Universidad de Salamanca

Departamento de Matemática Aplicada

Nuevas perspectivas en el estudio de amenazas persistentes avanzadas

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Tesis para optar al grado de

Doctor por la Universidad de Salamanca

Santiago Quintero Bonilla

Director: Dr. Ángel Martín del Rey

2021

Esta tesis está dedicada a mi esposa Farrah y a mi hija Fariely Sarai.

Familia esto también va por vosotros.

Declaración de autoría

D. Santiago Quintero Bonilla, presenta la tesis doctoral titulada "Nuevas perspectivas en el estudio de amenazas persistentes avanzadas", para optar al Grado de Doctor por la Universidad de Salamanca; y declara que este proyecto ha sido realizado bajo la dirección del Dr. Ángel Martín del Rey, profesor Titular de Universidad, del Departamento de Matemática Aplicada, Instituto Universitario de Física Fundamental y Matemáticas (IUFFyM), de la Universidad de Salamanca.

En Salamanca, a 21 de abril de 2021

Santiago Quintero Bonilla

Director

Dr. Ángel Martín del Rey

Agradecimientos

En primer lugar, he de agradecer a Dios por guiarme en todo momento durante día y noche para realizar este proyecto académico; por ser la luz en tiempos difíciles y permitirme estar a 8.200 kilómetros de distancia, lejos de mi hogar.

A mi tutor/director de tesis Ángel Martín del Rey, que desde el primer momento y luego en Salamanca me ha brindado sus consejos tanto personales como académicos. Esto me ha permitido crecer como persona e investigador; además gracias por vuestras sugerencias, revisiones y sabias correcciones durante todo el programa de doctorado.

A mi esposa Farrah por todo tu apoyo, cariño y motivación. Por ser mi compañera en este reto, tanto en las alegrías y en las no tan alegres; sin ti, esto no hubiese sido posible. Y a ti Fary, por haber llegado en el mejor momento de nuestras vidas a compartir esta experiencia con nosotros.

A mis padres, a mi hermano y a mis hermanas que en la distancia me han apoyado durante estos años.

A las personas que depositaron su confianza desde el principio en este proyecto de superación personal, Gustavo y Rafael a ustedes muchísimas gracias.

A todas y cada una de las personas que me han apoyado en este trabajo, sin los cuales, esta Tesis doctoral jamás hubiese sido terminada.

A la Universidad de Salamanca y al grupo de investigación GIDSIMAD por brindarme apoyo tanto en las gestiones académicas y administrativas; y por el espacio e infraestructura tecnológica puestas a disposición para desarrollar esta investigación.

Al Instituto de Tecnologías Físicas y de la Información "Leonardo Torres Quevedo" (ITEFI), instituto propio del Consejo Superior de Investigaciones Científicas (CSIC) por facilitarme sus instalaciones y permitirme realizar la estancia de investigación.

A la Universidad Tecnológica de Panamá por el apoyo administrativo recibido y la excedencia concedida para culminar estos estudios.

Y por último y no menos importante, agradezco a la Secretaría Nacional de Ciencia, Tecnología e Innovación (SENACYT) y al Instituto para la Formación y Aprovechamiento de Recursos Humanos (IFARHU), instituciones panameñas que me han otorgado la oportunidad de realizar este programa de doctorado.

En Salamanca, 21 de abril de 2021

Santiago Quintero Bonilla

Resumen

Una amenaza persistente avanzada es un ataque sofisticado, dirigido, selectivo y personalizado, que representa un riesgo para todas las organizaciones, especialmente aquellas que gestionan datos confidenciales o son infraestructuras críticas.

En los últimos años, el análisis de estas amenazas ha llamado la atención de la comunidad científica; los investigadores han estudiado el comportamiento de esta amenaza para crear modelos y herramientas que permitan la detección temprana de estos ataques.

El uso de la inteligencia artificial y el aprendizaje automático pueden ayudar a detectar, alertar y predecir automáticamente este tipo de amenazas y reducir el tiempo que el atacante puede permanecer en la red de la organización.

El objetivo de esta tesis es desarrollar un modelo teórico que permita detectar las amenazas persistentes avanzadas de manera temprana, basado en el ciclo de vida del ataque y utilizando métodos y técnicas de aprendizaje automático.

La metodología que se ha seguido para la realización de este trabajo comenzó con una revisión bibliográfica de los conceptos de amenaza persistente avanzada y de las aplicaciones de detección en el contexto de la ciberseguridad. Además, se analizaron los ciclos de vida existentes que explican el proceso que siguen estas amenazas durante su ejecución.

Posteriormente, se desarrolló un modelo para la detección temprana de las amenazas persistentes avanzadas basado en un ciclo de vida de 6 etapas, que han sido divididas en

etapas activas, pasivas y recurrentes; además, se han utilizado técnicas de aprendizaje automático para la detección de URL maliciosas, *phishing* y anomalías en la red.

En conclusión, los ataques de amenazas persistentes avanzadas son difíciles de detectar debido a la capacidad y los recursos con los que cuentan los grupos que las desarrollan. El objetivo de estos ataques es permanecer activos el mayor tiempo posible durante la ejecución de la intrusión.

Uno de los problemas detectados durante la realización de este trabajo ha sido que no se encuentran disponibles conjuntos de datos reales que permitan el entrenamiento de los algoritmos de aprendizaje automático de forma eficiente, por lo que ha sido necesario crear conjuntos de datos semi reales a partir de muestras de malware.

Finalmente, como trabajo futuro, se recomienda que el modelo que ha sido propuesto en este trabajo sea probado en un entorno informático controlado, para evitar ocasionar perjuicios.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La ciberseguridad es el área de la informática responsable de crear y gestionar mecanismos de seguridad, que establecen los pasos a seguir para garantizar la seguridad de los datos dentro de la infraestructura tecnológica de una organización. Sin embargo, algunos fallos y vulnerabilidades de seguridad, como pueden ser la utilización de equipos obsoletos, políticas de seguridad desactualizadas, instalación de las actualizaciones de software que no se realizan a tiempo, además de la falta de concienciación de los propios usuarios, permiten que los atacantes puedan realizar una intrusión a la red.

El creciente desarrollo de herramientas sofisticadas utilizadas por los ciberdelincuentes, como las vulnerabilidades de día cero y los ataques de denegación de servicio, hacen casi imposible que las soluciones convencionales, como los cortafuegos, antivirus o los sistemas de detección de intrusos, puedan hacer frente a la complejidad actual de este tipo de herramientas.

Una amenaza persistente avanzada (APT, en inglés *Advanced Persistent Threat*) es un ataque sofisticado, dirigido, selectivo y personalizado que busca obtener acceso no autorizado a los sistemas de información y comunicación, con la finalidad de filtrar datos confidenciales o causar daños a una empresa, industria u organización gubernamental. Desde la aparición de Stuxnet [28], estos ataques son cada vez más cautelosos y dañinos,

mostrando la facilidad de intrusión a sistemas de alto perfil y evadiendo muchas de las herramientas de defensa más sofisticadas utilizadas para proteger el entorno informático.

Este tipo de amenazas pueden permanecer por largos periodos de tiempo sin llegar a ser detectadas; sin embargo, cuando son detectadas, gran parte de ellas reaparecen con modificaciones para lograr su objetivo; algunos ejemplos de ataques que han causado importantes pérdidas de dinero, información confidencial y propiedad intelectual son FIN6 [30], APT28 [29], APT10 [19], APT1 [58] o APT41 [32].

Hoy en día, los ataques de APT representan una amenaza real para las entidades públicas y privadas de todo el mundo y seguirán siéndolo en el futuro. Estos ataques son una amenaza inminente, cuyo principal problema es la dificultad de detección temprana, ya que los atacantes utilizan diferentes técnicas, tanto para permanecer el mayor tiempo posible sin ser detectados, como para evadir de manera eficiente los sistemas de seguridad.

Consecuentemente, algunas herramientas de seguridad existentes, como las basadas en firmas, no son efectivas para detectar las amenazas persistentes avanzadas, debido a que estas utilizan patrones de detección predefinidos que pueden ser demasiado lentos para detectar ataques dirigidos. Los métodos de aprendizaje automático pueden ser más eficaces para detectar estas amenazas, puesto que buscan automáticamente patrones y eventos anómalos [44].

