DATASET PARA EL ANÁLISIS DE EVENTOS MALICIOSOS EN SISTEMAS WINDOWS BASADOS EN LA MATRIZ DE MITRE

Antonio Pérez Sánchez
Universidad Pontificia de Comillas
apsanchez@comillas.edu

Rafael Palacios Hielscher
Universidad Pontificia de Comillas
rafael.palacios@iit.comillas.edu

Gregorio Ignacio López López
Universidad Pontificia de Comillas
gllopez@comillas.edu

Resumen- **La evaluación de los sistemas de detección y prevención de amenazas requiere de la existencia de *datasets* que se encuentren actualizados y correctamente diseñados conforme las amenazas más comunes. Actualmente, la disponibilidad de *datasets* de eventos que contengan suficiente información para realizar estos análisis en sistemas Microsoft Windows es prácticamente inexistente. En este artículo llevaremos a cabo un análisis de los *datasets* ya existentes y haremos énfasis en los principales problemas para realizar estudios desde el punto de vista actual. Posteriormente, presentaremos el *dataset* que hemos generado a partir de la recolección de eventos en tiempo real y actualizados conforme las amenazas existentes en la actualidad. La principal ventaja del uso de este *dataset* con respecto a los ya conocidos es que permite la evaluación de sistemas de detección y prevención de amenazas basándose en técnicas y tácticas de la matriz de MITRE ATT&CK y donde se considera la componente temporal de los eventos maliciosos.**

***Index Terms*- Dataset, Detección de amenazas basadas en eventos, MITRE ATT&CK, Ataques Multietapa.**

Tipo de contribución: *Investigación original*

I. INTRODUCCIÓN

Los sistemas de detección y prevención de amenazas surgieron con el fin de solucionar los problemas relacionados con las actividades maliciosas en redes y sistemas informáticos [1]–[3]. Las diferentes amenazas a nivel de sistema operativo pueden ser identificadas a partir de Sistemas de Detección de Intrusiones (IDS) [4], sistemas antivirus (AV) [5] y sistemas de detección y respuesta en puntos finales (EDR) [6]. Cada uno de estos cuenta con unas capacidades distintas, siendo el último el que más unifica y añade además una capa de aprendizaje continuo mediante el uso de algoritmos de inteligencia artificial [7], [8].

Existen diferentes tipos de IDS: aquellos basados en detección de tráfico malicioso a nivel de red (NIDS) [9] mediante la monitorización de los flujos o *logs* de diferentes componentes de la misma. Por otro lado, los basados en la detección de eventos malintencionados en el propio sistema (HIDS) [10] a partir de la monitorización de archivos, eventos del sistema, etc. En cuanto a la detección podemos clasificarlos en dos tipos distintos, aquellos basados en firmas (S-IDS) [11], que hacen uso de reglas para la detección de comportamientos maliciosos, y los basados en anomalías (A-IDS) [12], que construyen un modelo a partir de datos de entrenamiento que decidirá posteriormente si un comportamiento es malicioso.

En cuanto a los sistemas antivirus [13], son el software principal para la clasificación y detección de malware y han evolucionado en las técnicas de detección y clasificación, utilizando mecanismos basados en firma [14]–[16] y heurística [17]–[19]. La detección por firmas es el método tradicional utilizado por los sistemas antivirus y se apoya en una base de datos generada por el proveedor. Cualquier archivo descargado en el sistema se compara con la base de datos y, si hay una coincidencia, se considera malware. El problema de este tipo de técnica es que sólo detectará aquellas muestras que hayan sido previamente identificadas y cuya firma esté almacenada en la base de datos del sistema antivirus. Para complementar la detección basada en firmas y ofrecer una solución a sus limitaciones, se diseñaron técnicas de detección heurística. El funcionamiento de los algoritmos heurísticos se basa en diferentes criterios, cada uno de ellos con una puntuación, que determinan si un fichero es malicioso.

Finalmente, los sistemas de detección y respuesta en puntos finales (EDR) combinan el monitoreo constante en tiempo real y el análisis de datos, incorporando el aprendizaje continuo mediante algoritmos de inteligencia artificial [20]–[22], de los diferentes sistemas y reacciona frente a las amenazas con respuestas automatizadas basadas en reglas definidas por el operador desde una consola centralizada.

Uno de los problemas para evaluar las capacidades de los sistemas de detección y prevención de amenazas, es proporcionar un conjunto de datos lo más actualizado y similar a las técnicas y tácticas representativas, usadas por los adversarios actualmente [23], [24]. Aunque existe un alto volumen de *datasets* [25] que pueden ser usados para este tipo de entrenamiento, cuentan con unas limitaciones, a nivel de sistema, que hacen difícil en la actualidad realizar una evaluación ajustada a los parámetros que nos encontramos en la realidad. Por ejemplo, existe un alto volumen de *datasets* que se centran en tráfico de red y no en eventos generados por los sistemas y aquellos que si lo hacen, se encuentran basados mayormente en sistemas Linux o no permiten la detección de ataques multietapa. Por ello, los puntos que consideramos durante la elaboración del *dataset* fueron, el uso de técnicas y tácticas definidas en la matriz de MITRE ATT&CK [26], el desbalanceo de eventos generados por los sistemas, la duplicidad de eventos y la temporalidad de los mismos.

II. REVISIÓN DE LOS DATASETS EXISTENTES

A continuación, realizamos una revisión de los *datasets* más relevantes en este campo durante los últimos años. A partir de estos, comentaremos los requisitos ideales para la generación de un nuevo *dataset* para la evaluación de sistemas de detección y prevención de amenazas. Finalmente, discutiremos las principales características de nuestra propuesta frente al conjunto de *datasets* comentados anteriormente.

