo que calcula la distancia de dicha muestra respecto al resto y mediante un algoritmo de votación por mayoría entre las k muestras más cercanas, se decidirá la clase a la que pertenece [29].
- Support Vector Machine (SVM). La idea fundamental del algoritmo SVM consiste en encontrar el hiperplano que mejor divida el conjunto de datos en un número determinado de clases o categorías [31].
- Descenso de Gradiente Estocástico (SGD). Se trata de una variación de SVM que se caracteriza por emplear un kernel lineal basado en el descenso de gradiente estocástico como función de optimización de la función de coste de un modelo [32].
- Decision Trees (DT). Los modelos basados en DT son clasificadores capaces de predecir la categoría de una muestra aplicando reglas de decisión simples aprendidas en la fase de entrenamiento [33].
- Multilayer Perceptron (MLP). El MLP es un tipo simple de red neuronal artificial compuesta de una capa de entrada, una de salida y múltiples capas intermedias, todas ellas completamente interconectadas.

La selección del mejor valor para los hiperparámetros de cada algoritmo se ha realizado mediante la técnica Grid Search y Cross Validation. Esta búsqueda de hiperparámetros permite probar múltiples combinaciones de valores habiendo definido previamente un espacio de búsqueda para cada uno de ellos.

III-C. Métricas de evaluación

En este estudio se han seleccionado varias métricas de evaluación para estimar el rendimiento del modelo de clasificación. Las métricas consideradas son *accuracy*, *precision*, *recall* y F1 que se calculan en función de los Verdaderos Positivos (TP), Verdaderos Negativos (TN), Falsos Positivos (FP) y Falsos Negativos (FN). A continuación se muestra una definición de cada métrica y cómo se calcula.

- Matriz de confusión. Matrix de dimensión $N * N$, siendo N el número de categorías del conjunto de datos. Sirve para representar el desempeño del modelo por clases.
- Accuracy. Métrica que mide el número de predicciones correctas sobre el total de muestras evaluadas por el modelo.

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- Precision. Sirve para cuantificar el número de muestras positivas clasificadas correctamente entre las predicciones positivas realizadas por el modelo.

$$\frac{TP}{TP + FP} \tag{2}$$

- Recall. Se utiliza para evaluar el número de muestras positivas clasificadas correctamente entre el total de muestras positivas en el conjunto evaluado.

$$\frac{TP}{TP + FN} \tag{3}$$

- F1. Medida estadística obtenida a partir de la media harmónica de *precision* y *recall*, resultando útil en dominios con distribuciones no balanceadas.

$$\sqrt{\frac{TP}{TP + TN} + \frac{TN}{FP + TN}} \tag{4}$$

III-D. Reducción de características

La reducción de dimensionalidad es el proceso de reducir el número de características del conjunto original para mejorar el rendimiento de un modelo de ML [15]. Esta técnica de preprocesado se puede abordar con dos enfoques, bien seleccionando un subconjunto de características (selección de características) o bien transformando las ya existentes en un espacio de menor dimensión (extracción de características). En esta sección se presentan los diferentes métodos de reducción de características aplicadas en la investigación.

- Test ANOVA (selección de características). Este método estadístico que realiza un análisis de la varianza de un conjunto de datos para ayudar en la selección de las mejores características para construir un modelo y determinar si una variable independiente está influyendo en una variable objetivo. En el caso de la detección de intrusiones el test ANOVA se encargará de identificar que variables son las más influyentes para determinar la categoría a la que pertenece una muestra de tráfico [34].
- Método de correlación de Kendall (selección de características). Se trata de una prueba estadística no paramétrica que mide la asociación entre variables. La relación entre dos variables se mide mediante los rangos de valores que toma una característica y no con los valores individuales. Puede tomar valores entre -1 y 1, siendo 1 en el caso de una asociación fuerte o -1 en caso contrario. A diferencia de la correlación de Pearson,

la correlación de Kendall mide la relación monótona entre variables y no la relación lineal, es decir, sirve para conocer si dos variables se mueven en la misma dirección, pero sin tener en cuenta el ritmo al que lo hacen [35].

- Análisis de componentes principales (PCA) (extracción de características). PCA es una técnica de reducción de dimensiones que transforma un conjunto de datos en uno nuevo con características no correlacionadas, denominadas componentes principales. Al aplicar PCA, las componentes principales se ordenan en función de la información que contienen de los datos originales. Por tanto, la primera componente calculada será la más descriptiva. A mayor número de componentes, menor es la pérdida de información [36].

III-E. Remuestreo

Al trabajar con conjuntos de datos no balanceados, existen dos enfoques que tratan de mitigar este problema: algorithm-level y data-level. En esta investigación se emplea una estrategia a nivel de datos ampliamente usada en la literatura científica, SMOTE [20]. Dada la naturaleza del conjunto de datos empleado en esta investigación, orientado a la ciberseguridad, las técnicas que se aplicarán serán de sobremuestreo, es decir, se busca aumentar el número de tuplas de ataque. Una de las técnicas más conocidas es Synthetic Minority Oversampling Technique (SMOTE) [20] y su funcionamiento se detalla brevemente:

1. Inicialmente, se identifican las muestras del conjunto de datos de entrada que pertenecen a la clase minoritaria.
2. A continuación, el procedimiento SMOTE itera sobre estas muestras para dibujar segmentos entre el vector de características de cada muestra minoritaria y sus correspondientes vecinos más cercanos K.
3. Entre los K vecinos calculados, se selecciona uno de ellos aleatoriamente.
4. La nueva muestra sintética se generará en una posición intermedia entre la muestra procesada y el vecino seleccionado.

Es preciso destacar que cada muestra sintética se crea entre dos muestras minoritarias reales. La nueva muestra sintética se sitúa en una zona aleatoria proporcional a la distancia euclídea que separa ambas muestras. La posición de la nueva muestra respecto a la procesada y su vecina dependerá de un número aleatorio entre cero y uno. De este modo, si el valor aleatorio es inferior a 0.5, la nueva muestra se creará más cerca de la muestra procesada. Análogamente, si es mayor que 0.5, la muestra sintética se situará más cerca de uno de sus vecinos. Los pasos 1, 2, 3 y 4 se repetirán hasta alcanzar el número de muestras especificado inicialmente.

IV. Diseño propuesto

En este trabajo se lleva a cabo un exhaustivo estudio comparativo con el fin de medir el impacto de la aplicación de técnicas para: disminuir la complejidad de los modelos a través de la reducción de dimensiones y abordar el problema de conjuntos de datos desbalanceados.

Esta investigación se compone de dos grupos de experimentos. El primer grupo de experimentos tiene como objetivo la evaluación del impacto de la reducción de dimensionalidad. Para lograr el objetivo, se aplicarán los métodos de reducción de dimensionalidad explicados previamente y, para cada conjunto de características generado, se entrenan varios modelos con los algoritmos mencionados en la subsección Algoritmos. Cada modelo generado, se evalúa mediante el cálculo de las métricas de evaluación más frecuentes en sistemas orientados a la clasificación de tráfico de red: accuracy, precisión, recall y F1-score. Así, se obtiene una estimación de la capacidad de clasificación de ciberataques, pudiendo determinar el modelo que muestra un mejor desempeño para cada subconjunto de características.

Al término de los experimentos relativos a la reducción de dimensiones, el segundo grupo de experimentos consiste en probar la influencia que tiene la generación de muestras sintéticas en el rendimiento de un sistema de clasificación de tráfico de red. La generación de nuevas muestras se lleva a cabo con SMOTE sobre el conjunto de entrenamiento original tras haber aplicado la reducción de dimensionalidad. Los algoritmos, junto a su configuración, que presenten un mejor desempeño en la primera fase de experimentación, se usarán para entrenar un nuevo modelo a partir del conjunto balanceado. Nuevamente, se obtendrá una evaluación en base a las métricas de evaluación que permita comparar la efectividad de la técnica de sobremuestreo aplicada. La Figura 1 muestra el proceso seguido para la realización de los experimentos.

Como es usual, es preciso realizar tareas de limpieza y normalización sobre los datos, previo a la aplicación de los algoritmos de clasificación. En este sentido, las características numéricas se han normalizado mediante el cálculo de la unidad tipificada, de forma que cada una seguirá una distribución gaussiana (media cero y varianza uno). Por otro lado, las características categóricas se han codificado usando etiquetas numéricas. Además, se han eliminado las tuplas redundantes.

Para probar cada uno de los métodos de reducción de dimensionalidad expuestos en la sección anterior es necesario establecer un criterio para la selección del número de características a considerar. Dado que el número de columnas del conjunto de datos es 44, se ha considerado que la reducción aplicada conserve un mínimo de 10 características y un máximo de 30. Para la construcción del mejor clasificador se han probado diferentes alternativas en función del método:

- El test ANOVA realizado devuelve una puntuación a cada característica según su importancia. Para la selección de características mediante el test ANOVA, se ordenarán por importancia, y se probará a seleccionar desde 10 hasta 30, dando lugar a un total de 5 experimentos.
- Para la reducción de la dimensionalidad mediante la correlación de Kendall se ha generado una matriz de correlación para las 44 características del conjunto de datos. Se han realizado dos grupos de experimentos, optando por eliminar aquellas características con un valor de correlación superior a 0.7 y a 0.8 ya que valores cercanos a 1 indican un elevado grado de asociación entre dos características y por tanto puede prescindirse de una de ellas. Los valores de correlación utilizados suponen una selección de 22 y 31 características para 0.7 y 0.8, respectivamente.
- La elección de experimentos a realizar para la extracción de características con PCA sigue el mismo criterio que

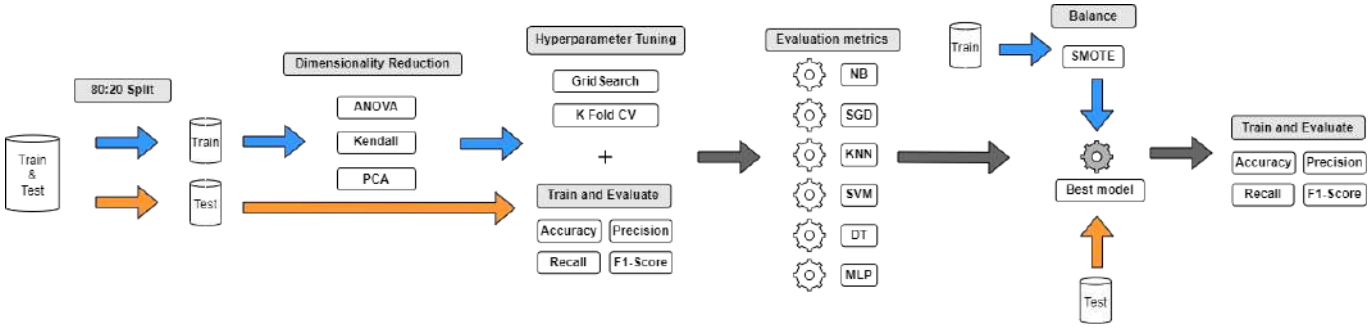


Figura 1. Sistema propuesto

para ANOVA. En este caso, una vez generadas las componentes principales y ordenadas según la varianza acumulada se probará a seleccionar desde 10 características hasta 30, resultando un total de 5 experimentos.

En este trabajo, tanto el dataset original como el preprocesado utilizado en los experimentos presenta un desbalanceo de categorías. Por ejemplo, el número de muestras para Shellcode o Worms es inferior al 1 % del total de muestras mientras que otras como Fuzzers o Exploits representan una proporción en torno al 15 % sobre el total. Para mitigar este problema, se aplica SMOTE para la generación de muestras sintéticas.

Para aplicar el balanceo, el número de muestras de cada categoría se ha calculado manualmente con el objetivo de no incrementar excesivamente el tamaño del nuevo conjunto de entrenamiento y aumentar el número de muestras de las clases minoritarias. La Tabla 10 muestra el número de muestras tras balancear el conjunto de entrenamiento multiplicando el número de tuplas original por un valor α, calculado tras varios experimentos.

V. Resultados y discusión

En esta sección se exponen los resultados para los experimentos realizados. En primer lugar, se detallan los resultados obtenidos tras la aplicación de distintas técnicas de reducción de dimensiones. Cada método de reducción de dimensionalidad se probará con distintos valores y, a su vez, se probarán con los algoritmos revisados anteriormente. En segundo lugar, aplicando los algoritmos que mejor resultado ofrecen tras la selección de características, se entrenan con un nuevo conjunto de datos balanceado y se evalúa su rendimiento para demostrar la influencia de los algoritmos de balanceo.

Las Tablas I, II, III muestran cómo afecta la selección de características en el rendimiento del modelo para clasificación binaria empleando ANOVA, correlación de Kendall y PCA, respectivamente.

En la clasificación binaria, tras la selección de características con ANOVA y basándose en la puntuación F1 los algoritmos que mejore resultado ofrecen son DT, DT, DT, MLP y MLP para 10, 15, 20, 25 y 30 componentes, respectivamente. Estos resultados se muestran en la Tabla I. Atendiendo al tiempo de entrenamiento, el algoritmo más eficiente es nuevamente Naïve Bayes. Los experimentos realizados demuestran que la mejor opción de reducción de dimensionalidad es reducir a 30 características donde el algoritmo MLP obtiene una evaluación de 0.9122, 0.9123, 0.9122, 0.9122 para accuracy, precision, recall y F1-Score.

Tabla I
Métricas de evaluación para clasificación binaria con ANOVA

	Algoritmo	Accuracy	Precision	Recall	F1-Score	Tiempo
10	Naïve Bayes	0.8181	0.8377	0.8181	0.8167	94 ms
	KNN	0.8530	0.8581	0.8530	0.8530	57.5 s
	SVM	0.8502	0.8825	0.8502	0.8482	19 min 51 s
	SGD	0.8337	0.8699	0.8337	0.8311	211 ms
	DT	0.8618	0.8635	0.8618	**0.8619**	415 ms
	MLP	0.8549	0.8653	0.8549	0.8545	33.6 s
15	Naïve Bayes	0.8007	0.8148	0.8007	0.7998	109 ms
	KNN	0.8700	0.8702	0.8700	0.8700	34.3 s
	SVM	0.8679	0.8710	0.8679	0.8680	8 min 43 s
	SGD	0.8324	0.8665	0.8324	0.8299	370 ms
	DT	0.8972	0.8981	0.8972	**0.8970**	926 ms
	MLP	0.8739	0.8751	0.8739	0.8740	44.2 s
20	Naïve Bayes	0.8138	0.8423	0.8138	0.8115	128 ms
	KNN	0.8770	0.8770	0.8770	0.8770	10.6 s
	SVM	0.8752	0.8764	0.8752	0.8753	7 min 58 s
	SGD	0.8506	0.8826	0.8506	0.8487	433 ms
	DT	0.8983	0.8991	0.8983	**0.8981**	829 ms
	MLP	0.8862	0.8861	0.8862	0.8861	54.2 s
25	Naïve Bayes	0.8176	0.8400	0.8176	0.8160	132 ms
	KNN	0.8989	0.8990	0.8989	0.8988	1 min 29 s
	SVM	0.9022	0.9030	0.9022	0.9023	7 min 43 s
	SGD	0.8600	0.8706	0.8600	0.8597	8.6 s
	DT	0.9079	0.9084	0.9079	0.9078	1.37 s
	MLP	0.9115	0.9118	0.9115	**0.9114**	1 min 39 s
30	Naïve Bayes	0.8029	0.8266	0.8029	0.8010	154 ms
	KNN	0.8877	0.8878	0.8877	0.8877	6 min 8 s
	SVM	0.9018	0.9028	0.9018	0.9019	9 min 23 s
	SGD	0.8567	0.8730	0.8567	0.8560	537 ms
	DT	0.9112	0.9122	0.9112	0.9110	2.72 s
	MLP	0.9122	0.9123	0.9122	**0.9122**	1 min 21 s

Continuando la presentación de resultados de la clasificación binaria, la Tabla II presenta la selección de características mediante el coeficiente de correlación de Kendall. Se han obtenido los mejores resultados para DT y MLP usando como coeficiente de correlación 0.7 y 0.8, respectivamente. Estos valores implican una selección de 22 y 31 características, para 0.7 y 0.8, respectivamente. Aunque Naïve Bayes es el algoritmo que menos tiempo de entrenamiento requiere, los resultados obtenidos quedan lejos de algoritmos como DT o SGD, con tiempos de entrenamiento también bajos. Los experimentos realizados demuestran que el valor 0.8 es el valor óptimo para la selección de características que, junto a MLP logran 0.9125, 0.9126, 0.9125, 0.9125 para accuracy, precision, recall y F1-Score, respectivamente.

El último grupo de experimentos realizado en la clasificación binaria tiene como objetivo evaluar el impacto de la extracción de características con PCA y se reflejan en la Tabla III. El algoritmo MLP es el que mejor resultados obtiene en todos los casos, a pesar de no ser el más eficiente. Los mejores resultados aplicando PCA se obtienen tras seleccionar 30 componentes y emplear MLP, alcanzando 0.9103, 0.9107,

Tabla II
MÉTRICAS DE EVALUACIÓN PARA CLASIFICACIÓN BINARIA CON CORRELACIÓN DE KENDALL

Coef. Corr.	Algoritmo	Accuracy	Precision	Recall	F1-Score	Tiempo
0.7	Naïve Bayes	0.7631	0.8137	0.7631	0.7564	128 ms
	KNN	0.8878	0.8880	0.8878	0.8879	1 min
	SVM	0.8905	0.8916	0.8905	0.8906	10 min 7 s
	SGD	0.8348	0.8613	0.8348	0.8330	330 ms
	DT	0.9065	0.9065	0.9065	**0.9064**	767 ms
	MLP	0.9048	0.9048	0.9048	0.9048	1 min 6 s
0.8	Naïve Bayes	0.7957	0.8240	0.7957	0.7931	269 ms
	KNN	0.8895	0.8895	0.8895	0.8895	55.7 s
	SVM	0.9039	0.9046	0.9039	0.9040	10 min 12 s
	SGD	0.8543	0.8575	0.8543	0.8543	1.8 s
	DT	0.9120	0.9122	0.9120	0.9119	1.3 s
	MLP	0.9125	0.9126	0.9125	**0.9125**	1 min 14 s

0.9103, 0.9102 para accuracy, precision, recall y F1-Score, respectivamente.

Tabla III
MÉTRICAS DE EVALUACIÓN PARA CLASIFICACIÓN BINARIA CON PCA

#	Algoritmo	Accuracy	Precision	Recall	F1-Score	Tiempo
10	Naïve Bayes	0.7577	0.7652	0.7577	0.7573	240 ms
	KNN	0.8755	0.8755	0.8755	0.8755	2.4 s
	SVM	0.8697	0.8785	0.8697	0.8695	7 min 15 s
	SGD	0.7989	0.8320	0.7989	0.7958	217 ms
	DT	0.8678	0.8685	0.8678	0.8678	1.6 s
	MLP	0.8887	0.8915	0.8887	**0.8888**	31.2 s
15	Naïve Bayes	0.6715	0.6918	0.6715	0.6565	120 ms
	KNN	0.8808	0.8808	0.8808	0.8808	8.9 s
	SVM	0.8840	0.8893	0.8840	0.8840	7 min 14 s
	SGD	0.8034	0.8398	0.8034	0.8000	344 ms
	DT	0.8726	0.8732	0.8726	0.8726	4.2 s
	MLP	0.8990	0.8993	0.8990	**0.8988**	39.1 s
20	Naïve Bayes	0.6353	0.6592	0.6353	0.6353	286 ms
	KNN	0.8830	0.8831	0.8830	0.8830	12 s
	SVM	0.8927	0.8951	0.8927	0.8928	7 min 38 s
	SGD	0.8197	0.8444	0.8197	0.8179	358 ms
	DT	0.8781	0.8784	0.8781	0.8781	4.4 s
	MLP	0.9047	0.9049	0.9047	**0.9046**	1 min 3 s
25	Naïve Bayes	0.6482	0.6774	0.6482	0.6245	197 ms
	KNN	0.8783	0.8786	0.8783	0.8784	30.1 s
	SVM	0.8958	0.8979	0.8958	0.8959	8 min 26 s
	SGD	0.8381	0.8568	0.8381	0.8371	723 ms
	DT	0.8785	0.8789	0.8785	0.8786	4.3 s
	MLP	0.9094	0.9094	0.9094	**0.9094**	1 min 14 s
30	Naïve Bayes	0.6958	0.7176	0.6958	0.6831	142 ms
	KNN	0.8793	0.8795	0.8793	0.8793	23.1 s
	SVM	0.8991	0.9005	0.8991	0.8991	9 min 27 s
	SGD	0.8446	0.8472	0.8446	0.8447	598 ms
	DT	0.8805	0.8806	0.8805	0.8805	5.1 s
	MLP	0.9103	0.9107	0.9103	**0.9102**	1 min 27 s

De forma general, se observa que a mayor número de características mejores son los resultados, a la vez que el tiempo de entrenamiento va incrementándose. Según los experimentos realizados, si el sistema que se construye debe detectar el mayor número de ataques la mejor opción es el método de correlación de Kendall, logrando un 0.9125 de F1-Score tras eliminar las características con un coeficiente de correlación superior al 0.8 (31 características).

A modo de discusión, la Figura 2 muestra una comparativa del F1-Score del mejor modelo según el número de características seleccionadas y para los tres métodos. Se ha optado por considerar esta métrica por ser una de las más representativas a la hora de evaluar en rendimiento de modelos clasificación. En el caso de la clasificación binaria los resultados óptimos se obtienen tras seleccionar 31 características mediante el coeficiente de correlación de Kendall que, junto al algoritmo MLP ofrece los mejores resultados, con un F1-Score de 0.9125. Sin embargo, el algoritmo MLP no es el más eficiente y, en caso de necesitar modelos más eficientes deberá considerarse el uso de DT para la clasificación binaria.

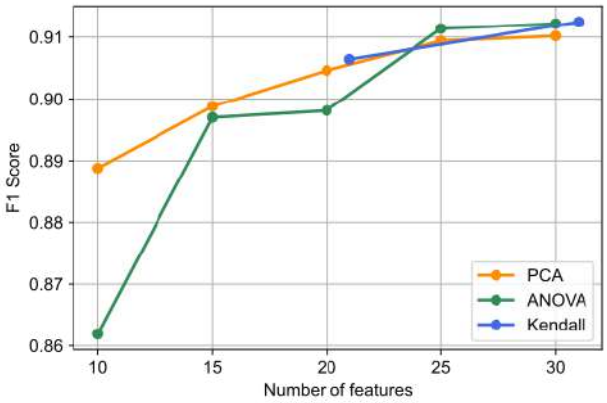


Figura 2. Comparativa de rendimiento según F1 por N.º de componentes y método

Una vez realizados los experimentos para la selección de características se procede a discutir los resultados obtenidos para el balanceo. Los métodos de balanceo se han aplicado para el algoritmo que mejor rendimiento ofrece para cada número de componentes. Los resultados se comparan con el rendimiento usando el conjunto de datos sin reducción de dimensiones y sin balancear (RAW) donde se obtiene 0.9122 para *accuracy*, *precision*, *recall* y F1.

Las Tablas IV, V, VI muestran la evaluación obtenida para clasificación binaria aplicando el algoritmo de balanceo SMOTE a diferentes conjuntos de datos.

La Tabla IV muestra los resultados de aplicar SMOTE tras haber reducido la dimensionalidad del conjunto de datos con la técnica ANOVA y efectuar una clasificación binaria entre comportamiento legítimo o de ataque. El algoritmo empleado en cada experimento es el que mejor resultados ofrece (Tabla I).

Tabla IV
COMPARATIVA ANOVA FRENTE ANOVA + SMOTE EN CLASIFICACIÓN BINARIA

N.º	Algoritmo	Accuracy	Precision	Recall	F1-Score
10	DT	0.8732	0.8829	0.8732	0.8730
15	DT	0.8880	0.8898	0.8880	0.8881
20	DT	0.9027	0.9028	0.9027	0.9027
25	MLP	0.9055	0.9067	0.9055	0.9055
30	MLP	0.9059	0.9073	0.9059	0.9059

La Figura 3 muestra la influencia de SMOTE al conjunto previamente reducido tras la selección con ANOVA. Los dos grupos de prueba se comparan con el rendimiento obtenido empleando todas las características del conjunto (línea roja discontinua). Aunque no se demuestra una clara mejora al balancear en conjunto, la selección de 30 características con mayor relevancia produce un modelo con rendimiento equiparable al uso de todas las características.

La Tabla V muestra los resultados de aplicar SMOTE tras haber efectuado una selección de características con Kendall y efectuar una clasificación binaria. El algoritmo empleado en cada experimento es el que mejor resultados ofrece (Tabla II) para cada coeficiente de correlación.

La Figura 4 muestra la influencia de SMOTE junto al

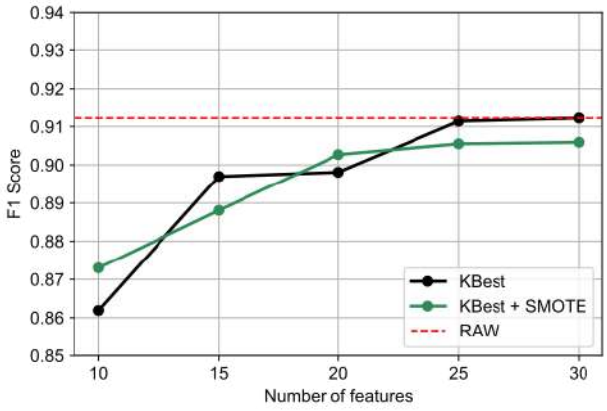


Figura 3. Comparativa ANOVA frente ANOVA + SMOTE en clasificación binaria

Tabla V
COMPARATIVA KENDALL FRENTE KENDALL + SMOTE EN CLASIFICACIÓN BINARIA

N.º	Algoritmo	Accuracy	Precision	Recall	F1-Score
22	DT	0.9062	0.9062	0.9062	0.9061
31	MLP	0.9043	0.9069	0.9043	0.9044

método de correlación Kendall. Similar al caso anterior, el rendimiento no mejora al balancear el conjunto de datos, e incluso disminuye la detección de ataques en el caso de usar mayor número de características. No obstante, la selección de 31 características tras la eliminación de las columnas con un coeficiente de correlación mayor a 0.8, produce un modelo con rendimiento equiparable al uso de todas las características.

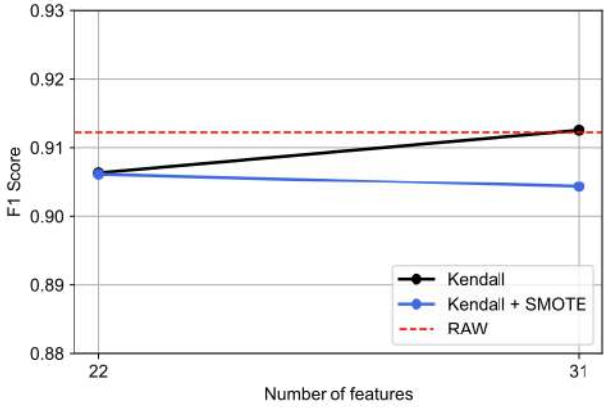


Figura 4. Comparativa Kendall frente Kendall + SMOTE en clasificación binaria

La Tabla VI muestra los resultados de aplicar SMOTE después de extraer las componentes principales con PCA y realizar la clasificación entre comportamiento legítimo o de ataque. El algoritmo empleado en todos los experimentos ha sido MLP por ser el que mejor resultados ofrece (Tabla III) para cualquier número de componentes.

La Figura 5 muestra la influencia de SMOTE junto PCA. Al igual que en el resto de los casos de clasificación binaria, el balanceo aplicado no repercute positivamente en la evaluación

Tabla VI
COMPARATIVA PCA FRENTE PCA + SMOTE EN CLASIFICACIÓN BINARIA

N.º	Algoritmo	Accuracy	Precision	Recall	F1-Score
10	MLP	0.8769	0.8883	0.8769	0.8766
15	MLP	0.8943	0.8981	0.8943	0.8943
20	MLP	0.8930	0.8987	0.8930	0.8930
25	MLP	0.9024	0.9047	0.9024	0.9025
30	MLP	0.9068	0.9077	0.9068	0.9069

final e incluso disminuye en comparación a la aplicación exclusiva de selección de características. Ninguna de las pruebas realizadas con PCA ha superado el rendimiento del modelo sin aplicar ninguna tarea de preprocesado (línea roja discontinua).

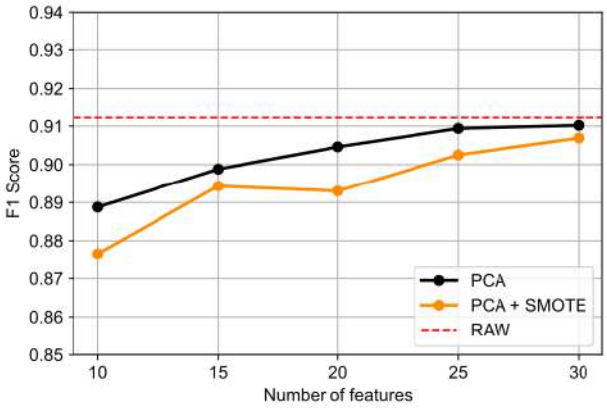


Figura 5. Comparativa PCA frente PCA + SMOTE en clasificación binaria

VI. CONCLUSIONES

En este trabajo se ha presentado un estudio exhaustivo del impacto de diferentes técnicas de preprocesado sobre un dataset de ciberseguridad, concretamente UNSW-NB15. En un primer grupo de experimentos se ha probado la influencia de tres técnicas de reducción de dimensiones cuyo uso está extendido en la literatura científica: ANOVA, correlación de Kendall y PCA. Una vez estudiado el método que mejor resultados ofrece, un segundo grupo de experimentos ha consistido en la aplicación del algoritmo de balanceo SMOTE.

En base a los resultados obtenidos, tras la aplicación de tres métodos de reducción de dimensiones se concluye que el método que mejores resultados obtiene para la detección de ataques es la selección de características basada en el método de correlación de Kendall. Además, respecto a la reducción de dimensionalidad en conjunción al algoritmo de balanceo SMOTE, este estudio ha evidenciado que la metodología seguida supone un disminución notable de la complejidad aunque ello conlleva una leve reducción en la capacidad de detección del modelo.

Como trabajos futuros se plantea el estudio de evaluación de técnicas de selección de características y balanceo en nuevos datasets orientados a la clasificación de tráfico de red. Además, se propone el uso de ténicas de reducción de dimensionalidad basadas en redes neuronales, autoencoders. Estas redes buscan convertir los datos a un espacio de características

de menor dimensión minimizando maximizando la información. Otra línea interesante sería el estudio de la influencia de otros métodos de sobremuestreo como variaciones de SMOTE o la generación sintética de muestras adaptativa (ADASYN).

Agradecimientos

Los autores agradecen la financiación recibida por (Fondo Europeo de Desarrollo Regional), Consejería de Economía, Ciencia y Agenda Digital, a través del proyecto GR21099.

Referencias

[1] P. Collela, "Ushering in a better connected future," Enero 2017, https://www.ericsson.com/en/about-us/company-facts/ericsson-worldwide/india/authored-articles/ushering-in-a-better-connected-future, Acceso: 20-04-2021.

[2] D. Paul, "Iot devices see more than 1.5bn cyberattacks so far this year," Septiembre 2021, https://www.digit.fyi/iot-security-kaspersky-research-attacks/, Acceso: 20-04-2021.

[3] M. Abomhara, G. M. Koien, and and, "Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks," *Journal of Cyber Security and Mobility*, vol. 4, no. 1, pp. 65–88, 2015. [Online]. Available: https://doi.org/10.13052/jcsm2245-1439.414

[4] A. Ponmalar and V. Dhanakoti, "An intrusion detection approach using ensemble support vector machine based chaos game optimization algorithm in big data platform," *Applied Soft Computing*, vol. 116, p. 108295, Feb. 2022. [Online]. Available: https://doi.org/10.1016/j.asoc.2021.108295

[5] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," *IEEE Access*, vol. 7, pp. 94 497–94 507, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2928048

[6] J. Kizza and F. M. Kizza, "Intrusion detection and prevention systems," in *Securing the Information Infrastructure*. IGI Global, 2008, pp. 239–258. [Online]. Available: https://doi.org/10.4018/978-1-59904-379-1.ch012

[7] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020. [Online]. Available: https://doi.org/10.1109/comst.2020.2988293

[8] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "TSDL: A two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30 373–30 385, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2899721

[9] T. Murovič and A. Trost, "Genetically optimized massively parallel binary neural networks for intrusion detection systems," *Computer Communications*, vol. 179, pp. 1–10, Nov. 2021. [Online]. Available: https://doi.org/10.1016/j.comcom.2021.07.015

[10] S. Jeon and H. K. Kim, "AutoVAS: An automated vulnerability analysis system with a deep learning approach," *Computers and Security*, vol. 106, p. 102308, Jul. 2021. [Online]. Available: https://doi.org/10.1016/j.cose.2021.102308

[11] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2895334

[12] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives," in *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*. IEEE, Oct. 2018. [Online]. Available: https://doi.org/10.1109/cccs.2018.8586840

[13] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE, Jul. 2009. [Online]. Available: https://doi.org/10.1109/cisda.2009.5356528

[14] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*. IEEE, Nov. 2015. [Online]. Available: https://doi.org/10.1109/milcis.2015.7348942

[15] P. Dhal and C. Azad, "A comprehensive survey on feature selection in the various fields of machine learning," *Applied Intelligence*, vol. 52, no. 4, pp. 4543–4581, Jul. 2021. [Online]. Available: https://doi.org/10.1007/s10489-021-02550-9

[16] S. M. Anouncia and U. K. Wiil, Eds., *Knowledge computing and its applications*. Singapore, Singapore: Springer, Feb. 2019.

[17] Y. Yang, K. Zheng, C. Wu, and Y. Yang, "Improving the classification effectiveness of intrusion detection by using improved conditional variational AutoEncoder and deep neural network," *Sensors*, vol. 19, no. 11, p. 2528, jun 2019. [Online]. Available: https://doi.org/10.3390%2Fs19112528

[18] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html

[19] A. Ali, S. M. Shamsuddin, and A. Ralescu, "Classification with class imbalance problem: A review," *International Journal of Advances in Soft Computing and its Applications*, vol. 7, no. 3, p. 176 – 204, 2015. [Online]. Available: http://home.ijasca.com/data/documents/13IJASCA-070301_Pg176-204_Classification-with-class-imbalance-problem_A-Review.pdf

[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, jun 2002.