Las diferencias entre una amenaza persistente avanzada y un ciberataque común son significativas; una de ellas es el número de recursos necesarios para llevar a cabo el ataque. Un ciberataque común puede dirigirse a entidades u organizaciones con políticas de ciberseguridad nulas o deficientes, con el fin de robar datos de clientes o de actividad financiera de una empresa; estos ataques suelen ser detectados y el daño causado no suele ser crítico. Sin embargo, una amenaza persistente avanzada puede tener como objetivo grandes organizaciones y sectores industriales, donde causará

graves daños, como el robo de propiedad intelectual, fallos en servicios esenciales o la destrucción de la infraestructura crítica. Estos ataques no suelen ser detectados tempranamente y el daño causado puede resultar crítico.

En los últimos años, se han propuesto ciclos de vida que intentan explicar el funcionamiento de las amenazas persistentes avanzadas, considerando el ciclo de vida como el tiempo que transcurre desde que se inicia el estudio del objetivo hasta que se extraen los datos de dicho objetivo hacia los servidores del atacante. El ciclo de vida identifica y describe las tácticas, técnicas y procedimientos que pueden ser utilizados en cada una de las etapas del proceso. Algunos ejemplos de ciclos de vida de un ataque de APT son: el modelo de tres etapas de Ussath [89], el modelo de cuatro etapas IKC [90], el modelo de cinco etapas AC [74], el modelo de siete etapas CKC [56] y el modelo de ocho etapas de Mandiant/Fireeye [58].

Por otro lado, se han propuesto diferentes enfoques para la detección de amenazas persistentes avanzadas utilizando aprendizaje automático, como los planteados por Bai [9], Ghafir [35] y Zhang [97].

El enfoque propuesto por Bai se ha basado en la detección de anomalías en el protocolo de sesiones remotas de Windows; estas sesiones remotas, pueden ser consideradas como un método de intrusión de la etapa de movimiento lateral en el ciclo de vida de una amenaza persistente avanzada. Esta propuesta, utiliza los algoritmos *logistic regression*, *decision tree*, *Gaussian naive bayes* y *LogitBoost* de aprendizaje automático [9].

El modelo propuesto por Ghafir detecta amenazas persistentes avanzadas a partir de alertas tempranas que se crean a partir de la correlación de varios módulos de detección. Los algoritmos de aprendizaje automático utilizados en esta propuesta son *decision tree*, *support vector machine*, *k*-NN y *ensamble learning*. Además, este modelo utiliza un ciclo de vida de seis etapas [35].

Finalmente, Zhang ha propuesto un método que simula escenarios de ataque sobre los registros de un sistema de detección de intrusos, que utiliza un ciclo de vida de una amenaza persistente avanzada de cuatro etapas, basado en el enfoque de IKC. Estos escenarios creados han servido como guía para la detección y mitigación de ataques dirigidos. El algoritmo de aprendizaje automático utilizado en esta propuesta es el *Fuzzy clustering* [97].

En este trabajo se plantea un modelo de ciclo de vida de ataque de APT que consta de seis etapas que han sido clasificadas en activas, pasivas y recurrentes. Este modelo se ha propuesto con el objetivo de facilitar la detección de las amenazas persistentes avanzadas de manera temprana y eficiente. Cabe destacar que este planteamiento se apoya en algoritmos y técnicas de aprendizaje automático y ha tenido en cuenta las tácticas, técnicas y procedimientos comúnmente utilizados por los atacantes de este tipo de amenazas.

1.1. Hipótesis

La hipótesis inicial de este trabajo es que las amenazas persistentes avanzadas pueden ser detectadas de manera temprana, a través del análisis de las herramientas utilizadas para realizar el ataque, y a partir del uso de las técnicas y métodos de aprendizaje automático aplicados a cada una de las etapas del ciclo de vida de un ataque de APT.

En esta tesis se ha propuesto un nuevo modelo de enfoque teórico para la detección de amenazas persistentes avanzadas, basado en el ciclo de vida de un ataque de APT, y utilizando para ello técnicas de aprendizaje automático.

Se espera que el modelo propuesto identifique los posibles ataques por etapas del ciclo de vida, facilitando la detección de amenazas persistentes avanzadas. Como

consecuencia, este modelo podrá facilitar la detección temprana de comportamientos anómalos en la red informática.

1.2. Metodología

La metodología que se ha seguido para la realización de esta tesis se describe a continuación:

- Revisión bibliográfica del concepto de amenaza persistente avanzada.
- Revisión bibliográfica de las aplicaciones de detección de ataques en ciberseguridad y de amenaza persistente avanzada.
- Análisis de los ciclos de vida existentes para explicar el proceso de un ataque de amenaza persistente avanzada.
- Definición de un modelo basado en el ciclo de vida, donde se utilizan técnicas de aprendizaje automático para la detección temprana de una amenaza persistente avanzada.

1.3. Objetivos

El objetivo general de esta tesis es desarrollar un modelo teórico basado en el ciclo de vida del ataque y utilizando métodos y técnicas de aprendizaje automático, que permita detectar las amenazas persistentes avanzadas de manera temprana.

Los objetivos específicos son los siguientes:

- Realizar una revisión bibliográfica de los modelos y técnicas existentes para la detección de una amenaza persistente avanzada.

- Analizar los ciclos de vida de los ataques de amenazas persistentes avanzadas para encontrar semejanzas entre las diferentes etapas e identificar las características más relevantes.
- Seleccionar los métodos y técnicas de aprendizaje automático que pueden ayudar a detectar las tácticas, técnicas y procedimientos utilizados durante un ataque de amenaza persistente avanzada.
- Definir un modelo basado en el ciclo de vida, que permita detectar amenazas persistentes avanzadas, utilizando para ello métodos y técnicas de aprendizaje automático.

1.4. Estructura de la tesis

De acuerdo con los temas que abarca esta tesis, la estructura que se ha propuesto es la siguiente:

Capítulo 2

Este capítulo define de manera general las amenazas persistentes avanzadas, así como sus características y las diferencias frente a otras amenazas o ataques. A continuación, se describe el proceso de un ataque, junto con los métodos y técnicas utilizadas. Se detallarán los problemas de atribución que presentan estas amenazas, junto con sus actores, campañas y grupos.

Capítulo 3

Este capítulo presenta el concepto de aprendizaje automático, sus diferentes aplicaciones, las técnicas y los algoritmos utilizados para la detección de amenazas persistentes avanzadas que ofrece este subcampo de la inteligencia artificial.

Capítulo 4

Este capítulo describe el papel que juega el aprendizaje automático en el campo de la ciberseguridad para la detección de amenazas persistentes avanzadas. También se analizan los diferentes enfoques que se han propuesto para detectar estas amenazas utilizando aprendizaje automático.

Capítulo 5

En este capítulo se realiza un análisis del ciclo de vida de un ataque de amenaza persistente avanzada, específicamente de los modelos y firmas de seguridad especializadas en este tipo de amenaza, que han sido propuestos por diferentes autores.

Capítulo 6

En este capítulo se incluye una discusión sobre los modelos propuestos anteriormente. A continuación, se describe una propuesta novedosa para la detección de amenazas persistentes avanzadas, utilizando las técnicas de aprendizaje automático y basado en el ciclo de vida de un ataque.

Capítulo 7

En este capítulo se presentan las conclusiones y contribuciones de esta tesis. Además, se mencionan las publicaciones que apoyan esta tesis, así como las posibles líneas de investigación futuras.

Capítulo 2

Amenazas Persistentes Avanzadas

Una amenaza persistente avanzada, se puede definir de diferentes maneras, en esta investigación, se optará por la definición propuesta por el *US National Institute of Standards and Technology (NIST)* [63], que establece que: "La amenaza persistente avanzada es un ataque dirigido con niveles sofisticados de pericia y recursos, que le permiten a los atacantes, por medio del uso de múltiples vectores de ataque (malware, vulnerabilidades, ingeniería social, entre otras), generar oportunidades para alcanzar sus objetivos, que habitualmente son establecer y extender su posicionamiento dentro de la infraestructura de tecnología de la información de organizaciones, con el objetivo de filtrar información hacia el exterior continuamente, minar e impedir aspectos importantes de una misión, un programa o una organización, o ubicarse en una posición que le permita hacerlo en el futuro. Además, la amenaza persistente avanzada persigue sus objetivos repetidamente durante un extenso periodo de tiempo, adaptándose a las medidas de defensa de la víctima, y con la determinación de mantener el nivel de interacción necesario para ejecutar sus objetivos".