Desde la aparición de las tecnologías de detección y prevención de amenazas, se han llevado a cabo diversas investigaciones que permiten publicar *datasets* [27] para realizar una evaluación ajustada a los parámetros reales en la actualidad. Aquellos que surgieron en primera instancia, se conoce como *datasets* de referencia, y a partir de los mismos se han podido realizar diferentes trabajos de investigación que han permitido el aprendizaje y mejora de los datos de evaluación.

A. Datasets de referencia

DARPA datasets [28]. El primer *dataset* que se creó con el fin de evaluar los sistemas IDS fue creado en 1999 por el MIT Lincoln Lab. Para ello, se implementó una red de pruebas y se recogió el tráfico y registros generados durante un periodo de tiempo, con el fin de identificar patrones maliciosos durante la actividad normal de la misma.

La recolección se realizó en diferentes fases. Las dos primeras semanas se ejecutaron ataques de red durante la actividad normal de la misma. La cuarta y quinta semana son los datos de test usados en la evaluación de 1999, desde el 16/9/1999 hasta el 01/10/1999. En total podemos encontrar 201 casos de unos 56 tipos de ataques distribuidos a lo largo de este periodo.

KDDCup'99 dataset [29]. Utilizado en el Tercer Concurso Internacional de Herramientas para el Descubrimiento del Conocimiento y la Minería de Datos, que se celebró conjuntamente con la KDD-99, en la Quinta Conferencia Internacional sobre Descubrimiento del Conocimiento y Minería de Datos. El objetivo consistía en construir un detector de intrusiones en la red, un modelo predictivo capaz de distinguir entre conexiones "malas", denominadas intrusiones o ataques, y conexiones normales "buenas". Esta base de datos contiene un conjunto estándar de datos a auditar, que incluye una amplia variedad de intrusiones simuladas en un entorno de red militar.

NSL-KDD dataset [30]. A partir de las investigaciones realizadas para la publicación del *dataset* KDDCup'99, Tavallaee et al. sugirió la creación de una evolución que resolvía algunos problemas inherentes a este. Aunque esta nueva versión del conjunto de datos de la KDD sigue mostrando algunos de los problemas comentados por McHugh [31] y puede que no sea un representante perfecto de las redes reales existentes, debido a la falta de conjuntos de datos públicos para IDS basados en redes, es posible que pueda aplicarse como un conjunto de datos de referencia efectivo para ayudar a los investigadores a comparar diferentes métodos de detección de intrusos.

B. Datasets actuales

Sperotto [32]. Fue generado en 2008 por la Universidad de Twente (UT) a partir de la recolección de datos a través de un *honeypot* que se encontraba conectado directamente a Internet, proporcionando la mayor exposición posible para la recolección de ataques reales. En este *dataset* se recogen ataques a tres servicios, FTP, SSH y HTTP, siendo parte de la información recolectada no maliciosa.

MAWI Working Lab Dataset [33]. Se trata de una base de datos creada en 2010 que ayuda a los investigadores a evaluar los métodos de detección de anomalías en tráfico de red. Se encuentra etiquetada para localizar las mismas en el archivo MAWI, en los puntos de muestreo B y F de la red. La construcción de las etiquetas se obtiene utilizando una metodología avanzada basada en gráficos que compara y combina detectores de anomalías en la red diferentes e independientes.

CTU-13 [34]. Fue generado en 2011 por la Universidad Técnica de Colorado (CTU) y contiene la captura de tráfico que genera una botnet. El objetivo era generar datos que contuviesen el tráfico de una botnet mezclado con tráfico normal. Se compone de trece escenarios distintos con muestras de diferentes botnet, en cada cual se ha usado diferente tipo de malware que usa protocolos y acciones diferentes. Para la representación y etiquetado del tráfico hace uso de NetFlows de tipo bidireccional, solventando algunos problemas que tendría el unidireccional, como la diferenciación de tráfico entre cliente y servidor, información adicional y mejora del detalle del etiquetado.

UNB-ISCX-IDS-2012 [35]. Generado en 2012, se compone de diferentes trazas de red que se encuentran etiquetadas, incluyendo el contenido de los paquetes en formato pcap. El enfoque usado su generación, se basa en el concepto de perfiles que contienen descripciones detalladas de intrusiones y modelos abstractos de distribución para aplicaciones, protocolos y redes. El tráfico analizado abarca los protocolos FTP, SSH, SMTP, POP3, IMAP y HTTP.

ADFA-LD12 [36]. Generado en 2013, tiene como objetivo la evaluación de sistemas de detección de intrusos en host (HIDS) y para ello se realizan diversos tipos de ataques contra un sistema Unix, en este caso la versión Ubuntu 11.04. Los datos se recopilan mediante la herramienta "auditd", que es un *framework* que permite auditar la seguridad a partir de los eventos generados en sistemas Unix. Se recogen diferentes tipos de ataques contra diferentes protocolos, como FTP y SSH, así como se hace uso de diferentes *frameworks* como Metasploit.

UNSW-NB15 [37]. Generado en 2015, este *dataset* fue creado usando la herramienta IXIA PerfectStor en el Cyber Range Lab de la UNSW de Canberra. Contiene tráfico híbrido entre actividades normales y diferentes tipos de ataques, en total nueve, donde podemos encontrar *Fuzzers*, *Backdoors*, *Exploits*, etc. La captura del tráfico en texto plano se realizó mediante la herramienta "tpcdump".