[21] M. M. Nwe and K. T. Lynnr, "Effective resampling approach for skewed distribution on imbalanced data set," *IAENG International Journal of Computer Science*, vol. 47, no. 2, p. 234 – 249, 2020. [Online]. Available: http://www.iaeng.org/IJCS/issues_v47/issue_2/IJCS_47_2_12.pdf

[22] P. Li, M. Abdel-Aty, and J. Yuan, "Real-time crash risk prediction on arterials based on LSTM-CNN," *Accident Analysis and Prevention*, vol. 135, p. 105371, Feb. 2020. [Online]. Available: https://doi.org/10.1016/j.aap.2019.105371

[23] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018. [Online]. Available: https://doi.org/10.1109/tnnls.2017.2732482

[24] D. Dasgupta, Z. Akhtar, and S. Sen, "Machine learning in cybersecurity: a comprehensive survey," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 19, no. 1, pp. 57–106, Sep. 2020. [Online]. Available: https://doi.org/10.1177/1548512920951275

[25] S. Manimurugan, "IoT-fog-cloud model for anomaly detection using improved naïve bayes and principal component analysis," *Journal of Ambient Intelligence and Humanized Computing*, Jan. 2021. [Online]. Available: https://doi.org/10.1007/s12652-020-02723-3

[26] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *Journal of Big Data*, vol. 8, no. 1, Jan. 2021. [Online]. Available: https://doi.org/10.1186/s40537-020-00390-x

[27] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, Mar. 2019. [Online]. Available: https://doi.org/10.3390/electronics8030322

[28] N. Moustafa, "The UNSW-NB15 Dataset (Reduced)," 2015. [Online]. Available: https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDdEECo4ys

[29] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. Springer Berlin Heidelberg, 2003, pp. 986–996. [Online]. Available: https://doi.org/10.1007/978-3-540-39964-3_62

[30] H. Zhang, "The optimality of naive bayes," vol. 2, 01 2004.

[31] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, Jul. 1998. [Online]. Available: https://doi.org/10.1109/5254.708428

[32] M. Fuchs, "Introduction to sgd classifier," Noviembre 2019, https://michael-fuchs-python.netlify.app/2019/11/11/introduction-to-sgd-classifier/ Acceso: 20-04-2021.

[33] J. R. Quinlan, "Induction of decision trees," pp. 81–106, mar 1986. [Online]. Available: http://dx.doi.org/10.1007/bf00116251

[34] D. C. Montgomery and G. C. Runger, *Applied statistics and probability for engineers*. Nashville, TN: John Wiley & Sons, Oct. 2013.

[35] M. G. Kendall, "Rank correlation methods," *Journal of the Institute of Actuaries*, vol. 75, no. 1, p. 140–141, 1949.

[36] D. Bartholomew, "Principal components analysis," in *International Encyclopedia of Education*. Elsevier, 2010, pp. 374–377. [Online]. Available: https://doi.org/10.1016/b978-0-08-044894-7.01358-0

Provision of Security-as-a-Service (SecaaS) in Lightweight Scenarios

Antonio López Martínez, Mattia Zago, and Manuel Gil Pérez

Faculty of Computer Science, University of Murcia, 30100 Murcia, Spain
Email: antonio.lopez41@um.es, mattia.zago@um.es, mgilperez@um.es

***Abstract*—The Small and Medium Enterprises (SMEs) landscape is nowadays crippled by cybersecurity threats and criminal attacks. Despite being a critical asset of any country's economy, cybersecurity protection of the SMEs is often overlooked due to lack or both financial and human resources and knowledge in the field. To this end, this paper reports on the authors' development in the H2020-EU project called "Practical Autonomous Cyberhealth for resilient SMEs & Microenterprises" (PALANTIR). It intends to offer protection and security to SMEs and Micro Enterprises (MEs) with the implementation of multiple Security Capabilities (SCs), composed of Security-as-a-Service (SecaaS) solutions to protect them with monitoring and reacting duties. This paper provides a first approach of the design and implementation of the SCs designed to provide affordable SecaaS, taking into account the current status of the development.**

***Index Terms*—SecaaS, Security Capabilities, Monitoring and Remediation, SME/MEs**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

The new scenarios appeared in the recent years imply a real change in the form of implementing and protecting the security and data privacy of the organizations. In this task, the resources (assets, budget, worker education, etc.) available by such entities have a great importance, and unfortunately there is a remarkable inequality among all of them. Considering large enterprises, they usually have in their staff experts in cybersecurity and implement the most recent technologies to protect their assets and processes. Nonetheless, the Small and Medium Enterprises (SMEs) are the main targets of the cyber criminals, due to the lack of resources that these organizations have in protecting their organization and assets. Therefore, there is a critical need in the protection of the SMEs to address the current threats that they are suffering [1].

Advances in the technologies used to implement security and data privacy also are a challenge for SMEs, because they require specific skills not widely available. For instance, technologies like Artificial Intelligence, Blockchain, Virtualization, IoT, and Big Data, are some of the recent technologies gradually adopted by the enterprises in the current scene [2]. Focusing on SMEs, the virtualization technology can allow them to use fewer resources and efforts for reaching successful actions, thereby reducing capital expenditure (CAPEX) and maximising operational expenditures (OPEX).

On the other hand, a new tendency so-called Security-as-a-Service (SecaaS) is appearing in the recent years in order to provide affordable security knowing the needs and resources that possible targets of this protection method have [3]. In the SME context, different SecaaS solutions are available in the literature trying to improve and protect the environment of these critical organizations [4].

Building upon this scenario, the H2020-EU project called "Practical Autonomous Cyberhealth for resilient SMEs & Microenterprises" (PALANTIR) [5] aims to protect SMEs and Micro Enterprises (MEs) by applying Network Function Virtualization (NFV) technology to implement Security and Data Privacy as-a-Service. These secure services will be the main enabler in ensuring compliance with the Security Capabilities (SCs) necessary for the protection of any SME/ME. In this vein, this paper presents the design of the first collection of SCs delivered in PALANTIR, composed of specific SecaaS solutions for monitoring and reaction, as well as the proposed architecture and their implementation.

In addition, the PALANTIR project consists of other components: a Threat Intelligence component, with Machine and Deep Learning support to spot complex threats; a Risk Management Framework to perform risk assessments of SME/ME clients and provide more specific SCs; an SC Orchestrator to manage the SC lifecycle and actions available in the SC; and other components which are not as relevant to this paper. Further information regarding PALANTIR components and functionalities may be found at [5].

The rest of the paper is structured as follows. Section II includes the design and creation of the SCs architecture with the technologies, subcomponents, and interfaces defined. Section III explains the implementation details, showing the technology used, structure defined, and the specific code details to create an SC. And finally, Section IV summarises the content presented in this paper of work-in-progress and explains the next steps and open fronts.

II. Design and Architecture

Before defining SC, the NFV technology must be presented as one of its key enablers. This technology defines the Virtual Network Functions (VNFs) which are virtualised services deployed in computing platforms running in dedicated hardware technology. Typically, VNF is named for the use of Virtual Machines in their developing, and Container Network Funtions (CNFs) [6] for the use of container technology. This last can be understood as a subset of VNFs where the technology applied in the implementation and deployment is the containerisation. If both technologies want to be named, the xNF nomenclature is selected. This technology has achieved a widely use in the recent solutions that are being deployed in virtualization scenarios, thanks to the lightweight use of resources and the interoperability offered. The SC is defined as the implementation of xNF technology to provide SecaaS solutions with an optimised use of the limited resources and knowledge that clients of the PALANTIR platform will have.

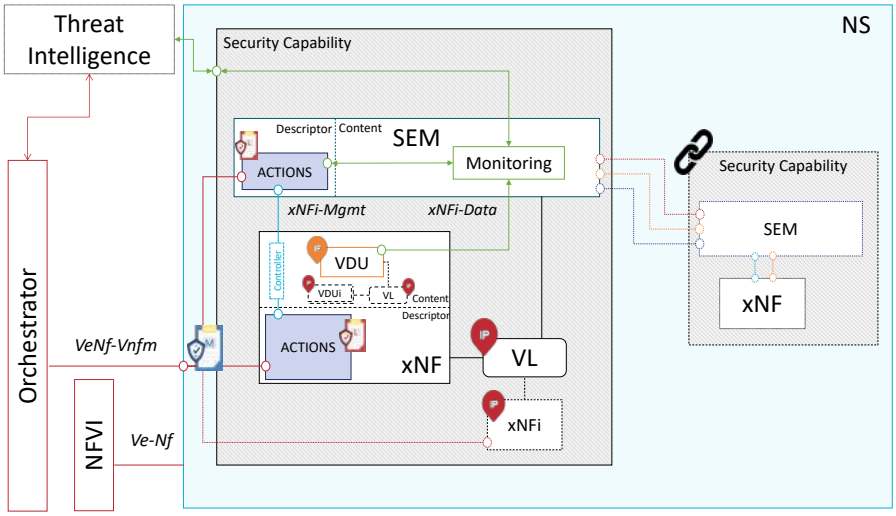


Figure 1. The architecture of a Security Capability.

The SC design and architecture is presented in Figure 1, where different subcomponents are presented:

- Virtual Deployment Unit (VDU): Allocated inside of the corresponding xNF, which includes the security service implemented in the form of a container. The VDU is the logic of the SC and defines the security character of the SC. One or more VDUs can be chained and connected by a given Virtual Link (VL), which manages the internal connectivity of the SC. Besides, the service can have implemented different actions that can be performed, such as reconfiguration, start commands, stop commands, etc., which shall be executed by the Orchestrator component belonging to the PALANTIR project.
- Security Element Manager (SEM) acts as an intermediary between the security service implemented and external components outside the SC. Mainly, the SEM exposes the data generated or collected in the services deployed in the SC. Furthermore, it is responsible for managing the SC lifecycle and security configurations through the communication with the Orchestrator component. Thanks to this component, the service configuration is abstracted for all external components that want to interact with the SC and manages the data privacy in an efficient way since certain data would be private in the SC context.
- The service implemented in a VDU contains the logic of a given security mechanism, such as a Firewall, an Intrusion Detection System (IDS), etc. The PALANTIR project pretends to offer the possibility of uploading new services created by other developers.
- Network Service (NS) is the most external subcomponent that the SC has in order to allocate all VDUs and the SEM inside. This can be understood as the wrapper of the SC. At the end, such NS is the element that finish to be instantiated by the Orchestrator.
- SC chain: Under the NS domain, more than one SC can be chained in order to provide a much more complete SecaaS solution with different characteristics given from each SC deployed. For example, a NS could contain a reacting SC with a monitoring SC allowing to join both characteristics in one SecaaS solution.

In addition, any SC exhibits two interfaces. First, it has a direct connection with the Orchestrator (*VeNf-Vnfm*), since this component is in charge of the lifecycle management of the SC. The Orchestrator has the responsibility for instantiating, managing, stopping, and deleting the SCs which are deployed. In addition, the Orchestrator is the input point to activate the actions available in the SCs so as to reconfigure and change the SecaaS services running on them. Besides, the SEM allows SC to transfer records and data collected with the services implemented, for instance, NetFlow records collected by a service or a log created by an IDS. This interface might be proven effective to detect possible threats which can be actuating in real time in the SME/ME infrastructure. In this case, the SEM does not need the connection with the Orchestrator because it can directly forward the information collected to the Threat Intelligence into the PALANTIR platform. The second interface (*Ve-Nf*) is the connection between the NFV Infrastructure (NFVI), where the SCs will be deployed.

Thanks to the virtualization technologies, the SCs might contain multiple security services, allowing the creation of a catalogue with different SC types that will be offered by the PALANTIR project. In this context, two large families are contemplated: the monitoring and reacting families. The monitoring family covers all security services which collect information and detect possible anomalous activities in the organization. The reacting family involves the mechanisms that stop and mitigate the threats previously detected, as well as the protection from non-produced attacks. Besides, each family can accommodate different types of SCs. In the first iteration of the PALANTIR project, the following types have been addressed to date:

- Intrusion Detection System (IDS): The IDS SC covers the threat detection techniques. There are two primary threat detection techniques: *signature-based* detection and *anomaly-based* detection. The former uses known characteristics of different cyber-attacks to recognise the threat produced, while the latter stores the normalised baseline and generates an alert when detecting an unknown behaviour or activity.
- Deep Packet Inspection (DPI): the DPI SC develops the

characteristics required to inspect IP packets. So, the DPI can be compared with an IDS since it performs traffic analysis and detects anomalous messages.

- Network NetFlow Sniffer (NNS): This SC can be implemented to support other components, both PALANTIR or external components. The NNS SC collects the traffic passing through it, converts it to NetFlow records, and sends it to an intermediate (e.g., Kafka) message broker. The components interested in such NetFlow records can obtain them through the specific broker channel.
- Firewall (FW): The FW SC is a key security service belonging to the reacting family, because traffic filtering turns out to be one of the most widely used security mechanisms in mitigating a threat. Besides, a FW SC offers integrated protection of the network environment through packet filtering, load balancing, and firewall rules created to avoid and mitigate malicious activities occurring to the organisation domain.

III. Implementation of Security Services

A complete review of the available solutions has been performed to implement the design and the architecture presented in Section II. Regarding this, the technology supported by the Orchestrator component is the key part of this researching, because this component mainly works with Virtual Machines and Containers. Between both, PALANTIR decides to use container technology to provide a lightweight solution. Therefore, Kubernetes (K8s) [7] is the selected technology for the deployment of SCs since the Orchestration component, which is implemented with Open Source MANO (OSM) software, only supports this technology.

Nevertheless, Kubernetes is the managing software for the containers, but the creation of the involved images inside of the containers are required. In this context, Docker [8] is the low-level technology used for the creation and implementation of the images that will be subsequently deployed in Kubernetes thanks to OSM. Docker provides the operating system images where it is possible to add the commands to customise and implement the specific SC image. The definition of these technologies allows developers to have a common base for the development of new SCs.

On the other hand, other components of PALANTIR can execute actions in the SCs in order to reconfigure the SC in execution time via:

- *day-0 actions*: they are understood as the tasks needed before instantiating the SC, for example, user/password creation, SSH keys establishment, etc.
- *day-1 actions*: are executed where the SC is already deployed. These actions are related to install packages, execute commands, etc.
- *day-2 actions*: These are referred to on-demand actions, such as real-time actions.

For the implementation of these actions, OSM presents two technology possibilities:

- Helm Charts [9]: It offers a way to package a collection of Kubernetes resources. Thanks to this, more complex Kubernetes deployments can be performed in a simple way, allowing OSM to abstract the low-level definitions that Kubernetes needs to create a container. Nonetheless, Helm Charts only permits the implementation of day-0 and day-1 actions.
- Juju Charms [10]: It implements a charmed operator framework to deploy, integrate, and manage Kubernetes containers and Virtual Machine (VM) natives applications, as well as the implementation of both day-0, day-1, and day-2 actions. This software needs a controller allocated in the Kubernetes cluster to deploy SCs and execute actions. OSM automatically creates the controller in a transparent way for the user. Besides, an operator (another container) is created along with the SC container to manage the actions that the developer has implemented in the definition of the charm for the SC.

```
FROM ubuntu:20.04

ENV DEBIAN_FRONTEND=noninteractive
RUN apt-get update -y && apt-get install python3 python3-pip wget -y
RUN apt-get install gcc openssl libssl-dev bison flex libdnet autoconf libtool -y
RUN apt-get install libpcap-dev -y
RUN apt-get install libnghttp2-dev -y
RUN apt-get install libdumbnet-dev -y
RUN apt-get install libpcre3-dev zlib1g-dev libluajit-5.1-dev -y

RUN wget https://www.snort.org/downloads/snort/daq-2.0.7.tar.gz
RUN wget https://www.snort.org/downloads/snort/snort-2.9.19.tar.gz

RUN tar xvzf daq-2.0.7.tar.gz
RUN tar xvzf snort-2.9.19.tar.gz

WORKDIR /daq-2.0.7
RUN autoreconf -f -i
RUN ./configure && make && make install

WORKDIR /snort-2.9.19
RUN ./configure --enable-sourcefire && make && make install

RUN ldconfig
RUN ln -s /usr/local/bin/snort /usr/sbin/snort
RUN mkdir -p /etc/snort/rules
RUN mkdir /var/log/snort
RUN mkdir /usr/local/lib/snort_dynamicrules
RUN touch /etc/snort/rules/white_list.rules
RUN touch /etc/snort/rules/black_list.rules
RUN touch /etc/snort/rules/local.rules
RUN cp etc/*.conf* /etc/snort
RUN cp etc/*.map /etc/snort

RUN alias python=python3

WORKDIR /opt/filebeat
RUN wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.13.3-amd64.deb
RUN dpkg -i filebeat-7.13.3-amd64.deb
COPY ./filebeat.yml /etc/filebeat/filebeat.yml
COPY ./portscanning.rules /etc/snort/rules/local.rules
RUN filebeat modules enable snort
```

Figure 2. The Snort Dockerfile.

After presenting the selection technologies, the showcase of an SC with the implementation of an IDS is presented in detail to see the different parts and elements needed to develop a new one. The full code presented here is publicly available in the official PALANTIR GitHub repository [11], which currently is in a work-in-progress state. Regarding SC creation, the first step is the creation of the docker image. In this case, the docker image corresponds to the preparation of the Snort IDS software [12]. Essentially, this step covers the development and testing of the Dockerfile with the commands required for the correct deployment and installation of the Snort IDS software in Ubuntu 20.04 operating system. In Figure 2, the content of the Snort IDS Dockerfile is presented.

The Dockerfile contains the commands needed to correctly install and configure the Snort IDS software. In addition, the Dockerfile includes the installation of the Filebeat software [13], the specific implementation of the SEM element (explained above), which is in charge of collecting the monitoring data generated in the SC. The SEM part involved in the communication with orchestration part is shown below.

In the current status, the docker image compiled from the Dockerfile and created in the local environment must be uploaded to the DockerHub [14], the official public repository

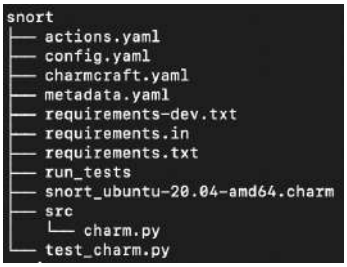


Figure 3. The Snort charm structure.

where many developers upload their images to be retrieved by the docker technology in the deployment moment, in order to be reachable in a next step when the charm is implemented. In a future state, the SC image will be allocated in a repository belonging to the PALANTIR infrastructure.

With the SC image created, the next step is the implementation of the Snort charm, the wrapper element that allows the docker image to be deployed into a Kubernetes cluster, as well as the development of different actions which can be executed into the SC image. The Snort IDS charm is composed by different files. In Figure 3, the structure of the Snort IDS charm is presented.

Inspecting the content presented in Figure 3, the files that can be highlighted are:

- *metadata.yaml*: The definition of the charm is indicated by the charm type (VM or K8s), the name, the creator, the description, the image resource (DockerHub repository), and the deployment type of Kubernetes (NodePort, LoadBalancer).
- *charmcraft.yaml*: This file is added in the recent versions of the Juju charm definition and includes the operating system base of the image (e.g., Ubuntu 20.04 LTS).
- *config.yaml*: The configuration parameters that the image needs in the instantiation time are defined in this file. For example, SSH keys used for remote connection, user and password of certain service required in the starting point.
- *actions.yaml*: This file defines the actions that can be triggered in the charm with the list of parameters, their format type (string, integer, boolean), the action name, and the description.
- *src/charm.py*: The "src" folder contains the key part of the charm, i.e., the code developed with the charm logic. The charm.py contains the different methods that define the charm, such as the K8s specification, action methods with the code needed to perform the action, etc.
- **.charm*: This file is the compression of the charm ready to be instantiated by the Orchestrator component. This file compresses all dependencies, libraries, and code that the charm needs to run in a correct form. It is the output of the command *charmcraft pack*.

With all content presented, the Snort IDS charm is ready for the instantiation and deployment in the K8s infrastructure. Executing the deployment, two containers are instantiated: one for the SC image and one with a Juju charm operator. The latter implements the communication with the orchestration part and the correct execution of the actions developed in the Snort IDS charm code, making up the second part of the SEM element together with the Filebeat software.

IV. Conclusion and Next Steps

In this paper, the criticality of the cybersecurity in the case of SMEs has been presented and the possible solution for the issues that such organizations suffer in this term. For that, the purpose, design, and implementation of the Security Capabilities defined in PALANTIR project have been presented. Thanks to this solution, different advantages are provided to SME/MEs, such as the affordable protection provided with a reduction of costs and resources, the lack of technical knowledge needed to implement the PALANTIR solution and the customisation of the solution offered (list of SCs) depending of each SME/ME requirements.

Different things are still pending to do and implement in the PALANTIR project. For instance, the SC catalogue needs to be extended with more types of SCs like a virtual Terminal Access Point to allow port mirroring, a Virtual Private Network to offer private communications for the SME clients, etc. Besides, the SCs must be attested in order to manage the integrity of such SC and detect anomalous behaviours inside it. Therefore, different mechanisms to provide a correct attestation of SCs will be implemented, such as the state of binary files and the checking of the services correctly executed into the SC. With regard to the IDS SC, it generates alerts when a cyber-attack is detected. These alerts should be incorporated into the Threat Intelligence component, which contains Machine and Deep Learning mechanisms to detect cyber-attacks and such alerts could help to provide a further stronger detection.

Acknowledgment

This work has been supported by the Cátedra SAES-Ciberseguridad and the European Commission Horizon 2020 Programme under grant agreement number H2020-SU-DS-2019/883335 - PALANTIR (Practical Autonomous Cyberhealth for resilient SMEs & Microenterprises).

References

[1] M. van Haastrecht, I. Sarhan, A. Shojaifar, L. Baumgartner, W. Mallouli, and M. Spruit, "A threat-based cybersecurity risk assessment approach addressing SME needs," in *16th International Conference on Availability, Reliability and Security*, 2021, pp. 1–12.
[2] C. Bai, P. Dallasega, G. Orzes, and J. Sarkis, "Industry 4.0 technologies assessment: A sustainability perspective," *International Journal of Production Economics*, vol. 229, p. 107776, 2020.
[3] W. Wang and S. Yongchareon, "A survey on security as a service," in *International Conference on Web Information Systems Engineering*, 2017, pp. 303–310.
[4] D. L. Nazareth, J. Choi, and T. L. Ngo-Ye, "The security-as-a-service market for small and medium enterprises," *Journal of Computer Information Systems*, pp. 1–11, 2021.
[5] The H2020-EU PALANTIR project, https://www.palantir-project.eu, 2022.
[6] R. Cziva and D. P. Pezaros, "Container network functions: Bringing nfv to the network edge," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 24–31, 2017.
[7] Kubernetes (K8s), https://kubernetes.io, 2022.
[8] Docker technology, https://www.docker.com, 2022.
[9] Helm Charts, https://helm.sh/docs/topics/charts, 2022.
[10] Juju Charms, https://juju.is, 2022.
[11] Official GitHub PALANTIR SCs repository, https://github.com/palantir-h2020/sc-secaas, 2022.
[12] Snort IDS, https://www.snort.org, 2022.
[13] FileBeat (SEM), https://www.elastic.co/es/beats/filebeat, 2022.
[14] DockerHub, https://hub.docker.com, 2022.

Benchmarking Ethereum security tools against major smart contract pitfalls and errors

Sergio Anguita Lorenzo
TECNALIA, Basque Research and Technology Alliance (BRTA)
Parque Científico y Tecnológico de Bizkaia
Astondo Bidea, Edificio 700
Derio, Spain
sergio.anguita@tecnalia.com

Aitor Gomez-Goiri
TECNALIA, Basque Research and Technology Alliance (BRTA)
Parque Tecnológico de Álava
Albert Einstein, 28
Vitoria-Gasteiz, Spain
aitor.gomez@tecnalia.com

José Miguel-Alonso
Department of Computer Architecture and Technology
University of the Basque Country (UPV/EHU), 20018
Donostia-San Sebastián, Spain
j.miguel@ehu.eus

***Abstract*—Ethereum is a Blockchain network where smart contracts are deployed and cannot be changed afterwards. This immutability forces developers to ensure correctness and security in their software before submission. Due to Ethereum's popularity and unique features, bugs in these contracts might result in big economic losses making their detection vital for the community.**

This paper reviews the major design and security flaws that have occurred throughout the history of Ethereum and analyzes the entry points and attack vectors used by cybercriminals to exploit existing flaws in smart contracts. In addition, whether existing tools could detect, at scale, such code vulnerabilities in exploited contracts measuring their reliability, accuracy and performance is evaluated.

Furthermore, we evaluate whether existing tools can detect such vulnerabilities in well-known exploited contracts, measuring their reliability, detection accuracy and performance.

Finally, we conclude that analyzed tools work with reasonable accurate results on smart contract deployed years ago. However they fail handling new Solidity language features, on complex logic, and with contracts created with latest Solidity compiler version.

***Index Terms*—Ethereum, Smart Contract, Security analysis**

I. Introduction

Ethereum network[1] has become an attractive target for cyberattacks thanks to its popularity and ability to execute arbitrary code. This code, also known as SC, cannot be modified after its deployment in the network (i.e., it's immutable). This makes bugs impossible to fix.

Bug detections and reliable code design is such a problem that in 2021, cybercriminals stole $3.2 billion worth in crypto assets[2]. In 2022 Q1 the total number of thefts and value stolen from exchanges and smart contracts peaked $1.3 billions. These bugs have a big economic impact since many of the decentralized applications (Dapps) build around these SCs involve payments between users.

For example, back in April 2016, attackers exploited a flaw categorized as a reentrancy attack in the smart contract *the DAO*, which resulted in the first ever largest ETH loss in history, with a final amount of 3.6M ETH.

As a consequence, it is necessary to detect malicious contracts and their bugs as early as possible. Such errors in the smart contracts can be detected by multiple frameworks and tools [3], but their accuracy and performance is limited.

The biggest challenge is to create accurate and precise analyzers with the ability to detect existing vulnerabilities in a turing-complete programming language and context. To address this situation, researchers tend to rely on well-known techniques such as fuzzing, symbolic execution, taint analysis or formal verification, among others. We make the following contributions in this paper.

- To design a systematic approach for evaluating EVM contract security tools against provided dataset.
- To provide an analysis of the accuracy of EVM security tools against real world exploited contracts.

The rest of the paper is organized as follows: first Section II provides a basic background to EVM and SC design. Section III explains the experimentation methodology and Section IV summarizes the results of such experimentation. Finally the Section V outlines the findings and Section VI enumerates the future work.

II. Background

A. Ethereum Virtual Machine

A smart contract is a program that runs on an Ethereum virtual machine (EVM) and implements business logic and rules. It is usually designed using a high level language, such as Solidity or Vyper [4], [5]. The high level design is then compiled into a set of low-level machine instructions. Each of the instructions is identified by a defined `OPCODE` and optional data parameter. At EVM runtime, each instruction is loaded, decoded and executed in a sandboxed environment. The sandboxed part of the EVM is a `1024` item size stack-based virtual machine. Those items can be SC function's input parameters, intermediate variables, return results and the result of any intermediate operations.

For instance, when executing the `SUB` opcode to subtract two operands, EVM will pop two values from the stack, subtract them together and then push the result back to the stack. All these `POP` and `PUSH` operations occur as part of the `SUB` instruction automatically.

Besides the stack, there are other four types of data sections in EVM: the state storage, the volatile memory, the input data, and the return value. The Blockchain state storage is a persistent key-value store that is used to maintain state variables of a smart contract in the ledger during the contract lifetime. For instance, a `ERC20` token contract uses the storage to keep a record of all balances.

In contrast, the memory, input data and return fields are used to store temporary and in-transit data such as the invoked function name, parameters, local variables, and return values. The values of these data locations are lost once their execution is finished and following EVM calls will use a new sandbox with an empty memory slot.

B. Software design defects and attack vectors

In Ethereum, SC vulnerabilities can be classified as Blockchain, Solidity or software issues. In this section, we summarize most critical issues affecting smart contracts.

Public access to private data: when a contract declares a field as private, it does not mean the content of the variable is private for the users, but only for the SC internal logic. Any attacker could inspect contract transactions and figure out the actual value of private variables. A common misconception is to think that private modifiers add privacy over attributes.

Gas related Denial of Service: the gas is the unit that measures the amount of computational effort required to execute specific operations in the EVM. All transactions require some gas value that depends on the smart contract execution logic. This value is paid by the account who invokes the transaction. *Out-of-Gas* issues may arise when the smart contract has defined an unbounded array on its state and a function that loops across the array's values is called. This may lead to a possible block gas limit exhaustion reverting the transaction.

Invalid constructor definition: constructors perform initialization tasks during the deployment of a SC. For instance, they often set the ownership of the contract to the deployer account, so it can perform privileged operations afterwards. Prior to Solidity *0.4.21*, the constructor name and the contract name were required to equal. Some developers fail to check this requirement before deploying the contract making the intended constructor work as a public function. In this situation any attacker can call the function which was supposed to be the constructor possibly gaining ownership privileges.

Insufficient entropy for random values: game or gambling related smart contracts tend to rely on pseudo random number generators to pick winners or to use it as input data for other purposes. Since smart contracts are public on the Blockchain, having a source of randomness is non-trivial, and hence, any seed used for randomness would be publicly known.

Timestamp manipulation: relying on the timestamp property as a random value is not recommended because miners, peers who are in charge of creating blocks, can arbitrarily manipulate their own local timestamp to some extent and affect to smart contract business logic.

Transaction ordering issues: when multiple users call the SC function at the same time, the pending transactions will wait in the pool until execution by the miner. The miner then, will choose those with higher gas, and hence, the execution order will be prioritized depending on the gas amount provided by the users to the miner in the transaction.

Improper exception handling: a contract calling other contracts or transferring ETH, must always check the return value to avoid unhandled errors or frozen funds.

Arithmetic errors: integer overflows and underflows may occur if no proper handling mechanisms are design. A relevant case in token related smart contracts for proper account balance management. For instance, trying to store 2^8 in a `uint8` type, would actually store a `0` because the value overflows and wraps around.

Unprotected self-destruct: this access control vulnerability allows an attacker to gain access to the smart contract and to self-destruct it, deleting all stored information and the contract bytecode (i.e., the compiled source code).

Re-entrancy: it occurs when calling an external contract allows an attacker to take control over the execution flow by recursively calling back into the contract before the initial call's appropriate state changes occur.

Flash loan attack: it is an abuse of the smart contract design in which an attacker borrows non collateral required funds from a third party and then manipulates a crypto asset (i.e, a token) spot price, rapidly increasing its value, to withdraw as many tokens as possible.

Rug-pull attack: it is a social attack used when the developer abandons the project and "runs away" with the investor's capital, usually thanks to backdoor functions included in the smart contract.

Oracle manipulation: An oracle is bridge between off-chain services and on-chain protocols used for information forwarding. Unprotected designs can lead to market spot price manipulation, data tampering and other price manipulation attacks.

C. Vulnerability analysis tools

This section introduces most relevant works and tools used to detect software vulnerabilities during secure software development lifecycle for EVM based smart contracts. These tools will be benchmarked in the evaluation.

DSol-decompiler[6] translates the raw EVM bytecode of a SC into a high-level language that resembles Solidity.

Ethersolve[7] extracts the control flow graph (CFG) from the bytecode of Ethereum contracts. The generated CFG is a directed graph which represents the execution flow. This outcome can be used later to apply more detection techniques in order to find reentrancy vulnerabilities and other security issues.

Evmdis[8] is a EVM disassembler which performs static analysis on the bytecode to provide a higher abstraction level than raw EVM operations. It separates bytecode into basic blocks, executes jump target analysis and assigns labels to targeted jumps for improved readability.

MAIAN[9] is a tool that uses systematic techniques to find contracts that violate specific properties of traces either safety violations or liveness violations.

Manticore[10] is an open-source dynamic symbolic execution framework with a flexible architecture and a satisfiability modulo theories (SMT) module.

Mythril[11] is a security analysis tool for EVM bytecode. It supports SCs built for Ethereum, Hedera, Quorum, Vechain, Roostock, Tron and other EVM-compatible Blockchains. It uses symbolic execution, SMT solving and taint analysis to detect a variety of security vulnerabilities in combination with other tools and techniques.

Oyente[12] Oyente is a symbolic execution tool that finds a set of predefined potential security bugs in contracts.

Osiris[13] is a tool specialized in the detection of integer overflow and underflow bugs based on Oyente.

Panoramix[14] is a Python based decompiler that allows the reconstruction, to some extent, of EVM bytecode to pseudo-code.

Rattle[15] is an EVM binary static analysis framework designed to work on deployed smart contracts. It takes EVM byte strings, uses a flow-sensitive analysis to recover the original control flow graph, lifts the control flow graph into an SSA/infinite register form, and optimizes the SSA – removing DUPs, SWAPs, PUSHs, and POPs.

Securify2[16][17] is a scanner supported by the Ethereum Foundation and ChainSecurity. It claims to support 37 vulnerabilities and implements datalog based static analysis.

Slither[18][19] is a Solidity static analysis framework written in Python 3. It runs a suite of vulnerability detectors, prints visual information about contract details, and provides an API to easily write custom analyses.

Smartcheck[20] is an extensible static analysis tool for discovering vulnerabilities and other code issues in SCs written in the Solidity programming language.

Vandal[21][22] is a static program analysis framework. It decompiles an EVM bytecode program to an equivalent intermediate representation that encodes the program's control flow graph. This representation removes all stack operations, thereby exposing data dependencies that are otherwise obscured. This information together with a Datalog specification is then fed into Souffle[23], a logic defined static analysis engine, for the extraction of program properties.

Tool name	Version/Date	Commit	Author
DSol-decompiler	Aug 2, 2018	`773e8f1`	tehybel
Ethersolve	Nov 2, 2021	`0840e9d`	SeUniVr
Evmdis	Mar 23, 2018	`0d14069`	Arachnid
MAIAN	Mar 19, 2018	`ab387e1`	ivicanikolicsg
Manticore	0.3.7	`9ed66b6`	trailofbits
Mythril	v0.22.1	`e4bbf70`	ConsenSys
Osiris	v0.0.1	`9079ab4`	christoftorres
Oyente	v0.2.7	`3d23264`	enzymefinance
Panoramix	May 15, 2020	`0e52ba4`	eveem-org
Rattle	Apr 21, 2020	`a3fa9c7`	Crytic
Securify2	Sep 5, 2021	`def1e30`	eth-sri
Slither	v0.8.3	`f962d6c`	Crytic
Smartcheck	v2.0	`4e2070a`	SmartDec
Vandal	Jul 29, 2020	`d2b0043`	usyd-blockchain

Table I
VERSIONS OF THE TOOLS ANALYSED.

III. METHODOLOGY

The benchmarking of Solidity and EVM security analyzers is based on the tools described in Table I. These tools have been individually sandboxed in Docker containers. Then, we measure execution time, CPU, RAM and disk usage during the analysis to detect any possible resource exhaustion. The evaluation process is described as follows:

1) **Tool sandboxing**: we compiled and build docker images from existing Dockerfiles when available. When no Dockerfile is provided, a custom configuration is created and built for the evaluation step.
2) **Dataset preparation**: input data needed by each tool in form of either bytecode or source code is prepared. A list of known vulnerable smart contracts[24][25] deployed on Ethereum Mainnet and other networks is collected. Then a code exploit attack vector suitable candidate's are selected for evaluation. Vulnerable contract bytecode is fetch from Etherscan[26] and Solidity source code if available. This data, is then prepared for each tool as part of a manual process to satisfy toolkit input conditions. In some cases, a manual process of EVM bytecode structure is needed, as shown in Figure 1 to remove `0x` prefix. In other cases, the process requires to split `.constructor` bytecode from `.runtime` bytecode and remove the `.metadata` and `.parameters` blocks.