En los últimos años, el número de informes relacionados con los ataques de APT ha aumentado considerablemente (ver Figura 2.1) [22, 33, 55]. Uno de los principales objetivos de los atacantes es permanecer sin ser detectados durante un largo periodo

de tiempo, lo que ha supuesto un gran reto en el mundo digital para afrontar la complejidad de este tipo de amenazas. En muchas ocasiones, la detección de ataques en etapas tempranas, es complicada, debido a que las técnicas utilizadas para efectuar el ataque cambian constantemente para no ser detectadas, y así mantener la presencia.

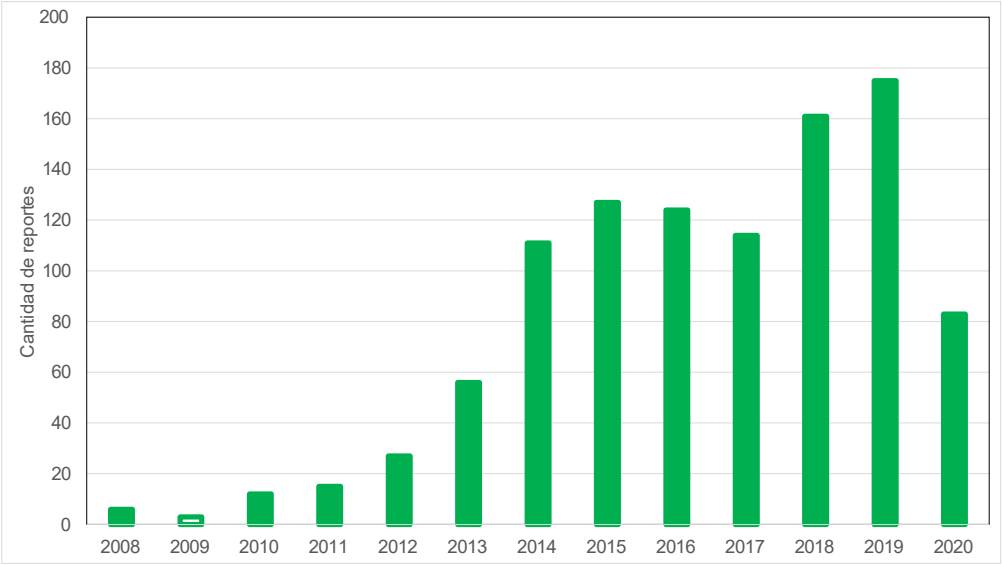


Figura 2.1 Informes de casos reportados de ataques de APT desde el 2008 al 2020 [22].

Un vector de ataque se puede definir como las técnicas y métodos utilizados durante la ejecución del ataque; estos vectores tiene características especiales y específicos para cada objetivo, por ejemplo, la ingeniería social puede crear mensajes de acuerdo a los intereses del usuario. Para crear un vector de ataque, los atacantes realizan una etapa de recopilación de información y reconocimiento del objetivo, a través de ataques a dispositivos móviles, explotación de vulnerabilidades de servicios, o incluso con la distribución de memorias USB infectadas.

Los atacantes realizan un estudio del comportamiento de la víctima, mostrando especial interés por los usuarios descuidados con su vida digital, que pueden ser utilizados como vector para la intrusión inicial, para no levantar sospechas. Una vez dentro de los sistemas de información de la organización, los atacantes intentan moverse

lateralmente dentro de la red, realizando búsquedas de credenciales con altos privilegios para acceder a servicios que almacenan información confidencial.

Como ya se ha mencionado anteriormente, uno de los primeros ataques de APT conocidos es Stuxnet, descubierto en septiembre de 2011. Este gusano informático, atacó las plantas industriales que utilizaban controladores Siemens, paralizando el funcionamiento de estos controladores durante un corto periodo de tiempo. En Irán, se vieron afectados más de 60.000 ordenadores, mientras que en Indonesia e India resultaron afectados unos 15.000 equipos. Este ataque se ha atribuido a los gobiernos de Estados Unidos e Israel, que buscaban detener el programa nuclear iraní [28, 94].

2.1. Características

Los ataques de APT poseen diversas características, una de ellas es que son ataques altamente organizados con objetivos específicos. Además, los atacantes suelen estar provistos de una gran cantidad de recursos económicos, utilizan vectores de ataque de manera persistente e implementan técnicas evasivas muy complejas [76].

En 2006, los analistas de la Fuerza Aérea de los Estados Unidos (USAF) acuñaron el término "amenaza persistente avanzada" para facilitar el debate sobre las actividades de intrusión con sus homólogos civiles no autorizados [42]. En este sentido, los equipos militares podían discutir las características del ataque, sin revelar identidades clasificadas. Se detallan a continuación los términos utilizados por la USAF:

- **Amenaza**: el enemigo está organizado, financiado y motivado.
- **Persistente**: el enemigo tiene la intención de cumplir una misión, recibe órdenes y trabaja para alcanzar metas específicas.
- **Avanzada**: el enemigo está familiarizado con las herramientas y técnicas de intrusión y es capaz de desarrollar *exploits* personalizados.

La ciberseguridad es importante en el ámbito gubernamental, y a su vez, ha adquirido interés en la sociedad en general. Consecuentemente, organizaciones e infraestructuras se han visto obligadas a permanecer conectadas a Internet para estar a la vanguardia de la demanda actual. Por este motivo, las amenazas cibernéticas preocupan a diversos sectores y las acciones que puede realizar un atacante para obtener ventaja sobre sus adversarios en el ciberespacio se denominan conflictos cibernéticos. Estos conflictos se pueden clasificar de la siguiente forma [95]:

- Hacktivismo/Cibervandalismo: son las personas o grupos de personas no estatales, cuya motivación es la protesta, reivindicación o reputación, y las acciones que realizan son generalmente por motivos ideológicos.
- Ciberdelincuencia: son los actos de delincuencia organizada, realizados por grupos que atacan a personas o empresas con el fin de obtener beneficios económicos; esto se consigue mediante el fraude, robo o extorsión.
- Ciberespionaje: generalmente es de carácter económico o político-militar. En el primer caso, se ataca a las empresas para obtener secretos comerciales o propiedad intelectual. En el ámbito político-militar, el ataque esta dirigido a naciones o sus representantes en el extranjero.
- Cibersabotage: puede dividirse en dos sectores, el de carácter económico y el político-militar. En el sector económico, los competidores deben ser eliminados, lo que se consigue, generalmente, alterando la integridad o disponibilidad de los sistemas o procesos, ya sea destruyéndolos o causando cierto daño. De igual forma, se utilizan las mismas premisas en el ámbito político-militar.
- Ciberterrorismo: describe a las personas o grupos terroristas que persiguen objetivos políticos e ideológicos, buscando causar el mayor daño posible. Sus

objetivos son las infraestructuras críticas, que son atacadas a través de acciones sistemáticas violentas.

- Guerra cibernética: son situaciones en las que una entidad, organización o individuo, ataca a otra entidad, a través del ciberespacio, con el propósito de robar información, afectar al rendimiento del entorno informático de sus adversarios, o sabotear sistemas físicos o centros de información. Sus objetivos tienen un enfoque político-militar.

Una vez detallados estos términos, se entiende como ciberataque a la ejecución práctica de acciones individuales que se utilizan en todos los diferentes tipos de conflictos cibernéticos. Un ciberataque se lleva a cabo mediante determinados programas maliciosos [40]. Algunos de los programas maliciosos más utilizados en los ataques de APT son los siguientes:

- Virus informáticos: son programas que tienen como objetivo propagarse a otros programas de manera oculta y maliciosa con una copia de ellos mismos, lo que ocasiona importantes daños en los sistemas informáticos pudiendo llegar a inutilizar una red local. El comportamiento del virus informático es semejante al del virus biológico.

- Gusanos informáticos: son programas independientes y que se auto-replican, propagándose de ordenador a ordenador a través de conexiones de red, aprovechando algún medio, como el correo electrónico, una red compartida, una memoria USB, etc. La propagación mediante la infección por una memoria USB se sigue utilizando actualmente. Este método es necesario para llegar a los sistemas informáticos que no están conectados a Internet.

- Troyanos: son programas que se hacen pasar por una aplicación legítima pero contiene otro programa o bloque de código indeseado, malicioso y destructivo,

es decir, es un código deliberadamente disfrazado. Un ejemplo de un troyano es un falso antivirus, es un programa que pretende ser un antivirus pero que en realidad es un malware en sí mismo.

Una de las características principales de los ataques cibernéticos clásicos es que se propagan ampliamente por Internet y, por lo general, su naturaleza y su comportamiento es el mismo en cada ejecución. Los virus informáticos, gusanos informáticos y troyanos atacan a todas las víctimas que encuentran a su paso, sin distinción alguna.