UGR'16 [38]. Generado en 2016 por la Universidad de Granada, usa tráfico real recogido a partir de nodos estratégicamente localizados en la red de un ISP español. Al tratarse de tráfico real, pueden contemplarse diferentes tipos de ataques de red que se observaron durante el periodo de evaluación. Está compuesto por dos tipos de conjuntos de datos, CALIBRATION, que contiene únicamente el tráfico real, y TEST, que contiene además tráfico generado sintéticamente. Además, considera la periodicidad en la evolución del tráfico y hace diferenciación entre el momento del día en que nos encontramos, así como si es laborable o no.

CIC-IDS2017 [39]. Generado en 2017, contiene tráfico normal como ataques comunes a diferentes protocolos, obteniéndose como resultado diferentes ficheros PCAP. Los

resultados además incluyes en análisis realizado mediante la herramienta CICFlowMetter con etiquetado basado en momento temporal, direcciones IP origen y destino, puertos origen y destino, protocolos y ataques. El objetivo era crear un conjunto de datos lo más realista posible y para ello usan un sistema propuesto denominado B-Profile, que permite perfilar el comportamiento abstracto de las interacciones humanas y genera tráfico normal. La generación se ha realizado mediante la creación de 25 tipos de usuarios basados en los protocolos FTP, SSH, HTTP, HTTPS y correo electrónico.

EVTX-ATTACK-SAMPLES [40]**:** Generado en 2020, contiene un conjunto de ejemplos de eventos maliciosos y que se encuentran categorizados en la matriz de MITRE ATT&CK. Cabe destacar que en ningún momento presenta la trazabilidad de los eventos generados durante un periodo de tiempo y no se podrá realizar un análisis realista que permita llevar a cabo la detección de ataques multietapa en entornos reales.

III. Requisitos para la evaluación de amenazas en sistemas de detección

Algunos autores han realizado diferentes evaluaciones, sobre aquellos *datasets* ya existentes, y han identificado un conjunto de características necesarias para poder realizar un estudio adecuado usando los mismos [41], [42]. Estos *datasets* de forma generalista, tal y como se muestra en el punto anterior, se centran en la evaluación de sistemas IDS y nosotros generalizamos a cualquier sistema de detección orientados al sistema operativo Microsoft Windows. Considerando esta diferencia, los requisitos que se pueden extraer son:

Características: El *dataset* debe incluir todos aquellos detalles asociados con el evento generado en Windows, información del usuario, *hostname*, acción ejecutada, fecha/hora del evento, proceso padre, identificador del proceso, etc. Aunque algunos *datasets* contienen características para que un tipo de sistema pueda ser entrenado, nosotros hemos recogido la información de la forma más generalista posible. De esta forma, permitimos que los datos sean transformados para poder entrenar cualquier sistema de detección basado en host y para sistemas operativos de Microsoft.

Etiquetado: Cada registro que se crea para cada evento recogido en el *dataset* se encuentra etiquetado como malicioso o no. Es posible conocer aquellos que son maliciosos a partir de las etiquetas “rule_technique_id” y “RuleName”. El resto de los registros se encuentran etiquetados acorde a lo comentado en el punto anterior.

Histórico: El *dataset* generado permite conocer un histórico de los eventos ocurridos de forma lineal. Esto es posible debido a que recogemos el proceso padre, en caso de existir, y el momento exacto en el que se generó cada evento en el sistema.

Documentación: Una descripción detallada del *dataset* es necesaria para comprender sus limitaciones y potencialidades. Algunos *datasets* publicados no proporcionan información suficiente para poder realizar una evaluación completa de los mismos. Por ejemplo, McHugh et al. [31] realizó un análisis sobre el *dataset* de DARPA y demostró algunos problemas asociados con su diseño y ejecución, los cuales no se resuelven de forma acertada. Durante este análisis comenta que algunas metodologías utilizadas en la evaluación son cuestionables y pueden haber sesgado sus resultados.

Formato: El formato seleccionado para la publicación del *dataset* es JSON. De esta forma es posible realizar transformaciones necesarias a otros formatos, como CSV, de forma sencilla. Se ha retirado del *dataset* toda aquella información de carácter sensible que pueda dar lugar a problemas graves de confidencialidad de los usuarios que han participado en la captura de eventos.

IV. Metodología para la generación del dataset

A continuación, describimos la metodología que se ha usado para producir los *datasets*. Además, se ofrece una descripción con todos los detalles relevantes acerca de las tecnologías que se han usado para la evaluación realizada.

A. Laboratorios para la recolección de eventos en tiempo real

En líneas generales, se han desarrollado dos laboratorios distintos para la generación de los *dataset*. Estos laboratorios pueden distinguirse entre: entorno real de trabajo y entorno de pruebas maliciosas. En el primero se han instalado las diferentes sondas de captura en una sala de ordenadores universitarios y se han monitorizado los mismos, recogiendo los eventos generados durante el plazo de 1 mes. Al tratarse de eventos que se generan de forma natural, sin forzar ningún tipo de actividad maliciosa, nos permite realizar un análisis realista sobre entornos que son usados por usuarios de forma habitual.