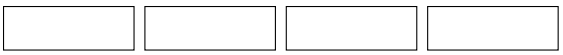


Figure 1. Structure of different sections of EVM bytecode.

3) **Collect execution results**: the execution of each tool reports valuable information, usually to system `stdout`, that needs to be collected and analyzed manually to check if the exploited vulnerability is detected or not. We also include whether a resource starvation error occurred during execution or not.

	A	B	C	D	E	F	G
DSol	*126	*10.6	-	0.0012	*20.4	*66	*72
Ethersolve	0.3	0.4	0.4	0.3	0.7	-	-
Evmdis	-	-	*0.045	*0.02	*0.018	*0.106	*0.114
Osiris	-	51.2	42.2	-	51.3	-	-
Oyente	-	3.7	3.4	-	23.4	-	-
Panoramix	*0.7	*5.2	-	-	-	*41.0	*39.6
Rattle	-	*3.7	*4.3	*6.5	*6.6	-	
Securify2	-	-	-	-	-	-	-
Slither	-	*0.6	*0.7	*2.0	0.0013	*0.9	*0.9
Smartcheck	3.8	2.5	2.4	4.4	3.2	3.1	3.0
Vandal	*3.6	*2.7	*1.5	*2.5	*6.8	*9.1	*9.6

	H	I	J	K	L	M	N
DSol	-	-	-	-	-	-	-
Ethersolve	0.4	13.6	20.4	0.5	2.3	73	0.2
Evmdis	0.048	-	-	-	-	-	0.039
Osiris	*51.2	-	-	-	-	-	-
Oyente	*5.5	-	-	-	-	-	-
Panoramix	-	-	*52	*1.2	*35.2	*486	*16.2
Rattle	-	-	-	-	-	-	*6.4
Securify2	-	-	-	-	-	-	-
Slither	-	-	-	-	2.2	144	-
Smartcheck	2.4	-	-	-	4.1	6.6	3.4
Vandal	*2.2	*25	*36.7	*8.2	*8.1	*28.1	-

Table II
PERFORMANCE AND ACCURACY RESULTS ON SMART CONTRACT DATASET

IV. EVALUATION

This section, presents our evaluation results. Table II shows the outcome and execution time, measured in seconds, of each of the evaluated tools against the testing dataset whereas Table III, details the real-world contracts used to represent each of the errors analyzed. Output of the analysis includes total execution time required by each tool per SC and whether the tool detected the expected error or not.

A good reliability and successful outcome is indicated with symbol **'*'**, meaning that expected bug was found. Compilation and analysis failures or unexpected outputs are flagged with the symbol **'-'**. For instance, `'*28.1'` indicates a execution time was of 28.1 seconds and successful analysis.

Many tools are prone to fail on SC compiled with latest version of Solidity compiler, as shown in analyzed DeFi (Decentralized Finance) related projects and contracts created after 2020. The output of our analysis also shows that MAIAN,

ID	Victim	Amount	Date	Type	Attack Vector	Contract Address
B	King of Ether Throne	101 ETH	Feb 2016	Game	Unchecked-send	0x2464d1d97f8d0180cfad67bdb19bc30cca69dda0
C	Rubixi	108 ETH	March 2016	Ponzi	Bad Constructor	0xe82719202e5965Cf5D9B6673B7503a3b92DE20be
D	SpankChain	165 ETH	March 2016	Token	Reentrancy	0xf91546835f756da0c10cfa0cda95b15577b84aa7
A	The DAO	$ 50M	June 2016	DAO	Reentrancy	0xbb9bc244d798123fde783fcc1c72d3bb8c189413
H	GovernMental	1.1K ETH	June 2016	Ponzi	Call Stack	0xf45717552f12ef7cb65e95476f217ea008167ae3
F	Parity Wallet	$ 30M	Jul. 2017	Wallet	Visibility	0xa657491c1e7f16adb39b9b60e87bbb8d93988bc3
G	Parity Deletage Call	$ 150M	Nov. 2017	Wallet	Delegate Call	0xa657491c1e7f16adb39b9b60e87bbb8d93988bc3
E	POWH	866 ETH	Feb 2018	Ponzi	Reentrancy	0xa7ca36f7273d4d38fc2aec5a454c497f86728a7a
M	Compound Finance	$ 90M	April 2021	DeFi	Flash loan	0x374ABb8cE19A73f2c4EFAd642bda76c797f19233
I	Poly Network	$ 611M	Aug. 2021	DeFi	Unchecked-send	0x838bf9e95cb12dd76a54c9f9d2e3082eaf928270
N	Grim Finance	$ 30M	Dec. 2021	DeFi	Reentrancy	0xFDc10560bd833B763352C481f5785Dd69C803429
J	Wormhole	$ 320M	Feb. 2022	DeFi	Input Validation	0x99309d2e7265528dc7c3067004cc4a90d37b7cc3
K	Ola Finance	$ 4.67M	March 2022	DeFi	Reentrancy	0x632942c9BeF1a1127353E1b99e817651e2390CFF
L	Uranium Finance	$ 50M	March 2022	DeFi	Design error	0xa08c4571b395f81fbd3755d44eaf9a25c9399a4a

Table III
ETHEREUM CONTRACT DATASET USED FOR BENCHMARKING.

Manticore and Mythril are not candidates for implementing smart contract analysis at scale, given the high demanding execution times for a single smart contract; creating big delays and waiting times in massive data analysis scenarios.

V. CONCLUSIONS

The resources and time required to analyze a single smart contract, ranges from seconds to minutes and hours, depending on selected tool. This is a considerable drawback which makes challenging to implement security measures on a massive scale. The time required per analysis makes difficult to perform real-time security evaluations for the contracts that are already deployed in the network. In addition to this, the low accuracy level reported by the tools and complexity of output data, requires to have deep expertise on Solidity, transaction flow and EVM design to detect and fix true positive bugs. However, a generalized use by developers during development cycle helps to avoid, to a large extent, possible failures that can exist, reducing the vulnerabilities and errors.

VI. FUTURE WORK

As for the future work, we plan to extend this analysis with more tests, tools and experiments that have been left out due to high computing requirements. With bigger dataset and more tools, a detailed analysis of failure causes can clarify and distinguish between *Out-of-Memory* errors, tool crashing or bad analysis results. This will help us to review where the detection algorithms are prone to fail and propose new techniques or alternative implementations.

REFERENCES

[1] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[2] chainalysis, "Hackers are stealing more cryptocurrency from defi platforms than ever before," https://blog.chainalysis.com/reports/2022-defi-hacks/, 2022, [Online; accessed June 6, 2022].

[3] A. Ghaleb and K. Pattabiraman, "How effective are smart contract analysis tools? evaluating smart contract static analysis tools using bug injection," in *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2020, pp. 415–427.

[4] R. Sierra, M. Eilers, and P. Müller, "Verification of ethereum smart contracts written in vyper," Ph.D. dissertation, Master's thesis, 2019.

[5] M. Kaleem, A. Mavridou, and A. Laszka, "Vyper: A security comparison with solidity based on common vulnerabilities," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. IEEE, 2020, pp. 107–111.

[6] T. Hybel, "Dsol-decompiler," https://github.com/tehybel/DSol-decompiler, 2018, [Online; accessed June 6, 2022].

[7] F. Contro, M. Crosara, M. Ceccato, and M. Dalla Preda, "Ethersolve: Computing an accurate control-flow graph from ethereum bytecode," in *2021 IEEE/ACM 29th International Conference on Program Comprehension (ICPC)*. IEEE, 2021, pp. 127–137.

[8] N. Jhonson, "evmdis," https://github.com/Arachnid/evmdis, 2018, [Online; accessed June 6, 2022].

[9] I. Nikolić, A. Kolluri, I. Sergey, P. Saxena, and A. Hobor, "Finding the greedy, prodigal, and suicidal contracts at scale," in *Proceedings of the 34th annual computer security applications conference*, 2018, pp. 653–663.

[10] M. Mossberg, F. Manzano, E. Hennenfent, A. Groce, G. Grieco, J. Feist, T. Brunson, and A. Dinaburg, "Manticore: A user-friendly symbolic execution framework for binaries and smart contracts," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2019, pp. 1186–1189.

[11] ConsenSys, "Mythril: a security analysis tool for evm bytecode," https://github.com/ConsenSys/mythril, 2022, [Online; accessed June 6, 2022].

[12] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 254–269.

[13] C. F. Torres, J. Schütte, and R. State, "Osiris: Hunting for integer bugs in ethereum smart contracts," in *Proceedings of the 34th Annual Computer Security Applications Conference*, 2018, pp. 664–676.

[14] Eveem.org, "Panoramix: the decompiler at the heart of eveem.org," https://github.com/eveem-org/panoramix, 2020, [Online; accessed June 6, 2022].

[15] Crytic, "Rattle: a evm binary static analysis tool," https://github.com/crytic/rattle, 2020, [Online; accessed June 6, 2022].

[16] E. Z. SRI Lab, "Securify 2.0: a security scanner for ethereum," https://github.com/eth-sri/securify2, 2020, [Online; accessed June 6, 2022].

[17] P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Buenzli, and M. Vechev, "Securify: Practical security analysis of smart contracts," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 67–82.

[18] Crytic, "Slither: a static analyzer for solidity," https://github.com/crytic/slither, 2022, [Online; accessed June 6, 2022].

[19] J. Feist, G. Grieco, and A. Groce, "Slither: a static analysis framework for smart contracts," in *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*. IEEE, 2019, pp. 8–15.

[20] S. Tikhomirov, E. Voskresenskaya, I. Ivanitskiy, R. Takhaviev, E. Marchenko, and Y. Alexandrov, "Smartcheck: Static analysis of ethereum smart contracts," in *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, 2018, pp. 9–16.

[21] ConsenSys, "Vandal: a static program analysis framework for ethereum smart contract bytecode," https://github.com/usyd-blockchain/vandal, 2018, [Online; accessed June 6, 2022].

[22] L. Brent, A. Jurisevic, M. Kong, E. Liu, F. Gauthier, V. Gramoli, R. Holz, and B. Scholz, "Vandal: A scalable security analysis framework for smart contracts," *arXiv preprint arXiv:1809.03981*, 2018.

[23] H. Jordan, B. Scholz, and P. Subotić, "Soufflé: On synthesis of program analyzers," in *International Conference on Computer Aided Verification*. Springer, 2016, pp. 422–430.

[24] chainalysis, "Hackenproof vulnerability database," https://hackenproof.com/vulnerabilities/, 2022, [Online; accessed June 6, 2022].

[25] rekt, "Rekt leaderboard," https://rekt.news/leaderboard//, 2022, [Online; accessed June 6, 2022].

[26] Etherscan, "Etherscan: Ethereum (eth) blockchain explorer," https://etherscan.io, 2018, [Online; accessed June 6, 2022].

Experimentación de un ataque sobre un sistema de comunicaciones cuánticas seguras

Alejandra Ruiz
TECNALIA
TECNALIA, Basque Research and Technology Alliance (BRTA)
Derio, España
alejandra.ruiz@tecnalia.com

Angel Rego
TECNALIA
TECNALIA, Basque Research and Technology Alliance (BRTA)
Derio, España
angel.rego@tecnalia.com

Resumen- **La llegada de las tecnologías cuánticas está impactando en la seguridad de las comunicaciones tal y como las conocemos, dada la potencial capacidad de los mismos para romper las claves cuánticas; se busca por ello, una evolución de las mismas de cara a asegurar la detección de intrusos en las comunicaciones. El paradigma *Quantum Key Distribution* (QKD) se perfila como un candidato que ofrece ciertas garantías ante este problema y está alcanzando un nivel de madurez como para ser introducido en sistemas operativos. Esta tecnología protocolo se aprovecha de la naturaleza intrínseca de la mecánica cuántica para detectar posibles intrusiones. En este artículo se describe un sistema comercial desplegado que implementa el protocolo QKD sobre una red de comunicaciones de laboratorio. Se describe una experimentación empírica realizada para analizar su comportamiento tanto en condiciones normales como ante un ataque. Las pruebas realizadas sobre el sistema confirman la detección de intrusos en las comunicaciones y muestran la reacción del sistema ante distintas duraciones de ataque.**

***Index Terms*- QKD, ciberseguridad, comunicaciones, cuántica, quantum, ataque**

Tipo de contribución: *Investigación en desarrollo*

I. INTRODUCCIÓN

Las tecnologías cuánticas representan el inicio de una nueva revolución industrial. A medida que estas alcanzan un nivel de madurez, la seguridad de nuestras comunicaciones tradicionales se ven amenazadas. La irrupción de los ordenadores cuánticos, acercan cada día el momento en que nuestras comunicaciones cifradas ya no sean fiables. Las comunicaciones cuánticas seguras, utilizan la fotónica como medio más habitual para transmitir y procesar la información. Tal y como explica Flamini et al. [2] la información cuántica basada en fotones se puede codificar utilizando los grados de libertad de luz:

- dirección de propagación (path encoding),
- impulso (*polarization* encoding),
- distribución espacial de la luz (*codificación de momento angular orbital*)
- y tiempo (*codificación time-bin y time-frequency*).

Todas las estrategias de codificación presentan sus ventajas e inconvenientes.

Cui et al. en [4], describen una estructura típica de un sistema Quantum Key Distribution (QKD) donde podemos dividir el sistema en tres grandes zonas tal y como vemos en la *Fig. 1*. En la parte superior nos encontramos con la zona que ponemos clasificar como de aplicación. Esta zona es la principal consumidora de las claves generadas por el sistema. En la zona intermedia, tenemos la zona de gestión de claves. Los equipos del sistema se conectan en esa zona a través un canal de fibra clásico. Finalmente, en la zona inferior tenemos la zona cuántica, es la encargada de transmitir fotones por el canal cuántico y principalmente controla la óptica necesaria para realizar la operación. Es en esta zona donde también se encuentra el componente de generador de números aleatorios puros.

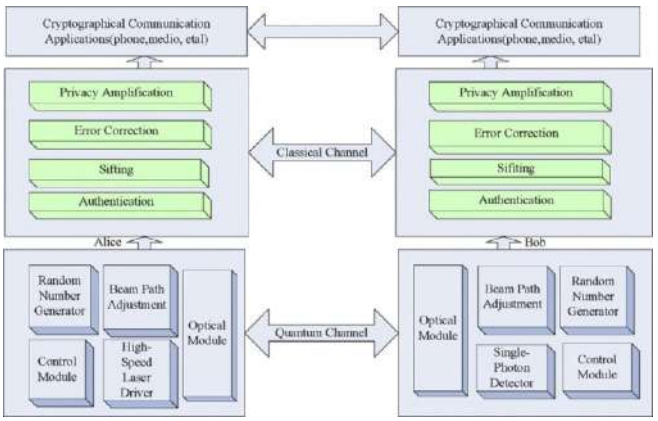


Fig. 1. Esquema típico de un sistema QKD [4]

Uno de los puntos fuertes reconocidos en el método de comunicaciones seguras QKD, es su capacidad de detectar intrusiones en las comunicaciones. En 1984 Bennet y Brassard propusieron el estándar BB84 basado en QKD [1] que es el más conocido e implementado. De manera que, en una comunicación entre dos partes, la interceptación silenciosa de los datos por un tercero es fácilmente detectada, mientras con tecnologías de comunicación convencionales esto no es posible. Desde entonces se han publicado distintos protocolos QKD que se clasifican en función del uso de propiedades durante la transmisión como la modulación aplicada, la codificación/decodificación y la implementación

del canal cuántico, etc. Así, existen varios tipos de enfoques QKD, como son:

- Gaussian-modulated CV-QKD
- Discrete-modulated CV-QKD
- Coherent one-way (COW) quantum key distribution
- Differential phase-shift (DPS) quantum key distribution
- Six-state quantum key distribution
- Decoy-state quantum key Distribution

En este artículo vamos a presentar en la sección 2, el sistema real de comunicaciones seguras sobre QKD, desplegado en un entorno controlado. Una vez desplegado pasamos a evaluar su rendimiento en la sección 3, tanto en una situación normal de operación, como cuando es introducido un atacante (Eve) en medio de la comunicación. Finalmente, en la sección 4 se describen unas conclusiones del experimento presentado, así como los siguientes pasos a realizar.

II. DESCRIPCIÓN DEL SISTEMA

El sistema desplegado consta de dos equipos comercializados por la empresa ID-Quantique[1] tal y como se ve en la *Fig. 2*. Estos equipos utilizan para codificar la información, *time-bin encoding*. Este tipo de codificación es adecuado para dispositivos fotónicos integrados, donde los fotones pueden generarse, manipularse y medirse sin la necesidad de dispositivos de codificación externos. Además, el time-bin es un buen candidato para aplicaciones comunicaciones y distribución de claves cuánticas dada su resiliencia al ruido; el ruido puede afectar a la polarización, pero al no polarizar los fotones, este tipo de codificación resiste a medios despolarizantes o decoherencia y modo de dispersión [2]. Para la emisión física de los fotones, utilizan el protocolo Coherent on-way (COW) quantum key distribution [3]. COW-QKD consiste en codificar bits lógicos y emitirlos en una secuencia de pulsos débiles y modulando su intensidad. Entre las ventajas de este protocolo están la posibilidad de ofrecer una gran eficiencia para el alcance (se han publicado estudios donde se puede llegar a distancias superiores a 300km [6], en el caso de los equipos desplegados se alcanzan los 100km), la facilidad en su implementación y experimentación, se reduce el impacto de las interferencias en la visibilidad y se evita en gran medida los ataques por división del número de fotones; razones clave por lo que es el principal protocolo a la hora de desplegar un sistema QKD para su uso real y no experimental. Como inconvenientes de este protocolo se puede mencionar que los pulsos vacíos contienen una luz que puede introducir ruido, lo que aumenta las tasas de error o que su rendimiento disminuye con un aumento de las perturbaciones. No obstante, las pequeñas perturbaciones no afectan al rendimiento, sólo afectan cuando superan un umbral [3].

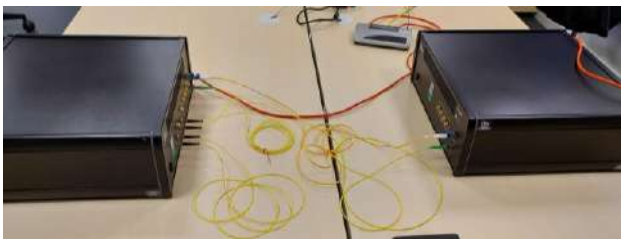


Fig. 2. Sistema QKD de dos nodos desplegado en el laboratorio.

[1] https://www.idquantique.com/quantum-safe-security/products/clavis3-qkd-platform-rd/

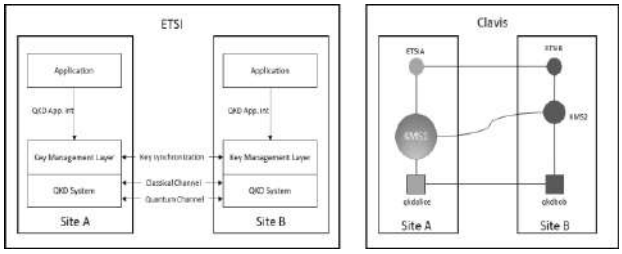


Fig. 3. Configuración lógica del sistema QKD desplegado.

Los dos equipos Alice y Bob, están conectados mediante fibra óptica por dos canales, por un lado, un par de fibras conectan un canal de comunicación clásico y por otro lado está el canal cuántico compuesto por una única fibra aislada del canal clásico.

En la Fig. 3 se puede ver a la izquierda la propuesta del organismo de estandarización ETSI [5] de interfaz de aplicación y la configuración lógica del sistema y a la derecha la configuración lógica específica del sistema bajo estudio. En la capa superior, se encuentra la capa o nivel de aplicación que consumirá las claves generadas por los equipos. En este nivel de aplicación, los nodos ETSIA y ETSIB proporcionan claves y certificados siguiendo con el estándar ETSI [5]. La capa central es la encargada de la gestión de las claves, la comunicación entre los nodos (KMS1 y KMS2) y responsable de la sincronización de las claves. Finalmente, la capa más baja es la dedicada al sistema del sistema QKD, donde se da la comunicación por el canal cuántico.

La distancia actual entre los nodos es de 2 metros, por lo que errores quedan minimizados una vez las fibras ópticas de comunicación están correctamente instaladas. No obstante, debido a la regulación de los detectores internos, configurados para distancias largas de fibra, se ha hecho uso de atenuadores ópticos, a fin de no deslumbrar a los detectores situados en el equipo receptor (Bob).

III. EVALUACIÓN DE RENDIMIENTO

Con el objetivo de comprobar el funcionamiento de un sistema para uso extensivo y de manera comercial, se ha propuesto un experimento donde primeramente analizamos el uso normal del sistema y luego comprobamos la reacción del sistema ante un atacante.

A. Evaluación de rendimiento en condiciones normales

En un primer momento, se configuró el sistema para tener una comunicación directa entre Alice y Bob, separados por 2m, y guardando una clave de un tamaño de 256 bits. Los equipos estuvieron comunicándose de manera continuada durante aproximadamente 24 horas.

Una vez pasadas 24 horas, se extrajeron los logs y los datos de monitorización del sistema para su posterior estudio. Los principales valores medidos fueron el quantum bit error rate (QBER), la visibilidad y el key rate en cada uno de los equipos. El QBER indica el ratio de error detectado entre las claves intercambiadas entre los equipos. Puede darse el caso que se pierda algún fotón de manera natural y con ello se introduzcan errores. Es por lo que no es posible tener un QBER con valor 0, aunque se trata de conseguir valores cercanos a 0. La visibilidad indica lo bueno que es el medio para transmitir

información y finalmente, el key rate mide la velocidad de envío de bits por segundo para enviar claves, que sirve para indicar si el sistema podrá ser usado en condiciones de producción.

Tabla I
EVALUACIÓN DE PARÁMETROS DEL SISTEMA EN UN ENTORNO DE LABORATORIO

	QBER(%)	Visibilidad (%)	Key rate(bits/s)
Máximo	5	100	2.877,61
Mínimo	0,9	39	382,82
Media	2,5	99	2543,88

En la *Tabla I* se muestran los valores de los parámetros medidos tanto en Alice, como en Bob. Como se ve, el ratio de error (QBER) no sube del 5% con lo que los errores se consideran suficientemente bajos como para considerar que no hay nadie escuchando la conversación como para considerarla segura.

Cabe comentar que se ha detectado interferencia entre la temperatura y el ratio de error pero dadas las condiciones en las que se trabaja en el laboratorio (temperatura ambiente), no es un parámetro limitante aunque puede ser una de las razones para las pequeñas fluctuaciones.

B. Evaluación de rendimiento introduciendo un ataque

Tras haber evaluado el sistema y su comportamiento en situaciones normales, se propone monitorizar el sistema cuando un atacante (Eve) es introducido en el sistema para capturar la información. Para ello se conecta un tercer equipo se interconecta en medio del canal cuántico entre Alice y Bob. El esquema del sistema resultante puede verse en la *Fig. 4.*

Primeramente, se arranca el sistema y a pesar de estar el atacante en medio del canal, se deja en modo pasivo con el fin de permitir la configuración y sincronización correcta en los nodos del sistema (Alice y Bob). Al estar en modo pasivo, Eve deja pasar los fotones sin actuar sobre ellos.

Tras la configuración inicial y mientras está funcionando correctamente, se realiza un ataque de 30 segundos de duración, donde Eve es capaz de capturar el 45% de los fotones que se transmiten y los envía de nuevo a Bob, pero con un retraso. Aunque Eve no modifica el valor de estos fotones, dado que la información está codificada en los fotones utilizando *time-bin*, esto genera una diferencia entre los fotones enviados por Alice y los recibidos por Bob. Durante este ataque de corta duración, el atacante pretende enmascararse y simular una perturbación natural en la fibra óptica. No obstante, el sistema ha detectado fallos y ha dejado de almacenar nuevas claves, sin embargo, ha podido seguir funcionando gracias al buffer donde había almacenadas claves, si bien en modo degradado, al no reponer las claves utilizadas durante el ataque. Tras los 30 segundos de ataque, el sistema se recupera.

Fig. 4. Sistema QKD en el laboratorio con un atacante en medio del canal cuántico

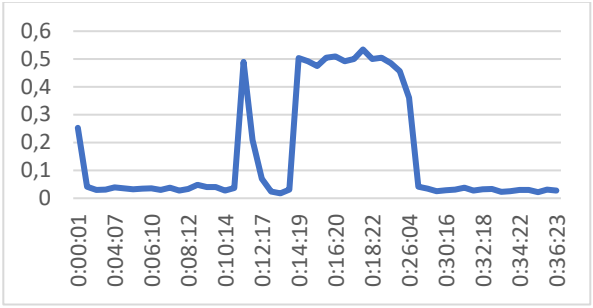


Fig. 5. Gráfica de valores de QBER durante el ataque al sistema

Pasados 2 minutos con el sistema recuperado y volviendo a funcionar a toda su capacidad, se realiza un segundo ataque, este de mayor duración (hacía el minuto 12). Tras 6 minutos de ataque, el sistema se reinicia, y se da por terminado el ataque, de modo que el sistema puede volver a configurarse y ponerse en marcha de nuevo y funcionar correctamente. Se ha comprobado que, si el ataque se da durante la configuración inicial del sistema o tras un reinicio, el sistema no logra ponerse en marcha y vuelve a reiniciarse indefinidamente.

En la Fig. 5 pueden verse valores de ratio de error (QBER) que se corresponden con la descripción previa. En el primer minuto el sistema termina de configurarse y a su término los valores de QBER están bajos, entorno a cero. Al comienzo del primer ataque (minuto 10), los valores QBER; empiezan a subir, se considera que hay un atacante cuando el QBER sube por encima de 0,045. Tras el fin del ataque comienzan a recuperarse. Hacía el minuto 14 se ve una subida brusca y prolongada en el tiempo que se corresponde con el segundo ataque de larga duración. Tras la reconfiguración, el QBER vuelve a estar en valores bajos.

En la gráfica que se muestra en la Fig. 6, se puede ver la visibilidad entre los equipos. Al principio los valores son bajos, incluso por debajo de cero, porque el sistema está en una fase de configuración. Alrededor del minuto 1, los valores suben, ya que se ha terminado la fase de configuración y comienza la fase de trabajo. Hacia el minuto 10 puede verse un pico de bajada que se da durante el ataque de corta duración. Tras este ataque hay una subida, recuperando los valores en torno a 1, para volver a bajar hacia el minuto 12, donde aparece una bajada brusca de valores, y tras ellos los valores siguen con una bajada más prolongada, incluso terminan por tener picos por debajo de 0. Esto se debe al ataque de larga duración que obliga al sistema a reiniciarse (momento que los valores bajan de 0), y el sistema vuelve a configurarse tras lo que los vuelve a valores cercanos a 1 que son los valores esperados cuando no hay agentes tratando de escuchar.

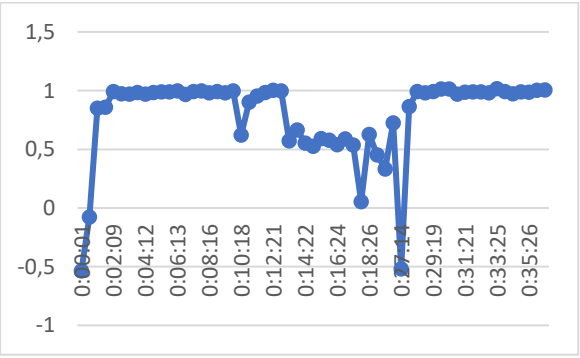


Fig. 6. Gráfica de valores de visibilidad durante el ataque al sistema

IV. CONCLUSIONES

Se ha desplegado un para un uso intensivo, que implementa el paradigma QKD y se ha experimentado sobre él un ataque de escucha en el canal de comunicaciones (eavesdropping). Tras analizar los resultados del ataque, se puede afirmar, que las comunicaciones cuánticas se perfilan como una alternativa más segura y válida a las comunicaciones clásicas. La validación de ha realizado teniendo presente que son equipos para uso genérico, independientemente del lugar donde se despliegan y la distancia entre puntos (dentro del rango admisible) y que deben de tratar ofrecer un servicio de manera continuada. El sistema es resiliente a problemas transitorios ya que almacena una cache, claves para ser consumidas mientras dura la perturbación. Esto mejora los resultados de prototipos que no tienen en cuenta la reducción de tiempos inoperativos. Las pruebas se han realizado en condiciones normales de operación, y realizando un ataque plausible sobre el canal de comunicaciones.

Mientras que, en condiciones habituales de operación, este sistema presenta un ratio aceptable de generación de claves, la introducción de un agente externo provoca una notable disminución de la coherencia cuántica necesaria para aceptar las claves transmitidas como válidas, contemplándose en la brusca subida de indicadores de funcionamiento como el QBER. Aunque el atacante consiga inhabilitar el sistema (no genera más claves y si el problema persiste, se agotan las claves o el sistema se reinicia indefinidamente), el objetivo principal del ataque, es decir la captura silenciosa de los datos secretos, nunca es conseguido, ya que el atacante desconoce los parámetros necesarios para interpretar correctamente los datos fotónicos que se transmiten en por el canal cuántico.

Tras estas primeras experimentaciones, las líneas de trabajo que se espera realizar en un futuro cercano: desplegar el sistema en un entorno real con una distancia mayor, se espera comunicar dos edificios situados a kilómetros de distancia y experimentar sobre fibra desplegada con anterioridad. Esto permitirá estudiar cómo combinar las comunicaciones existentes con comunicaciones cuánticas. Otra de las líneas a trabajar es la combinación del este tipo de paradigma QKD, con Blockchain a fin de aumentar la seguridad. También es necesario estudiar nuevos parámetros de monitorización de este tipo de sistemas, dado que no son habituales.

AGRADECIMIENTOS

Esta investigación ha sido financiada gracias al proyecto QUANTEK, bajo el programa de financiación del Gobierno Vasco ELKARTEK, expediente número KK- 2021/00070.

REFERENCIAS

[1] C. H. Bennett and G. Brassard. “Quantum cryptography: Public key distribution and coin tossing”. In Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, volume 175, page 8. New York, 1984

[2] Flamini F, Spagnolo N, Sciarrino F.: “Photonic quantum information processing: a review”. Rep Prog Phys. 2019 Jan;82(1):016001. doi: 10.1088/1361-6633/aad5b2. Epub 2018 Nov 13. PMID: 30421725.

[3] N Gill, SS, Kumar, A, Singh, H, et al.: “Quantum computing: A taxonomy, systematic review and future directions”. Softw: Pract Exper. 2022; 52 (1): 66- 114. doi:10.1002/spe.3039

[4] Ke Cui, Jian Wang, Hong-fei Zhang, Sheng-zhao Lin, Dong-xu Yang, Teng-yun Chen: “An authentication scheme with high throughput based on FPGA for a practical QKD system.” Optik 126 (2015) 4747–4750.

[5] Quantum Key Distribution (QKD); Application Interface. Standard ETSI, ETSI GS QKD 004 V1.1.1 (2010-12).

[6] Korzh et al. Nature Photonics 9, 163–168 (2015).

$S_c^e martWiFi$: Monitorización dinámica del entorno inalámbrico para mejorar la seguridad

Víctor José López Marín, Pedro García Teodoro

Network Engineering & Security Group (https://nesg.ugr.es)
ETS Ingenierías Informática y de Telecomunicación - Universidad de Granada
victorjlopez@correo.ugr.es, pgteodor@ugr.es

***Resumen*—Con el paso del tiempo los entornos inalámbricos resultan cada vez más complejos, incorporando un mayor número de dispositivos que enriquecen los servicios que nos rodean y mejoran nuestra vida diaria como, por ejemplo, entornos IoT, 5G/6G. Ello, sin embargo, no siempre viene acompañado de unos mecanismos de administración, configuración y securización adecuados que permitan alcanzar los objetivos perseguidos, sin precisar un conocimiento técnico avanzado de los sistemas. Para facilitar al usuario final la operación adecuada de todo ello se pueden adoptar soluciones automáticas para localizar dispositivos problemáticos que, por una pobre configuración u operación, provocan una degradación de las capacidades y prestaciones del entorno, impactando de forma negativa en el resto de dispositivos y, en general, en la experiencia del usuario. En este ánimo, el objetivo de este trabajo es la identificación y seguimiento de parámetros relativos a dispositivos de un entorno inalámbrico que permitan mantener, a un bajo coste técnico y computacional, un nivel de operación adecuado del sistema global, pudiendo incorporar soluciones que den respuesta oportuna en caso contrario.**

***Index Terms*—WiFi, RaspberryPi, Seguridad, Monitorización.**

Tipo de contribución: Investigación en desarrollo.

I. Introducción

Es conocido que cada vez son más los dispositivos que agregamos a nuestra red inalámbrica y mayor es la interacción usuario-dispositivo y dispositivo-dispositivo [1]. Dichos sistemas no siempre incorporan de serie mecanismos de seguridad y configuración adecuados y, en casi la totalidad de las situaciones, los usuarios desconocen o no son conscientes de este hecho [2].

Este trabajo, planteado en el marco de un Trabajo Fin de Máster del Máster en Ciberseguridad de la Universidad de Granada, ha sido pensado para atender a una necesidad actual que carece de una solución tangible a fecha de hoy en el mercado para cualquier tipo de usuario, ya sea con un perfil técnico o un perfil, usualmente, lego. Como requerimiento necesario, una característica importante es que la solución no tenga un alto coste, resulte viable para cualquier entorno inalámbrico y que la interacción del usuario con la solución/herramienta sea sencilla, pero no por ello limitada o incompleta en su funcionalidad.

Como veremos más adelante, la propuesta aquí planteada resulta abierta para entornos inalámbricos de forma general; es decir, aplicaría tanto a entornos domésticos como corporativos ya que, en ambos casos, surge la necesidad de conocer el estado de los dispositivos de la red, tener un control sobre ellos y disponer de una realimentación hacia el usuario/administrador acerca del funcionamiento y operación de los mismos.

Para una correcta estimación del estado del entorno, será necesario implementar mecanismos de adquisición de información de los dispositivos inalámbricos como, por ejemplo, datos sobre el sistema operativo o el *firmware* del equipo, el tipo de dispositivo (móvil, tablet, TV, electrodoméstico, sensor, etc.), operaciones que está realizando, estimación del ancho de banda consumido en la red, entre otros. A partir de ello, se tratará de identificar la causa del descenso de calidad o posibles eventos anómalos que provocan la degradación del servicio WiFi o de los servicios digitales en general. Desde este punto de vista, es importante destacar el hecho de que el concepto de *seguridad* en el contexto que nos ocupa se utiliza de modo mucho más amplio que el habitual, siendo el fin último en nuestro caso no tanto la solución de posibles eventos maliciosos intencionados como la de situaciones que, provocadas externamente o no, ponen en riesgo el *performance* general del entorno.

En las siguientes secciones se realizará, en primer lugar, una descripción general del funcionamiento pretendido para la solución de monitorización que se propone, a la que hemos bautizado como $S_c^e martWiFi$, y seguidamente se detallará de una forma más técnica la arquitectura sobre la que se prevé el desarrollo de este trabajo, así como la programación a realizar en los distintos módulos software de los que constará el sistema.