Si se compara con un ciberataque, un ataque de amenaza persistente avanzada es elaborado y adaptado a una o más víctimas específicas; mientas que el malware que se utiliza en un ciberataque no está dirigido a las víctimas seleccionadas y no se distribuye de forma masiva (ver Figura 2.2).

Durante un ataque de APT se utilizan técnicas complejas para superar las tecnologías de defensa convencionales. Después de la intrusión exitosa en una red, se utilizan técnicas de ocultación para permanecer durante mucho tiempo en ella y robar datos confidenciales. La detección de las amenazas persistentes avanzadas suele ser más difícil que la detección de los ataques cibernéticos clásicos.

La Oficina Federal de Seguridad de la Información en Alemania (BSI), identifica los ataques de APT de los ciberataques teniendo en cuenta las características específicas que se detallan a continuación [14].

- El objetivo de los atacantes es obtener acceso a la red de la víctima durante el mayor tiempo posible, con la finalidad de robar datos confidenciales.
- Se utiliza una combinación de ingeniería social, herramientas y diferentes técnicas para llevar a cabo el ataque.
- Se utilizan correos electrónicos dirigidos con código malicioso adjunto, especialmente adaptados a la víctima.

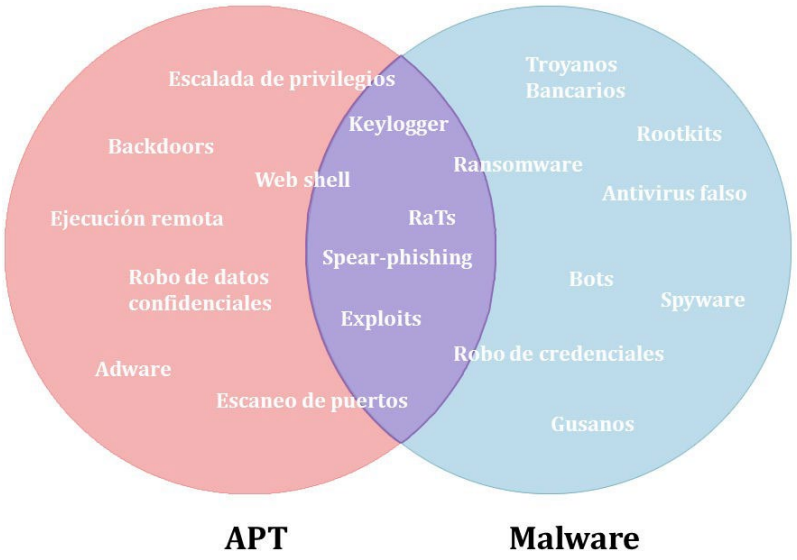


Figura 2.2 Ataque cibernético con malware vs. ataque de APT

- Se explotan vulnerabilidades desconocidas para las que no existen parches de seguridad.
- Se desarrollan funciones para el camuflaje y la ocultación de huellas, de modo que el malware no se detecte, y se convierta en un espía persistente.

Entre las características que se pueden considerar para establecer una clara diferencia entre un ciberataque y un ataque de APT se pueden considerar el tipo de atacante, el objetivo, el propósito y el ciclo de vida. En la Tabla 2.1 se muestran las diferencias entre un ataque de amenaza persistente avanzada y un ataque cibernético clásico.

2.2. Proceso de un ataque de APT

En los ataques de APT, se denomina campaña a una serie de operaciones emprendidas en las que un intruso, o un grupo de intrusos, establecen una presencia ilícita, a largo plazo, en una red para lograr un objetivo a gran escala.

Tabla 2.1 Diferencias entre un ataque de amenaza persistente avanzada y un ataque cibernético clásico [17].

Características	**Ataque de APT**	**Ataques cibernéticos clásicos**
Definición	Un ataque de APT, es un ataque sofisticado, dirigido y altamente organizado (por ejemplo, Stuxnet).	El malware es un software malicioso que se utiliza para atacar y desactivar cualquier sistema (por ejemplo, el ransomware).
Atacante	Los actores gubernamentales y los grupos de delincuencia organizada.	Un *cracker* (hacker en actividades ilegales).
Objetivo	Organizaciones diplomáticas, industria de la tecnología de la información y otros sectores.	Cualquier ordenador personal o de negocios.
Propósito	Filtrar datos confidenciales o causar daños a un objetivo específico.	Reconocimiento personal.
Ciclo de vida del ataque	Mantener la persistencia en la medida de lo posible utilizando diferentes formas.	Termina cuando es detectado por las acciones de seguridad (por ejemplo, software de antivirus)

Cada campaña actúa de manera diferente, ya que los ataques se personalizan para la víctima u organización específica.

Generalmente, como en cualquier ataque, el primer paso en un ataque ATP es crear un punto de acceso a la red. A continuación, el malware personalizado crea una red de comunicación para mantener dicho acceso, que permite a los atacantes inyectar código malicioso en múltiples ocasiones. Este malware se mueve lateralmente de manera sigilosa a través del sistema, detectando las vulnerabilidades que puede explotar e infectando a otros ordenadores en la red. Además, hace copias de sí mismo para mantener la persistencia dentro del sistema. El malware puede establecer otras conexiones salientes a medida que se accede al sistema y se obtiene la mayor cantidad de datos posible.

Se denomina ciclo de vida de un ataque, a las etapas funcionales en las que se ejecuta un componente importante del ataque y al tiempo en el que el ataque se está ejecutando.

En el estudio realizado por Mandiant (actualmente, FireEye) se presenta un informe sobre el ataque realizado por el grupo APT1, en el cual se propone una visión general del modelo de ciclo de vida de un ataque de ATP, que consta de ocho etapas detalladas en la Figura 2.3: (1) Reconocimiento inicial, (2) compromiso inicial, (3) establecimiento de un punto de apoyo, (4) escalado de privilegios, (5) reconocimiento interno, (6) desplazamiento lateral, (7) mantenimiento de la presencia y (8) misión completada. De estas ocho etapas, las primeras se consideran comunes a todos los ataques ATP, pero las etapas (3) a (8) pueden no tener lugar en el orden propuesto, una vez iniciado el ataque.

A medida que se tiene conocimiento de nuevas campañas de ataques de APT, se observa que su anatomía es diversa y además cambia según el objetivo para el que ha sido diseñada, utilizando diversos vectores de ataque [58].

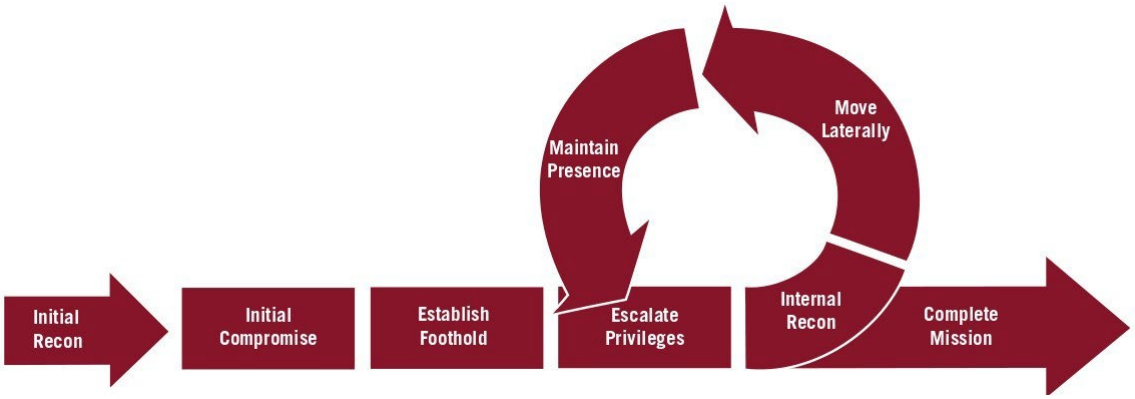


Figura 2.3 Modelo Mandiant del ciclo de vida de un ataque de APT [58].

2.3. Métodos y técnicas

El proceso de un ataque de APT, detallado en la Figura 2.4, se inicia con la realización de un estudio de la víctima, en muchos casos se utiliza *spear-phishing* o correos electrónicos dirigidos, junto con técnicas de ingeniería social, con el objetivo de que la víctima descargue un fichero infectado, para luego vulnerar el ordenador y de esta forma, obtener acceso a otros ordenadores dentro de la organización por medio de la red.

Figura 2.4 Técnicas utilizadas, objetivos e intención de un ataque de APT.