El segundo, es un laboratorio controlado y específicamente creado para la ejecución de pruebas de Ciberseguridad Ofensiva y se ha monitorizado la ejecución de las mismas por distintos usuarios durante 3 semanas. En este caso, al tratarse de un laboratorio preparado para la ejecución de diferentes técnicas maliciosas, nos permite realizar una diferenciación clara de qué es un evento malicioso y que no, permitiendo por ello el análisis y entrenamiento de diferentes algoritmos de inteligencia artificial de forma precisa. Se puede observar a continuación, en la Fig.1, de forma detallada la infraestructura y técnicas que podían usarse sobre el mismo.

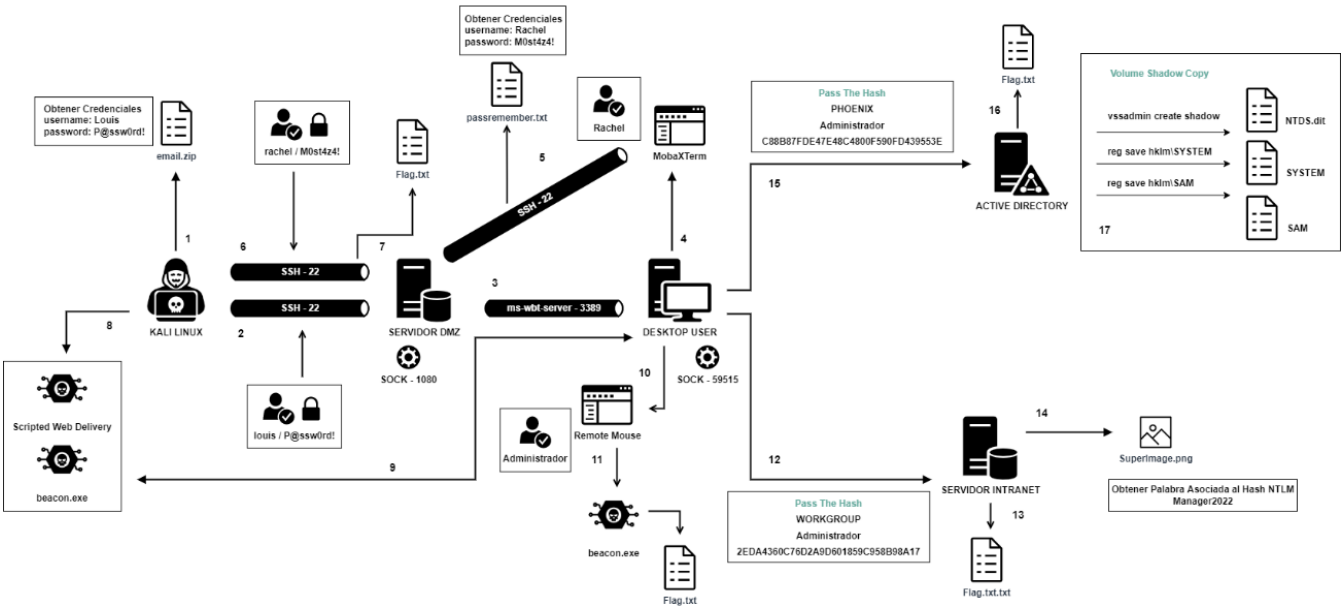

Fig 1. Infraestructura del laboratorio de pruebas de pruebas para la recolección de eventos

B. Infraestructura para la detección de eventos maliciosos (EDBS)

Cómo parte fundamental para la obtención de los eventos generados por los usuarios durante la fase de recolección se implementó el siguiente sistema de detección. Este se implementó basándose en la unificación de un conjunto de herramientas de código abierto, para automatizar el análisis de las técnicas maliciosas explicadas anteriormente. Como núcleo central se utilizó Sysmon [43], [44], que una vez instalado se ejecuta como un servicio residente en el sistema y proporciona información de la actividad a través del registro de eventos de Windows. Para la recepción y análisis centralizado de la información también se implementó un sistema basado en ELK (ElasticSearch, Logstah y Kibana) [45]. Finalmente, la comunicación entre ambos sistemas para el envío de información al sistema ELK utiliza Winlogbeat [46]. A este sistema de detección lo hemos denominado EBDS (Event-Based Detection System) [47]. El diagrama de funcionamiento del mismo puede observarse en Fig.2:

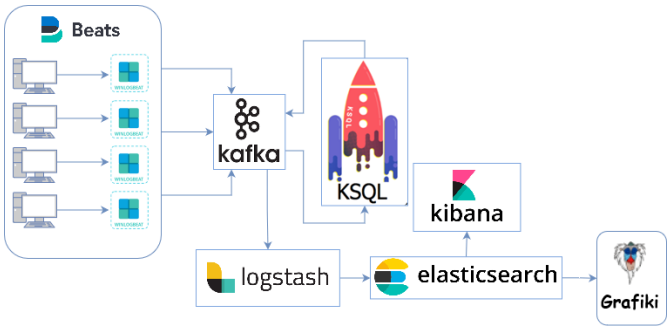

Fig 2. Infraestructura del sistema de detección de eventos (EDBS)

C. Procesamiento de los datos del dataset

La información maliciosa recolectada para los entornos real de trabajo y pruebas maliciosas se encuentra en formato JSON en dos ficheros diferenciados y recolectados en dos periodos de tiempo distintos. El *dataset* correspondiente al entorno real de trabajo, comprende el periodo comprendido entre las fechas 01/07/2022 y 30/07/2022, siendo el tamaño del *dataset* de 914GB. El *dataset* correspondiente al entorno de pruebas maliciosas, corresponde al periodo comprendido entre las fechas 16/11/2022 y 25/11/2022, siendo el tamaño del *dataset* de 155GB. La Tabla I resume la información de cada uno de los ficheros:

Tabla I
RESUMEN DE LA INFORMACIÓN DE LOS DATASETS

Características del dataset	Entorno real	Entorno pruebas
Inicio monitorización	06:26h 01/07/2022	08:25h 16/11/2022
Fin de monitorización	12:33h 30/07/2022	22:56h 25/11/2022
Nº de eventos	202304794	49914325
Nº de eventos maliciosos	631606	1713709
Tamaño	914GB	155GB

Cada evento en formato JSON que se encuentra en cada fichero tiene un formato similar al que se muestra. En este caso se muestra la parte correspondiente a un evento etiquetado como malicioso. El evento que se observa en la Fig.3, muestra la detección de un *beacon* generado con "Cobalt Strike" [48] en formato ejecutable.

```
"_source": {
  "process_guid": "4FD6B357-4B34-6375-2801-000000002900",
  "DestinationHostname": "-",
  "rule_technique_name": "Masquerading",
  "@timestamp": "2022-11-16T20:43:43.564Z",
  "process_path": "c:\\users\\louis\\desktop\\tools\\beacon.exe",
  "etl_host_agent_ephemeral_uid": "ecc16bcb-39c4-4de0-b10a-ed3c7d9bf3c3",
  "thread_id": 3244,
  "SourceHostname": "-",
  "user_domain": "desktop-4pvps6e",
  "@version": "1",
  "src_ip_rfc": "RFC_1918",
  "RuleName": "technique_id=T1036,technique_name=Masquerading",
  "event_recorded_time": "2022-11-16T20:43:44.706Z",
  "Initiated": "true",
  "event_id": "3",
  "event_timezone": "UTC",
  "meta_user_name_is_machine": "false",
  "record_number": 579925,
  "user_account": "desktop-4pvps6e\\louis",
  "process_name": "beacon.exe",
  "etl_version": "2020.04.19.01",
  "beat_version": "8.5.0",
  "rule_technique_id": "T1036",
  "event_original_time": "2022-11-16T20:43:43.564Z",
  "etl_processed_time": "2022-11-16T20:43:48.174Z",
  "user_name": "louis",
  "dst_ip_addr": "192.168.137.129",
  "src_ip_addr": "192.168.137.131",
  "SourcePort": "49791",
  "type": "wineventlog",
  "version": 5,
  "process_id": "2668",
  "host_name": "desktop-4pvps6e.phoenix.local",
  "src_ip_type": "private",
  "level": "information",
  "SourceIsIpv6": "false",
  "etl_host_agent_type": "winlogbeat",
```


Fig 3. Detección de evento malicioso. Beacon de “Cobalt Strike”

D. Información recolectada para la construcción de los datasets

El resultado es la construcción de un nuevo *dataset* que contiene dos ficheros generados a partir de los eventos recolectados en tiempo real en los diferentes sistemas Microsoft Windows. Aquellos eventos considerados maliciosos han sido etiquetados con diferentes técnicas de la matriz de MITRE ATT&CK. De esta forma, ha sido posible llevar a cabo el conteo de las tácticas y técnicas que se encuentran en cada *dataset*. En la Tabla II, se muestran los valores correspondientes al conteo de las tácticas que se encuentran registradas en los eventos maliciosos.

Tabla II
EVENTOS DETECTADOS EN LOS ENTORNOS REAL Y DE PRUEBAS Y CLASIFICADOS USANDO LA MATRIZ DE MITRE

MITRE ATT&CK Tácticas	Entorno real	Entorno pruebas
Reconocimiento	0	0
Desarrollo	0	0
Acceso inicial	0	197
Ejecución	12506	449483
Persistencia	43276	400327
Escalada de privilegios	595990	459247
Evasión de defensas	567857	1043839
Obtención de credenciales	13927	3674
Descubrimiento	572	211
Movimiento lateral	1909	1010
Recolección	28	862
Comando y control	0	120
Exfiltración	0	0
Impacto	446	0

Realizando un análisis de estas tablas, se puede observar que la mayoría de los eventos recolectados se encuentran comprendidos entre el conjunto de tácticas de Ejecución, Persistencias, Escalada de Privilegios y Evasión de Defensas. Este resultado es lógico debido a que las sondas instaladas pueden realizar la recolección de información a partir de la táctica de Acceso Inicial y un adversario realizaría de forma habitual diferentes técnicas comprendidas entre las tácticas señaladas para conseguir sus objetivos.

Como resultado del análisis exploratorio se han obtenido los siguientes mapas de calor para mostrar de forma clara cuales son las técnicas de la matriz de MITRE ATT&CK más usadas.

La Fig 4. muestra el mapa de calor correspondiente al entorno real de trabajo. Se puede diferenciar como la técnica “T1055” correspondiente a “Inyección de Procesos” y que se encuentra dentro de las tácticas “Escalada de Privilegios” y “Evasión de Defensas”, tienen un alto volumen de eventos detectados.