II. Descripción de la Propuesta

Nuestra solución de monitorización, a la que hemos denominado $S_c^e martWiFi$, persigue ser implementable en cualquier tipo de red inalámbrica, doméstica o corporativa. Los usuarios, de perfil técnico o, como es más habitual, lego, lo último que quieren y necesitan es realizar grandes esfuerzos e inversiones en modificar la arquitectura inalámbrica de la que disponen. Por ello, esta solución ha de integrarse en la misma red WiFi que el resto de dispositivos de los que se desea tener control, y conocer de forma ágil si están realizando alguna actividad o están operando de algún modo que no debieran para influir negativamente en la calidad y estado generales del entorno.

Este proyecto va a estar fundamentado en una plataforma hardware Raspberry Pi [3] que permita todo el desarrollo. La elección de ella como núcleo de este proyecto es debido a la versatilidad que aporta en cuanto a programación, funcionalidades y ventajas frente a otros dispositivos en un

rango adecuado coste-beneficio. Además, soporta multitud de lenguajes de programación y distintas herramientas *open source* que facilitarán el desarrollo del proyecto.

Aunque con la reciente salida del sistema operativo Raspberry Pi OS de 64 bits [4], este dispositivo ya no tiene límite en la cantidad de memoria RAM que se le asigna a un solo proceso, en la versión previa de 32 bits cada proceso se limitaba a acceder hasta un máximo de 3 GB. Así, utilizando el modelo de 8 GB de RAM se obtendrá un mejor rendimiento en los procesos que se definirán en el transcurso de este proyecto.

Este pequeño pero altamente versátil computador será, pues, el dispositivo que se integre en el ecosistema de red inalámbrica, cuyo objetivo será vigilar y verificar la correcta funcionalidad de cada uno de los dispositivos inalámbricos del entorno, con opción de notificar al usuario de una forma clara, comprensible y concisa cuando algo no esté funcionando como se espera. La Fig. 1 muestra en una primera aproximación la integración de la Raspberry en la red inalámbrica de, por ejemplo, un hogar doméstico. Este tipo de entornos tienen dispositivos de todo tipo, desde *smartphones* y portátiles hasta electrodomésticos, *smart TVs* y dispositivos IoT. La principal idea es que la Raspberry sea un dispositivo más en el ecosistema inalámbrico pero que, ante eventos anómalos en relación al funcionamiento y operación del resto de dispositivos, recopile la información precisa para que el usuario sea conocedor del hecho y, en su caso, pueda adoptar contramedidas.

Otro aspecto importante a tener en cuenta es que la adquisición de información sobre los dispositivos inalámbricos del entorno no tiene como objetivo vulnerar la privacidad del usario en relación a sus datos personales, ya que los datos a tener en cuenta serán asépticos en relación a su contenido y significado. Esto se refiere a que exclusivamente se trabajará con información recopilada sobre las características, configuración, funcionamiento y operación de los diferentes dispositivos, analizando aquellos que disminuyan la calidad de señal del entorno inalámbrico, incluso analizando aquellos que están debidamente conectados pero que no son capaces de aprovechar todo el ancho de banda.

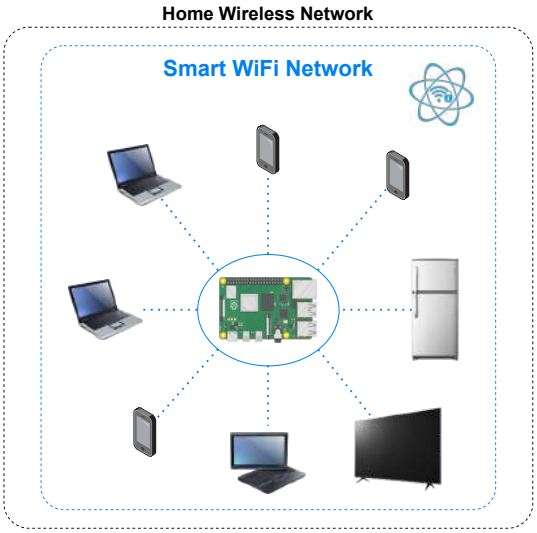


Figura 1. Arquitectura general del sistema $S^e_c martWiFi$.

Adicionalmente a ello, se podría realizar una búsqueda inversa de los eventos que afectan al correcto funcionamiento del entorno en aras de una mayor precisión del análisis e identificación de los dispositivos.

III. Arquitectura Operacional

En esta sección llevaremos a cabo una descripción breve de la arquitectura operacional de $S^e_c martWiFi$. Como se ha comentado con anterioridad, la Raspberry Pi es el núcleo de este desarrollo, operando en modo "*Plug & Play*". También cabe destacar que las pruebas llevadas a cabo se han realizado en un entorno doméstico donde la conectividad inalámbrica se ofrece a los dispositivos a través de un *router* del ISP (*Internet Service Provider*) con funcionalidad de AP (*Access Point*).

III-A. Configuración

La Raspberry se puede disponer como un dispositivo cliente más del entorno inalámbrico. Aunque esto puede parecer la opción más simple y viable, existen ciertas limitaciones. La mayor limitación se produce a la hora de analizar el tráfico del resto de dispositivos, ya que, en la mayoría de entornos inalámbricos, todo el tráfico está cifrado y la comunicación es directa con el AP. Este esquema de red solo permitiría obtener algunos parámetros de los dispositivos tales como dirección IP, dirección MAC y puertos y servicios accesibles. En líneas generales, y más allá de la firme intención de no acceder a información sensible, el control que se consigue así resulta reducido.

Una segunda opción de operación es agregar dos dispositivos de red adicionales al entorno: un *switch* y un AP, según se muestra en la Fig. 2. El *router* del ISP se conectaría a un *switch* que, al menos, tuviese otras dos interfaces de salida. Una de ellas se destina a la conexión vía Ethernet con la Raspberry y la otra interfaz se conecta a un AP, el cual facilita el servicio de conectividad inalámbrica al resto de dispositivos que demanden acceso externo a la red. Es importante que el *switch* tenga la funcionalidad de *port mirroring* en alguna de sus interfaces, a través de cuyo servicio se permite copiar el tráfico de paquetes de un puerto en otro. De este modo, la interfaz que conecta con la Raspberry será el puerto espejo que copie el tráfico que entra y sale de la red inalámbrica.

Esta aproximación solventa la problemática del escenario primero ya que sería posible monitorizar el tráfico con destino a Internet de la red inalámbrica. Pero de nuevo surgen algunas limitaciones. El objetivo del proyecto es identificar cada

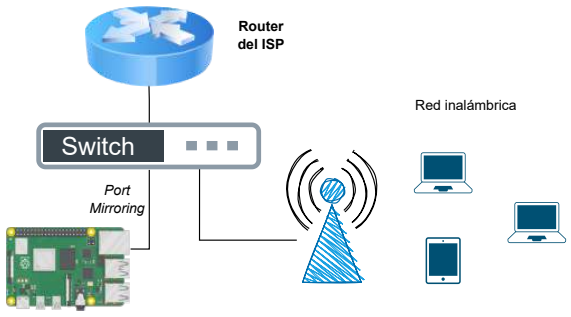


Figura 2. Segunda opción de configuración del entorno de trabajo.

dispositivo de forma individual con la captura de diferentes parámetros e información. Sin embargo, con este escenario no se permitiría clasificar el tráfico para cada dispositivo y queda oculto el tráfico que queda dentro de la red inalámbrica. Consecuentemente, el análisis que se realiza de la *performance* del entorno no sería completo e implicaría un mayor costo dado que se agregan más cantidad de dispositivos de red.

Queda claro, pues, que incluir más dispositivos no aporta una mejor funcionalidad de la que se desea. Por tanto, la Raspberry formará parte del ecosistema actuando como un AP en modo *bridge*. Gracias a la herramienta *hostapd* [5], la Raspberry es capaz de actuar como AP en dos modos: *router* (*routed*) y puente (*bridge*).

El modo *routed* implica que se crea una nueva subred, se asignan direcciones IP, se realiza el enmascaramiento y demás funcionalidades típicas de un AP WiFi. Con el modo *bridge* [6] se libera a este de varias de las funcionalidades y se destina todo el trabajo computacional a identificar los dispositivos por separado, obtener la información y modo de operación de los mismos y analizar el estado y operación de cada uno. Así, la elección de configuración en este modo en nuestro caso se debe simplemente al hecho de que se quiere mantener la misma configuración de red que ya hay establecida y solamente se necesita que la Raspberry sea un "intermediario" entre el *router* del ISP y la red inalámbrica.

En la Fig. 3 se representa el esquema final sobre el que se va a desarrollar la propuesta. Dado que la Raspberry se comportará como un AP en modo puente, las direcciones IP que había previamente establecidas para cada dispositivo quedan inalteradas porque el servicio de DHCP lo realiza el *router* ISP, además del enrutamiento, servicio de DNS y resto de funcionalidades. De esta forma, todo dispositivo conectado a la red $S_c^e martWiFi$ podrá ser identificado de forma correcta por la Raspberry, tanto para analizar su configuración como el tráfico que genera en la red.

III-B. Operación

Una vez identificada la arquitectura final a usar, se precisa definir la programación que se va a realizar en la Raspberry. En líneas generales, el procedimiento de estimación de la *performance* del entorno $S_c^e martWiFi$ se dividirá en cuatro etapas principales, representadas en la Fig. 4, como sigue:

1. *Identificación*. La principal ventaja de configurar la Raspberry como *bridged AP* es que se tiene un control completo sobre los dispositivos que están conectados en la red $S_c^e martWiFi$. Cada vez que demanda conexión un nuevo dispositivo se conoce si la autenticación ha sido exitosa, fallida o si se ha desconectado de la red. Este será un primer indicativo para conocer si hay dispositivos que estén intentando acceder a la red de manera legítima o si se trata de dispositivos que no deberían estar en dicho entorno. Cada uno de ellos será identificado de manera unívoca para su diferenciación respecto del resto.
2. *Adquisición*. El siguiente paso será definir el procedimiento a seguir para la recopilación de parámetros relativos a los dispositivos. Estos serán recopilados tanto al inicio de la asociación a la $S_c^e martWiFi$ como de manera periódica a lo largo del tiempo. Entre otros parámetros a extraer podemos mencionar [7]: tipo de dispositivo, SO, versión software, puertos abiertos, servicios accedidos y equipos en las comunicaciones, volumen de datos (ancho de banda), paquetes de subida y bajada, etc.
   A este fin podemos usar herramientas diversas como *nmap, wireshark, tcpdump*, etc.
3. *Visualización*. Toda la información citada será almacenada para un acceso histórico, a fin de conseguir un análisis de la evolución temporal del entorno. Esta información podrá visualizarse a través del empleo de herramientas gráficas como la denominada Grafana [8], de carácter *open source*, a través de la cual se podría desplegar una aplicación web donde se represente, para cada dispositivo, la información más relevante en forma de tablas o gráficas.
   También podría utilizarse una representación en forma de gráficos circulares [9] o tela de araña [10], donde se visualice de manera conjunta la evolución de cada uno de los parámetros monitorizados.
4. *Notificación*. La elección de Grafana como herramienta de visualización permite, además, integrar numerosos métodos de notificación según el umbral que se establezca para una potencial métrica, relativa a cada parámetro monitorizado o a un conjunto de ellos. Algunos de estos métodos de notificación son: email, Telegram, Discord, Slack o Microsoft Teams. Muchos de estos servicios son conocidos por multitud de usuarios y puede personalizarse el canal de notificación sobre el que se desee recibir la información acerca del estado del entorno.
   Sea como fuere, ante la recepción de una notificación de alerta, el usuario o administrador del entorno podrá poner en marcha contramedidas que den solución a la situación concreta evidenciada: activación de reglas de cortafuegos, ejecución de antivirus, desconexión de

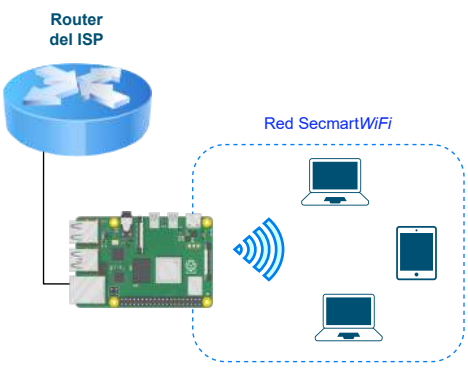


Figura 3. Entorno final $S_c^e martWiFi$ planteado como solución.

Figura 4. Etapas operacionales del sistema $S_c^e martWiFi$.

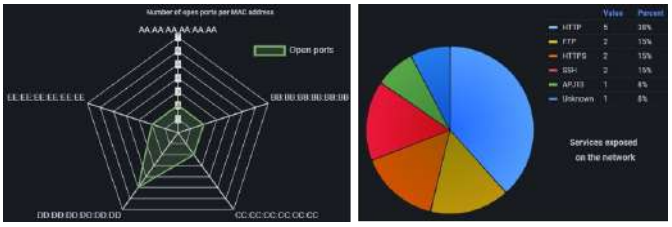


Figura 5. *Dashboards* en forma de tela de araña y gráfico circular.

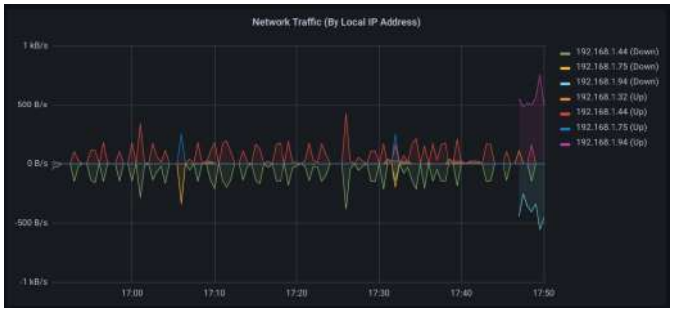


Figura 6. Tráfico *upstream* y *downstream* para cada dispositivo.

equipos, etc. Estas opciones podrán ser propuestas de manera pro-activa por la herramienta a fin de ayudar al usuario en la toma de decisiones.

IV. Resultados Experimentales

Una vez descritos los procedimientos de configuración y operación de la propuesta $S_c^e martWiFi$, es necesario evaluar el sistema para comprobar que satisface los requisitos pretendidos. Las pruebas preliminares realizadas a modo de prueba de concepto se centran en las fases de adquisición y visualización de los parámetros recopilados de cada dispositivo por la Raspberry Pi. El acopio de información será el primer desafío que deba realizarse de forma correcta.

Una vez configurado correctamente el escenario general, donde se han considerado *smartphones*, ordenadores portátiles y algunos dipositivos IoT, y tras la fase de identificación de dispositivo, se procede a recopilar parámetros tales como su dirección MAC, IP, puertos abiertos, servicios que ofrece, protocolos utilizados, versión del sistema operativo, entre otros. Esta información se almacena y actualiza periódicamente en una base de datos SQL para su posterior análisis.

En la Fig. 5 se observan dos gráficas simples, pero muy ilustrativas, que muestran dos tipos de información de interés:

- A la izquierda se encuentra un *dashboard* estilo tela de araña que muestra el número de puertos abiertos que se han detectado para un dispositivo inalámbrico identificado por su dirección MAC concreta.
- A la derecha se sitúa otro *dashboard*, pero esta vez con un estilo de gráfico circular, que muestra cuantitativamente los servicios que están expuestos en los dispositivos de la red.

Por otro lado, a través de un *script* escrito en Python y junto a la plataforma Prometheus [11], se recopilan métricas de tráfico para todos los dispositivos objetivo del sistema $S_c^e martWiFi$. Se pueden definir diferentes paneles de monitorización como número de bytes transmitidos o *bandwidth* consumido para cada dispositivo, distinguiendo tanto tráfico de subida como de bajada. En la Fig. 6 se muestra un panel mediante Grafana que refleja el tráfico *upstream* y *downstream* de cada dispositivo monitorizado.

Puesto que todos los gráficos recogen la información dinámicamente de la base de datos, conforme se actualizan estos valores, los cambios se ven reflejados de forma automática e instantánea.

V. Conclusiones

Ante el masivo crecimiento de las redes inalámbricas, resulta muy complicado conocer si su funcionamiento es adecuado o si, incluso, algún dispositivo está entrando en conflicto con otro; derivando en una mala experiencia del usuario final o en un potencial problema de seguidad en el entorno. Así, generalmente, cuando sucede alguna problemática con los dispositivos del entorno WiFi, no hay un mecanismo de *feedback* que muestre al usuario de forma clara lo que está sucediendo.

Es por todo lo anterior que entendemos que este proyecto cubre una necesidad importante en los entornos inalámbricos actuales, permitiendo que cualquier perfil de usuario conozca si su red WiFi opera de manera adecuada. Nuestra propuesta en esta línea, $S_c^e martWiFi$, resulta sencilla y versátil, pudiendo operar en cualquier tipo de entorno inalámbrico, especialmente en los entornos corporativos.

Gracias al uso de la Raspberry Pi, con un sistema operativo de código libre, la programación de las características que ofrece el sistema resulta de una enorme potencialidad, siendo múltiples las posibilidades que se abren al desarrollo y despliegue de este proyecto.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Gobierno de España, con fondos FEDER, a través del proyecto PID2020-114495RB-I00.

Referencias

[1] Statistica: “Global IoT and non-IoT connections 2010-2025”. Disponible en https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/.
[2] M.B. Barcena, C. Wueest: “Insecurity in the Internet of Things”. Informe Symantec, 2015. Disponible en https://docs.broadcom.com/doc/insecurity-in-the-internet-of-things-en.
[3] Raspberry Pi: “Raspberry Pi”. Disponible en https://www.raspberrypi.com/products/raspberry-pi-4-model-b/.
[4] Raspberry Pi: “Raspberry Pi OS (64-bit)”. Disponible en https://www.raspberrypi.com/news/raspberry-pi-os-64-bit/.
[5] J. Malinen: “hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator”. Disponible en https://w1.fi/hostapd/.
[6] Rapberry Pi: “Setting up a Bridged Wireless Access Point”. Disponible en https://www.raspberrypi.com/documentation/computers/configuration.html#setting-up-a-bridged-wireless-access-point.
[7] J.A. Gómez-Hernández, P. García-Teodoro, J.A. Holgado-Terriza, G. Maciá-Fernández, J. Camacho, M. Robles-Carrillo: “AMon: A Monitoring Multidimensional Feature Application to Secure Android Environments”. WTMC-42th IEEE Symposium on Security and Privacy, pp. 1-12, 2021.
[8] GrafanaLabs: “Grafana: The open observability platform”. Disponible en https://grafana.com/.
[9] GrafanaLabs: “Pie Chart plugin for Grafana”. Disponible en https://grafana.com/grafana/plugins/grafana-piechart-panel/.
[10] GrafanaLabs: “Radar Graph plugin for Grafana”. Disponible en https://grafana.com/grafana/plugins/snuids-radar-panel/.
[11] Prometheus: “Monitoring system & time series database”. Disponible en https://prometheus.io/.

FENIoT: Un nodo edge computing para las investigaciones forenses en el Internet de las Cosas

Juan Manuel Castelo Gómez, Sergio Ruiz Villafranca y José Luis Martínez Martínez
Instituto de Investigación en Informática de Albacete
Universidad de Castilla-La Mancha
Investigación 2, Albacete 02071
juanmanuel.castelo@uclm.es, sergio.rvillafranca@uclm.es, joseluis.martinez@uclm.es

***Resumen*—La aparición del Internet de las Cosas (IoT) ha traído consigo grandes cambios en el ámbito de la ciberseguridad. Dentro del campo del análisis forense, estos cambios han afectado a la manera de proceder a la hora de realizar las investigaciones en este entorno, requiriendo modificaciones en fases clave del proceso forense como la identificación y la adquisición de fuentes de evidencia. Aspectos como el número de dispositivos que encontramos en la redes IoT, su dinamicidad o el reducido tiempo de vida que tienen los datos hacen que se necesiten de nuevas técnicas forenses que permitan llevar a cabo las investigaciones de forma efectiva y completa. Intentado dar solución a este problema, este artículo presenta FENIoT (Forensic Edge Node for the Internet of Things), un nodo IoT que se apoya en la tecnología edge computing para automatizar el proceso de detección de anomalías, identificación de fuentes de evidencia y adquisición de las mismas, tanto desde una perspectiva proactiva como reactiva. Además, se presentan una serie de casos de estudio en los que se evalúa el comportamiento de dicho nodo, probando su utilidad para ser usado en procesos forenses.**

***Index Terms*—Análisis Forense, Internet de las Cosas, Edge Computing, Forense en IoT**

Tipo de contribución: *Investigación original*

I. Introducción

Cuando hablamos del Internet de las Cosas, IoT por sus siglas en inglés, lo hacemos del entorno con mayor número de dispositivos conectados a Internet desde el año 2020 [1]. Por tanto, aunque por su novedad podría dar lugar a pensar que su relevancia no es tan grande, datos como este nos indican la repercusión que tiene este escenario dentro del mundo de la tecnología. Por si esto fuera poco, su crecimiento continúa teniendo el ritmo fulgurante de los últimos años, con las predicciones apuntando a una duplicación en el número de unidades IoT conectadas, alcanzando así los 27 mil millones [2].

La rápida adopción de este entorno por parte de los usuarios ha tenido grandes repercusiones desde el punto de vista de la ciberseguridad. Debido a su popularidad, el encontrar un incidente informático en el que se ha visto involucrado un dispositivo IoT ha pasado de ser una situación excepcional a una muy plausible. Prueba de ello es la cantidad de ataques detectados en el año 2019 [3], que superó los cien millones. Con el ratio de crecimiento que ha sufrido el IoT, es lógico pensar que este número ha crecido enormemente en años posteriores y lo seguirá haciendo en el futuro, pues es bien sabido que las medidas de seguridad de estos dispositivos no son los suficientemente fuertes como deberían.

De la mano de la materialización de los incidentes encontramos las investigaciones forenses, que se encargan de obtener respuestas sobre lo acontecido en ellos. La unión entre el aumento del número de ataques dirigidos a dispositivos IoT con el gran uso de éstos por parte de los usuarios, ha hecho que este entorno se convierta en un escenario de interés considerable para la comunidad forense. Al igual que el IoT ha traído consigo cambios en la forma de hacer uso de la tecnología, también lo ha hecho en la forma de investigar los datos que se generan en este entorno. Elementos como su heterogeneidad, interoperabilidad, el número de dispositivos o su capacidad de cómputo afectan a la manera en la que enfocar los procesos forenses. Estos aspectos hacen que las técnicas utilizadas en los entornos convencionales, es decir, aquellos en los que se venían realizando análisis hasta la aparición del IoT, no sean del todo efectivas, pudiendo dar lugar a la omisión de datos que podrían ser de interés en una investigación. A su vez, debido a la novedad del entorno IoT, los investigadores deben apoyarse en las técnicas convencionales en sus análisis, puesto que no existen unas específicamente diseñadas para este escenario. El ejemplo más claro lo vemos a la hora de realizar la adquisición física de la memoria no volátil de los dispositivos IoT, debiendo utilizar técnicas como Joint Test Action Group (JTAG), In-System Programming (ISP) o chip-off, como se puede ver en [4], [5], [6], [7]. Estos métodos, comúnmente usados en smarthpones, se vuelven difícilmente ejecutables debido a la posibilidad de no tener acceso físico al dispositivo.

Para solventar este tipo de problemas y dotar a las investigaciones IoT de una solución diseñada para en este entorno, este artículo presenta FENIoT (Forensic Edge Node for the Internet of Things), un nodo edge que realiza funciones clave en los procesos forenses y permite automatizar el proceso de detección de anomalías, identificación, y adquisición de fuentes de evidencia. Además, la propuesta puede funcionar tanto en modo reactivo, utilizándose el nodo edge una vez que el incidente ya sea ha producido, o bien de forma proactiva, instalándolo en una red IoT para hacerla más amigable a la hora de realizar investigaciones forenses.

Contribuciones. Las contribuciones de esta propuesta se pueden resumir en los siguientes puntos:

- Se hace un breve recorrido por las propuestas realizadas por la comunidad científica centradas en el uso de soluciones externas en las investigaciones forenses en el Internet de las Cosas.
- Se presenta una solución forense en forma de nodo IoT que hace uso de la tecnología edge computing para asistir al investigador en las fases de detección de incidentes, identificación y adquisición de un proceso forense. Las

funcionalidades clave de este nodo son:

- Permite detectar la aparición de anomalías tanto en la red IoT como en dispositivos individuales mediante la monitorización del tráfico y de los sistemas de ficheros.
- Reduce el tiempo de respuesta ante un incidente al mínimo, comenzando el proceso forense de forma automática cuando se detecta una anomalía.
- Estudia el tráfico generado en la red IoT para determinar la cantidad de dispositivos en la misma, su elemento identificador, su comportamiento y, en algunos casos, el modelo del mismo.
- Capaz de analizar los protocolos más comunes utilizados por dispositivos IoT como WiFi, Zigbee, Z-Wave o Bluetooth.
- Permite la adquisición de forma remota del sistema de ficheros de dispositivos IoT que sean compatibles con esta técnica, tomando una serie de precauciones para garantizar la correcta preservación de los datos obtenidos.

- Se somete la propuesta a una evaluación de rendimiento, usando el nodo FENIoT en varios contextos IoT, probando que su uso es útil en investigaciones forenses.

Este artículo se estructura de la siguiente forma. En la Sección II se describen los elementos que motivan el desarrollo de este trabajo. La Sección III presenta un breve análisis de las propuestas realizadas por la comunidad científica que hacen uso de elementos externos en las investigaciones forenses IoT. Las características del nodo FENIoT se detallan en la Sección IV, mostrando su rendimiento en la Sección V. Finalmente, el artículo presenta las conclusiones que podemos extraer tras la realización de esta investigación, las cuales se indican en la Sección VI.

II. Motivación

Las nuevas funcionalidades que caracterizan a los dispositivos IoT no solo afectan a la forma en la que los usuarios interactúan con ellos, sino que también lo hacen con la manera de proceder de los investigadores forenses, los cuales intentan extraer la mayor cantidad de datos de la manera más completa, efectiva y fiable posible, encontrando en los siguientes aspectos los cambios más relevantes en la forma de afrontar las investigaciones:

- Rango de la escena: el incremento del número de dispositivos en las redes IoT, las cuales suelen estar formadas por múltiples unidades, trae consigo un aumento del rango de la escena a investigar. Este aumento no se produce solo a nivel físico, sino que también lo hace a nivel lógico. Mientras que en entornos convencionales el rango lo marcaba la longitud la conexión cableada o la cobertura de la red WiFi, en el IoT vemos que los dispositivos pueden estar separados por una gran distancia debido a que son compatibles con tecnologías de telefonía móvil como 5G. Del mismo modo, también encontramos protocolos IoT como Zigbee, Z-Wave, LoRaWAN o SigFox, que, aunque su rango es menor que el 5G, es superior al que tenían los protocolos usados por dispositivos convencionales. A esto se le suma que las redes IoT son muy dinámicas, con dispositivos saliendo y entrando a las mismas en cortos espacios de tiempo, como puede ocurrir en las redes vehiculares. Ante estas circunstancias, el realizar un seguimiento del estado de una red se vuelve una tarea compleja. Estos aspectos hacen que la fase de identificación, en la cual se determinan qué dispositivos son susceptibles de contener evidencias, aumente en complejidad y duración.
- Tiempo de vida de las evidencias: debido a que los dispositivos IoT están diseñados para cooperar entre sí para proporcionar funcionalidades en lugar de realizar tareas complejas de forma individual, la mayoría de datos en una red IoT se intercambia en la forma de paquetes de red sin almacenarse en memoria. Además, ya que las unidades tienen una cantidad pequeña de memoria volátil y no volátil, en el caso de que se almacene algún dato, la probabilidad de que este sea reemplazado es muy alta. Esto requiere que el proceso de adquisición se realice lo más pronto posible, de cara a evitar la pérdida de información. De hecho, en el caso del tráfico de red, los datos deberán ser capturados justo cuando se genera el paquete si queremos acceder a ellos.
- Dificultad para realizar el proceso de adquisición: dentro del entorno IoT nos encontramos con dos métodos de adquisición: física y remota. La primera de ellas requiere que el investigador tenga acceso físico al dispositivo, lo que, como hemos comentado anteriormente, se ve dificultado debido a la separación que puede haber entre los miembros de una red. Además, existe la posibilidad de que una unidad IoT esté embebida dentro de un objeto; incluso en el caso de que podamos tener acceso físico al dispositivo, el poder acceder a la placa a la cual están soldados los chips de memoria no es algo trivial y requerirá de un desensamblado, aumentando la posibilidad de que se produzca un daño que pueda impedir la lectura de los datos. Ante esta tesitura, el segundo método, la adquisición remota, se erige como una alternativa muy interesante al método físico. En este caso, hay dos requisitos para poder ejecutar esta técnica. El primero es que el dispositivo sea accesible de forma remota. El segundo, que el sistema operativo que esté ejecutando sea compatible con las herramientas forenses que permiten la adquisición del sistema de ficheros. Lamentablemente, no todos los dispositivos IoT ejecutan un sistema operativo y, si lo hacen, la cantidad de recursos que este ofrece puede que sea tan limitada que las herramientas forenses no sean ejecutables. Además, muchos fabricantes lanzan sus modelos en un entorno cerrado, solo ofreciendo al usuario la posibilidad de interactuar con los dispositivos mediante una aplicación o servicio propietario e impidiendo la posibilidad de conectar de forma remota a los dispositivos usando servicios como SSH o Telnet. De este modo, vemos como el proceso de adquisición se convierte en una tarea ardua cuando trabajamos con dispositivos IoT, siendo muy difícil asegurar con total certeza la viabilidad del mismo.
- Tipos de fuentes de evidencias analizables: otra consecuencia directa del aspecto anterior es que la cantidad de datos a los cuales un investigador tendrá acceso en

un proceso forense IoT se ve limitado. En el caso de la memoria no volátil ya hemos hablado de los problemas que encontramos, pero si nos centramos en la memoria volátil encontramos una situación más difícil todavía. En este caso, además de la necesidad de tener acceso remoto al dispositivo, se necesita de herramientas forenses específicas, que permitan, primero, la adquisición de la memoria y, segundo, la creación de un perfil de la misma. Debido a estos requisitos, el adquirir la memoria volátil se convierte en una tarea casi imposible, solo realizable mediante el uso de técnicas de debugging, que requieren de acceso físico al dispositivo. Sin embargo, encontramos en el tráfico de red la mayor fuente de datos en un entorno IoT debido a su gran interoperabilidad. En este caso, la adquisición de estos datos es relativamente simple, puesto que nos podemos apoyar en un dispositivo externo como un ordenador al que, conectándole el adaptador correspondiente para hacerlo compatible con el protocolo que deseemos, nos permitirá leer los paquetes de red intercambiados. Ante estas circunstancias, se necesitan de técnicas que faciliten la captura y el análisis de este tipo de datos, puesto que en muchos escenarios será la única fuente de evidencias.

III. Estado del Arte

Existen muy pocas propuestas centradas en el uso de elementos externos para ayudar en las investigaciones IoT, aunque hay varias muy interesantes. La primera de ellas es [8], que propone un software denominado ProFIT que, una vez instalado en el dispositivo, es capaz de recolectar información a alto nivel sobre el comportamiento del sistema y almacenarla en el sistema de ficheros, creando así un repositorio de evidencias. Sin embargo, se trata de una propuesta sin implementar y que requiere de tener el software instalado en el dispositivo antes de que se produzca el incidente.

Centrado en el Internet de Vehículos (IoV), [9] propone un framework para la recolección de datos en vehículos inteligentes, el cual está dividido en dos partes. La primera, un dispositivo que se encarga de adquirir los datos del vehículo. La segunda, una infraestructura distribuida para el almacenamiento seguro de los datos adquiridos, para la que formula un algoritmo centrado en la verificación de su integridad.

Por último, [10] presenta un breve framework que hace uso de un nodo fog capaz de analizar los datos generados en la red y actuar en caso de detectar actividad sospechosa, aunque lo hace de forma teórica.

Todas estas propuestas muestran el interés de la comunidad forense en diseñar nuevos tipos de soluciones para las investigaciones IoT, además de remarcar la utilidad que tendría apoyarse en un elemento externo para hacerlo.

IV. Descripción del Nodo FENIoT

El nodo FENIoT está pensado para interactuar de forma directa con la red IoT presente en la escena a investigar para proporcionar diversos servicios forenses. Dicho nodo se encuadraría dentro de la misma de la forma en la que se puede apreciar en la Figura 1. El núcleo de FENIoT está desarrollado con el lenguaje de programación Python [11], por lo que sería posible utilizarlo en cualquier dispositivo que sea compatible con él, incluso en uno convencional. Sin embargo, debido a que la idea es integrarlo dentro de la red IoT, lo conveniente es hacerlo en un dispositivo IoT. Además, se hace uso de una serie de herramientas, las cuales se describirán más adelante, que deben ser compatibles con el sistema elegido o el funcionamiento será incorrecto. En nuestro caso, hemos utilizado una Raspberry Pi Modelo 3 B+ [12] con el sistema operativo Ubuntu Core [13] instalado.

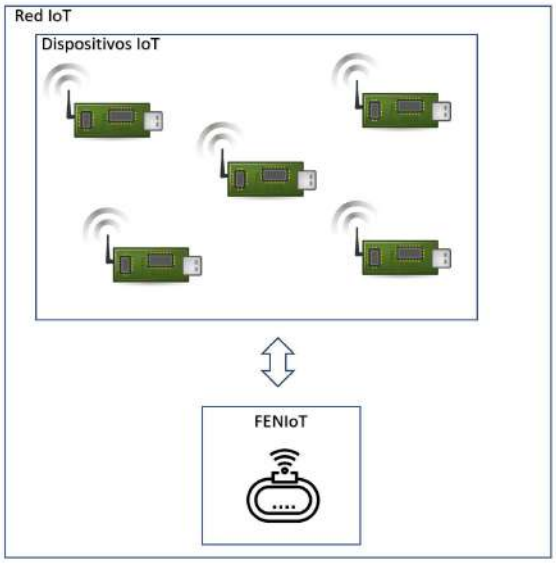


Figura 1. Representación gráfica del funcionamiento del nodo FENIoT.

En el resto de la sección se detalla el funcionamiento de FENIoT, comenzando por sus modos de operación, y continuando por todas las actividades que realiza en las distintas fases del proceso forense.

IV-A. Modos de Funcionamiento

Una de las características determinantes de esta propuesta es la capacidad del nodo de poder ser usado tanto con un enfoque reactivo como en uno proactivo. La diferencia principal radica en el momento en el que FENIoT forma parte de la investigación. En el primer caso, el nodo se utiliza como una herramienta forense cuando el proceso ya ha comenzado. Es decir, el investigador acude a la escena a investigar y hace uso del nodo edge como si de cualquier otra utilidad forense se tratase. Tras ello, y dependiendo del tipo de información que el investigador desee estudiar, añadirá FENIoT a la red bajo examen o lo ejecutará desde fuera de ella.