Los métodos que caracterizan a los grupos que realizan ataques de APT son el uso de *exploits* de día cero (vulnerabilidades desconocidas utilizadas para atacar a un sistema informático), vectores de ataque desconocidos (métodos que los atacantes utilizan para comprometer un ordenador), código malicioso en memoria (*fileless*, tipo de malware que se ejecuta en la memoria del ordenador, sin dejar rastro de su actividad) y herramientas no identificadas previamente. Las técnicas comúnmente utilizadas para llevar a cabo un ataque de APT se adaptan o combinan dependiendo del objetivo.

Algunos ejemplos de estas técnicas para realizar ataques de APT son los siguientes:

- Ingeniería social: es el arte de conseguir que un usuario comprometa los sistemas de información. Esta técnica está dirigida a usuarios con acceso privilegiado, que son manipulados para que divulguen información personal para llevar a cabo un ataque malicioso, lo que se consigue a través del control y la persuasión, en lugar de implicar ataques aleatorios a los sistemas de información [52]. El objetivo del atacante es recopilar la mayor cantidad de información posible, que supuestamente es irrelevante, creando con ella una especie de rompecabezas; esto puede proporcionar un conocimiento generalizado de la estructura y el proceso interno de una organización u objetivo. Un ejemplo de esta técnica, se lleva a cabo durante una conversación con la víctima, tratando de obtener información relevante de una manera sutil [61].

- *Spear-phishing*: este procedimiento utiliza un correo electrónico dirigido intencionadamente a una organización, con la finalidad de recoger credenciales de usuario, información financiera u otra información confidencial [4]. Inicialmente, se envía a la víctima un correo personalizado que parece provenir de proveedores o clientes conocidos (ver Figura 2.5). Dicho correo suele contener un enlace o algún fichero adjunto, que puede estar manipulado para que ejecute un *exploit* en el ordenador de la víctima [82].

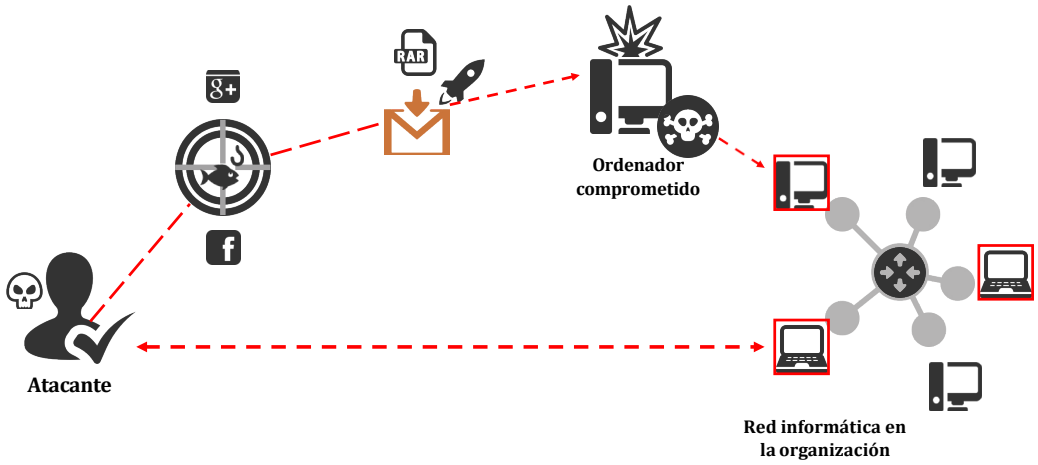


Figura 2.5 Ejemplo de un ataque *spear-phishing*.

- *Watering hole*: es una técnica de ciberespionaje similar a *spear-phishing*. En este caso los ataques se adaptan a las necesidades de las víctimas; es decir, los atacantes intentan obtener información sobre los intereses personales de su víctima [85]. Además, los atacantes pueden identificar sitios web de terceros que son visitados con frecuencia por las víctimas y, una vez identificados, intentan infectar uno o más de estos sitios web con malware. El uso de este tipo de ataque se ha visto en varias campañas de ataques de APT.

- *Drive-by-download*: esta técnica realiza la descarga y ejecución no intencional de software malicioso cuando se visita una página web comprometida [86]. El malware se descarga "sigilosamente" sin el conocimiento de los usuarios, aprovechando las brechas de seguridad, los *exploits* en los navegadores o los *plugins* integrados como ActiveX, Java/JavaScript o Adobe Flash player [67].

- *Rootkit* y otros malware: los rootkits se pueden utilizar para ocultar la presencia de malware o para instalar puertas traseras [57]. Además de los *rootkits*, los ataques de APT utilizan a menudo otros derivados del malware, como los *key-logger* que pueden captar y registrar las entradas (pulsaciones) de teclado, los spywares, utilizados para recopilar información sin conocimiento de la víctima, o el ransomware, que cifra información en los discos duros.

2.4. Problemas de atribución

La tarea de identificar un ciberataque como un ataque de APT es complicado. Además, atribuir dicho ataque a un actor conlleva a una problemática cuando se trata de buscar una correlación en base a un ataque previo ejecutado por algún grupo cibercriminal, estado o nación en particular.

2.4.1. Actores

En un ataque de amenazas persistentes avanzadas se definen los "actores" como los individuos que ejecutan el ataque. Los equipos de vigilancia encargados de la seguridad pueden, en general, observar diferentes evidencias para identificar atacantes, como las direcciones IP, los correos electrónicos o el código malicioso que se ha utilizado. Los atacantes suelen utilizar el concepto de bandera falsa, es decir, se hacen pasar por un tercero para camuflar sus operaciones. En los últimos años, los ataques atribuidos a actores gubernamentales y grupos organizados han aumentado considerablemente.

Los principales actores que intervienen en un ataque de APT se pueden dividir en dos grandes grupos: actores gubernamentales y grupos delictivos organizados. Se incluye a continuación información detallada de ambos grupos.

Los ciberataques llevados a cabo por diferentes gobiernos son cada vez más frecuentes, las sospechas de interferencia en los procesos electorales o de interrupción del suministro de energía, están generando una amplia preocupación pública, debido a las altas capacidades de interferencia cibernética de estos actores. Algunos de los gobiernos que han participado como actores de los ataques de APT son los siguientes:

- China: su principal campo de ciberataque observado es el espionaje industrial, que ha tenido como objetivo el robo de propiedad intelectual. La ciberamenaza más persistente de este actor ha sido el grupo APT1.

- Estados Unidos: este actor podría haber perpetrado los ciberataques más sofisticados. Los ataques han sido perjudiciales y se han utilizado tecnologías avanzadas, lo que significa que han utilizado una considerable cantidad de recursos para el desarrollo de este tipo de ataque. Las campañas de los ataques de APT han servido principalmente para hacer valer intereses geopolíticos. Un ejemplo, fue la operación Stuxnet [28], que apuntaba a los sistemas SCADA para causar un daño sustancial en el programa nuclear de Irán.

- Rusia: este actor es muy activo en cuanto a la actividad de ataques de APT patrocinada por el estado. Además, ha sido objeto de intensas investigaciones debido a su participación en intrusiones consideradas de perfil alto [55]. Recientemente, Microsoft ha detectado ataques de "*spear-phishing*" ejecutados por el grupo APT28, en los que los los empleados del gobierno alemán han sido el objetivo; este grupo ha intentado acceder a las credenciales de los empleados e infectar los sitios con malware [87].

- Irán: este actor de Oriente Medio posee la capacidad de ataque más definida, con varios incidentes y grupos atribuidos al país [55]. Los expertos han monitorizado las operaciones del grupo de atacantes denominado APT33, que ha mejorado

recientemente su infraestructura. Los principales objetivos de este grupo han sido la industria de la aviación y las empresas de energía con conexiones a la producción petroquímica. Las últimas campañas de malware se han dirigido a organizaciones de Oriente Medio, Estados Unidos y Asia [68].

- Corea del Norte: los grupos cibernéticos asociados con este actor han realizado una amplia gama de operaciones, incluyendo ataques destructivos, operaciones de espionaje convencional y piratería bancaria [33]. El ransomware Wannacry es un ejemplo de los ataques perpetrados por este actor. [2].