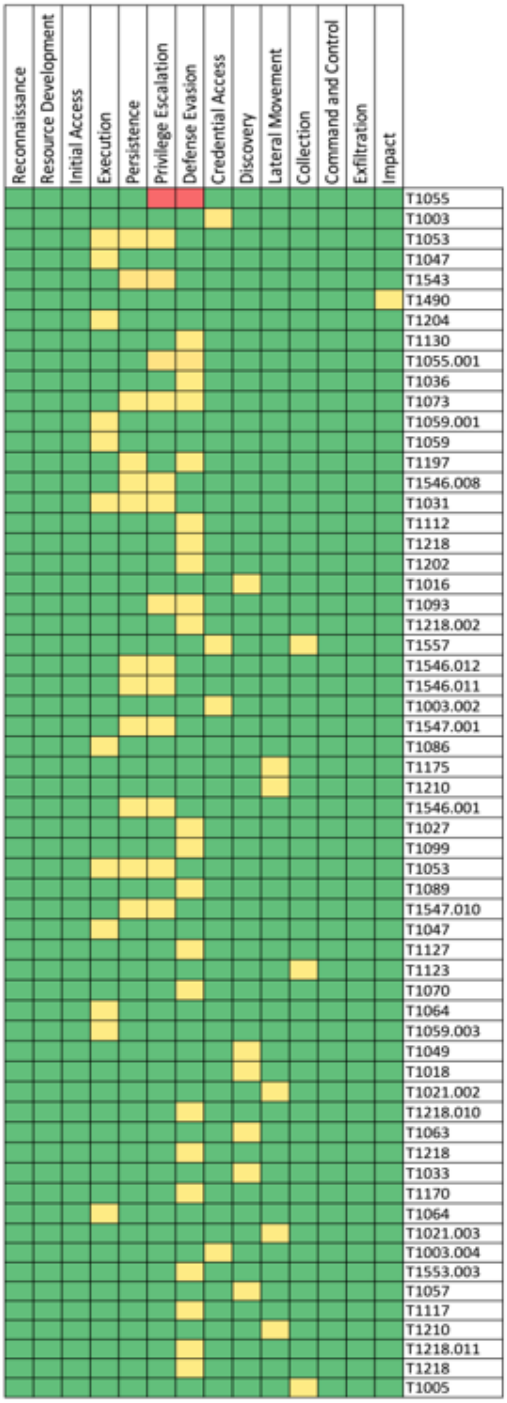

Fig 4. Mapa de calor del Entorno Real

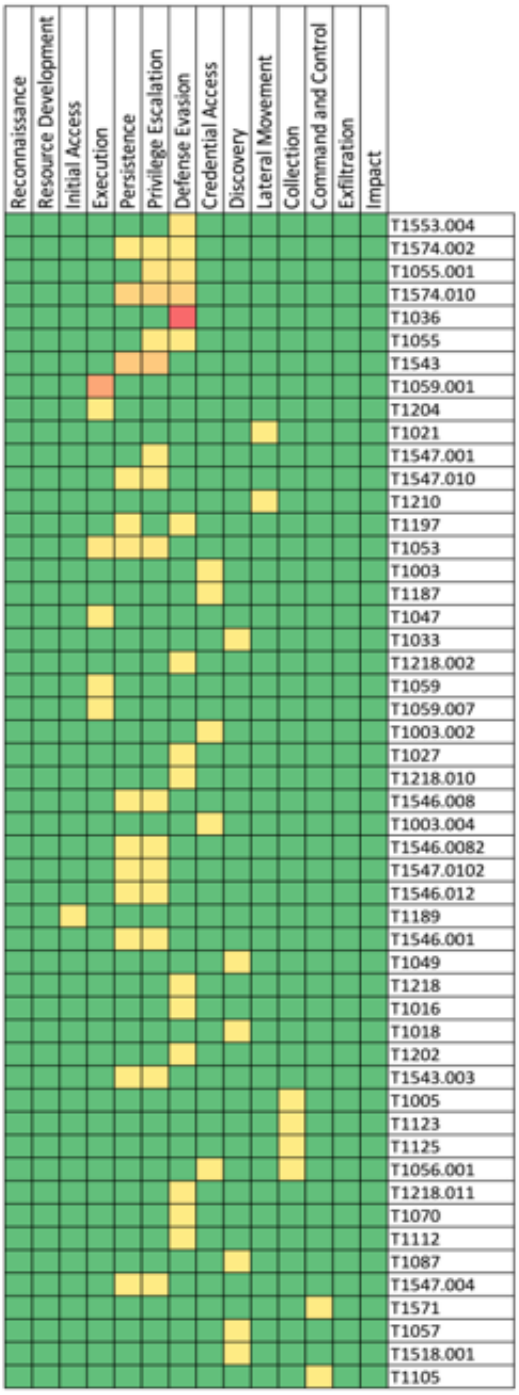

Fig 5. Mapa de calor del Entorno de Pruebas

La Fig 5. muestra el mapa de calor correspondiente al entorno de pruebas maliciosas. Se puede diferenciar como la técnica “T1036” correspondiente a “Enmascaramiento” y que se encuentra dentro de las tácticas “Evasión de Defensas”, tienen un alto volumen de eventos detectados.

V. Conclusiones

En este artículo, la principal contribución es la generación de un nuevo *dataset*, formado por dos ficheros con eventos generados sobre dos entornos distintos, entorno real de trabajo y entorno de pruebas maliciosas. Todos los eventos recolectados son a nivel de sistema y basados en el sistema operativo Microsoft Windows. Además, destacamos que no se han encontrado *datasets* con características similares al generado en la actualidad. Este *dataset* ha sido creado a partir de la monitorización de eventos en tiempo real de diferentes sistemas durante diferentes periodos de tiempo. Estos eventos se han generado tanto con acciones que usuarios realizarían de forma normal como por técnicas

maliciosas sobre el sistema.

Además, este *dataset* se encuentra etiquetado basándose en las tácticas y técnicas de la matriz de MITRE ATT&CK, permitiendo de esta forma poder obtener la trazabilidad de las acciones realizadas por los atacantes en el sistema. Para poder realizar este etiquetado de forma automática ha sido necesaria la instalación de diferentes herramientas de captura de eventos, las cuales no suponen una sobrecarga para el sistema.