En el caso proactivo, FENIoT forma parte de la red IoT antes de que la investigación sea necesaria. Este enfoque permite dotar al entorno de la propiedad conocida como *forensic-by-design*, que viene a indicar que este ha sido diseñado contando con la posibilidad de que sea necesario realizar un análisis forense, adaptándolo en consecuencia y haciendo el proceso de investigación mucho más amigable para el investigador al facilitar la realización de ciertas tareas forenses. De este modo, el nodo estaría en funcionamiento constante monitorizando el comportamiento de la red IoT.

- Modo reactivo.
  - Ventaja: no requiere de una instalación previa, lo que es menos intrusivo para el usuario y más plausible en un escenario real.

- Desventaja: puede dar lugar a la pérdida de información, puesto que pasa un tiempo desde que la investigación se requiere hasta que el investigador acude a la escena.

- Modo proactivo.
  - Ventajas:
    - Añade un método de monitorización de la red IoT.
    - Permite detectar anomalías en el tráfico de red o en alguno de los dispositivos que la componen.
    - Reduce el tiempo de respuesta ante un incidente al mínimo, puesto que, en cuanto detecte una anomalía, comenzará a ejecutar las tareas correspondientes.
  - Desventaja: su uso como herramienta de monitorización individual de dispositivos requerirá la generación de tráfico adicional en la red y la ejecución de una tarea costosa computacionalmente, lo que puede causar una ralentización en el funcionamiento del mismo. Por tanto, se debe optar por programar esta funcionalidad cuando el sistema IoT no vaya a ser usado.

En cualquiera de los dos casos, y tal como se puede apreciar en la Tabla I, ambos modos automatizan tanto la fase de identificación como la de adquisición, por lo que el investigador sigue obteniendo un gran beneficio independientemente del modo en el que utilice FENIoT.

IV-B. Monitorización

El proceso de monitorización tiene como objetivo obtener información sobre el comportamiento de la red de forma que, si se necesita realizar un análisis forense del entorno, los datos más relevantes estén disponibles para ser estudiados. Además, FENIoT ofrece un método de localización temprana de incidentes mediante la detección de anomalías. Los dos tipos de evidencia tratados por él nodo son el tráfico de red y la memoria no volátil, trabajando de la forma descrita a continuación.

Tráfico de Red. Para llevar a cabo la captura del tráfico de red es necesario conocer qué protocolos se utilizan en la red IoT a analizar, puesto que puede que se necesite de hardware adicional para poder acceder a los paquetes. Por ejemplo, para capturar el tráfico Zigbee y Z-Wave es necesario disponer de adaptadores específicos. En cambio, si se desea trabajar con WiFi o Bluetooth, el estudio de los paquetes de red se hace de forma nativa si se replica el escenario comentado anteriormente. En caso de trabajar en una red WiFi, FENIoT informará al investigador de los protocolos detectados, por si desea filtrar por alguno en concreto. Las capturas se pueden lanzar y parar de forma manual o automatizarlas para que se realicen por intervalos de tiempo. Cuando finaliza una captura, se genera el log correspondiente y se transforma el archivo generado al formato Comma-Separated Values (CSV) para ser tratado usando la biblioteca Scikit-learn [14] para extraer la información.

Memoria no Volátil. Para poder analizar el contenido de la memoria no volátil es necesario que el dispositivo a estudiar permita el acceso remoto usando el servicio SSH. En este caso, se programa una adquisición automática del dispositivo de forma periódica y se utiliza como base la información sobre los ficheros almacenados en la memoria como sus fechas de creación, acceso y modificación, ruta de almacenamiento y tamaño. Dependiendo del tamaño del sistema de ficheros, se puede optar por utilizar copias incrementales o completas, aunque la adquisición siempre se hará de forma completa, como veremos más adelante. De igual manera que en el caso del tráfico de red, también se genera un log por cada adquisición. Además, estas copias periódicas sirven como punto de restauración del dispositivo en caso de que sea necesario realizar una tarea de recuperación.

Detección de Anomalías. Para realizar el proceso de detección de anomalías se emplea el algoritmo Isolation Forest [15], el cual se basa en la idea de que los datos anómalos se pueden aislar de los normales dividiendo el conjunto de forma recursiva, dado que hay muchos menos y tienen valores muy diferentes [16]. El algoritmo opera seleccionando aleatoriamente un campo del conjunto de datos y escogiendo un valor de corte aleatorio que se encuentra entre el máximo y el mínimo de los encontrados en todos los registros de dicho campo [17]. Este proceso se repite varias veces, apoyándose en otro algoritmo muy utilizado en la ciencia de datos como es Random Forest [18] para realizar el particionado, obteniendo una puntuación normalizada para cada dato en el que se indica con un 1 si es normal y con un -1 si representa una anomalía.

El motivo para escoger este algoritmo es que, como se puede ver en [19], Isolation Forest consigue unos buenos resultados teniendo en cuenta la capacidad de computación requerida para su ejecución, la cual no es muy alta, por lo que es idóneo para ser ejecutado en un dispositivo IoT.

La detección de anomalías en el caso del tráfico de red se puede hacer a nivel de conexiones o de paquetes, o incluso ambos, dependiendo de lo que el investigador considere oportuno. En el caso de la memoria no volátil, se hace a nivel de fichero. En caso de detectar una anomalía, se procedería a ejecutar las siguientes fases del proceso forense.

IV-C. Identificación de Dispositivos

La obtención de información para la fase de identificación tiene su base en el análisis del tráfico de red. Dependiendo del modo de funcionamiento de FENIoT, se procederá a trabajar de una forma u otra. Si está trabajando en modo proactivo, se utilizan las mismas capturas obtenidas durante el proceso de monitorización. En cambio, si lo hace en modo reactivo, se opera de forma similar a como se hace en la fase de monitorización: el investigador pone el nodo a capturar el tráfico del protocolo/s que desea y para la captura cuando considera conveniente. Al igual que en el caso de la monitorización, cuando termina la captura se genera el log correspondiente.

En concreto. se extrae la siguiente información:

- Dispositivos que han enviado o recibido algún paquete de red junto su identificador, que variará dependiendo del protocolo a analizar.
- Paquete enviados y recibidos por cada dispositivo.
- Número de conexiones establecidas entre dispositivos.
- Fecha y hora en la que se realizó el último envío de un paquete por cada dispositivo.

De este modo, se consigue asistir al investigador a la hora de determinar el rango de la escena, puesto que se consigue

Tabla I
RESUMEN DE LAS FUNCIONALIDADES DISPONIBLES EN CADA MODO DE FUNCIONAMIENTO.

Modo de funcionamiento	Monitorización y detección de anomalías		Identificación de dispositivos	Adquisición de fuentes de evidencia	
	Tráfico de red	Sistema de ficheros		Tráfico de red	Sistema de ficheros
Proactivo	✓	✓	✓	✓	✓
Reactivo	✗	✗	✓	✓	✓

extraer el número de dispositivos en funcionamiento en la red y su comportamiento. Además, en caso de que se utilice en su versión proactiva, se podrá mantener un historial de los dispositivos que han formado parte de la red.

IV-D. Adquisición de Fuentes de Evidencia

Finalmente, FENIoT también puede asistir en la fase de adquisición del proceso forense. Esto permite, primero, automatizar uno de las tareas más complejas cuando se realizan investigaciones IoT como es la captura de la memoria no volátil, y, segundo, acceder a un tipo de fuente de evidencia que, como ya hemos comentado, puede ser clave en este entorno: el tráfico de red. Con esto conseguimos asegurar que el investigador, como mínimo, tendrá acceso una fuente de evidencia. Su funcionamiento es el siguiente:

Tráfico de Red. Complementando lo descrito en la Sección IV-B, indicar que se pueden capturar varios protocolos a la vez, puesto que los procesos se ejecutan en segundo plano. La herramienta utilizada para capturar los paquete de red es *tcpdump* [20], que generalmente está disponible de forma nativa en sistemas Linux. Además, puesto que es desde el propio nodo edge desde donde se realiza la captura, esta queda almacenada en su sistema de ficheros, lo que también facilita al investigador la extracción de los datos adquiridos, pues solo tendrá que conectar la tarjeta microSD, la cual actúa como almacenamiento en la Raspberry Pi Modelo 3 B+, a su ordenador para acceder a los mismos.

Memoria no volátil. Como se ha indicado anteriormente, para poder realizar la adquisición remota es necesario que el dispositivo IoT permita el acceso remoto mediante SSH. Además, el comando *dd* [21], el cual se encuentra de forma nativa en sistemas Linux, debe poderse ejecutar. Para establecer la conexión con el dispositivo IoT, se emplea la librería Paramiko [22], para lo cual se necesita tener acceso al método de autenticación (en caso de que lo tenga) del dispositivo. Una vez conectado, FENIoT mostrará las particiones disponibles en el sistema para que el investigador seleccione la deseada. Tras ello, ejecutará el comando *dd* y, ayudándose del comando *netcat* [23], enviará el fichero capturado al nodo edge o al método de almacenamiento que se desee.

Preservación de las evidencias. Para asegurar la integridad de los datos adquiridos, por cada captura de red o imagen de disco se genera un log con la siguiente información:

- Fecha y hora en la que se inició la adquisición.
- Fecha y hora en la que finalizó la adquisición.
- Fichero que se ha generado tras la captura.
- Hashes MD5 y SHA-1 del fichero generado.

De este modo, FENIoT proporciona datos suficientes como para que el investigador pueda comprobar, en caso de que quiera analizar los ficheros adquiridos de forma independiente, si la integridad de los mismos se ha visto afectada.

V. PRUEBAS DE RENDIMIENTO

De cara a probar el funcionamiento de FENIoT y evaluar su rendimiento, la propuesta ha sido estudiada en distintos casos de estudio representando varios escenarios IoT. En esta sección, se describirán las pruebas realizadas y se mostrará la información más relevante que se puede extraer de ellas.

V-A. Escenario de Pruebas

Para ofrecer una evaluación lo suficientemente completa, se han diseñado tres escenarios distintos de evaluación. El objetivo es poder estudiar el comportamiento del sistema en distintos contextos IoT. En concreto, los escenarios son los siguientes:

- El Xiaomi Mi Smart Sensor Set [24], utilizado en el contexto smart home, compuesto por un nodo central, dos sensores de presencia, un interruptor inteligente y dos sensores de apertura de puertas y ventanas. El objetivo de este caso es trabajar con los protocolo de red Zigbee (usado para comunicarse entre nodo central y sensores) y WiFi (usado para enviar datos desde el nodo central a la nube).
- Un escenario IoT industrial simulado en openLEON [25], compuesto por un nodo central con un sistema basado en Linux, dos sensores comunicándose entre sí usando el protocolo Modbus, y dos sensores haciendo lo propio con el protocolo Message Queuing Telemetry Transport (MQTT). El objetivo en este escenario es trabajar con distintos protocolos de red y con una unidad IoT que proporciona acceso remoto mediante SSH.
- Un sistema de videovigilancia con varias cámaras Google Nest [26]. El objetivo de esta prueba es trabajar con tráfico WiFi y Bluetooth.

Todos ellos se evalúan usando FENIoT de forma reactiva y proactiva.

V-B. Detección de Anomalías

Para la detección de anomalías se ha utilizado como referencia el escenario IoT industrial. En la Figura 2, se muestra el resultado de usar el nodo para analizar el tráfico MQTT, tanto a nivel de paquete, como a nivel de conexión. En el primer caso, tras analizar todos los paquetes, se detecta que uno de ellos tiene un comportamiento anómalo. En el segundo caso, FENIoT es capaz de detectar los dos dispositivos que están utilizando el protocolo MQTT y, tras agrupar los paquetes capturados en conexiones y aplicar el algoritmo Isolation Forest sobre ellas, la solución detecta que la última de ellas tiene características notablemente diferentes a las demás (debido al bajo número de conexiones realizadas), por lo que la identifica como anómala.

Esta prueba se trata de un breve ejemplo con pocos paquetes de red, por lo que la cantidad de anomalías detectadas es baja,

	No.	Time	Source	Destination	Protocol	Length	Info	Anomaly
80	81	2021-07-01 23:51:39.358335064	172.17.0.4	172.17.0.2	MQTT	78	Subscribe Request (id=1) [topic]	-1

((a)) Anomalías detectadas en los paquetes MQTT

	Source	Destination	Connections	Anomaly
0	172.17.0.2	172.17.0.3	80	1
1	172.17.0.2	172.17.0.4	59	1
2	172.17.0.3	172.17.0.2	52	1
3	172.17.0.4	172.17.0.2	30	-1

((b)) Anomalías detectadas en las conexiones MQTT

Figura 2. Información generada por FENIoT en la fase de detección

pero sería perfectamente extrapolable a un caso con mayor cantidad de conexiones, paquetes y dispositivos.

V-C. Identificación de Dispositivos

Para mostrar el funcionamiento de FENIoT en el proceso de identificación de dispositivos, utilizamos el mismo escenario que en el caso anterior, puesto que es el que más protocolos utiliza. Como se puede apreciar en la Figura 3, una vez que el nodo realiza una captura inicial, este informa sobre los protocolos detectados y el número de paquetes totales intercambiados usando cada uno de ellos. Tras ello, pasa a proporcionar información sobre los dispositivos identificados, su ID, en este caso en forma de dirección IP, los paquetes enviados y recibidos y la fecha y hora en la que cada unidad envió el último paquete de red.

V-D. Adquisición de Fuentes de Evidencia

Para finalizar, se muestra el comportamiento de FENIoT en el proceso de adquisición. En este caso, se ofrecen ejemplos de los dos casos de estudio restantes. En la Figura 4 podemos observar, primeramente, el resultado de adquirir el tráfico de red Zigbee junto con el log generado tras su captura. En segundo lugar, vemos la capacidad del nodo para conectarse a un dispositivo IoT por SSH, listar sus particiones y realizar el proceso de adquisición[1].

VI. Conclusiones

En este trabajo se ha presentado una solución IoT para el desarrollo de investigaciones forenses en este entorno. Tras analizar las propuestas de la comunidad científica, se ha detectado que el uso de dispositivos externos para asistir en las investigaciones forenses es una solución interesante para facilitar el manejo de fuentes de evidencia. Generalmente, las soluciones adoptadas se han basado en el uso de servidores o algún tipo de software que necesita estar ejecutándose antes de que la investigación comience, por lo que su utilidad se ve limitada. Además, muchas de las propuestas se centran en la obtención de logs a partir de los datos generados en la red, añadiendo una capa de abstracción que impide acceder a los datos en bruto.

[1]Se ha recortado la figura para no sobrecargar el artículo, eliminando algunas de las particiones presentes en el sistema que no aportaban datos relevantes.

Tratando de combinar el uso de dispositivos externos en investigaciones forenses, y abordando algunas de las limitaciones detectadas en las propuestas de la comunidad forense, este artículo ha presentado un nodo IoT, denominado FENIoT, que permite asistir al investigador en las fases de detección, identificación y adquisición del proceso forense. Dicho nodo puede ser usado con un enfoque proactivo, instalándolo en la red IoT antes de que el incidente ocurra, por lo que puede contribuir a la detección del mismo, o de forma reactiva, utilizándose una vez que el proceso forense ha comenzado. En este segundo caso, aunque se pierde la capacidad de detección de anomalías, el nodo es capaz de asistir en el proceso de identificación y adquisición.

Para realizar la detección de incidentes, FENIoT hace uso del algoritmo Isolation Forest para localizar anomalías, tanto a nivel de red, mediante el análisis de los paquetes intercambiados en la misma, como a nivel de dispositivo, estudiando los sistemas de ficheros de los dispositivos que permiten la conexión remota. Durante la fase de identificación, el nodo ayuda a determinar el rango de la escena estudiando los paquetes de red intercambiados y enumerando los dispositivos detectados junto con información sobre la cantidad de tráfico enviado y recibido o la última hora de conexión. Finalmente, en la fase de adquisición, FENIoT permite la captura remota de los sistemas de ficheros de los dispositivos IoT de la red, así como la de los paquetes intercambiados en la misma. En ambos casos, se genera un log con los códigos hash MD5 y SHA-1, la fecha de comienzo y fin de adquisición y el nombre del archivo. Esto se realiza por cada fuente de evidencia capturada.

El nodo FENIoT ha sido evaluado en diversos casos de estudio, probando su utilidad para ser usados en procesos forenses, tanto en su vertiente proactiva como la activa. Además, su rendimiento se ha estudiado en entornos simulados y reales representando varios contextos IoT.

VI-A. Trabajo Futuro

De cara a extender la funcionalidad del nodo edge presentado en esta propuesta, se podrían abordar las siguientes temáticas:

- Añadir y evaluar algoritmos de detección de anomalías adicionales, de cara a determinar cuál es el que mayor rendimiento obtiene.

```
The protocol TCP has been detected with 192 packets exchanged

The protocol MQTT has been detected with 76 packets exchanged

The protocol Modbus/TCP has been detected with 181 packets exchanged

The following devices have been detected:

       Device
0  172.17.0.3
1  172.17.0.2
2  172.17.0.4
3  172.17.0.6
4  172.17.0.5

The number of packets sent by each device is:

            Packets Sent
Source
172.17.0.2           173
172.17.0.3           100
172.17.0.4            96
172.17.0.5            22
172.17.0.6            58

The number of packets received by each device is:

             Packets Received
Destination
172.17.0.2                276
172.17.0.3                 66
172.17.0.4                 65
172.17.0.5                 12
172.17.0.6                 30


The last packet sent by the device with IP 172.17.0.3 was in 2021-07-02 00:13:42.619244018 UTC Time
The last packet sent by the device with IP 172.17.0.2 was in 2021-07-02 00:13:42.737941166 UTC Time
The last packet sent by the device with IP 172.17.0.4 was in 2021-07-02 00:13:42.619419899 UTC Time
The last packet sent by the device with IP 172.17.0.6 was in 2021-07-02 00:13:42.725023755 UTC Time
The last packet sent by the device with IP 172.17.0.5 was in 2021-07-02 00:13:42.737954526 UTC Time
```

Figura 3. Resultado de usar FENIoT en el caso de estudio IoT industrial.

- Extender el estudio del tráfico WiFi para permitir el análisis de los paquetes de red cuando están encriptados.
- Implementar en el nodo la posibilidad de realizar ataques por diccionario o fuerza bruta para poder intentar conectar de forma remota a aquellos dispositivos que lo permiten, pero para los cuales no se dispone de los credenciales necesarios.
- Mejorar el proceso de preservación de las fuentes de evidencia empleando técnicas adicionales de preservación.
- Incluir una función de restauración de los dispositivos a partir de las adquisiciones del sistema de ficheros realizadas en la fase de monitorización.

Agradecimientos

Este trabajo ha sido realizado con la colaboración de Felix Freiling y Philipp Klein, miembros de Lehrstuhl für Informatik 1 de la Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). Además, esta investigación cuenta con el apoyo de la Universidad de Castilla-La Mancha, mediante los contratos con referencia 2021-POST-20518 y PI001482, el proyecto con referencia 2021-GRIN-31042, del del Ministerio de Asuntos Económicos y Transformación Digital, mediante el proyecto con referencia RTI2018-098156-B-C52, y de la Consejería de Educación, Cultura y Deportes de la Junta de Comunidades de Castilla-La Mancha, mediante los proyecto con referencias SBPLY/17/180501/000353 y SBPLY/21/180501/000195.

Referencias

[1] Knud Lasse Lueth. IoT Analytics, “State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time,” 2020, Last accessed on Sep. 2021. [Online]. Available: https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/

[2] Satyajit Sinha. Iot Analytics, “State of IoT 2021: Number of connected IoT devices growing 9 % to 12.3 billion globally, cellular IoT now surpassing 2 billion,” Last accessed on Sep. 2021. [Online]. Available: https://iot-analytics.com/number-connected-iot-devices/

[3] Dan Demeter and Marco Preuss and Yaroslav Shmelev, “IoT: a malware story - Securelist,” https://securelist.com/iot-a-malware-story/94451/, 2019.

[4] J. Wurm, K. Hoang, O. Arias, A. Sadeghi, and Y. Jin, “Security analysis on consumer and industrial iot devices,” in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2016, pp. 519–524.

[5] N.-A. Le-Khac, D. Jacobs, J. Nijhoff, K. Bertens, and K.-K. R. Choo, “Smart vehicle forensics: Challenges and case study,” *Future Generation Computer Systems*, vol. 109, pp. 500–510, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17322422

[6] C. W. Badenhop, B. W. Ramsey, B. E. Mullins, and L. O. Mailloux, “Extraction and analysis of non-volatile memory of the zw0301 module, a z-wave transceiver,” *Digital Investigation*, vol. 17, pp. 14 – 27, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287616300214

	No.	Time	Source	Destination	Protocol	Length	Info
0	1	2021-02-10 19:02:13.062753	0x0000	Broadcast	ZigBee	47	Command, Dst: Broadcast, Src: 0x0000
1	2	2021-02-10 19:02:28.067074	0x0000	Broadcast	ZigBee	47	Command, Dst: Broadcast, Src: 0x0000
2	3	2021-02-10 19:02:43.064974	0x0000	Broadcast	ZigBee	47	Command, Dst: Broadcast, Src: 0x0000
3	4	2021-02-10 19:05:26.769104	0xf969	0x0000	ZigBee	52	Data, Dst: 0x0000, Src: 0xf969
4	5	2021-02-10 19:05:26.771153	NaN	NaN	IEEE 802.15.4	5	Ack
5	6	2021-02-10 19:06:28.076042	0x0000	Broadcast	ZigBee	47	Command, Dst: Broadcast, Src: 0x0000
6	7	2021-02-10 19:08:13.083403	0x0000	Broadcast	ZigBee	47	Command, Dst: Broadcast, Src: 0x0000
7	8	2021-02-10 19:09:46.053523	0xf969	0x0000	IEEE 802.15.4	12	Data Request
8	9	2021-02-10 19:09:46.260254	0xf969	0x0000	ZigBee	74	Data, Dst: 0x0000, Src: 0xf969
9	10	2021-02-10 19:09:46.263007	NaN	NaN	IEEE 802.15.4	5	Ack

((a)) Tráfico ZigBee capturado.

```
Acquisition started: 2021-02-10 19:02:13.062753
Acquisition ended: 2021-02-10 20:09:13.274963
File generated: Capture_2021-02-10 19:02:13.062753.pcap
MD5 cheksum: ec03001e09077996f244e99393463412
SHA1 checksum: 19b9bea97bfd5fd563704a760d8327e0b0e66e28
```

((b)) Log generado por FENIoT tras realizar una adquisición.

```
Connecting to: 192.168.1.58

Connected to: 192.168.1.58

The following filesystems are available:

Filesystem     1K-blocks    Used Available Use% Mounted on

udev              393424       0    393424   0% /dev

tmpfs              92792    8552     84240  10% /run

/dev/mmcblk0p2  29699400 1120648  27057128   4% /writable

/dev/loop0         50176   50176         0 100% /

/dev/loop1        189696  189696         0 100% /lib/modules

/dev/mmcblk0p1    258095  128818    129278  50% /boot/uboot

Acquiring /dev/mmcblk0p1

524288+0 records in

524288+0 records out

268435456 bytes (268 MB, 256 MiB) copied, 16.0184 s, 16.8 MB/s


/dev/mmcblk0p1 acquired
```

((c)) Adquisición del sistema de ficheros.

Figura 4. Información generada por FENIoT en la fase de adquisición.

[7] J. Elstner and M. Roeloffs, "Forensic analysis of newer tomtom devices," *Digital Investigation*, vol. 16, pp. 29 – 37, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S174228761630010X

[8] A. Nieto, R. Rios, and J. Lopez, "A methodology for privacy-aware iot-forensics," in *2017 IEEE Trustcom/BigDataSE/ICESS*, Aug 2017, pp. 626–633.

[9] M. Hossain, Y. Karim, and R. Hasan, "Fif-iot: A forensic investigation framework for iot using a public digital ledger," in *2018 IEEE International Congress on Internet of Things (ICIOT)*, 2018, pp. 33–40.

[10] E. Al-Masri, Y. Bai, and J. Li, "A fog-based digital forensics investigation framework for iot systems," in *2018 IEEE International Conference on Smart Cloud (SmartCloud)*, 2018, pp. 196–201.

[11] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[12] Raspberry Pi Foundation, "Buy a Raspberry Pi 3 Model B – Raspberry Pi," https://www.raspberrypi.org/products/raspberry-pi-3-model-b/, 2020.

[13] Canonical Group, "Ubuntu Core - Ubuntu," https://ubuntu.com/core.

[14] "scikit-learn: machine learning in Python — scikit-learn 1.0.2 documentation." [Online]. Available: https://scikit-learn.org/stable/

[15] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data*, 2012.

[16] ElevenPaths, "Redes más seguras con Machine Learning: Una prueba de concepto," http://blog.elevenpaths.com/2017/07/redes-mas-seguras-con-machine-learning.html.

[17] Alejandro Correa Bahnsen, "Isolation forests for anomaly detection improve fraud detection," http://blog.easysol.net/using-isolation-forests-anamoly-detection/.

[18] T. Yiu, "Understanding Random Forest," Sep. 2021. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[19] C. Stanton, G. Katz, and D. Song, "Isolation forest for anomaly detection," 2015.

[20] tcpdump, "Tcpdump/Libpcap public repository," https://www.tcpdump.org, 2020. [Online]. Available: https://www.tcpdump.org

[21] Computer Hope. Computerhope.com, "Linux and Unix dd Command," http://www.computerhope.com/unix/dd.htm, 2020.

[22] J. Forcier, "Welcome to Paramiko! — Paramiko documentation." [Online]. Available: https://www.paramiko.org/

[23] "The GNU Netcat – Official homepage." [Online]. Available: http://netcat.sourceforge.net/

[24] Xiaomi, "Mi Global Home," 2021. [Online]. Available: https://www.mi.com/global/mi-smart-sensor-set/

[25] C. Fiandrino, A. Pizarro, P. Mateo, C. Andrés Ramiro, N. Ludant, and J. Widmer, "openleon: An end-to-end emulation platform from the edge data center to the mobile user," *Computer Communications*, vol. 148, pp. 17–26, 12 2019.

[26] G. Inc., "Nest Cam (de exterior o interior, con batería)." [Online]. Available: https://store.google.com/es/product/nest_cam_battery

Diseño de un ecosistema DTN basado en NFV para el análisis de ciberseguridad en redes 5G

Mario Sanz-Rodrigo, Diego Rivera-Pinto, José Ignacio Moreno-Novella, Xavier Larriva Novo
Universidad Politécnica de Madrid
mario.sanz@upm.es, diego.rivera@upm.es, joseignacio.moreno@upm.es,
xavier.larriva.novo@upm.es

Resumen- **Una red gemela digital (Digital Twin Network – DTN) es una representación virtual de una red de telecomunicaciones que modela con precisión los dispositivos, los enlaces de comunicación, el entorno operativo y las aplicaciones que se ejecutan en la red. Al replicar diferentes entornos en un laboratorio y ejecutar múltiples escenarios, los gemelos digitales ofrecen una forma rentable de evaluar el rendimiento, predecir el impacto de los cambios ambientales (como las amenazas cibernéticas) y optimizar los procesos de la red, así como la toma de decisiones en consecuencia. El proyecto B5GEMINI, pretende el desarrollo de una DTN aplicada a un entorno de núcleo 5G y su evolución hacia 6G donde se apliquen casos de uso de interés para el despliegue de escenarios avanzados tales como la ciberseguridad o la gestión de red, como principal objetivo de este trabajo se pretende tener un sistema de auto captación de entornos reales, el cual modele el gemelo digital a desplegar en la infraestructura propuesta, cubriendo tanto la parte de virtualización como la de interconexión bidireccional entre el gemelo físico y el gemelo digital.**

Index Terms- **Gemelos Digitales, DTN, redes 5G, emulación, redes, ciberseguridad**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

En la situación actual, donde la tecnología 5G en modo Standalone (SA) estará disponible comercialmente en los próximos años, este trabajo se centra en la investigación y desarrollo de un entorno tecnológico, el cual pese a tomar como caso de uso las redes 5G, sirva como plataforma de soporte capaz de emular un escenario de red específico de forma dinámica y generar el tráfico requerido para ser utilizado posteriormente para diferentes actividades de análisis centradas en ciberseguridad y gestión de red, tomando la característica principal de los gemelos digitales, la cual interconecta de manera bidireccional ambos mundos, el físico y su contraparte digital. La característica clave del proyecto B5GEMINI consiste en desarrollar un núcleo de red bajo tecnología 5G/6G con la capacidad de desarrollar experimentos bajo una red replicable, de modo que las mismas condiciones ambientales permitan la evaluación y análisis de diferentes respuestas basadas en patrones estadísticos similares, y el impacto de variaciones controladas en el ambiente. El proyecto tiene como objetivo la innovación en varios campos de la nueva red 5G/6G como [1],[2],[3],[4],[5] y [6]:

- Evaluación de soluciones de circuito cerrado para el Monitorización continua y automatización de redes involucradas en los gemelos, en las que los motores de inteligencia artificial permitan el análisis del estado de la red en base a los datos de gestión recopilados en la red y ofrecer retroalimentación al sistema para la toma de decisiones.
- Evaluación del rendimiento, escalabilidad y resistencia cibernética de las redes antes del lanzamiento a gran escala de actualizaciones / tecnologías nuevas, como 5G/6G, IoT, V2X o nuevas versiones en los elementos de red.
- Entorno seguro para las pruebas de seguridad cibernética sin afectar al gemelo físico.
- Entorno controlado para la formación en nuevas tecnologías, operaciones, ciberdefensa, etc, con opción de aislar el gemelo digital del gemelo físico.

II. Análisis de tecnologías

A. *Network Functions Virtualization - NFV*

La tecnología NFV [7] permite el despliegue de servicios y funciones de red basados en software, las cuales pueden ser desplegadas en hardware no específico. NFV hace uso de funciones de red denominadas VNFs (Virtual Network Functions). Estás VNFs se encargan de realizar diferentes tareas de red, como pueden ser routers, switches, firewall, etc. En este contexto, la combinación de diferentes VNFs dentro de la tecnología NFV permiten implementar lo que se conoce como NS (Network Service) virtualizado.

NFV es actualmente liderada por la ETSI (European Telecommunications Standars Institute) [7], la cual propone la siguiente arquitectura de referencia en la cual aparecen definidos los distintos bloques funcionales e interfaces para la gestión y orquestación de servicios de red virtualizados.

- ***Virtualizated Network Funciton (VNF)***. Implementación virtual de una función de red vía software. Hace uso de las máquinas virtuales proporcionadas por el bloque NFVI.
- ***Network Functions Virtualization Infrastructure (NFVI).*** Constituye la base general de la arquitectura NFV. Encargada de contener el hardware para el alojamiento de las máquinas virtuales, el software dedicado a la virtualización y los recursos virtualizados.
- ***Management and Orchestration (MANO)***. Plataforma encargada de proporcionar un marco para la gestión de la infraestructura general de NFV y los servicios virtualizados en ella. Cubre tanto el apartado de orquestación como el de la gestión del ciclo de vida de

los recursos interactuando tanto con el bloque NFV como con las distintas VNFs presentes en la arquitectura NFV. Se encarga de realizar el control sobre todas las entidades dentro de la arquitectura NFV. A su vez, dentro de MANO, nos encontramos con los siguientes bloques funcionales:

- ***VNF Manager (VNFM).*** Bloque encargado de la gestión de las VNFs, entre sus funciones destacan:
- ***NFV Orchestrator (NFVO).*** Bloque encargado de gestionar los NS (Network Services).
- ***Virtual Infrastructure Manager (VIM).*** Bloque encargado de la gestión de recursos NFVI en un dominio concreto. Es posible la coexistencia de distintos VIM en una misma arquitectura NFV, encargándose cada uno de ellos, de gestionar su dominio NFVI asociado. Las funciones más relevantes de este bloque son:

B. Open Source MANO - OSM

Open Source MANO (OSM) [8] es un proyecto abierto promovido por la ETSI, el cual ofrece una pila de software Open Source para la gestión y orquestación de NFV, capaz de consumir modelos de información publicados abiertamente, disponibles y alineado con los modelos de información ETSI NFV.

OSM hace referencia a la implementación, orquestación y administración del ciclo de vida de los recursos lógicos y físicos, así como de la infraestructura NFV. Este proyecto incluye la creación de instancias de VNF, encadenamiento de servicios VNF, supervisión, reubicación y cierre de instancias. OSM interactúa con OSS/BSS, el cual permite a la NFV ser integrada en un entorno de gestión más amplio.

C. OpenStack

OpenStack [9] es una solución Open Source que engloba un conjunto de herramientas software enfocadas a la construcción y administración de plataformas de computación, tanto en nubes públicas como privadas. Actualmente se trata de una plataforma clave en entornos NFV, ya que se utiliza como VIM, ofreciendo una interfaz estandarizada para la administración, monitorización y evaluación de los recursos dentro de la infraestructura NFV. A continuación, se muestran los componentes que conforman OpenStack.

- *Nova (Compute Service).* Módulo encargado del control general de OpenStack, permite trabajar con diferentes hypervisores, encargado de la creación y gestión de instancias.
- *Swift (Object Store Service).* Módulo encargado del almacenamiento de objetos y archivos. Para facilitar las tareas de escalado, es posible referirse a un identificador único que hace referencia al archivo o elemento de información y OpenStack se encarga de almacenar dicha información.
- *Cinder (Block Storage Service).* Módulo encargado de proporcionar dispositivos de almacenamiento a nivel de bloque.
- *Neutron (OpenStack Networking Service).* Módulo encargado de la gestión relacionado con el networking.
- *Horizon (Dashboard).* Intefáz gráfica para gestionar el acceso, provisión, etc.
- *Keystone (Identity Service).* Módulo encargado con la autenticación de usuarios y políticas de acceso.
- *Glance (Image Service).* Módulo encargado de la gestión de imágenes de sistemas operativos en formato plantilla.
- *Ceilometer (Telemtry Service).* Módulo encargado de proporcionar servicios de telemetría.
- *Heat (Orchestration Service).* Módulo de orquestación OpenStack, permite almacenar los requisitos de aplicación en un archivo, el cual define los recursos necesarios para dicho elemento.

III. SISTEMA PROPUESTO

La propuesta de diseño basada en un sistema modular y escalable para la creación de una arquitectura DTN aplicada a redes 5G, tiene como objetivo modelar cada uno de los elementos presentes en una red 5G en forma de gemelo digital, estableciendo un modelo completo de DTN capaz de emular con precisión el comportamiento de una red 5G bajo determinados casos de uso o estudio, los cuales harán uso de la DTN para poder aplicar tareas basadas en Inteligencia Artificial. La Figura 1 muestra el diseño conceptual del sistema completo. A continuación, se revisarán los distintos módulos de la arquitectura DTN B5GEMINI-INFRA.