- Israel: este actor ha sido identificado como un posible co-autor del ataque de Stuxnet [28]. Se conoce públicamente el alto potencial de los servicios de inteligencia de este país, como la Unidad 8200 del ejército israelí [20], el equivalente a los servicios de inteligencia de los Estados Unidos. El ataque del grupo Duqu 2.0 [47] ha sido patrocinado por este actor, y ha infectado numerosos sistemas en varios países en los últimos años. Este malware utilizaba vulnerabilidades de día cero para comprometer los ordenadores, luego los datos robados eran extraídos a los servidores de comando y control (C&C, *Command and Control*). Además, se utilizaron diferentes técnicas para comprometer otros ordenadores dentro de la red.

2.4.2. Campañas

Se denominan campañas a las acciones, métodos y técnicas personalizadas que realiza un actor contra un objetivo para ejecutar un ataque de APT, con la finalidad de extraer datos altamente sensibles. Además de los actores mencionados, existen grupos de ciberdelincuentes organizados con financiación privada y que no responden a los intereses de un gobierno; estos grupos han llevado a cabo diferentes campañas. En los últimos años, se han descubierto nuevas campañas de ataques de APT; estas campañas,

detalladas en la Tabla 2.2, siguen activas en su mayoría y se desconoce el número de objetivos afectados. Estas campañas utilizan diferentes métodos de propagación como los *exploits*, ficheros infectados o malware personalizado, y están diseñadas para el ciberespionaje, siendo sus principales objetivos las organizaciones diplomáticas y la industria de la tecnología de la información.

La investigación de estas campañas fue llevada a cabo por Kaspersky, utilizando una metodología de 15 pasos para identificar un ataque como ataque de APT, en la que se diseccionaron muestras de malware, de tráfico generado y los protocolos de comunicación utilizados por los atacantes en un incidente [48].

Tabla 2.2 Últimas campañas de ataques de APT descubiertas [48].

Descubrimiento	**Primera muestra conocida**	**Nombre**	**Estado**	**Plataforma objetivo**
2019	2019	Topinambour	Activo	Windows
2019	2013	TajMahal	Activo	Windows
2018	2018	ShadowHammer	Inactivo	Windows
2018	2018	FruitArmor	Activo	Windows

2.4.3. Grupos de atacantes

Identificar el origen de un ataque sin un análisis previo resulta difícil, y aún más difícil puede ser el atribuir la operación en curso a un grupo concreto de individuos responsables del ataque. Estos grupos, están patrocinados en su mayoría por los actores, que poseen una infraestructura que permite efectuar el ataque. A continuación, se enumeran los grupos más importantes ordenados por actores y en la Tabla 2.3 se detallan sus características principales.

2.4.3.1. APT1

La organización FireEye, describe en su informe al grupo APT1, como el grupo de ciberespionaje más persistente de los actores en China [58]. El APT1 es un grupo

que tiene vínculos con el ejército chino (Unidad 61398). Este grupo estuvo activo desde aproximadamente 2006 hasta 2010 y ha recopilado datos corporativos sensibles y propiedad intelectual de al menos 141 organizaciones norteamericanas [87]. La infraestructura del grupo APT1 implica una gran organización con al menos cientos de operadores humanos. Además de los servidores de C&C, sus ataques se extendieron a más de 900 servidores y 849 direcciones IP distintas, ubicados en 13 países distintos.

2.4.3.2. APT40

También conocido como "Leviatán", APT40 es un grupo de espionaje cibernético de China, activo desde al menos durante el 2014 [31, 87]. Este grupo ha apuntado específicamente a sectores como la ingeniería, el transporte y la industria de defensa, especialmente donde estos sectores se superponen con las tecnologías marítimas. El grupo APT40 tiene como objetivo las tecnologías críticas y objetivos de inteligencia tradicionales.

2.4.3.3. APT29

APT29, también conocido como "TheDukes", es un grupo de ciberespionaje provisto de importantes recursos, altamente dedicado y organizado. Tiene como objetivo buscar y recopilar información de inteligencia en apoyo de la adopción de decisiones en materia de política exterior y de seguridad. Se cree que ha estado trabajando para la Federación Rusa al menos desde 2008. Este grupo oculta su actividad en la red del objetivo, realizando pocas comunicaciones y simulando el tráfico legítimo de la red; utiliza múltiples servicios web populares y manipula certificados SSL (*Secure Sockets Layer*), lo que dificulta su detección. Se caracteriza por ser uno de los grupos con más capacidad y que más ha evolucionado, por la cantidad de herramientas disponibles para desplegar el ataque [53, 87].

2.4.3.4. APT28

APT28 o "Fancy Bear" es un grupo de ciberespionaje con vínculos con el gobierno ruso. Su forma de trabajo demuestra que están directamente financiados por una organización bien establecida. Tienen como objetivo recopilar información de inteligencia sobre temas de defensa y geopolíticos de múltiples organizaciones gubernamentales, militares y de seguridad. Se cree que está funcionando desde enero de 2007, y cuenta con un numeroso grupo de desarrolladores y operadores para llevar a cabo sus operaciones. Una de sus operaciones más recientes, fue la campaña de injerencia contra Hillary Clinton, el Comité Nacional Demócrata y el Comité de Campaña del Congreso Demócrata en 2016, para afectar las elecciones presidenciales de los Estados Unidos de ese mismo año [31, 87].

2.4.3.5. APT38

APT38, también conocido como "Lazarus", es un grupo vinculado al gobierno de Corea del Norte con una motivación económica. Ha realizado operaciones en más de 16 países demostrando su capacidad de despliegue para lograr sus objetivos. Se caracteriza por ser un grupo cuidadoso en mantener la persistencia el tiempo que sea necesario dentro de la red a atacar. Además de extraer datos sensibles de sus objetivos, también se ha comprobado que irrumpen de forma agresiva con el fin de inhabilitar los servicios de algunas organizaciones. Con un vasto arsenal de herramientas y capacidades, han llevado a cabo operaciones como ataques coordinados contra una serie de emisoras e instituciones financieras surcoreanas utilizando el malware, *DarkSeoul*; además, se le atribuye una importante intrusión y filtración de datos en las redes de Sony Pictures y también, el brote del ransomware Wannacry [31, 53, 87].

2.4.3.6. APT37

El grupo APT37 o "Reaper" es un grupo también vinculado al gobierno de Corea del Norte, activo desde 2012, que ha atacado a sectores públicos y privados principalmente en Corea del Sur. Diferentes informes confirman que dicho grupo esta expandiendo sus operaciones más allá de la península de Corea, utilizando un conjunto de herramientas que incluyen vulnerabilidades de día cero y malware para el borrado de huellas [31].

2.4.3.7. APT39

El APT39, también conocido como "Chafer", es un grupo atribuido a Irán. Tiene como objetivo las industrias de telecomunicaciones, empresas de tecnología informática (TI) e industrias de alta tecnología, con el fin de obtener datos con fines comerciales, además de realizar seguimientos, vigilancia y supervisión de objetivos específicos. Dentro de sus operaciones, se encuentra el de reunir datos geopolíticos que beneficien a la toma de decisiones del país. El alcance de este grupo es global, sus actividades se concentran en Oriente Medio y utiliza principalmente variantes de puertas traseras para cumplir con sus objetivos [53].

2.4.3.8. APT34

El APT34, también conocido como "OilRig", es un grupo de ciberespionaje involucrado en operaciones centradas en beneficiar los intereses de el gobierno iraní y está activo al menos desde el 2012 [87]. Tiene como objetivos diferentes sectores industriales, incluyendo gubernamentales, energéticos, de telecomunicaciones y químicos, centrando sus operaciones en Oriente Medio. Varios informes afirman que según la infraestructura que utiliza este grupo, cuenta con un fuerte vínculo con la nación iraní. Dentro de las operaciones realizadas por este grupo, se han documentado los ataques Shamoon entre

2012 y 2020, en los que se utilizó malware de tipo destructivo (*wipers*) dejando los ordenadores sin funcionar [53].

2.4.3.9. APT32

APT32, también conocido como "OceanLotus", es un grupo de ciberespionaje con objetivo en intereses privados, en diferentes sectores de consumo, fabricación y hostelería y su objetivo es opacar la ventaja competitiva de la organización objetivo. Además, han llevado a cabo operaciones que apuntan a gobiernos extranjeros, disidentes y periodistas. El conjunto de herramientas utilizado por este grupo se mezcla con herramientas disponibles comercialmente para efectuar operaciones dirigidas que cumplen con los intereses del estado vietnamita [31].