Por otro lado, el elevado tamaño de los ficheros que componen el *dataset*, permite que contengan un alto volumen de eventos correspondientes a muchas de las técnicas y tácticas que se encuentran en la matriz de MITRE ATT&CK. Esto permitirá realizar análisis que se aproximen a la realidad, dado que cuenta con la ejecución de técnicas que realizan los atacantes en entornos del mundo real.

Finalmente, resaltar que el *dataset* no se encuentra disponible aún porque está siendo utilizado en la investigación que se está llevando a cabo como parte de la tesis doctoral del primer autor.

REFERENCIAS

[1] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, Dec. 2019, doi: 10.1186/s42400-019-0038-7.

[2] W. U. Hassan, A. Bates, and D. Marino, "Tactical Provenance Analysis for Endpoint Detection and Response Systems," in *2020 IEEE Symposium on Security and Privacy (SP)*, May 2020, pp. 1172–1189, doi: 10.1109/SP40000.2020.00096.

[3] N. S. Bhati, M. Khari, V. García-Díaz, and E. Verdú, "A Review on Intrusion Detection Systems and Techniques," *Int. J. Uncertainty, Fuzziness Knowlege-Based Syst.*, vol. 28, no. Supp02, pp. 65–91, 2020, doi: 10.1142/S0218488520400140.

[4] I. Debicha, B. Cochez, T. Kenaza, T. Debatty, J.-M. Dricot, and W. Mees, "Review on the Feasibility of Adversarial Evasion Attacks and Defenses for Network Intrusion Detection Systems," Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.07003.

[5] A. Kamruzzaman, S. Ismat, J. C. Brickley, A. Liu, and K. Thakur, "A Comprehensive Review of Endpoint Security: Threats and Defenses," in *2022 International Conference on Cyber Warfare and Security (ICCWS)*, Dec. 2022, pp. 1–7, doi: 10.1109/ICCWS56285.2022.9998470.

[6] Waldemar Graniszewski, "Review and analysis of EDR systems," Faculty of Electrical Engineering (FoEE), 2022.

[7] N. N. A. Sjarif *et al.*, "Endpoint Detection and Response: Why Use Machine Learning?," in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2019, pp. 283–288, doi: 10.1109/ICTC46691.2019.8939836.

[8] H. Kaur and R. Tiwari, "Endpoint detection and response using machine learning," *J. Phys. Conf. Ser.*, vol. 2062, no. 1, p. 012013, Nov. 2021, doi: 10.1088/1742-6596/2062/1/012013.

[9] N. T. Singh and R. Chadha, "A Review Paper on Network Intrusion Detection System," 2023, pp. 453–463.

[10] P. Khandelwal, P. Likhar, and R. S. Yadav, "Machine Learning Methods leveraging ADFA-LD Dataset for Anomaly Detection in Linux Host Systems," in *2022 2nd International Conference on Intelligent Technologies (CONIT)*, Jun. 2022, pp. 1–8, doi: 10.1109/CONIT55038.2022.9848305.

[11] S. Jin, J.-G. Chung, and Y. Xu, "Signature-Based Intrusion Detection System (IDS) for In-Vehicle CAN Bus Network," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2021, pp. 1–5, doi: 10.1109/ISCAS51556.2021.9401087.

[12] V. Jyothsna, V. V. Rama Prasad, and K. Munivara Prasad, "A Review of Anomaly based Intrusion Detection Systems," *Int. J. Comput. Appl.*, vol. 28, no. 7, pp. 26–35, Aug. 2011, doi: 10.5120/3399-4730.

[13] M. Y. Wanjala and N. & M. Jacob, "Review of Viruses and Antivirus Patterns," *Glob. J. Comput. Sci. Technol.*, vol. 17, no. 3, pp. 1–5, 2017.

[14] M. Al-Asli and T. A. Ghaleb, "Review of Signature-based Techniques in Antivirus Products," in *2019 International Conference on Computer and Information Sciences (ICCIS)*, Apr. 2019, pp. 1–6, doi: 10.1109/ICCISci.2019.8716381.

[15] J. Scott, "Signature Based Malware Detection is Dead," 2017.

[16] V. S. Sathyanarayan, P. Kohli, and B. Bruhadeshwar, "Signature Generation and Detection of Malware Families," in *Information Security and Privacy*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 336–349.

[17] Z. Bazrafshan, H. Hashemi, S. M. H. Fard, and A. Hamzeh, "A survey on heuristic malware detection techniques," in *The 5th Conference on Information and Knowledge Technology*, May 2013, pp. 113–120, doi: 10.1109/IKT.2013.6620049.

[18] S. Treadwell and M. Zhou, "A heuristic approach for detection of obfuscated malware," in *2009 IEEE International Conference on Intelligence and Security Informatics*, Jun. 2009, pp. 291–299, doi: 10.1109/ISI.2009.5137328.

[19] A. L. D. Harley, "Heuristic Analysis – Detecting Unknown Viruses." 2007.

[20] Xiao-bin Wang, Guang-yuan Yang, Yi-chao Li, and Dan Liu, "Review on the application of artificial intelligence in antivirus detection systemi," in *2008 IEEE Conference on Cybernetics and Intelligent Systems*, Sep. 2008, pp. 506–509, doi: 10.1109/ICCIS.2008.4670733.