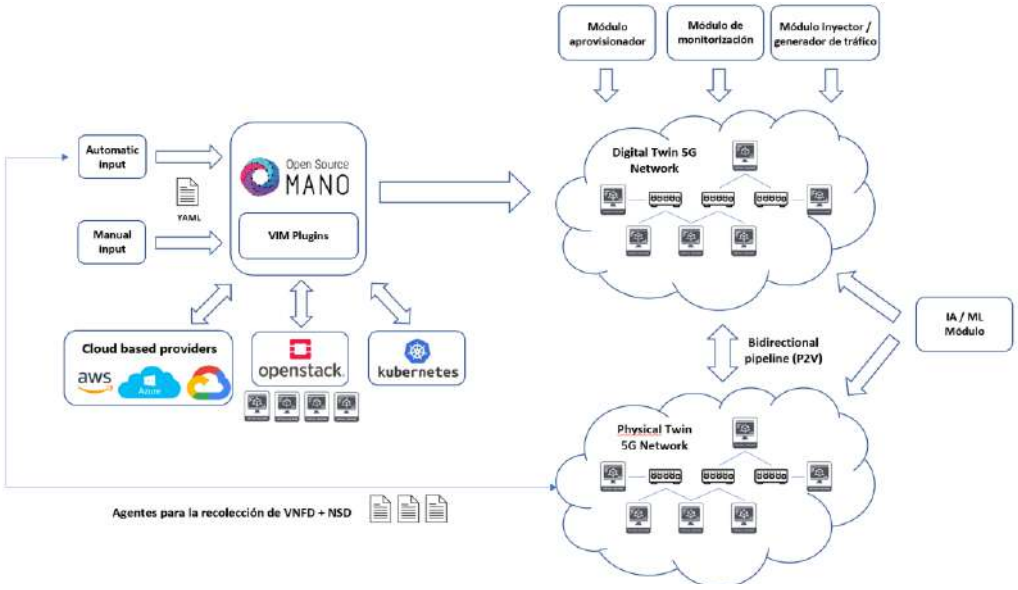


Fig. 1. Diagrama arquitectura DTN B5GEMINI.

A. Input del sistema para la generación de la DTN

En la fase de creación de los diferentes DT que conformarán la DTN se establecen dos entradas generales del sistema, entrada automática y entrada manual.

- Entrada automática: Se define con el objetivo de obtener una réplica exacta del objeto físico a modelar. Para ello, se propone el desarrollo de agentes inteligentes que puedan ser desplegados en la red objetivo, encargados de recolectar toda la información necesaria para la generación del DT (información topológica, información de hardware y software, estados, etc). Esta información se introduce en el módulo de implementación (OpenMANO) en formato YAML para rellenar las plantillas de descriptores VNFD (Virtual Network Function Descriptor) y NSD (Network Service Descriptor) con el objetivo de desplegar las VM asociadas a través de OpenStack.

- Entrada manual: En el caso de no tener los permisos suficientes, o querer completar la información, recolectada por los agentes inteligentes descritos en el punto anterior, se habilita la opción para que los operadores del sistema carguen estos archivos manualmente, evitando la limitación en la capacidad de emulación del sistema DTN

B. Módulo de despliegue NFV basado en OpenMANO y OpenStack

Una vez finalizada la fase inicial de recolección de información para la generación de los DT, se utiliza el módulo de despliegue, basado en arquitectura NFV junto a OpenMANO, utilizando las tecnologías OpenStack, Kubernetes y opcionalmente Cloud (AWS, Google Cloud, Azure, etc) como sistemas VIM (Virtual Intrastructure Manager), el cual supervisará la implementación de la infraestructura virtual que albergará los distintos DT en formato VNF para la creación de la DTN completa. Una de las ventajas de realizar el despliegue con estas tecnologías es la capacidad de realizar slicing, permitiendo la creación de múltiples despliegues DTN aislados entre sí. Este módulo de despliegue hace uso de un catálogo de imágenes y contenedores, los cuales servirán de base para el aprovisionamiento y configuración de los diferentes DT a implementar mediante tecnologías como Kubernetes o Docker.

Para el caso concreto de DTN aplicado a redes 5G, se propone el uso VNFs de la red core 5G implementadas mediante Kubernetes. Permitiendo desplegar todas las VNFs en una única instancia virtual (arquitectura monolítica) o de manera distribuida, permitiendo la comunicación entre los distintos DT que albergan las VNFs correspondientes. Dentro de la DTN 5G Core, con el objetivo de aplicar contexto real, se permite la inclusión de dispositivos, tanto virtuales como hardware, que actúan como clientes o servidores para la emulación de distintos tipos de tráfico (video Streaming, páginas web, etc) dentro de la DTN. A su vez, se desplegaría un orquestador que se comunica con todos los dispositivos DT que conforman la DTN a través de una red de administración con la capacidad de monitorear y capturar todo el tráfico con el objetivo de realizar análisis, obtención de estadísticas y la generación de datasets para su posterior uso por los algoritmos de IA.

C. Módulo de configuración y aprovisionamiento DT

Tras completar la fase de despliegue de infraestructura, este módulo es el encargado, en base a la información proporcionada a la entrada del sistema en formato YAML, de modelar cada uno de los DT y configurar las interconexiones necesarias para la emulación completa de la red 5G objetivo. La infraestructura 5G objetivo se basa en la virtualización de la red central 5G. Las funciones de red (VNF) que componen el núcleo interactúan en el entorno virtual de la infraestructura DTN B5GEMINI. La implementación de estas VNFs se basa en el proyecto Open Source free5GC [10]. Este proyecto ofrece una implementación abierta de las principales VNF necesarias para la operación del core 5G.

D. Módulo de monitorización de red

Este módulo es el responsable de monitorizar el funcionamiento completo de toda la información intercambiada dentro de la DTN. Una vez provisionados y configurados todos los gemelos, el módulo de monitorización controlara la activación de las funcionalidades de port-mirroring dentro de cada una de las subredes presentes en la DTN con el fin de poder utilizar ese tráfico de información para la generación de datasets que permitan la mejora en la emulación de la DTN.

E. Módulo de inyección y generación de tráfico

Este módulo ofrece la opción de habilitar la generación o inyección de tráfico en base a diferentes modelos de tráfico 5G sobre la DTN, permitiendo la validación y el análisis del desempeño de la red. El objetivo principal de este módulo es proporcionar un método de inyección de tráfico en la infraestructura virtualizada de la red central 5G sin tener que lidiar con redes de acceso radio reales y hardware 5G. Este módulo actúa como un generador de tráfico NAS de señalización que puede comunicarse con la VNF AMF en el núcleo 5G y emular las operaciones del equipo de usuario en un entorno real. Este módulo puede realizar la gestión de sesiones, registro y registro de UE, etc. Adicionalmente, puede mantener túneles GTP con la VNF UPF de la red central y enviar datos a través de ellos. Los datos de usuario que se envían a través de este módulo, que actúa como intermediario, se capturan en una interfaz de red y pueden tener como fuente cualquier dispositivo virtual o máquina hardware.

Tanto la emulación de señalización NAS (Signalling Traffic Generator – STG) como el data broker de usuarios (User Traffic Generator – UTG) se implementan como un solo proceso, habilitando la posibilidad de registrar UEs, establecer sesiones PDU y enviar tráfico de usuarios a una red desde la misma herramienta software, asociado la información del tráfico de señalización al túnel generado para el tráfico de los usuarios.

F. Pipeline bidireccional

Una vez desplegada y aprovisionada la infraestructura DTN, es necesario establecer los diferentes pipelines a utilizar en el sistema completo. Según el tipo de elementos

interconectados y sus necesidades de comunicación, se distinguen dos tuberías o pipelines diferenciados: V2V (Virtual a Virtual) y P2V (Físico a Virtual). La primera de ellas se realiza dentro de la DTN, interconectando cada uno de los DT que componen el sistema completo. Estos pipelines permiten reflejar el comportamiento de la comunicación que tiene lugar en el mundo físico, sin la limitación de tiempo presente en él, permitiendo la comunicación de grandes flujos de datos en tiempos reducidos, lo que facilita la tarea de emulación, acelerando la obtención de resultados. La segunda, habilita la interconexión entre el mundo virtual de la DTN y el mundo físico, permitiendo un bucle de retroalimentación continua entre la DTN y su contraparte física, proporcionando al sistema capacidades de coevolución y cooperación continuas.

G. Módulo de Inteligencia Artificial (IA) / Machine Learning (ML)

Este módulo es el encargado de realizar acciones inteligentes sobre los gemelos, como puede ser la optimización de políticas de red (enrutamiento de tráfico, estrategias de implementación de recursos, etc) o predecir fallas en la red. Para este propósito, este módulo monitorea continuamente la DTN y actuará en consecuencia para realizar tareas arbitrarias. Estos cambios se replicarán en el mundo físico a través del pipeline P2V.

IV. Casos de uso aplicados a ciberseguridad en 5G

A través del sistema propuesto basado en DTN, enfocado al análisis desde la perspectiva de la ciberseguridad en redes 5G, se plantean dos casos de uso, descubrimiento de cripto minado malicioso en el core 5G y los ataques a infraestructura DNS, en los que este tipo de tecnologías pueden aportar un gran valor. En el primero de los casos, debido a que el core 5G, específicamente los microservicios y la arquitectura que componen permiten el uso de soluciones de nube pública o hibrida, las cuales pueden facilitar la introducción de malware, como la criptominería. En este tipo de entornos, tanto el tráfico legal como el ilegal utilizan tráfico encriptado. El entorno DTN propuesto en este trabajo permitiría las capacidades de analizar mediante IA / ML los patrones de tráfico con el objetivo de realizar acciones de prevención sobre el gemelo físico. El segundo caso de uso, parte de las nuevas implementaciones asociadas a 5G, en este caso, en uno de los componentes SBA (Service-Based Architecture) de la arquitectura 5G, los servidores DNS. En esta tecnología se adopta la implantación de DoH (DNS Over HTTPS) (RFC8484), en la cual el entrenamiento de la DTN permitiría la detección y mitigación de ataques DNS a estas nuevas implementaciones, realizando acciones sobre los gemelos físicos a través del bucle de realimentación continua propio de la tecnología DTN.

V. Conclusiones

La implantación de este tipo de tecnologías, aparte de brindar grandes ventajas en ámbitos como el desarrollo, la ciberseguridad, la validación, la certificación o el análisis de datos recolectados en redes reales, presenta una serie de retos tecnológicos importantes los cuales han de tenerse en cuenta.

El primero de ellos, se enfoca en el pipeline de comunicación P2V que intercomunica ambos gemelos, el digital y el físico. Debido a que se están conectados sistemas reales, a sistemas virtualizados, típicamente en nubes, es de vital importancia establecer mecanismos de seguridad que permitan cumplir con la integridad y confidencialidad de los datos que fluyen entre ambos gemelos. Otro punto importante, dentro de la comunicación P2V son los tiempos de latencia, pese a no ser sistemas de tiempo real, para sacar el máximo partido a los resultados obtenidos del procesamiento de datos por parte de los modelos de IA y ML es crucial contar con enlaces de baja latencia, para poder adaptar de forma dinámica tanto el gemelo digital como el gemelo físico. El segundo reto, se enfoca en la recolección y procesamiento de datos físicos, ya que las redes reales están compuestas por una gran cantidad de dispositivos, tanto hardware como softwares heterogéneos, los cuales han de ser recolectados, analizados y formateados para que los módulos de la DTN puedan arrojar información útil. Por último, el último reto se enfoca en el propio modelado y creación de la DTN, se trata de un sistema vivo, el cual tiene que actualizarse continuamente, no únicamente a nivel de datos que recibe o envía, sino a cualquier variación o modificación de la topología de red física que está emulando, alcanzando un compromiso de coste y recursos dedicados para evitar caer en la paradoja de replicar completamente la red real.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el Ministerio de Asuntos Económicos y Transformación Digital del Gobierno de España a través del programa UNICO-5G I+D, a través del contrato B5GEMINI-INFRA (TSI-063000-2021-81) y mediante el programa Horizon 2020 a través del proyecto SPIDER (Grant agreement 833685)

Referencias

[1] H. X. Nguyen, R. Trestian, D. To and M. Tatipamula, "Digital Twin for 5G and Beyond," in IEEE Communications Magazine, vol. 59, no. 2, pp. 10-15, February 2021, doi: 10.1109/MCOM.001.2000343.

[2] Concepts of Digital Twin Network. [Available: https://tools.ietf.org/id/draft-zhou-nmrg-digitaltwin-network-concepts-03.txt]

[3] Vakaruk, S., Mozo, A., Pastor, A. & Lopez, D. R. "A Digital Twin Network for Security Training in 5G Industrial Environments", IEEE International Conference on Digital Twins and Parallel Intelligence, July 2021, Beijing, China

[4] Pastor, A., Mozo, A., Lopez, D. R., Folgueira, J., & Kapodistria, A. (2018, August). The Mouseworld, a security traffic analysis lab based on NFV/SDN. In Proceedings of the 13th International Conference on Availability, Reliability and Security (pp. 1-6).

[5] Digital Twin Network: Concepts and Reference Architecture. [Available: https://datatracker.ietf.org/doc/draft-zhou-nmrg-digitaltwin-network-concepts/] [Date: November 2021]

[6] Y. Wu, K. Zhang, and Y. Zhang, 'Digital Twin Networks: A Survey', *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13789–13804, Sep. 2021, doi: 10.1109/JIOT.2021.3079510.

[7] ETSI GS NFV 002, "Network Functions Virtualization (NFV); Architectural Framework," v. 1.1.1, Dec. 2014

[8] OpenStack URL: http://www.openstack.org/

[9] Open Source MANO, White Paper, ETSI OSM Community, Apr. 2017. [Online]. Available: https://osm.etsi.org/images/OSM-WhitepaperTechContent-ReleaseTWO-FINAL.PDF

[10] free5GC: an open-sourec 5G core network. Retrieved April 19, 2022 from https://www.free5gc.org/

Advisory: Análisis de vulnerabilidades en proyectos de desarrollo software

Antonio Germán Márquez Trujillo, Ángel Jesús Varela-Vaca, José A. Galindo, María Teresa Gómez López, y David Benavides
IDEA Research Group, Universidad de Sevilla
{amtrujillo, ajvarela, jagalindo, maytegomez, benavides}@us.es

***Abstract*—La seguridad se ha convertido en un factor crucial en el desarrollo de los sistemas software. La cantidad de dependencias existentes en los sistemas software se está convirtiendo en una fuente de innumerables errores y vulnerabilidades. En el pasado, la comunidad de líneas de producto ha propuesto varias técnicas y mecanismos para hacer frente a los problemas que surgen al tratar la variabilidad y la gestión de dependencias en dichos sistemas. En este artículo presentamos *Advisory*, una solución que permite el análisis automatizado de las dependencias en busca de vulnerabilidades dentro de los proyectos de software basándose en técnicas de la comunidad de líneas de producto. *Advisory* inspecciona primero las dependencias del software, luego genera un grafo de dependencias, al que posteriormente se le atribuye información de seguridad sobre las vulnerabilidades y se traduce a un modelo formal, en este caso, basado en SMT. Por último, *Advisory* ofrece un conjunto de operaciones de análisis y razonamiento sobre estos modelos que permiten extraen información útil sobre el conjunto de vulnerabilidades del espacio de configuraciones del proyecto, así como de información para el asesoramiento sobre el riesgo de seguridad de estos proyectos y sus posibles configuraciones. Así mismo comparamos nuestra solución con otras existentes en el mercado, en las que observamos que nuestra solución es capaz de detectar más vulnerabilidades en las dependencias en diferentes proyectos.**

***Index Terms*—Proyecto Software, Librería, Dependencia, Vulnerabilidad, CVE, Seguridad, Verificación, Riesgo, Impacto**

Tipo de contribución: *Investigación original*

I. Introducción

Los proyectos software normalmente delegan gran parte de la funcionalidad en librerías externas, lo que hace que las vulnerabilidades de éstas, puedan afectar al proyecto en desarrollo. En la actualidad, se identifican múltiples vulnerabilidades cada día [1] que deben ser conocidas y gestionadas rápidamente por los desarrolladores. Siendo conscientes de que las cadenas de ciberataques utilizadas por los atacantes para penetrar en los sistemas son cada vez más sofisticadas [2]. Así, los atacantes puede hacer de versiones de dependencias con vulnerabilidades conocidas, como la reciente vulnerabilidad CVE-2021-44228[1] detectada en Log4j[2], que ha afectado al menos a 186,352 proyectos en el ecosistema Java [3] debido a la complejidad del análisis de sus dependencias. Por tanto, una mala configuración (según OWASP Top-10 vulnerabilidades) en un componente software (dependencia), puede utilizarse como punto de entrada (vector de ataque) para un atacante. Debido a la gran variedad de opciones de configuración de las dependencias es todo un reto analizar las posibles vulnerabilidades de un proyecto software [4][5][6].

Los sistemas de alta variabilidad (Variability-intensive System - VIS) son aquellos sistemas software que, para funcionar de manera correcta, deben gestionar y lidiar con un gran número de dependencias [7][8]. En la literatura encontramos proyectos con centenares de dependencias y opciones de configuración, como por ejemplo el Kernel de Linux con más de 10^{60} configuraciones distintas. Una configuración se define en este contexto como una combinación válida de versiones de las librerías y artefactos software de los que un proyecto es dependiente. Esta cantidad de dependencias y librerías dificulta que los desarrolladores sean conscientes de qué vulnerabilidades afectan al software que están desarrollando y cómo tomar medidas para paliar los riesgos de seguridad.

Para ayudar a encontrar vulnerabilidades en las dependencias de un proyecto software encontramos soluciones tecnológicas como Snyk[3], OWASP Dependancy Check[4] o DependantBot de Github[5] que ayudan en la identificación de vulnerabilidades debido al uso de dependencias en un proyecto. No obstante, las herramientas y soluciones existentes en el mercado tienen ciertas limitaciones, por ejemplo, no permiten explorar el espacio completo de configuraciones existentes dada la complejidad del mismo [9]. Esta diferencia hace que pueda haber configuraciones que se estén usando y que sean vulnerables, pero no sean detectadas por estos sistemas. Normalmente, y hasta donde sabemos, las herramientas actuales se centran en el análisis de una única configuración dejando al lado, por ejemplo, los posibles entornos de despliegue, que podrían o no estar afectados por vulnerabilidades.

Dada la dificultad del análisis de dependencias de manera manual, la comunidad de líneas de producto propuso el análisis automático de la variabilidad en los VIS, por ejemplo con técnicas como los AAFM (Automated Analysis of Feature Models) [10]. Los AAFM habilitan el razonamiento sobre los VIS mediante el uso de sistemas de inteligencia artificial o algoritmos ad-hoc, para extraer información relevante del conjunto de dependencias descritas en un VIS. Unido al análisis de variabilidad, han surgido aproximaciones que intentan analizar

[1]https://nvd.nist.gov/vuln/detail/CVE-2021-44228
[2]https://logging.apache.org/log4j/2.x/
[3]https://www.snyk.io
[4]https://owasp.org/www-project-dependency-check/
[5]https://github.com/dependabot

las vulnerabilidades de una línea de productos software para optimizar el conjunto de pruebas a realizar [11].

En este artículo presentamos Advisory, una solución que permite analizar y razonar sobre el espacio completo de configuraciones de las dependencias de un proyecto software teniendo en cuenta sus vulnerabilidades. Para ello, Advisory se presenta como una solución para:

- Modelar las dependencias, es decir, el espacio de configuraciones de un proyecto software, y atribuir ese espacio de configuración con información de seguridad relacionada con sus vulnerabilidades.
- Habilitar técnicas y operaciones que permitan razonar sobre el espacio de dependencias de un proyecto software teniendo en cuenta la información de seguridad relacionada con las vulnerabilidades (p.ej., no usar Log4j).
- Evaluar y verificar como se comporta nuestra solución Advisory con respecto a las soluciones existentes en el mercado.

El resto del artículo está organizado de la siguiente forma: La sección II presenta un ejemplo motivador usado a lo largo del artículo como ejemplo. La sección III presenta Advisory, una solución para el análisis de proyectos software que tiene en cuenta las dependencias y vulnerabilidades que pueden afectar al mismo. La sección IV detalla resultados empíricos sobre la solución propuesta. Finalmente, la sección VI presenta nuestras conclusiones y lecciones aprendidas.

II. Ejemplo motivador

Los proyectos de desarrollo software actuales se construyen sobre otras herramientas existentes, a conocidas como dependencias. Una dependencia d_x puede tener asociado un conjunto valores de versión V_x válidas para el proyecto. Esto es, $d_x \mapsto V_x$. Para nuestra dependencia d_x, todas las configuraciones estarán definidas por un conjunto de pares de la forma $\{(d_x, v_1), \ldots, (d_x, v_n)\}$ con respecto al conjunto de versiones válidas. La combinatoria entre las diferentes versiones de las diferentes dependencias definen el espacio de configuración del proyecto completo. Si nuestro proyecto tuviera tres dependencias d_x, d_y, y d_z el espacio de configuraciones vendría definido de la siguiente manera: $\{d_x \mapsto V_x\} \times \{d_y \mapsto V_y\} \times \{d_z \mapsto V_z\}$. Por ejemplo, supongamos que tenemos las dependencias d_x y d_y, tomando cada una un número diferente de versiones en el rango 1.0 a 2.0 incluidas, habiendo hasta n versiones intermedias; el espacio de configuraciones completo del proyecto se puede representar de la siguiente manera: $\{(d_x, 1.0)$, $(d_y, 1.0)$,..., $(d_x, 1.n)$ y $(d_y, 1.n)$,..., $(d_x, 2.0)$ y $(d_y, 2.0)\}$.

Este espacio de configuraciones permite al desarrollador elegir para su proyecto una determinada configuración que se adapte a su entorno. Podemos pensar todo esto para un entorno de despliegue o integración continuo (CI/CD), que dé servicios a clientes. Sin embargo, el espacio de configuraciones crece exponencialmente en función del número de versiones de cada dependencia, quedando para n dependencias:

$$proyecto_{EspacioConfig} = \prod_{i=1}^{n} |V_i| \quad (1)$$

Por ejemplo, en el caso de tener 4 dependencias con 10 versiones cada una, tendríamos teóricamente 10.000 configuraciones posibles. Esta cota superior sobre el número de configuraciones se puede reducir mediante distintas restricciones que se suelen especificar en un archivo dentro de los proyectos. Por ejemplo, la dependencia d_x debe ser mayor o igual que la versión 1.5, quedando de la siguiente manera $d_x \geq 1.5$, lo que implica que la elección de dicha versión continúe siendo amplia. Esto requiere un análisis complejo de qué versiones son o no son válidas. El hecho de tener un espacio de configuraciones elevado dificulta al desarrollador la elección de la óptima para sus necesidades.

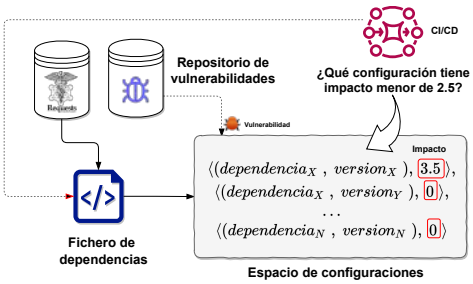


Fig. 1. Ejemplo motivador

Como ya hemos comentado, las dependencias pueden verse afectadas por vulnerabilidades conocidas. Actualmente el *Common Vulnerabilities and Exposures (CVE)* de Mitre[6] es el estándar de facto usado para representar la información de las vulnerabilidades, y es usada en casi todas las bases de datos como la de *National Vulnerability Database (NVD)* de NIST[7]. Generalmente, un CVE viene definido por un identificador de la forma CVE–$\langle year \rangle$–$\langle identificador \rangle$, por ejemplo, CVE-2022-27528. Además, cada CVE tiene asociado una descripción detallada, un impacto o severidad expresado mediante el Common Vulnerability Scoring System (CVSS)[8], y una lista de Common Platform Enumeration (CPE)[9] con características de las aplicaciones software, hardware o sistemas que se ven afectados por esta vulnerabilidad. Por ejemplo, la vulnerabilidad CVE-2021-43546, tiene un impacto CVSS de 4.3 sobre 10 (nivel medio), y uno de sus CPE es para la aplicación Mozilla Firefox, expresado de la siguiente manera: *cpe:2.3:a:mozilla:firefox:*:*:*:*:*:*:*:*:*.*

Como hemos comentado antes, las vulnerabilidades tienen asociadas un nivel de impacto o severidad (CVSS) que hacen que debamos tenerlas en cuenta para evaluar la seguridad de las configuraciones seleccionadas en los proyectos. Con lo que a la complejidad de seleccionar una configuración, hay que añadirle la consideración de evaluar el impacto producido por las vulnerabilidades asociadas.

En la Fig. 1 tenemos un ejemplo donde un sistema CI/CD toma un fichero de dependencias que usa la dependencia

[6]http://cve.mitre.org/
[7]http://nvd.nist.gov
[8]https://www.first.org/cvss/
[9]https://cpe.mitre.org/

de Requests de Python. El paquete Requests[10] de llamadas HTTP, es usado en aproximadamente 1.200.000 repositorios y 65.000 paquetes[11], lo que da una idea de su repercusión sobre otros proyectos. Podemos comprobar que Requests a su vez depende (indirectamente) de urllib3 y flask. A partir de ahora, usaremos Requests, urllib3 y flask, y asumiremos un fichero de dependencias que define las siguientes dependencias: $urllib3 \geq 1.21.1$ y $urllib3 < 1.27$, y $flask > 1.0$ y $flask < 2.0$. Si analizamos el espacio de configuraciones, podemos comprobar que hay un total de 29 versiones posibles para urllib3 y 9 para flask, lo que significa un total de 261 configuraciones de versiones posibles. Además, de entre las 261 configuraciones detectamos que al menos una vulnerabilidad está asociada a una de las dependencias. Es decir, si el desarrollador eligiera aleatoriamente una configuración de entre las 261, aproximadamente un 83% de las veces estaría asumiendo un impacto de seguridad en su configuración. Es decir, la configuración que eligiera podría tener una vulnerabilidad, y por tanto un impacto sobre su seguridad.

Una posible pregunta que se podría plantear cualquier desarrollador o un sistema CI/CD, es si podríamos asegurar minimizar el riesgo o impacto, seleccionando una configuración cuyo impacto de seguridad sea inferior a 2.5. Para esto, tendremos que analizar el espacio de configuraciones y las vulnerabilidades.

III. Advisory: Análisis De VulnerabilIdades en pRoyectos de deSarrollo softwaRe

La Fig. 2 muestra una visión general del proceso soportado por Advisory. El proceso se divide en cuatro componentes principales: a) Extracción del grafo de dependencias, por ejemplo, de un fichero requirements.txt de un proyecto Python, extraemos su grafo de dependencias usando la información de GitHub[12]; b) Atribuir el grafo con información relativa a las vulnerabilidades desde la base de datos de vulnerabilidades de NVD del NIST; c) Codificar la información en un modelo formal basado en SMT solver [12], que nos permite razonar sobre las dependencias y sus vulnerabilidades, y finalmente; d) Aplicar un conjunto de operaciones que nos facilite el análisis de la información de dependencias y sus vulnerabilidades.

A continuación, describiremos los componentes de manera pormenorizada siempre basándonos o usando como soporte el ejemplo motivador de la Sección II.

A. Extraer el grafo de dependencias

En el primer componente (Extraer) construimos el grafo de dependencias de un proyecto software alojado en un repositorio. Este proceso consiste en los siguientes pasos: 1) obtenemos la información desde el repositorio; 2) filtramos las versiones validas para esas dependencias; y 3) construimos nodo a nodo las dependencias del grafo. Como se describe con más detalle a continuación:

1) En primer lugar, obtenemos la información sobre las dependencias desde el repositorio de código del proyecto software, extrayéndola de manera automática, por ejemplo, requirements.txt o septup.py. De estos archivos extraemos el nombre de la dependencia y la URL para acceder a ella. En caso de que las dependencias tengan otras sub-dependencias (indirectas), se realizará un proceso recursivo, extrayendo las restricciones sobre los valores de versiones que puede tomar éstas. Actualmente Advisory es capaz de obtener esta información de repositorios alojados en GitHub y de naturaleza Python. Para ello, se apoya en la API GraphQL de GitHub que nos permite hacer llamadas usando el lenguaje GraphQL[13]. Para el caso del ejemplo motivador de la sección anterior, se ha realizado una única llamada al repositorio que contiene los ficheros con las dependencias, en este caso de urllib3 y flask.
2) Después utilizaremos las restricciones para filtrar las versiones válidas. Primero extraemos la información de todas versiones disponibles de cada dependencia. Luego las filtramos una a una y escogemos las que satisfagan las restricciones del fichero de dependencias. Para el ejemplo, estamos trabajando con proyectos de naturaleza Python, por lo que utilizamos el Python Package Index (PyPI)[14] para extraer todas versiones dichas dependencias, para luego filtrarlas según las restricciones del fichero de dependencias. Para las del ejemplo motivador, obtenemos las siguientes versiones: flask={1.0.1, 1.0.2, 1.0.3, 1.0.4, 1.1.0, 1.1.1, 1.1.2, 1.1.3, 1.1.4} y urllib3={1.21.1, 1.22, 1.23, 1.24, 1.24.1, 1.24.2, 1.24.3, 1.25, 1.25.1, 1.25.10, 1.25.11, 1.25.2, 1.25.3, 1.25.4, 1.25.5, 1.25.6, 1.25.7, 1.25.8, 1.25.9, 1.26.0, 1.26.1, 1.26.2, 1.26.3, 1.26.4, 1.26.5, 1.26.6, 1.26.7, 1.26.8, 1.26.9}.
3) Por último, con esta información y las versiones filtradas construimos un nuevo nodo del grafo por cada dependencia y para cada versión, y agregamos los arcos dirigidos para relacionar cada nodo de dependencia con los de sus versión, además de incluir otras relaciones con nodos padres, e hijos (si fuera necesario). Así lo haremos con todas las dependencias extraídas. Nótese que dada la más que probable explosión combinatoria de dependencias, será necesario especificar un nivel de profundidad del grafo. Empezando por la raíz que será nuestro proyecto (Request para ejemplo) que tendrá profundidad 0, realizamos una llamada que construirá tanto el nodo raíz como todos los nodos de profundidad 1, que serán las dependencias de nuestro proyecto (o dependencias directas). Sí definimos que nuestro grafo debe tener profundidad 2, posteriormente haremos una llamada por cada nodo de profundidad 1 para construir sus dependencias, que serán subdependencias o dependencias indirectas de nuestro proyecto. El grafo a nivel 1 del ejemplo motivador quedaría como en la Fig. 3.

[10]https://docs.python-requests.org/
[11]https://github.com/psf/requests/network/dependents
[12]https://docs.github.com/es/graphql
[13]https://graphql.org/
[14]https://pypi.org/

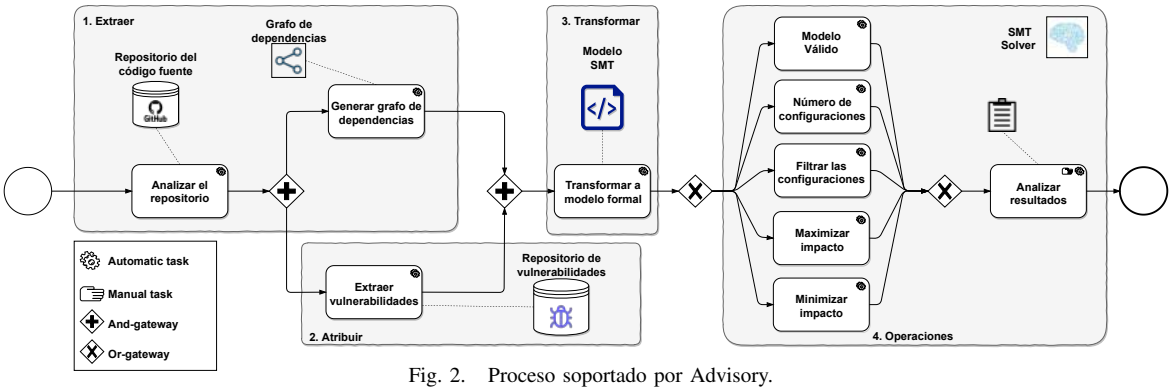


Fig. 2. Proceso soportado por Advisory.

Notar que en la figura se han añadido información de las vulnerabilidades que se detallará a continuación.

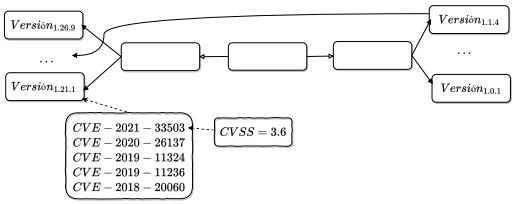


Fig. 3. Grafo de dependencias atribuido del ejemplo motivador

B. Atribuir con información relativa a las vulnerabilidades

Ahora describiremos como se atribuyen los nodos del grafo, es decir las dependencias, con información relativa a las vulnerabilidades (CVE). Para este proceso actualmente utilizamos la API de la base de datos de NVD del NIST. En primer lugar, utilizaremos el nombre de la dependencia, como clave para encontrar esas vulnerabilidades a la hora de realizar las búsquedas. De estas búsquedas extraemos todos los CVE que incluyen en alguno de sus CPE el nombre dicha dependencia. En el caso de flask y urllib3, NVD nos devuelve 2 y 8 CVE respectivamente asociados a esas dependencias[15].

Estos CVEs podrán asociarse a una versión o no incluidos de nuestro grafo, por lo que tendremos que analizar, si alguna de las versiones de las dependencias está incluida en algún CPE de estos CVE, ya que de lo contrario ese CVE no debería ser asignado a ninguna versión de nuestro grafo. Por ejemplo, para flask ninguno de los 2 CVE detectados será asociado porque no coincide con alguna versión contemplada en sus CPEs. Además, para urllib3, 7 de 8 CVEs si son asociados a las versiones válidas del ejemplo motivador. Esto lo hacemos así porque hacer peticiones individuales por cada versión ralentizaría el proceso de atribución debido a que cada consulta es una petición a la API de NVD y consumiría demasiado tiempo.

De esta forma relacionamos las versiones de las dependencias con los CVE que hemos extraído y que les afectan de forma directa. Además, vamos a almacenar para cada CVE el valor de CVSS (impacto), así como el conjunto de métricas de su vector de ataque. Por ejemplo, podemos ver los CVE asociados para la dependencia 1.21.1 de urllib3 y el impacto del CVSS de uno de ellos en la Fig. 3. Este impacto también viene con el CVE, y será fundamental para determinar el impacto del espacio de configuraciones.

[15]Estos datos son los obtenidos en el momento de redactar este artículo.