2.4.3.10. Shadow Brokers

TSB (en ingles, *The Shadow Brokers*) es un grupo de hackers conocido desde el 2016. Fueron los causantes de la brecha de seguridad a "Equation Group", grupo vinculado a la Agencia de Seguridad Nacional de los Estados Unidos, en la que filtraron varias herramientas de hacking, incluyendo vulnerabilidades de día cero [87]. Esto afecto a varias firmas de seguridad en diferentes productos como cortafuegos, antivirus y productos de Microsoft. Entre las más notables vulnerabilidades filtradas se encuentra "ETERNALBLUE", un *exploit* que se utilizó para la creación de los ransomware WannaCry y NotPetya.

2.4.3.11. The Lamberts

"The Lamberts", también conocido como "Longhorn", es un actor vinculado a la CIA del gobierno de los Estados Unidos, que ha estado activo al menos desde el 2011. Su objetivo han sido organizaciones en diferentes ámbitos como el académico,

las telecomunicaciones o el sector energético, en más de 15 países del Oriente Medio, Europa, Asia y África; específicamente en operaciones de ciberespionaje. Todas las organizaciones atacadas serían de interés para un atacante de cualquier nación. Dentro del conjunto de herramientas se encuentran una serie de troyanos de puertas traseras y vulnerabilidades de día cero para comprometer sus objetivos. Algunas de estas herramientas utilizadas por este actor, fueron publicadas por Wikileaks bajo el nombre de "Vault 7".

Tabla 2.3 Grupos APT

Actor	Grupo APT	Otros nombres	Descubierto desde	Motivación	Objetivo (sectores)	Objetivo (regiones)	Vectores de ataque	Malware asociado	Vulnerabilidades explotadas	Última operación reportada
China	APT1	Comment Panda; TG-8223; BrownFox	2006	Robo de información y espionaje	Aeroespacial; telecomunicaciones; gobierno; investigación	Asia oriental; América del Norte	Correo dirigido; acceso no autorizado; robo de datos	Herramientas de robo de contraseñas; múltiples C&C; troyanos; puertas traseras	No especificado	"Oceansalt"(mayo 2018)
	APT40	Leviathan; Temp.Periscope; Bronze Mohawk	2013	Robo de información y espionaje	Capacidades navales; ingeniería; gobierno; investigación	Europa occidental; América del Norte; Asia sudoriental	Correo dirigido; señuelos; correo basura	Troyanos; puertas traseras; malware personalizado; herramientas de robo de contraseñas	CVE-2012-0158; CVE-2017-0199; CVE-2017-8759; CVE-2017-11882	Injerencia electoral en Camboya (julio 2018)
Rusia	APT29	Cozy Bear; The Dukes; CloudLook	2008	Robo de información y espionaje	Energía; farmacéutico; gobierno; transporte	Europa occidental; América del Norte	Ataques de watering hole; correo dirigido	Troyanos; puertas traseras; malware personalizado	CVE-2009-3129; CVE-2013-2729; CVE-2015-1641; CVE-2016-7855	Campaña de *phishing* en los EE.UU. (noviembre 2019)
	APT28	Fancy Bear; Sofacy; Sednit	2007	Robo de información y espionaje	Química; ingeniería; industrial; organizaciones de inteligencia	Europa occidental; América del Norte; América del Sur	Correo dirigido; robo de datos; señuelos	Puertas traseras; malware personalizado; troyanos	CVE-2014-4076; CVE-2015-1701; CVE-2017-0262; CVE-2017-0263	Campana dirigida a miembros de la OTAN (feb 2019)
Irán	APT39	Chafer; Remix Kitten; Cobalt Hickman	2015	Robo de información y espionaje	Aerolíneas; gobierno; telecomunicaciones; logística	Oriente Medio	Correo dirigido; enmascaramiento del dominio	Puertas traseras; malware de limpieza; troyanos	No especificado	Espiar a las entidades diplomáticas extranjeras con sede en Irán (otoño 2018)
	APT34	OilRig; Helix Kitten; Crambus	2014	Robo de información y espionaje	Química; energía; financiero; gobierno	Oriente Medio; Europa; América del Norte	Correo dirigido; credenciales falsas	puertas traseras; malware de limpieza; troyanos	CVE-2017-0213; CVE-2017-11774; CVE-2017-11882; CVE-2018-20250	Shamoon v3 (diciembre 2018)
Corea del Norte	APT 38	Bluenoroff; Lazarus; Hidden Cobra	2014	Robo de información y espionaje, interrupción, sabotaje y ganancias financieras	Financiero; gobierno; tecnología; Intercambios de BitCoin	En todo el mundo	Correo dirigido; ransomware; credenciales falsas	Malware de limpieza; troyanos; puertas traseras	CVE-2017-0144	Wannacry (mayo 2017); Ataque de SWIFT al Cosmos Bank en la India (agosto 2018)
	APT37	Reaper; Group 123; Ricochet Chollima	2012	Robo de información y espionaje	Aeroespacial; financiero; gobierno; salud	Asia sudoriental; Europa del norte	Ingeniería social; correo dirigido; distribución de malware	DDoS botnets; troyanos; malware de limpieza	CVE-2016-4117; CVE-2017-0199; CVE-2018-0802	Black Banner(abril 2019)
EEUU	Shadow Brokers	-	2016	Ganancia financiera	-	-	-	Malware personalizado	Publicados: CVE-2017-0143	Lanzamiento malware UNITEDRAKE
	Longhorn	Lamberts; APT-C-39	2011	Robo de información y espionaje	Financiero; gobierno; tecnología; educación	Oriente Medio; Europa; Asia; África	-	Malware personalizado	CVE-2014-4148	-
Vietnam	APT32	OceanLotus; SeaLotus; APT-C-00	2013	Robo de información y espionaje	Gobierno; hospitales; fabricación; periodistas	Asia sudoriental; América del Norte	Ingeniería social; payload malicioso remoto; correo dirigido	Puertas traseras; malware personalizado; troyanos	CVE-2016-7255; CVE-2017-11882	Ataques a la Península de Indochina (mayo 2019)

Capítulo 3

Aprendizaje automático

El aprendizaje automático (ML, del inglés *Machine Learning*), es un subcampo de la inteligencia artificial, que da lugar al proceso computacional de inferir y generalizar automáticamente un modelo de aprendizaje a partir de datos de una muestra. El ML estudia algoritmos y técnicas para automatizar soluciones de problemas complejos que son difíciles de programar con métodos de programación convencionales. Los modelos de aprendizaje automático utilizan funciones y técnicas matemáticas y estadísticas para describir las dependencias de los datos, las causalidades y las correlaciones entre los datos de entrada y los de salida.

Teóricamente, dados un conjunto de datos observados, D, un conjunto de parámetros, θ, y un modelo de aprendizaje, $f(\theta)$, se utilizan como método de aprendizaje automático para minimizar los errores de aprendizaje, $E(f(\theta), D)$, entre el modelo de aprendizaje y los datos observados (por ejemplo, evidencia empírica). Consecuentemente, se obtienen los errores de aprendizaje utilizando la diferencia entre el resultado previsto por $f(\theta)$ y los datos de la muestra observada, donde θ representa el conjunto de parámetros aproximados que se derivan de los procedimientos de optimización empleados para la minimización de la función objetivo de los errores de aprendizaje. Así pues, la diferencia

entre los distintos métodos de aprendizaje automático se debe a la selección del modelo de aprendizaje, los parámetros y la expresión del error de aprendizaje [26].

Formalmente, el aprendizaje automático se puede definir de la siguiente manera: se dice que un programa informático aprende de la experiencia, R, con respecto a alguna clase de tareas, T, con la medida de desempeño, P, si su desempeño en T, según lo medido por P, mejora con R [60].

Las técnicas de aprendizaje automático son bastante genéricas y se pueden aplicar en diferentes entornos. Para utilizar estos tipos de algoritmos es necesario traducir el problema al dominio del aprendizaje automático, que normalmente espera un conjunto de características y un criterio de salida o agrupamiento deseable.

Entre los paradigmas de aprendizaje cabe citar los algoritmos supervisados, que utilizan exclusivamente información externa para inducir o entrenar sus hipótesis, y por otra parte, los métodos de aprendizaje no supervisados, que se guían exclusivamente por la estructura intrínseca de los datos a lo largo del proceso de aprendizaje; o lo que es lo mismo, sin ningún tipo de conocimiento externo. Entre estos dos modelos de aprendizaje se encuentra el aprendizaje semi-supervisado, que emplea tanto los datos etiquetados (son aquellos datos que presentan una etiqueta de acuerdo a sus características) como los no etiquetados (datos que no presentan una etiqueta de acuerdo a sus características) en el proceso de aprendizaje.