[21] P. Singhal, "Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks," *Int. J. Netw. Secur. Its Appl.*, vol. 4, no. 1, pp. 61–67, Jan. 2012, doi: 10.5121/ijnsa.2012.4106.

[22] S. M. L. de Lima *et al.*, "Artificial intelligence-based antivirus in order to detect malware preventively," *Prog. Artif. Intell.*, Oct. 2020, doi: 10.1007/s13748-020-00220-4.

[23] E. LeMay, W. Unkenholz, D. Parks, C. Muehrcke, K. Keefe, and W. H. Sanders, "Adversary-driven state-based system security evaluation," in *Proceedings of*

the 6th International Workshop on Security Measurements and Metrics, Sep. 2010, pp. 1–9, doi: 10.1145/1853919.1853926.

[24] K. S. Babu and Y. N. Rao, "MCGAN: Modified Conditional Generative Adversarial Network (MCGAN) for Class Imbalance Problems in Network Intrusion Detection System," *Appl. Sci.*, vol. 13, no. 4, p. 2576, Feb. 2023, doi: 10.3390/app13042576.

[25] A. Alshaibi, M. Al-Ani, A. Al-Azzawi, A. Konev, and A. Shelupanov, "The Comparison of Cybersecurity Datasets," *Data*, vol. 7, no. 2, p. 22, Jan. 2022, doi: 10.3390/data7020022.

[26] R. Al-Shaer, J. M. Spring, and E. Christou, "Learning the Associations of MITRE ATT&CK Adversarial Techniques," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2005.01654.

[27] O. Yavanoglu and M. Aydos, "A review on cyber security datasets for machine learning algorithms," in *2017 IEEE International Conference on Big Data (Big Data)*, Dec. 2017, pp. 2186–2193, doi: 10.1109/BigData.2017.8258167.

[28] R. P. Lippmann *et al.*, "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, vol. 2, pp. 12–26, doi: 10.1109/DISCEX.2000.821506.

[29] S. J. Stolfo, Wei Fan, Wenke Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: results from the JAM project," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, vol. 2, pp. 130–144, doi: 10.1109/DISCEX.2000.821515.

[30] 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*, Jul. 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[31] J. McHugh, "Testing Intrusion detection systems," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, Nov. 2000, doi: 10.1145/382912.382923.

[32] A. Sperotto, R. Sadre, F. van Vliet, and A. Pras, "A Labeled Data Set for Flow-Based Intrusion Detection," 2009, pp. 39–50.

[33] J. Mazel, R. Fontugne, and K. Fukuda, "A taxonomy of anomalies in backbone network traffic," in *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug. 2014, pp. 30–36, doi: 10.1109/IWCMC.2014.6906328.

[34] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, Sep. 2014, doi: 10.1016/j.cose.2014.05.011.

[35] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, May 2012, doi: 10.1016/j.cose.2011.12.012.

[36] G. Creech and J. Hu, "A Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguousand Discontiguous System Call Patterns," *IEEE Trans. Comput.*, vol. 63, no. 4, pp. 807–819, Apr. 2014, doi: 10.1109/TC.2013.13.

[37] 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)*, Nov. 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

[38] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, M. Fuentes-García, P. García-Teodoro, and R. Theron, "UGR'16: Un nuevo conjunto de datos para la evaluación de IDS de red," in *Proceedings XIII Jornadas de Ingenieria Telematica - JITEL2017*, Sep. 2017, pp. 71–78, doi: 10.4995/JITEL2017.2017.6520.

[39] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116, doi: 10.5220/0006639801080116.

[40] S. Bousseaden, "EVTX-ATTACK-SAMPLES," 2020. https://github.com/sbousseaden/EVTX-ATTACK-SAMPLES.

[41] B. Sangster *et al.*, "Toward instrumenting network warfare competitions to generate labeled datasets," in *CSET'09: Proceedings of the 2nd conference on Cyber security experimentation and test*, 2009, p. 9, doi: https://dl.acm.org/doi/10.5555/1855481.1855490.

[42] A. Thakkar and R. Lohiya, "A Review of the Advancement in Intrusion Detection Datasets," *Procedia Comput. Sci.*, vol. 167, pp. 636–645, 2020, doi: 10.1016/j.procs.2020.03.330.

[43] A. Sapegin, D. Jaeger, F. Cheng, and C. Meinel, "Towards a system for complex analysis of security events in large-scale networks," *Comput. Secur.*, vol. 67, pp. 16–34, Jun. 2017, doi: 10.1016/j.cose.2017.02.001.

[44] T. Tsigkritis and G. Spanoudakis, "Assessing the genuineness of events in runtime monitoring of cyber systems," *Comput. Secur.*, vol. 38, pp. 76–96, Oct. 2013, doi: 10.1016/j.cose.2013.03.011.

[45] V. Mavroeidis and A. Jøsang, "Data-Driven Threat Hunting Using Sysmon," in *Proceedings of the 2nd International Conference on Cryptography, Security and Privacy*, Mar. 2018, pp. 82–88, doi: 10.1145/3199478.3199490.

[46] Microsoft, "Sysmon." 1996, [Online]. Available: https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon.

[47] A. Pérez-Sánchez and R. Palacios, "Evaluation of Local Security Event Management System vs. Standard Antivirus Software," *Appl. Sci.*, vol. 12, no. 3, p. 1076, Jan. 2022, doi: 10.3390/app12031076.

[48] L. . Pin Mera and F. X. Pinargote Espinoza, "Análisis y simulación de COBALT STRIKE, metasploit y PUPYRAT para conocer las características o patrones de ataques.," 2022.