C. Transformar en modelo SMT

Por último, el grafo atribuido lo transformaremos a un modelo Satisfiability Modulo Theories (SMT) [12]. Un modelo SMT es un modelo formal de satisfacción de restricciones generalizado (SAT) que permite usar fórmulas más complejas que implican números reales, enteros y/o diversas estructuras de datos como listas, matrices, vectores de bits y cadenas. Para nuestro aproximación, un SMT vendrá definido mediante la tupla: $\langle D,\ V,\ DC,\ Vul,\ f_d(D,\ V),\ f_t(I)\rangle$. Donde:

- D es el conjunto de nodos del grafo, y representará el conjunto de variables del modelo. Para una dependencia $d_i \in D$ si $d_i = 0$ significa que no es seleccionado para el análisis, y si $d_i = 1$, sí es seleccionados. Si escogemos flask y urllib3, ambas dependencias deben aparecer en el modelo SMT como variables.
- V es el conjunto de valores de versiones de cada dependencia escogida del grafo. La dependencia sólo podrá tomar los valores de las versiones que tiene disponible en el grafo. Como mencionamos, si se escogen flask y urllib3, estas deben tomar como valor alguna de las dependencias válidas que ya vimos en la sección anterior.
- DC representa las restricciones que aplicamos al conjunto global de versiones de una dependencia. En este caso, al tratarse de variables de tipo numérico podemos hacer uso de operaciones lógicas de rango. Por ejemplo, para el caso de urllib3 del ejemplo motivador, $urllib3 \geq 1.21.1$ y $urllib3 < 1.27$. Estas restricciones definen el cojunto de versiones V.
- Vul es el conjunto de valores de impacto, es decir, el impacto definido en cada CVSS asociado a un CVE.
- $f_d(D,V)$ es una función que permite calcular el impacto para cada dependencia escogida del grafo en función de la configuración de versiones que tome el resolutor. En nuestra caso, vamos a considerar todos los CVE con sus impactos (CVSS), para lo que debemos agregar esta información por cada nodo. Para ello, podemos utilizar diferentes funciones que calculen el impacto de la dependencia agregando los impactos, por ejemplo, podemos usar la media, la mediana o la moda. Si usáramos la media para determinar el impacto de la

dependencia d_x en la versión v_s que tiene *n* vulnerabilidades (CVE), implica que si la dependencia toma esa versión, quedaría tal que para cada dependencia el calculo del impacto sería:

$$Impacto_{(d_x,v_s)} = \frac{\sum_{i=1}^{n} CVSS_{CVE_i}}{n} \quad (2)$$

Por ejemplo, si la dependencia urrlib3 tomase la versión 1.21.1, el impacto se calcularía de la siguiente manera: $\frac{CVSS_{CVE-2021-33503}+...+CVSS_{CVE-2018-20060}}{5}$

- $f_t(I)$ es una función que calcula el impacto total de un proyecto agregando el impacto de todas las dependencias, es decir, si nuestro proyecto tuviera para m dependencias cada una con sus versiones quedaría como sigue: $Impacto_{total} = \frac{\sum_{i=1,j=1}^{m,s} Impacto_{(d_i,v_j)}}{m}$

Al igual que la anterior función, podemos utilizar diferentes funciones que calculen el impacto de la dependencia como la media, la mediana o la moda. Para el ejemplo motivador, usando una media, el impacto total se quedaría como sigue: $\frac{Impacto_{urllib3}+Impacto_{flask}}{2}$

Para aplicar esta transformación en la versión actual de Advisory hemos usado modelos para el resolutor Z3[16].

D. Aplicar operaciones

Una vez construimos el modelo SMT a partir del grafo, podemos aplicar operaciones de razonamiento. De entre ellas, las actualmente implementadas por Advisory son:

- **Modelo válido:** Conocer si el modelo puede encontrar alguna solución que satisfaga todas las restricciones y operaciones creadas. Esta operación devolvería un booleano $True$ o $False$ indicando que el proyecto es válido o no. El ejemplo motivador es válido dado que existe al menos una configuración, por ejemplo, la compuesta por {urllib3 = 1.21.1, flask = 1.0.1}.
- **Número de configuraciones:** si el modelo es válido, extraer el número de configuraciones posibles satisfactorias. Nótese que esta operación no es viable ejecutarla en proyectos con muchas dependencias siempre que usemos SMT solvers. En el ejemplo de la Sección II existen 261 configuraciones. El número máximo de configuraciones que Advisory puede devolver es el representable en un entero de Z3 en Python definido por la funcion *sys.maxsize()*, un total de $9.223372036854775807 \times 10^{18}$ configuraciones.
- **Análisis de las configuraciones:** si las anteriores operaciones no nos aportan la suficiente información sobre el impacto de la seguridad de nuestras dependencias. Hemos desarrollado las siguientes operaciones que permiten refinar el análisis de una configuración. Estas operaciones son los siguientes: a) **Filtrar configuraciones por un umbral mínimo y un umbral máximo**, que nos permite crear un rango de impacto total sobre las configuraciones. Por ejemplo, obtener todas las configuraciones cuyo impacto esté entre 0 y 1.5. Si no se especifican, de forma predeterminada el mínimo se asigna a 0 y el máximo se asigna a 10, que son los valores máximos y mínimo de impacto para una vulnerabilidad según CVSS. Por ejemplo, en el caso del ejemplo motivador, si fijamos el umbral máximo a 0 obtendríamos un total de 45 configuraciones. Esta operación devuelve un conjunto de configuraciones; b) **Minimizar o Maximizar** si queremos sacar las configuraciones con menor impacto o las configuraciones con mayor impacto, podemos aplicar una de las dos operaciones de optimización para el impacto. Por ejemplo, en el caso del ejemplo motivador, podríamos maximizar el impacto para obtener la configuración $\{urllib3 = 1.22, flask = 1.1.3\}$ con impacto máximo de 1.83. Estas operaciones devuelven un conjunto de configuraciones; y c) **Limitar exploración.** Estas tres operaciones cuentan con un parámetro para limitar el número de configuraciones que pueden devolver Advisory. Actualmente, analizar el número de configuraciones válidas para problemas de satisfacibilidad, se convierte en casos en los que haya mucha combinatoria, en un problema no determinista en tiempo (NP-completo). Por lo que necesitamos limitar el número de configuraciones que queremos recibir. En otras palabras, dada a imposibilidad de poder enumerar todas las configuraciones existentes que cumplan un criterio de seguridad, permitimos limitar el número de soluciones a un valor especificado por el usuario. Por ejemplo, podríamos obtener 3 configuraciones que cumplan con el umbral máximo de 2.5, o minimizar el impacto y devolver las 3 configuraciones con menor impacto.

Una vez que el resolutor resuelve el modelo junto con la operación requerida, éste arrojará los resultados en términos de lógica proposicional, es decir, a nivel de asignación de valores a cada una de las variables y Advisory se encarga entonces de interpretar los resultados y presentarlos como dependencias y versiones a usar para cumplir los objetivos marcados por el usuario. Nótese que las distintas funciones de optimización están implementadas en lógica proposicional dentro del propio solver (Z3), lo que permite reducir el espacio de búsqueda a la vez que definimos las condiciones del umbral, las funciones de maximización y minimización, y finalmente el número de configuraciones a devolver.

IV. Validación y resultados

A. Plataforma de experimentación

Para la realización de la experimentación debemos seleccionar un conjunto representativo de pruebas con la limitación de que deben ser proyectos Python. Para la elección de los proyectos, se ha usado la lista de los 10 proyectos más usados según la web https://pypistats.org/top. Las características de vulnerabilidades identificadas y del número de configuraciones de los mismos se presenta en la Tabla I. En este caso mostramos los resultados del número teórico de configuraciones totales, y el número de configuraciones alcanzables para umbrales de impacto 0, (0-2.5], (2.5,10]. Estos datos serán analizados más adelante en los experimentos.

Los siguientes experimentos se realizaron en una máquina con CPU Intel(R) Core(TM) i7-10510U (1.80GHz hasta 4,90GHz, 8MB caché, 4 núcleos), una tarjeta gráfica integrada

[16]https://github.com/Z3Prover/z3

TABLE I
CANTIDAD DE CONFIGURACIONES POR UMBRALES.

Proyecto	Total Configs.	Impacto $= 0$	Impacto > 0 y ≤ 2.5	Impacto ≥ 2.5
boto3	$1.88 \cdot 10^6$	+1000	+1000	0
urllib3	$1.35 \cdot 10^{13}$	0	+1000	0
botocore	$2.37 \cdot 10^7$	+1000	+1000	0
s3transfer	$5.46 \cdot 10^5$	0	+1000	0
requests	$2.02 \cdot 10^{19}$	+1000	+1000	0
six	151	151	0	0
typing	$7.04 \cdot 10^7$	+1000	0	0
dateutil	$1.05 \cdot 10^{22}$	0	+1000	0
aws-cli	$5.21 \cdot 10^8$	+1000	+1000	0
charset-normalizer	$2.79 \cdot 10^{12}$	+1000	+1000	0
click	$1.4 \cdot 10^7$	+1000	+1000	0
numpy	$5.56 \cdot 10^{23}$	+1000	+1000	0
cryptography	86,580	+1000	+1000	0
packaging	$2.7 \cdot 10^5$	+1000	0	0

Intel UHD Graphics, una memoria RAM de 16GB 2666MHz DDR4-SDRAM y un almacenamiento de 750GB NVMe PCIe SSD, bajo un entorno Ubuntu 20.04.4 LTS, y Python versión 3.9.12. Advisory y los experimentos descritos en este trabajo están disponibles como código abierto para la comunidad en https://doi.org/10.5281/zenodo.6479353

B. Experimento 1: Comparación con otras herramientas existentes

En el mercado ya existe una serie de herramientas que detectan vulnerabilidades en proyectos software y sus dependencias, por eso es interesante posicionar el rendimiento de Advisory con respecto a ellas. En concreto analizaremos Snyk y DependanBot de GitHub dada su popularidad, uso, y soporte de análisis de proyectos tipo Python. Se ha descartado OWAS Dependacy Check porque ésta se basa en proyectos de naturaleza Java y Advisory sólo soporta actualmente proyectos tipo Python.

En este experimento intentaremos verificar si Advisory puede detectar más vulnerabilidades (cuantificadas como número de CVE detectados) en el análisis de las vulnerabilidades de las dependencias de un proyecto, extraídas en nuestro caso de la API GraphQL de GitHub. Es decir, si somos capaces de producir más información sobre las vulnerabilidades de un proyecto que otras opciones. Esto ayudaría a los desarrolladores a tener una información más ampliada de las posibles vulnerabilidades que pueden afectar a sus proyectos. Para poder compararnos minimizando los sesgos, configuraremos nuestra herramienta para que sólo busque en el primer nivel de profundidad (dependencias directas) del grafo.

Hipótesis: Advisory detecta más vulnerabilidades que las soluciones alternativas para los proyectos analizados. En nuestra hipótesis nula es que nuestra herramienta puede detectar más vulnerabilidades en las dependencias de los proyectos software más usados.

Resultados del experimento 1. Los resultados que obtenidos para cada proyecto analizado se puede ver en la Fig. 4. Vemos que Snyk detecta más vulnerabilidades que DependantBot para todos los proyectos, pero además Advisory consigue detectar más vulnerabilidades que Snyk para todos los proyectos, excepto para el proyecto *cryptography*, sólo 1 más. Este último caso, parece ocurrir debido a las diferencias entres las bases de datos de vulnerabilidades usadas por Snyk y Advisory o posibles inconsistencias [13] entre las base de datos de Snyk y NVD. Otro posible motivo es que Synk, por defecto, busca en más de un nivel de dependencias (dependencias indirectas) las vulnerabilidades, mientras que Advisory sólo se ha ejecutado en este experimento configurado para explorar el primer nivel de dependencias. También cabe mencionar que Snyk y Dependant Bot no analizan el espacio completo de configuraciones, solo la configuración que consideran latest.

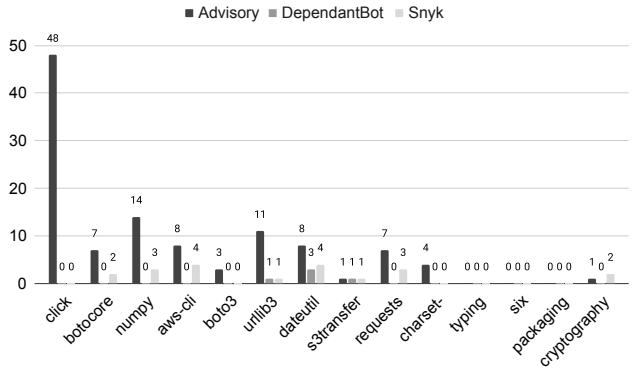


Fig. 4. Cantidad de vulnerabilidades que detecta cada herramienta.

C. Experimento 2: Datos generales de los grafos construidos

En el segundo experimento mostramos los datos procedentes de la extracción de los grafos y la atribución de los mismos. Probamos así el funcionamiento de Advisory, en las etapas de extracción de la información descritas en la Sección III-A, utilizando los 10 proyectos más usados segun las estádisticas de PyPI.

Hipótesis: Advisory es capaz de construir con información de seguridad coherente las grafos de dependencias. Nuestra hipótesis es que con Advisory extraemos y atribuimos una cantidad de dependencias, restricciones y vulnerabilidades coherente entre ellos, es decir, que mientras más dependencias encontramos más restricciones somos capaces de extraer y más vulnerabilidades atribuimos.

Resultados del experimento 2. Realizamos la extracción del grafo y la atribución sobre los proyectos, con los resultados de la Fig. 5 y la Tabla I. Como vemos, en todos los casos, se aprecia una relación entre el número de dependencias detectadas y el número de restricciones y CVEs. Mientras más dependencias extraemos, más restricciones y mientras más dependencias más posibilidades de tener vulnerabilidades. Pero a la vez a más restricciones, menor número de versiones, lo que reduce el número de vulnerabilidades asociadas. En proyectos como *botocore* o *aws-cli* donde el número de restricciones crece más que las dependencias, el espacio de configuraciones es más pequeño y por lo tanto es atribuido con menos vulnerabilidades. Esto no pasa en los proyectos como *click* o *numpy* donde las dependencias y el número de restricciones se mantiene equivalente.

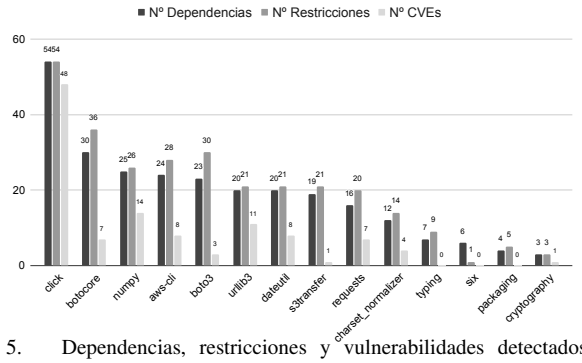


Fig. 5. Dependencias, restricciones y vulnerabilidades detectados por proyecto.

D. Experimento 3: Realización de operaciones

En el tercer experimento analizamos las operaciones mencionadas en la Sección III. Para ello, probamos el funcionamiento de la solución tanto en el conteo de configuraciones que cumplan ciertas condiciones como para la identificación del impacto máximo y mínimo. Para ello, ejecutamos las operaciones de optimización sobre los proyectos y reportamos el valor medio del impacto detectado.

Hipótesis: Nuestras operaciones procesan la suficiente información relativa a la seguridad y aportan información al desarrollador. Nuestra hipótesis pretende dar a entender que las operaciones desarrolladas son herramientas útiles para desarrollar proyectos software seguros.

Resultados del experimento 3. El resultado esperado si podemos detectar si nuestro proyecto tiene configuraciones vulnerables o no. Y además, en caso de tener configuraciones vulnerables cuáles son las que tiene el impacto máximo y mínimo de entre todas. Teniendo en cuenta la problemática del número de configuraciones que tiene cada proyecto de la Tabla I. Comenzamos aplicando las dos operaciones maximización y minimización del impacto, para extraer la configuración con mayor impacto y la configuración con menor impacto. Como vemos en la Fig. 6, sólo 3 de los 14 proyectos analizadas no cuentan con configuraciones vulnerables, 8 de ellas cuentan con configuraciones vulnerables pero también con configuraciones sin vulnerabilidades, y otras 3 de ellas no cuentan con configuraciones sin vulnerabilidades.

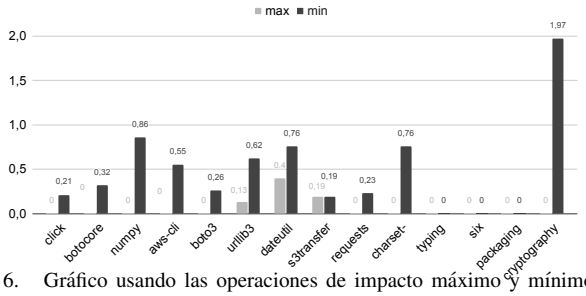


Fig. 6. Gráfico usando las operaciones de impacto máximo y mínimo por proyecto.

Realizando la operación de filtro con umbrales sobre estos proyectos, extraemos los datos que vemos en la Tabla I. Podemos observar que existen proyectos, como *urllib3* de la que no se puede obtener una configuración con impacto 0. También podemos ver casos donde hemos acotado el total de configuraciones entorno a un rango, como es el caso de *boto3* o *botocore*, donde todas sus configuraciones están entre 0 y 2.5 de impacto, o el de *packaging* y *six* donde todas sus configuraciones están en impacto 0. Estos datos nos permiten ver y analizar de un simple vistazo a modo de cuadro de mando el conjunto de configuraciones, comprobando qué conjunto de configuraciones están afectados por vulnerabilidades o no.

E. Amenazas a la validez

Validez externa: Los datos de los experimentos son realistas, ya que son copias exactas de los repositorios de los proyectos analizadas. Sin embargo, para la extracción de los datos del grafo y la atribución de vulnerabilidades se han utilizado APIs que pueden contener errores externos a nosotros, o que pueden dejar de funcionar, así como inconsistencias entre los datos que obtenemos. La amenazas que nos afectan a la que nos es enfrentamos es sobre *validez de la población*, ya que la cantidad de vulnerabilidades atribuidas a un grafo pueden no ser todas las existentes en el momento del análisis. Además, entre diferentes bases de datos pueden existir inconsistencias. Para reducir esta amenaza en el pensamos nutrirnos de más de una bases de datos, de forma que, si una sufre una inconsistencia pueda ser corregida por el resto.

Validez interna: Los recursos de CPU requeridos para analizar el espacio de configuraciones de un grafo depende del número de versiones y restricciones que se aplican a cada dependencia. Sin embargo, para minimizar estos efectos, hemos introducido parámetros para elegir la profundidad deseada o el número límite de configuraciones a devolvemos. Así controlamos aquellos proyectos con un coste computacional muy elevado. Por otro lado, no estamos exentos de cometer errores en nuestra herramienta que invaliden los resultados de la misma. Por tanto, las amenazas que nos afectan son: 1) *Regresión estadística*: En el caso del uso de la media como factor de agregación de los diferentes impactos de las vulnerabilidades, hace que los casos extremos tengan demasiada influencia en el cálculo del impacto agregado; y 2) *Instrumentación*: Los proyectos escogidas para el análisis han sido elegidas siendo las más descargadas a lo largo del presente mes, y podría originar que con el paso del tiempo hayan cambiado.

V. Trabajos relacionados

El análisis de vulnerabilidades en dependencias de proyectos software es problema abierto en la comunidad [14]. En dicho estudio, después de analizar más de 10.000 artefactos software se ha determinado que más del 80% de las dependencias incluidas contiene vulnerabilidades. Las problemáticas del análisis de vulnerabilidades de las dependencias vienen determinadas por múltiples factores: 1) cómo se realiza el análisis de las dependencias; 2) cómo se identifican las vulnerabilidades asociadas a las dependencias; 3) tipo de proyecto software y repositorio analizado.

Con respecto a cómo se analizan las dependencias en la literatura se puede hacer un análisis sólo de dependencias directas [15] o también dependencias transitivas o indirectas

[16]. Existe cierta controversia de cómo "tratar" el impacto asociado de estas dependencias [16], ya que las dependencias pueden considerarse de manera agrupada o no. Sin embargo, en nuestra propuesta usamos un grafo de dependencias que si refleja las dependencias transitivas y además podemos configurar el nivel de profundidad con respecto al análisis, y todas son tenidas en cuenta para determinar el impacto en el proyecto.

Con respecto a la forma de identificar vulnerabilidades para dependencias, se observan diferentes aproximaciones como la basada en el emparejamiento de nombres, términos y valores con los de CVEs y CPEs de base de datos de vulnerabilidades [17][18]. Mientras que otras aproximaciones se basan en aspectos semánticos [19] o en el análisis del código de las dependencias [20].

Otro aspecto fundamental es la naturaleza del proyecto, la mayoría de estudios se centran en proyectos Java usando Maven [14] como repositorio de artefactos principales, mientras que otros analizan de naturaleza Javascript como npm [18]. Hasta donde nosotros hemos podido estudiar, nuestra propuesta es la única de las estudiadas que abarca proyectos de código basados en Python y dependencias en repositorios como PyPI.

VI. Observaciones Finales y Lecciones Aprendidas

El análisis de vulnerabilidades en proyectos software resulta en una tarea compleja y crucial debido al elevado número de posibilidades del espacio de configuraciones definido por sus ficheros de dependencias. Además, es un problema real para los desarrolladores a priori escoger una configuración de dependencias cuyo impacto se desconoce o que esté libre de vulnerabilidades.

Advisory resuelve este problema mediante el análisis de este espacio de configuraciones. Primero construyendo un grafo de dependencias atribuido con información de las vulnerabilidades y luego analizando las configuraciones mediante un modelo formal basado en SMT. Esto permite dar al desarrollador obtener información relevante sobre las vulnerabilidades de las configuraciones analizadas. Hemos comprobado en entornos reales, que somos una solución competitiva y que puede aportar a la comunidad. Sin embargo, tenemos ciertas limitaciones y aspectos mejorables como la utilización de diferentes bases de datos de vulnerabilidades para evitar inconsistencias en la información extraída.

En resumen, hemos aprendido las siguientes lecciones importantes: 1) **Acercamiento entre variabilidad y seguridad.** Podemos analizar la variabilidad que se encuentra en las dependencias de los proyectos software y extraer información relativa a su seguridad; 2) **Inconsistencias en las bases de datos de vulnerabilidades.** La necesidad de tener que incorporar múltiples bases de datos para evitar inconsistencias que pueden originarse al indexar una vulnerabilidad en una base de datos y en otras no; y 3) **Imposibilidad de analizar el espacio completo de configuraciones.** El análisis de espacio de configuraciones muy elevados hace imposible para un resolutor analizar todas las posibles configuraciones. Lo que resulta un problema abierto para el análisis completo de la seguridad de una herramienta con una gran cantidad de dependencias y versiones.

Material y agradecimientos

Este trabajo ha sido financiado por los proyectos AETHER-US (PID2020-112540RB-C44/AEI/10.13039/501100011033), COPERNICA (P20_01224) y METAMORFOSIS (US-1381375).

References

[1] J. R. Jones, "Estimating software vulnerabilities," *IEEE Security Privacy*, vol. 5, no. 4, pp. 28–32, 2007.

[2] T. Yadav and A. M. Rao, "Technical aspects of cyber kill chain," in *Security in Computing and Communications*, 2015, pp. 438–452.

[3] W. Hu, Y. Wang, X. Liu, J. Sun, Q. Gao, and Y. Huang, "Open source software vulnerability propagation analysis algorithm based on knowledge graph," in *2019 IEEE SmartCloud*, 2019, pp. 121–127.

[4] S. Dass and A. S. Namin, "Vulnerability coverage for adequacy security testing." New York, NY, USA: ACM, 2020.

[5] P. Murthy and R. Shilpa, "Vulnerability coverage criteria for security testing of web applications," in *2018 ICACCI*, 2018, pp. 489–494.

[6] S. M. Perez, V. Cosentino, and J. Cabot, "Model-based analysis of java EE web security misconfigurations," *Comput. Lang. Syst. Struct.*, vol. 49, pp. 36–61, 2017.

[7] J. A. G. Duarte, "Evolution, testing and configuration of variability systems intensive." Ph.D. dissertation, University of Rennes 1, France, 2015.

[8] J. A. Galindo, D. Benavides, and S. Segura, "Debian packages repositories as software product line models. towards automated analysis," in *ACOTA, Belgium, September, 2010*, vol. 688. CEUR-WS.org, 2010, pp. 29–34.

[9] R. Heradio, D. Fernández-Amorós, J. A. Galindo, D. Benavides, and D. S. Batory, "Uniform and scalable sampling of highly configurable systems," *Empir. Softw. Eng.*, vol. 27, no. 2, p. 44, 2022.

[10] J. A. Galindo, D. Benavides, P. Trinidad, A.-M. Gutiérrez-Fernández, and A. Ruiz-Cortés, "Automated analysis of feature models: Quo vadis?" *Computing*, vol. 101, no. 5, pp. 387–433, 2019.

[11] Á. J. Varela-Vaca, R. M. Gasca, J. A. Carmona-Fombella, and M. T. G. López, "AMADEUS: towards the automated security testing," in *24th ACM SPLC '20, Montreal, Quebec, Canada, October 19-23, 2020, Volume A*. ACM, 2020, pp. 11:1–11:12.

[12] P. Arcaini, A. Gargantini, and P. Vavassori, "Generating tests for detecting faults in feature models," in *2015 IEEE 8th ICST*. IEEE, 2015, pp. 1–10.

[13] V. H. Nguyen, S. Dashevskyi, and F. Massacci, "An automatic method for assessing the versions affected by a vulnerability," vol. 21, no. 6, 2016.

[14] I. Pashchenko, H. Plate, S. E. Ponta, A. Sabetta, and F. Massacci, "Vulnerable open source dependencies: Counting those that matter," in *12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2018, pp. 1–10.

[15] J. Cox, E. Bouwers, M. van Eekelen, and J. Visser, "Measuring dependency freshness in software systems," in *2015 IEEE/ACM 37th ICSE*, vol. 2, 2015, pp. 109–118.

[16] R. G. Kula, D. M. Germán, A. Ouni, T. Ishio, and K. Inoue, "Do developers update their library dependencies? an empirical study on the impact of security advisories on library migration," *CoRR*, vol. abs/1709.04621, 2017.

[17] M. Cadariu, E. Bouwers, J. Visser, and A. van Deursen, "Tracking known security vulnerabilities in proprietary software systems," in *2015 IEEE 22nd SANER*, 2015, pp. 516–519.

[18] J. Hejderup, "In dependencies we trust: How vulnerable are dependencies in software modules?" 2015.

[19] "Tracing known security vulnerabilities in software repositories – a semantic web enabled modeling approach," *Science of Computer Programming*, vol. 121, pp. 153–175, 2016.

[20] S. Ponta, H. Plate, and A. Sabetta, "Beyond metadata: Code-centric and usage-based analysis of known vulnerabilities in open-source software," in *2018 ICSME*, 2018, pp. 449–460.

Secure Platform for ICT Systems Rooted at the Silicon Manufacturing Process (SPIRS)

Piedad Brox, Macarena C. Martínez-Rodríguez
Instituto de Microelectrónica de Sevilla
(CSIC / Universidad de Sevilla)
Américo Vespucio 28, 41092 Sevilla
{brox, macarena}@imse-cnm.csic.es

David Arroyo Guardeño
Instituto de Tecnologías Físicas
y de la Información (CSIC)
Serrano 144, 28006 Madrid
david.arroyo@csic.es

***Abstract*—Internet of Things and ubiquitous/pervasive computing are shaping our world where smart devices enter every aspect of our everyday life. This is why privacy-enhancing technologies are all the more important. In this context, the EU-funded *"Secure Platform for ICT Systems Rooted at the Silicon Manufacturing Process"* project will design a platform that integrates a hardware dedicated Root-of-Trust and a processor core with the capability of offering a full suite of security services. The platform will be able to leverage this capability to support privacy respectful attestation mechanisms and enable trusted communication channels across 5G infrastructures. The project will also provide solutions to integrate the platform in the deployment of cryptographic protocols and network infrastructures in a trustworthy way.**

***Index Terms*—ICT systems, cybersecurity, Root-of-Trust.**

Tipo de contribución: *Investigación en desarrollo (límite 4 páginas)*

I. Introduction

Our society is continuously demanding more and more intelligent devices, along with network infrastructures and distributed services that make our daily lives more comfortable. This revolution has also reached the industrial sector, transforming traditional factories into smart ones, with the objective of enhancing supply chain and manufacturing. However, the frantic adoption of Internet of Things (IoT) technologies in multiple application domains has led to widespread implementations without a deep analysis about the vulnerabilities that IoT devices are exposed to. The *"Secure Platform for ICT Systems Rooted at the Silicon Manufacturing Process (SPIRS)"* project addresses innovative approaches to provide security and data privacy to future Information and Communications Technology (ICT) elements [1]. This project encompasses the complete design of a platform, so called SPIRS platform, which integrates a hardware dedicated Root of Trust (RoT) and a processor core with the capability of offering a full suite of security services. Furthermore, the SPIRS platform is designed to properly address the challenge of providing privacy respectful remote attestation in 5G and Industry 4.0.

RoT is implemented in hardware with a dedicated circuitry to extract a unique digital identifier for the SPIRS platform during its entire lifetime. A silicon CMOS Physical Unclonable Function (PUF) is used to derive the device's identity, exploiting tiny variations in CMOS manufacturing process. The RoT also integrates attack resistant cryptographic hardware cores that incorporate countermeasures, so minimizing the vulnerability against side channel (SCA) and fault injection attacks (FA), and increasing the performance of the approach in terms of timing and power consumption. The security of the core is reinforced with the implementation of a mutual authentication scheme between the RoT and the embedded software. To build a complete solution, the project also features a Trusted Execution Environment (TEE), secure boot, and runtime integrity. Furthermore, resilience and privacy protection are major concerns in this project, and it endeavours to the design of a decentralized trust management framework targeted to minimize the impact of Single Point of Failure (SPOF) risks and achieve adequate security and privacy trade offs. To facilitate the tasks of validation and testing, the SPIRS platform is conceived as an open platform that can easily integrate other IP modules and facilitates upgrades.

The project goes beyond the construction of the SPIRS platform and it provides solutions to integrate it in the deployment of cryptographic protocols and network infrastructures in a trustworthy way, leveraging the RoT provided by the platform. This includes the implementation of chains of trust using the physical identity, also known as trust anchor, derived from the PUF. On this basis, remote and direct anonymous attestation are integrated with network orchestration mechanisms to foster security and privacy protection in edge and cloud infrastructures. The project will extend existing open source network orchestration frameworks to demonstrate the applicability of these trust anchors and attestation procedures in the development of IoT network gateways building a Trusted Network Environment for Devices (TNED). In addition, the ambition of the project is to ease the design of secure and privacy friendly IoT architectures by developing a hardware design integration framework. This framework will bring flexibility to the design of the SPIRS architecture, will automate its validation and participate to the continuum of the chain of trust from the hardware up to the network.

II. Concept

During recent years, electronic devices are increasingly being transformed from isolated systems to networked Internet-enabled devices that can communicate with each other, or even providing connections to remote cloud environments, trending towards a full network integration with the advent of edge computing technologies. The proliferation of these embedded systems, so-called IoT devices, has emerged in multiple sectors focusing efforts in operative solutions whereas security issues have been postponed. Although IoT

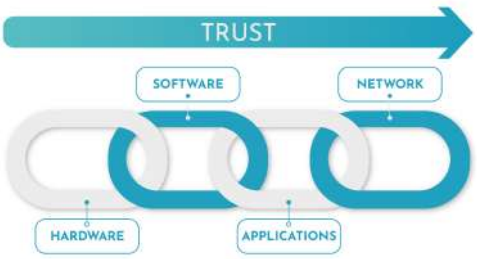


Fig. 1: Chains of trust in embedded systems.

devices are becoming dispensable for our daily activities, they present critical security gaps that could even compromise their functionality. For example, in late 2016, researchers were able to demonstrate one way that hackers can put people in harm's way, by accessing a Tesla Model S car control system remotely [2]. The vulnerability allowed the researchers to access and control the braking system, engine, sunroof, door locks, trunk, side-view mirrors, and more. The vulnerability was through the infotainment and WiFi connection where the researchers were able to attack it and use it to gain access to the car's controller area network bus.

Although the scientific community has provided progress in security and privacy technologies over the years, interconnect devices and systems in a broad spectrum of application domains is an open research challenge [3]. Several issues must be addressed to build privacy-friendly and secure devices: device authentication, communication channels, handling of sensitive data, prevention of attacks, capacity to add additional functionality after deployment, or flexibility to include future modifications in regulations. All of the above aspects have to be considered taking into account the heterogeneity of the ecosystem, with a wide variety of different device features depending on the application case, ranging from large-scale grid-based solutions down to resource-constrained portable devices. Another critical aspect to consider is that they are integrated into network infrastructures, exposing them to more potential security risks [4]. Such security risks and vulnerabilities are further exacerbated by the increased softwarization and virtualization of the application functions and the underlying network protocols [5]. Agile and flexible procedures are required to govern such virtualized and highly distributed functions by means of adequate Management and Orchestration (MANO) systems. The investigation of vulnerabilities in a multi-vendor, multi-domain virtualized networked eco-system of 5G and beyond is currently under investigation in the 3GPP (the reference standards group for 5G technologies) and the ETSI (European Telecommunications Standards Institute) NFV (Network Functions Virtualisation) group. Conventional security and trust paradigms will not be sufficient to address the new challenges brought about by the enhanced Service Based Architecture (SBA) that will be the foundational architectural blueprint for the network architecture of 5G and beyond [6]. The development of secure and privacy-respectful solutions is also demanded to protect the device during all its lifecycle phases. The end-to-end security strategy must address all phases during its lifetime including operation and maintenance tasks. SPIRS-platform will become the unique existing solution covering all levels of trust from silicon up to application levels.

Due to the hybrid (hardware/software) nature of an embedded system, establishing a trusted chain is a requirement without a straightforward implementation. The first link in the chain of trust is system hardware. The next one is software (bootloader and operating system) that trusts in hardware, assuming that it is working properly. Next, applications trust that the operating system does not get corrupted. And beyond the device itself, a remote system trusts the device's identity to which it is connected (see Figure 1).

To extend trust, a Public Key Infrastructure (PKI) is employed to mutually-authenticate endpoints. Therefore, each link in the system can authenticate the identity of another component of the system before accepting any data from it. The process to establish trust is authentication at every level. The starting point of authentication is the RoT, and then the rest of the system layers must prove trustworthiness before allowing them to act.

Device authentication must adhere to RoT that can be implemented on software or hardware. To provide authenticity, a proof-of-knowledge approach based on a challenge-response protocol is usually implemented. During the verification process, a host authority generates a random challenge for the device. Then, the device performs a cryptographic function using secret data provided by the RoT and returns an output response. If the output satisfies the challenge, the verifying authority treats the target as authentic.

First approaches use software implementations of RoT, where the secret keys are usually stored in a Non-Volatile Memory (NVM) by a trusted manufacturer. However, these approaches are vulnerable to device impersonation attacks since a counterfeit (malicious) system that integrates this NVM could be used without being identified as an impostor. An alternative is a hardware RoT that uses, as a device secret, a digital identifier derived from hardware. This could be implemented using a PUF that exploits tiny variations in the silicon manufacturing processes of semiconductor circuitry.

III. Objectives

SPIRS has identified the following project objective to face the posed research and innovation challenges: establish chains of trust rooted in the silicon manufacturing process for ICT systems, and apply them in improving the supply chain for networked infrastructures. To achieve this main objective, SPIRS has to handle the following specific objectives:

1. Design of a platform with a tamper-proof silicon RoT that:
   - Derives its cryptographic identity (device's secret key) from manufacturing variation measured by a PUF.
   - Incorporates crypto hardware modules with Differential Power Analysis (DPA) and FA protection.
   - Generates a Trust Flag if all internal blocks are operating properly and no attacks are detected.
2. Design of a TEE using the silicon RoT.
   - To provide a secure boot process in either a local or remote client setting.
   - To ensure runtime integrity of executed files.
   - To provide the client with an attestation to its state that can be compared against a manufacturer list of trustworthy states.

- Token generation from privacy-friendly credentials, on the grounds of data minimality and assurance by TEE.
- Deployment of privacy respectful audit trails to identify security threats and monitor performance.

3. Integration of the platform into network infrastructures using the silicon RoT.

- To extend the attestation mechanisms in order to support remote attestation.
- To integrate the attestation procedures with network orchestration, operation and management.
- To provide a set of (virtualized) network functions supporting the extended security model enabled by the above.
- Leverage PUFs and Distributed Ledger Technologies (DLTs) for security monitoring in Systems Development Life Cycle (SDLC).

4. Implementation of the platform.

- Design a secure IoT hardware platform generation framework with RoT capabilities.
- Using a processor core with an open-source hardware Instruction Set Architecture (ISA) based on established reduced instruction set computer (RISC) principles.
- Evaluating its performance on programmable devices (FPGAs).
- Evaluating its performance on Integrated Circuits using a nanometer technology.
- Leverage PUFs and DLTs for security monitoring in High-level Data Link Control (HDLC).

5. Evaluation of the platform in different scenarios.

- Data minimization and privacy respectful Authentication, Authorization and Audit (AAA).
- Extended network gateway supporting enhanced security and privacy in IoT device connections.
- Industry 4.0.
- 5G Communication Infrastructure and management systems.

IV. Scenarios

SPIRS platform will be validated in the following scenarios:

A. Industry 4.0

SPIRS platform will be used to secure the network connectivity and smart devices within the warehouse and production area of the NEXT company that participates in SPIRS as end-user. The company makes use of two interconnected execution systems: (1) Logistic Execution System (LES) and (2) Manufacturing Execution System (MES). SPIRS will provide a secure and privacy friendly communication of both execution systems (LES and MES) into network infrastructures, enabling the connection to remote servers in a safe way and performing event recording for audit and accountability on the grounds of the RoT. This will foster trustworthiness, ease the maintenance and update of smart devices through the implementation of techniques for predictive maintenance of machinery, and exchange of information with other external plants, which will eventually increase the overall production of the NEXT company in a short/medium term.

B. 5G Infrastructure Management

The SPIRS platform will be used to develop, test and validate security models to protect the multifarious assets in a complex 5G infrastructure. The scenario abstracts a multi-tenant, multi-vendor, multi-site, multi-domain 5G eco-system consisting of multiple network slices deployed across multiple NFV Infrastructure (NFVI) providing compute, network and storage resources. The tenants are connected to the end-users over one or more of the network slice(s), where each slice instance delivers a 5G service that is most suitable to the tenant business/service requirements. Network slices are composed of Virtualized Network Functions (VNFs) interconnected over Virtual Links (VL) to enable the service profile of a respective slice. Each slice can support multiple tenants, and each tenant can be a customer of multiple slices. The tenant, via these network slices, delivers their services to end-user domain, which comprises a diverse set of devices ranging from aerial and terrestrial drones/robots, connected autonomous vehicles, used communication devices, etc.

V. Innovation potential

Innovation potential: Embedded security for IoT is highly-demanded by the rising number of devices and the increasing strength of cyber-attacks (on IoT devices as well as on networks). Several key market players are identified on this topic. Some examples of commercial products are: $Rambus\ Cryptomanager\ RoT$, an independent hardware security co-processor to be integrated into semiconductor devices; $EmSPARK^{TM}$, an embedded security platform from NXP. None of these solutions, and those provided by the above mentioned companies, is as ambitious as SPIRS proposal, since it will provide: 1) a hardware RoT with a PUF to derive keys instead of the classical storage of keys, 2) hardware crypto modules with special circuitry design to prevent tampering and aging effects, 3) mixed SW-HW based chain of trust from device up to application, 4) guaranteeing privacy requirements at all levels, 5) all embedded in an Open Source Platform. The development of TEE and secure booting using derived keys from hardware RoT, as well as the build of trusted applications for privacy and RA will make position SPIRS platform ahead of all existing solutions available on the market. Additionally, its secure integration into network infrastructures with the development of a TNED will provide a disruptive end-to-end innovative solution. Industry 4.0 and 5G use cases will demonstrate the ability of SPIRS platform to be placed as an operative solution at the top of this exigent market.

VI. Consortium

SPIRS is a collaborative project gathering the significant expertise of entities that are internationally recognized in their specific fields of applied and fundamental research. The core competence of the SPIRS team originates from the fact that its members represent the entire value chain to build a secure platform for ICT systems, from research activities to use cases that provide end-user solutions. The Consortium of the SPIRS Project is composed of 4 industrial companies, 3 Research and Technology Organisations and 2 academic institutions from 5 European countries (Spain, Finland, Italy, France and Germany) [1].

VII. The role of CSIC in SPIRS

Two research groups of CSIC are involved in the SPIRS proposal, namely the research group in Digital and Mixed Integrated Circuits Design (UDDM), and the research group in Cryptology and Information Security (GiCSI). UDDM belongs to the Institute of Microelectronics in Seville (IMSE), and GiCSI belongs to the Institute of Physical and Information Technologies (ITEFI).

A. Microelectronics Institute of Seville (IMSE)

IMSE is a joint Institute from CSIC and University of Seville that has a broad experience in the design of advanced Integrated Circuits under different technologies. The IMSE group is very active in the design of VLSI hardware for security. Among its research lines, we can highlight the design of PUFs and high-performance ciphers, vulnerability analysis and countermeasures design against physical attacks (side channel analysis and fault attacks) of hardware implementations of cryptocircuits.

B. Institute of Physical and Information Technologies (ITEFI)

GiCSI has a broad experience in the design and cryptanalysis of cryptosystems and cryptographic protocols. It also deals with activities related to generation of (pseudo)random bits, vulnerability analysis of cryptographic devices against side-channel attacks, blockchain and Privacy-Enhancing Technologies (PETs), security analysis of web-based applications, anomaly detection, characterization of misinformation phenomena, and techniques of accountability in communication networks.

In close collaboration with UDDM, the software engineers and cryptographers of the GiCSI team will develop and implement cryptographic functions for digital signatures, and performing side-channel assessment of the implementations. The knowledge and experience of both CSIC teams in the fields of random number generators, digital signatures, and side-channel attacks perfectly match the tasks in the proposal, since those are building blocks to effectively develop a secure hardware root of trust ensuring privacy preservation.

UDDM is the leader of project coordination with the support of GiCSI. As experts in silicon technologies, UDDM also leads the design of hardware RoT components. GiCSI contributes to SPIRS by extending its previous work on the design of PETs [7] and leading its integration as blockchain-based protocols for digital evidence and audit trails gathering [8], [9]. GiCSI provides trustworthy software lifecycle for the TNED by deploying formal methods and metrics for security and privacy assessment, and to persist and manage security incidents by blockchain and DLTs protocols. GiCSI also contributes to testing privacy protection means and blockchain and DLT protocols in the different use cases.

VIII. Current activities

SPIRS started in October 1st, 2021 and it has a duration of 36 months. The research teams are focussed on:

- Work package 2 (WP2): Design of a RoT suitable for building secure execution environments, protocols, and integration into network infrastructures.
- Work package 3 (WP3): Development of secure booting mechanisms using the RoT and TEE to build high-level protocols supporting privacy enhancement, and remote attestation protocols.
- Work package 5 (WP5): Implementation of a RISC-V processor bridging WP2 and WP3 results.

From the results coming from WP2 have arisen the early scientific publications. In [10], the evaluation of a Ring Oscillator PUF to be integrated in the HW RoT has been presented. The acceleration of post-quantum encryption NTRU algorithm has been explored in [11]. And a revision of fault injection and power analysis attacks have been reported in [12], [13].

IX. Conclusions

SPIRS will provide a dedicated hardware RoT that actively interacts with the rest of components (software, protocols, and network) in a safe way. This gives rise to attractive challenges to scientific and industrial communities that are looking for urgent solutions: SPIRS will address these challenges and provide these solutions. Additionally, SPIRS is a good opportunity to foster CSIC in Europe as main actor in the design of technologies to protect ICT systems.

Ackowledgments

This research is supported by the SPIRS Project with Grant Agreement No. 952622 under the EU H2020 research and innovation programme. M.C.M.R. holds a Postdoc fellowship from the Andalusian Government with support from PO FSE of EU.

References

[1] https://www.spirs-project.eu
[2] https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus-wp.pdf
[3] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of things: The road ahead", Computer Networks, 2015.
[4] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A Survey on Security and Privacy Issues in Internet-of-Things", IEEE Internet Things J., 2017.
[5] M. Condoluci, and T. Mahmoodi, "Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges", Computer Networks, vol 146, pp. 65-84, 2018.
[6] S. Zhang, Y. Wang, and W. Zhou, "Towards secure 5G networks: A Survey", Comput. Networks, 2019.
[7] J. Diaz, S. G. Choi, D. Arroyo, A. D. Keromytis, F. B. Rodriguez, and M. Yung, "A Methodology for Retrofitting Privacy and Its Application to e-Shopping Transactions, " Advances in Cyber Security: Principles, Techniques, and Applications, pp. 143-183, 2019.
[8] M. Gimenez-Aguilar, J. M. de Fuentes, L. Gonzalez-Manzano, D. Arroyo, "Achieving cybersecurity in blockchain-based systems: A survey", Future Generation Computer Systems, vol. 124, pp. 91-118, 2021.
[9] A. Marín-López, S. Chica-Manjarrez, D. Arroyo, F. Almenares-Mendoza, D. Díaz-Sánchez, "Security information sharing in smart grids: Persisting security audits to the blockchain", Electronics, vol. 9 (11), pp. 1865, 2020.
[10] M.C. Martínez-Rodríguez, E. Camacho-Ruiz, S. Sánchez-Solano and P. Brox, "Design Flow to Evaluate the Performance of Ring Oscillator PUFs on FPGAs", Conference on Design of Circuits and Integrated Systems (DCIS), 2021.
[11] S. Sánchez-Solano, E. Camacho-Ruiz, M.C. Martínez-Rodríguez and P. Brox, "Multi-Unit Serial Polynomial Multiplier to Accelerate NTRU-Based Cryptographic Schemes in IoT Embedded Systems", Sensors, vol. 22, no. 5, article 2057, 2022.
[12] F.E. Potestad-Ordóñez, E. Tena-Sánchez, A.J. Acosta, C.J. Jiménez-Fernández and R. Chaves, "Hardware Countermeasures Benchmarking agains Fault Attacks", Applied Sciences, vol. 12, no. 5, article 2443, 2022.
[13] E. Tena-Sánchez, F.E. Potestad-Ordóñez, C.J. Jiménez-Fernández, A.J. Acosta and R. Chaves, 'Gate-Level Hardware Countermeasure Comparison against Power Analysis Attacks", Applied Sciences, vol. 12, no. 5, article 2390, 2022.

Premios RENIC

Premio a Mejor Tesis Doctoral en Ciberseguridad

Towards decentralized and scalable architectures for access control systems for IIoT scenarios

Santiago Figueroa-Lorenzo [1,2]
sfigueroa@ceit.es

[1]CEIT-Basque Research and Technology Alliance (BRTA), Manuel Lardizabal 15, 20018 Donostia / San Sebastián, Spain.
[2]Universidad de Navarra, Tecnun, Manuel Lardizabal 13, 20018 Donostia / San Sebastián, Spain.

***Abstract*—The Industrial Internet of Things (IIoT) architecture is complex due to, among other things, the convergence of protocols, standards, and buses from such heterogeneous environments as Information Technology (IT) and Operational Technology (OT). IT – OT convergence not only makes interoperability difficult but also makes security one of the main challenges for IIoT environments. In this context, this thesis starts with a comprehensive survey of the protocols, standards, and buses commonly used in IIoT environments, analyzing the vulnerabilities in assets implementing them, as well as the impact and severity of exploiting such vulnerabilities in IT and OT environments. The Vulnerability Analysis Framework (VAF) methodology used for risk assessment in IIoT environments has been applied to 1,363 vulnerabilities collected from assets implementing the 33 protocols, standards and buses studied. On the other hand, Access Control Systems emerges as an efficient solution to mitigate some of the vulnerabilities and threats in the context of IIoT scenarios. Motivated by the variety and heterogeneity of IIoT environments, the thesis explores different alternatives of Access Control Systems covering different architectures. These architectures include Access Control Systems based on traditional Authorization policies such as Role-based Access Control or Attribute-based Access Control, as well as Access Control Systems that integrate other capabilities besides Authorization such as Identification, Authentication, Auditing and Accountability. Blockchain technologies are incorporated into some of the proposals as they enable properties not achievable in centralized architectures, at different levels of complexity: they can be used just as a verifiable data registry, executing simple off-chain authorization policies, up to scenarios where the blockchain enables on-chain an Identity and Access Management System, based on Self-Sovereign Identity.**

***Index Terms*—Security, Blockchain, Industrial Internet of Things, Access Control Systems, Identity and Access Management Systems, Self-Sovereign Identity.**

Type of contribution: *PhD Award RENIC*

I. Introduction

The accelerated development of Information and Communication Technologies (ICT) has resulted in the transformation of the industrial production, traditionally based on Operation Technology (OT), towards the incorporation of information technologies (IT) to achieve further automation of manufacturing into the industry 4.0 paradigm, which attempts to build a controllable, credible, scalable, secure and efficient interaction of physical devices in the industrial scene, trying to fundamentally change the traditional human understanding of the industry. Industry 4.0 is merely a term for the fourth industrial revolution, which includes interoperability, autonomy, information transparency, technical assistance and distributed decisions, making traditional manufacturing smart, representing, in addition, a broader concept that encompasses both smart manufacturing and the Industrial Internet of Things (IIoT) [1].

IIoT is defined by Enisa as IoT applied in the industrial environment [1]. IIoT represents a subset of IoT, encompassing the domains of machine-to-machine (M2M) and industrial communication technologies (ICT). IIoT, therefore, is paving the way for enhanced manufacturing process performance, improving operational efficiency. In this regard, the enabling technologies that ensure the industrial transformation represented by IIoT, integrating virtual space with the physical world, are mainly cyber-physical systems (CPS) and the Internet of Things (IoT). CPS interconnects physical assets with computational capabilities, while IoT can be considered a global network infrastructure composed of several connected devices based on sensory, communication, networking and information processing technologies. By the end of 2020, IIoT projected more than 10 billion devices, accounting for 57% of IoT spending [2].

In particular, the role of IIoT is to connect all industrial assets, including machines and control systems, with the corresponding information systems and business processes, i.e., to adapt business and operating models by integrating information technologies (IT) and operation technologies (OT). IIoT devices are therefore responsible for monitoring, collecting, exchanging and analyzing information so that they can control their behavior without human intervention. This huge amount of data collected can feed analytical solutions and lead to optimal industrial operations and all that information needs to be collected and analyzed to create significant information for future decision-making.

Despite the benefits of IIoT for Industry 4.0, the technology faces several challenges in terms of energy management, evolution to new generations of industrial systems to cope with the complexity of production in cyber-physical environments and the capacity to perform diagnoses, such as health management of machine tools, enhancing productivity and increasing the quality of processes and services, based on data analytics and artificial intelligence (AI). However, one of the main challenges presented by the IIoT is security. The scientific literature is clear in highlighting, on the one hand, that data confidentiality, cyber-physical systems integrity and device management are challenges inherited from the IoT [3]. On the other hand, it defines specific challenges related to industrial security, such as the security of industrial control systems (ICS), connectivity between field devices and gateways, the increasing integration of IT monitoring of physical production processes, but above all, the fact that IIoT has

blurred the traditional boundaries of IT and OT infrastructures, enabling the convergence of both environments [4]. In this regard, the threats and vulnerabilities, not only of design but also of implementation and configuration, in both IoT and IIoT environments have been properly documented [5]. Considering that the severity of these threats on the IIoT environment can be characterized around three tenets: confidentiality, integrity and availability, so that, on the one hand, the impact of vulnerability exploitation can be measured, depending on the security requirements of an organization from them and on the other hand, these three tenets, allow to evaluate the security controls, i.e., the countermeasures that are adopted to protect IIoT environments, where, access control system emerges as one of the few common solutions for the protection of these three tenets [6]. In this sense, Access Control Systems can be part of scenarios that include only authorization processes, i.e., the data protection against unauthorized access, use, or disclosure while in storage, in-process or in transit and therefore considered simply as enforcement of access control model, as well as scenarios consisting of Access Control Systems, where in addition to the usual authorization, the phases of identification, authentication and accountability are involved. Additionally, these Access Control Systems are integrated into the blockchain, which is a disruptive technology consisting of states, operations and transactions replicated in a distributed system, on which security is guaranteed based on cryptographic techniques both to preserve the integrity (e.g. digital signatures) and to preserve privacy, which in turn, uses protocols for the establishment of consensus between the operations performed by the nodes of the distributed network and finally presents a business layer that represents the logic embedded in the peers, e.g., smart contracts that ensure the development of decentralized applications, enabling decentralized Access Control Systems.

II. Thesis Motivation

The IIoT environments are specialized in improving the productivity and efficiency of industrial processes and are particularly related to Industry 4.0. For these environments, security constitutes one of the main challenges so that numerous issues have been identified that bring associated risks, among which are the security of industrial control systems, the connectivity between field devices and gateways, the increasing integration of IT monitoring of physical production processes. However, the major security risk is represented by the fact that IIoT has blurred the traditional boundaries of IT and OT infrastructures, allowing the convergence of both environments. In that sense, while IIoT convergence is central to the capabilities that IIoT builds, breaking down the boundaries between these two worlds exposes significant aspects such as unprotected network connections, deployment of technologies with identified vulnerabilities that bring together formerly unidentified risks into the OT environment and inadequate perception of requirements for ICS settings. Therefore, these scenarios can result in brand and reputational damage, material financial loss and potential damage to critical infrastructures. Considering that confidentiality, integrity and availability are parameters that emerge as mechanisms with applicability in the context of categorization of both risks and protection mechanisms, the first motivation of this thesis is the need for a thorough analysis of the vulnerabilities and their impact on both IoT and IIoT.

It has also been mentioned that access control emerges as enabling technology in the security management of IIoT environments, both at the level of communication and connectivity protection and at the level of data and endpoint protection [7]. Hence, the second motivation of the thesis is the exploration of the Access Control Systems for IIoT scenarios due to the current heterogeneity and variety of this environment. For this purpose, Access Control Systems are studied with an approach related to the execution of authorization policies based on traditional models such as RBAC or ABAC, as well as Access Control Systems that involve not only authorization capabilities but also identification, authentication, auditing and accountability. Additionally, it is also considered of interest the research not only from the access management perspective but also from the identity management perspective. For the scenarios analyzed and proposals realized, the discussion in terms of the degree of decentralization, information storage and performance will be also relevant.

III. List of contributions

The Ph.D. Thesis was organized by articles. The list of contributions is presented below:

1) S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Survey of IIoT Protocols: A Measure of Vulnerability Risk Analysis Based on CVSS," ACM Computer Surveys, vol. 53, no. 2, p. 53, 2020, doi: doi.org/10.1145/3381038. [JCR. 7.990, Q1].
2) S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Role-Based Access Control Model in Modbus SCADA Systems. A Centralized Model Approach," Sensors, MDPI, vol. 19, no. 20, p. 4455, 2019, doi: doi.org/10.3390/s19204455. [JCR. 3.576, Q1].
3) S. Figueroa Lorenzo, J. Añorga, and S. Arrizabalaga, "An Attribute-Based Access Control Model in RFID Systems Based on Blockchain Decentralized Applications for Healthcare Environments," Computers, MDPI, vol. 8, no. 3, p. 19, 2019, doi: doi.org/10.3390/computers8030057. [SJR. 3.3, Q2].
4) S. Figueroa-Lorenzo, Santiago; Goya, Jon; Añorga, Javier; Adin, Iñigo; Mendizabal, Jaizki; Arrizabalaga, "Alarm collector in Smart Train based on Ethereum blockchain events-log," IEEE Internet of Things Journal, vol. 08, no. 17, pp. 13306 – 13315, 2021, doi: doi.org/10.1109/JIOT.2021.3065631. [JCR. 9.471, Q1].
5) S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "Methodological performance analysis applied to a novel IIoT access control system based on permissioned blockchain," Information Processing & Management., vol. 58, no. 4, p. 102558, 2021, doi: doi.org/10.1016/j.ipm.2021.102558. [JCR. 6.222, Q1].
6) S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "Modbus access control system based on SSI over Hyperledger Fabric Blockchain," Sensors, MDPI, vol. 21, no. 16, p. 5438, 2021, doi: doi.org/10.3390/s21165438. [JCR. 3.576, Q1].
7) S. Figueroa, J. Añorga, S. Arrizabalaga, I. Irigoyen, and M. Monterde, "An Attribute-Based Access Control using Chaincode in RFID Systems," in 2019

10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2019, pp. 1–5, doi: doi.org/10.1109/NTMS.2019.8763824. [CONFERENCE].

8) S. Figueroa, J. F. Carías, J. Añorga, S. Arrizabalaga, and J. Hernantes, "A RFID-based IoT Cybersecurity Lab in Telecommunications Engineering," in 2018 XIII Technologies Applied to Electronics Teaching Conference (TAEE), 2018, pp. 1–8, doi: doi.org/10.1109/TAEE.2018.8475973. [CONFERENCE].
9) S. Figueroa Lorenzo, J. Añorga Benito, P. García Cardarelli, J. Alberdi Garaia, and S. Arrizabalaga Juaristi, "A Comprehensive Review of RFID and Bluetooth Security: Practical Analysis," Technologies, 2019, vol. 7, no. 1, p. 15, doi: doi.org/10.3390/technologies7010015. [JCR. 0.73, Q2].
10) Y. S. Mingsheng, S. Figueroa-Lorenzo, J. Añorga, S. Arrizabalaga, Y. Sun, "IACAP: Internet-exposed Assets Cybersecurity Analysis Platform," International Journal of Interdisciplinary Telecommunications and Networking (IJITN), 2020, vol. 12, no. 4, p. 14, doi: doi.org/10.4018/IJITN.2020100109. [JCR. 0.13, Q4].
11) S. Figueroa-Lorenzo, S. Arrizabalaga, J. Añorga, "Towards decentralized and scalable architectures for Access Control Systems in IIoT environments," VI Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), 2021, vol. 34, no. 33, p. 145, doi: doi.org/10.18239/jornadas_2021.34.33. [CONFERENCE].

IV. Link to the PhD thesis

The PhD thesis was published on August 14, 2021 by Servicio de Publicaciones, Universidad de Navarra. It can be downloaded at the following link: https://dadun.unav.edu/handle/10171/62234.

Acknowledgment

To my PhD thesis supervisors: Saioa Arrizabalaga Juaristi and Javier Añorga Benito.

References

[1] European Union Agency for Cybersecurity (ENISA), Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures, no. November. 2017.
[2] J. Navarro-Ortiz, S. Sendra, P. Ameigeiras, and J. M. Lopez-Soler, "Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things," IEEE Commun. Mag., vol. 56, no. 2, pp. 60–67, Feb. 2018, doi: doi.org/10.1109/MCOM.2018.1700625.
[3] V. Sklyar and V. Kharchenko, "ENISA Documents in Cybersecurity Assurance for Industry 4.0: IIoT Threats and Attacks Scenarios," in 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2019, vol. 2, pp. 1046–1049, doi: doi.org/10.1109/IDAACS.2019.8924452.
[4] X. Yu and H. Guo, "A Survey on IIoT Security," in 2019 IEEE VTS Asia Pacific Wireless Comm. Symposium (APWCS), 2019, pp. 1–5, doi: doi.org/10.1109/VTS-APWCS.2019.8851679.
[5] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," IEEE Internet Things J., vol. 6, no. 5, pp. 8182–8201, Oct. 2019, doi: doi.org/10.1109/JIOT.2019.2935189.
[6] M. Chapple, J. M. Stewart, and D. Gibson, (ISC)2 CISSP Certified Information Systems Security Professional Official Study Guide. 2018.
[7] S. Schrecker et al., "Industrial Internet of Things Volume G4: Security Framework," Ind. Internet Consort., pp. 1–173, 2016, doi: doi.org/10.13140/RG.2.2.28143.23201.

Premio a Mejor Trabajo Fin de Máster sobre Ciberseguridad

Extracción y Análisis de Artefactos de Memoria de la Aplicación Telegram Desktop

Pedro Fernández-Álvarez
Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza, Spain
pfernandez@unizar.es

***Resumen*—Las aplicaciones de mensajería instantánea se han convertido en una manera muy común para comunicarse. A nivel global, *Telegram* es una de las plataformas de mensajería instantánea más populares, y es el servicio en el cual se centra este proyecto. En este trabajo se ha desarrollado un entorno de análisis forense para la obtención de artefactos digitales presentes en memoria RAM relativos a aplicaciones de mensajería instantánea. Concretamente, el foco se ha puesto en la extracción de artefactos de memoria pertenecientes a la aplicación `Telegram Desktop`, como contactos del usuario o contenidos de conversaciones que han tenido lugar, entre otros. La posibilidad de conseguir estos datos resulta de gran ayuda para un analista forense a la hora de esclarecer o resolver un caso.**

***Index Terms*—Análisis forense digital, análisis forense de memoria, mensajería instantánea, Telegram Desktop**

Tipo de contribución: *Investigación ya publicada [1]*

I. Introducción

Las aplicaciones de mensajería instantánea (MI) permiten comunicarse de una manera rápida y cómoda. Hoy en día, una parte notable de la sociedad hace uso de este tipo de aplicaciones para mantener conversaciones. Sin embargo, estas aplicaciones también son usadas en ocasiones como medio para cometer o esclarecer delitos. Es en estos últimos casos cuando el análisis forense de los dispositivos del criminal o de la víctima puede ser de gran ayuda, proporcionando evidencias vitales para la resolución o esclarecimiento del posible crimen acontecido.

Uno de los factores que en ocasiones dificulta la obtención de artefactos digitales relativos a aplicaciones de MI es la existencia de cifrado, tanto en las bases de datos locales de estas aplicaciones como en la información que las aplicaciones transmiten a través de la red. No obstante, los contenidos presentes en la memoria RAM deben encontrarse descifrados para que la aplicación pueda trabajar con ellos, haciendo que el análisis forense de la memoria RAM sea especialmente interesante cuando exista cifrado tanto de los datos almacenados de manera local como de las comunicaciones.

Esta investigación se focaliza en la plataforma de MI *Telegram*, la cual se encuentra entre las 5 más populares a nivel global. En particular, se ha analizado el cliente multiplataforma oficial de *Telegram* para ordenadores, llamado `Telegram Desktop`. El objetivo de este trabajo es investigar los contenidos presentes en memoria RAM relativos a la aplicación `Telegram Desktop` de cara a identificar artefactos digitales de interés para una investigación forense. La base de datos local de `Telegram Desktop` se encuentra cifrada [2] y las comunicaciones con sus servidores se llevan a cabo también de manera cifrada [3], provocando ambos hechos que el análisis de la memoria RAM se considere de gran importancia.

Durante la realización de este proyecto se ha trabajado con la versión 2.7.1 de `Telegram Desktop` para el sistema operativo Windows 10. Se ha elegido Windows dado que es el sistema operativo para ordenadores con más cuota de mercado en la actualidad y Windows 10 dado que es la versión más popular a día de hoy [4].

Esta investigación se ha presentado en la conferencia DFRWS EU 2022 [5] y también se ha publicado un artículo en la revista internacional con revisión por pares Forensic Science International: Digital Investigation (FSIDI) [1].

II. Entorno de Análisis Desarrollado

De cara a analizar los contenidos que se encuentran en la memoria RAM de un sistema Windows es necesario, en primera instancia, obtenerlos. Además, dado un volcado de memoria RAM de un proceso, hay que identificar la localización con la que se corresponden las direcciones virtuales de dicho proceso. De esta manera, es posible acceder al elemento almacenado en una dirección virtual concreta. Con el objetivo de extraer los contenidos de la memoria RAM relativos a un proceso determinado en un formato que cumpla con el requisito previamente mencionado se ha elaborado la herramienta llamada `Windows Memory Extractor` [6], puesto que ninguna de las herramientas existentes para volcar memoria permitían obtener el volcado de una manera que sea navegable a través de sus direcciones virtuales.

La segunda herramienta desarrollada en este trabajo, denominada `Instant Messaging Artifact Finder` (abreviado como `IM Artifact Finder`) [7], se encarga de analizar un volcado de memoria de un proceso relativo a una aplicación de MI (obtenido con la herramienta `Windows Memory Extractor`) y de generar un informe en el que se refleje la información de los artefactos obtenidos. Estas dos herramientas elaboradas componen el entorno de análisis desarrollado en el trabajo. Por otro lado, ambas herramientas son utilidades de línea de comandos, lo cual permite que puedan ser integradas en flujos de análisis más amplios. De manera adicional, el código fuente de estas herramientas se ha liberado bajo licencia GNU/GPLv3.

La herramienta `Windows Memory Extractor` está implementada en C++ y es de utilidad versátil, ya que puede extraer tanto módulos completos como regiones de memoria que cuenten con unas protecciones determinadas. Por su parte, la herramienta `IM Artifact Finder` se ha desarrollado en Python y no se ha confeccionado únicamente para analizar `Telegram Desktop`, sino que se ha desarrollado como un

framework que pueda ser extensible a otras aplicaciones de MI.

III. Resultados y Discusión

Respecto a la identificación de artefactos relevantes desde un punto de vista forense se ha analizado el código fuente de `Telegram Desktop`, el cual se distribuye de manera libre. Para encontrar en un volcado de memoria objetos de interés se han realizado búsquedas de patrones de números de teléfono y de patrones horarios. Una vez que se han encontrado objetos, se han identificado dentro de ellos punteros a otros objetos, y de esta manera se han podido identificar nuevos objetos de interés para el análisis forense.

A continuación se resumen los resultados obtenidos tras evaluar el entorno de análisis elaborado (una discusión más extensa sobre los resultados puede consultarse en [1]):

- **Cuentas:** Identificación del número de cuentas añadidas a la aplicación e información acerca de sus correspondientes propietarios (identificador, nombre completo, número de teléfono y nombre de usuario).
- **Conversaciones:** Reconstrucción de las conversaciones accedidas por el usuario, tanto en el caso de conversaciones individuales como en el caso de grupos y canales. No obstante, los mensajes eliminados y editados no se han podido recuperar. Por otro lado, es posible recuperar parte de una conversación tras eliminarla, aunque los contenidos recuperados no son completamente fiables.
- **Usuarios:** Se han logrado detectar aquellos usuarios que comparten su número de teléfono y también algunos que no lo comparten, siendo posible diferenciar si un usuario es un contacto o no. Adicionalmente, se han recuperado contactos tras haber sido eliminados.
- **Privacidad:** En relación con la privacidad del número de teléfono si un usuario decide no compartirlo, tras los experimentos llevados a cabo, no se ha encontrado en los datos obtenidos correspondientes a dicho usuario su número de teléfono. Este hecho es coherente con la expectativa de privacidad del usuario.
- **Multimedia:** En relación con los mensajes donde no solamente se transmite texto, se ha podido recuperar el nombre y el tipo de los archivos transmitidos, el nombre y el número de teléfono en el caso de los contactos compartidos, y en cuanto a las localizaciones geográficas, la latitud y la longitud. Adicionalmente, si una localización tiene información adicional asociada, como el nombre del lugar o la dirección, también se ha podido identificar.
- **Bloqueo:** En las pruebas llevadas a cabo se ha observado que se puede recuperar la misma información de la memoria RAM sin importar que la aplicación esté bloqueada o no. Tras desbloquear la aplicación se ha identificado la contraseña en memoria de manera manual, aunque no se ha podido recuperar de manera automática. Adicionalmente, dicha contraseña deja de estar en memoria tras un corto periodo de tiempo.
- **Sesión:** Ha sido posible identificar artefactos tras cerrar sesión. Sin embargo, los artefactos recuperados no son todos los que se obtienen antes de cerrar sesión, sino solamente un subconjunto. Por otro lado, la recuperación de los mismos no es completamente fiable, ya que en ocasiones los datos recuperados son incorrectos o incompletos.

La posibilidad de recuperar esta información presente en la memoria volátil de un ordenador incautado puede proporcionar a un analista forense evidencias cruciales para la resolución de un caso. Entre otros, el hecho de saber datos sobre el propietario de una cuenta puede ayudar a identificar a quién pertenece un equipo. Por otra parte, la identificación de contactos puede proporcionar individuos relacionados con un sospechoso a los que poder investigar. De manera adicional, se considera importante la recuperación de información a la que un analista forense no podría acceder desde la interfaz de usuario, como los contactos borrados.

IV. Conclusiones

En este proyecto se ha elaborado un entorno de análisis destinado a la obtención de artefactos forenses presentes en la memoria RAM del proceso de `Telegram Desktop` en sistemas Windows. Todas las herramientas que componen el entorno son de código abierto y están liberadas bajo licencia GNU/GPLv3. Los resultados obtenidos tras la evaluación de dicho entorno han proporcionado una serie de artefactos relacionados con `Telegram Desktop` cuya relevancia puede ser notable de cara a la resolución de un caso forense.

Agradecimientos

Me gustaría agradecer a Ricardo J. Rodríguez toda su ayuda y dedicación a lo largo de esta investigación.

Referencias

[1] Pedro Fernández-Álvarez y Ricardo J. Rodríguez. «Extraction and analysis of retrievable memory artifacts from Windows Telegram Desktop application». En: *Forensic Science International: Digital Investigation* 40 (2022). Selected Papers of the Ninth Annual DFRWS Europe Conference, pág. 301342.

[2] J. Gregorio, A. Gardel y B. Alarcos. «Forensic analysis of Telegram Messenger for Windows Phone». En: *Digital Investigation* 22 (2017), págs. 88-106.

[3] G. B. Satrya, P. T. Daely y S. Y. Shin. «Android forensics analysis: Private chat on social messenger». En: *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2016, págs. 430-435.

[4] *Evolución de la cuota de mercado de las versiones del sistema Windows*. [Online: https://www.statista.com/statistics/993868/worldwide-windows-operating-system-market-share/]. Accedido el 1 de junio de 2021.

[5] *Programa de DFRWS EU 2022*. [Online: https://dfrws.org/eu-program-2022/]. Accedido el 2 de junio de 2022.

[6] *Repositorio en GitHub de la herramienta Windows Memory Extractor*. [Online: https://github.com/reverseame/windows-memory-extractor]. Accedido el 3 de junio de 2022.

[7] *Repositorio en GitHub de la herramienta Instant Messaging Artifact Finder*. [Online: https://github.com/reverseame/instant-messaging-artifact-finder]. Accedido el 3 de junio de 2022.

Patrocinadores


accenture


BASQUE
CYBERSECURITY
CENTRE


indra


Ibermática
EMBRACING THE FUTURE


RENIC
Red de Excelencia Nacional de
Investigación en Ciberseguridad


VIEWNEXT
AN IBM SUBSIDIARY


ZIUR
INDUSTRIAL
CYBER SECURITY
CENTER-GIPUZKOA