3.1. Aplicaciones del aprendizaje automático

El aprendizaje automático y sus diferentes algoritmos detectan patrones naturales en los datos, que generan información aportando la toma de decisiones, y además predicen qué situaciones podrían validarse o no.

La aplicación de las técnicas y algoritmos de aprendizaje automático en grandes cantidades de datos se denomina minería de datos (en inglés, *data mining*). El procesa-

miento de un volumen de datos grande permite construir un modelo para aprender, para que tome decisiones y genere al final resultados fiables. Con el aumento incesante de datos, el aprendizaje automático se ha convertido en un factor importante para resolver problemas en áreas tan diversas como la informática, la ciberseguridad, los negocios, la publicidad o la medicina.

Entre las múltiples utilidades del ML está el sirvir para resolver problemas cotidianos y apoyar a los responsables en la toma de decisiones. Algunos problemas que el ML puede resolver son los siguientes: reconocimiento facial, detección de noticias falsas, análisis de sentimientos, sistemas de recomendación, sistemas de detección de fraudes, traducción de idiomas o *chatbots* (bot conversacional).

3.2. Técnicas y algoritmos de aprendizaje automático

Antes de describir los modelos de aprendizaje automático es necesario introducir el concepto de datos etiquetados y datos no etiquetados. Así, cuando se conoce la respuesta correcta a una pregunta relacionada con los datos analizados, se dice que se obtienen datos etiquetados; sin embargo, cuando se desconoce cuál es la respuesta correcta, se dice que los datos son datos no etiquetados.

La elección del algoritmo adecuado no resulta tarea fácil. Existen decenas de algoritmos de aprendizaje automático supervisados y no supervisados, cada uno de ellos adopta un enfoque diferente del aprendizaje. No existe un método perfecto para todos los casos. Encontrar el algoritmo correcto conlleva ensayos de prueba y error, incluso los analistas de datos, denominados científicos de datos (del inglés, *data scientist*), altamente experimentados, no pueden saber si un algoritmo funcionará sin probarlo. Sin embargo, la elección de algoritmos depende del tamaño y tipo de datos con los que

se está trabajando, de las predicciones que se quieran obtener de los datos y cómo se usarán.

Los algoritmos de ML derivan su poder de la capacidad de aprender de los datos disponibles. Tal como se ha comentado anteriormente, los principales modelos de ML pueden clasificarse en modelos de aprendizaje supervisado y modelos de aprendizaje no supervisado. La Figura 3.1 incluye el esquema de estos modelos que se detallan a continuación.

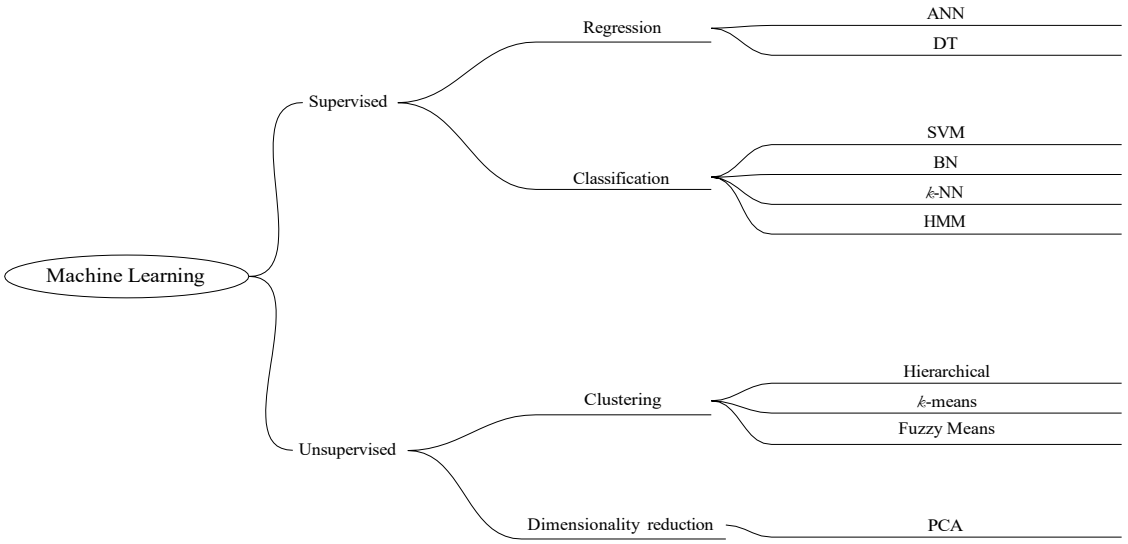


Figura 3.1 Algoritmos de aprendizaje automático.

3.2.1. Aprendizaje supervisado

El objetivo del aprendizaje automático supervisado es construir un modelo que cree predicciones basadas en la evidencia, en presencia de incertidumbre. Estos algoritmos utilizan como entrada un conjunto de datos conocidos y devuelven como salida las respuestas conocidas a los datos, posteriormente entrenan el modelo para, finalmente, generar predicciones analíticas como respuesta a nuevos datos. Un algoritmo de este tipo es el que se utiliza en la previsión meteorológica.

El aprendizaje supervisado utiliza técnicas de clasificación y regresión para desarrollar modelos de predicción. Los métodos más populares de aprendizaje automático supervisado son las redes neuronales artificiales (ANN, del inglés *artificial neural network*), las máquinas de soporte vectorial (SVM, *support vector machine*), el árbol de decisión (DT, *desicion Tree*), las redes bayesianas (BN, *bayesian network*), el vecino más cercano a *k* (*k*-NN) y el modelo oculto de Markov (HMM, *hidden markov model*) [26]. A continuación se detalla cada uno de estos algoritmos mencionados:

- Las redes neuronales artificiales son modelos computacionales inspirados en las interconexiones de las neuronas del cerebro humano, denominadas sinapsis artificiales, de neuronas artificiales (denominadas nodos de la red) capaces de realizar cálculos específicos con sus entradas [49]. Las neuronas biológicas (Figura 3.2), son las células que forman la corteza cerebral en los seres vivos y cuentan con tres partes principales: las dendritas, el cuerpo de la neurona o soma y el axón. Las dendritas son partes de las neuronas que forman una estructura de filamentos muy finos que las rodean y se encargan de cambiar el estado de la neurona dependiendo del tipo de impulso que reciben. El axón es un tubo largo y delgado que típicamente conduce impulsos eléctricos. Cuando una neurona recibe un impulso, dependiendo de la fuerza del estímulo, provocará un cambio en el estado de la neurona y esto a su vez, podrá ser propagado a otras neuronas.

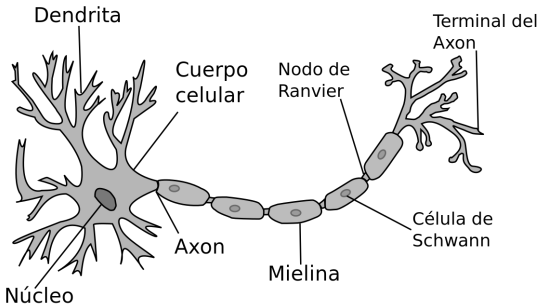


Figura 3.2 Esquema de una neurona biológica.

Una neurona artificial se compone de tres o más capas: una capa de entrada, una o más capas ocultas y una capa de salida. Una ANN es capaz de crear modelos no lineales para obtener las relaciones entre los atributos de entrada y la clasificación de los datos etiquetados [51]. Las principales características de la ANN son la adaptación a partir de la experiencia, la capacidad de aprendizaje, la capacidad de generalización, la organización de los datos, la tolerancia a los fallos, el almacenamiento distribuido y la facilidad de la creación de prototipos [23]. Los algoritmos basados en ANN son útiles para el reconocimiento del habla y de patrones [24], el pronóstico del clima [90] y el diagnóstico de enfermedades [27]; y además, resuelven también problemas de clasificación y regresión.

- La máquina de soporte vectorial es uno de los métodos más precisos y robustos de los algoritmos de ML, que funciona como clasificador, identificando un hiperplano, es decir, entre dos clases de datos etiquetados en un conjunto de datos de entrenamiento. Además, está diseñado para la clasificación binaria, que, en este caso, consiste en separar un conjunto de vectores de entrenamiento que pertenecen a dos clases diferentes. El clasificador de SVM utiliza varios métodos, como el de no linealidad y uso de núcleos, el de separabilidad o el de márgenes o minimización de riesgos.

  La no linealidad y el uso de núcleos son algunos de los descubrimientos pioneros en el campo del ML. Este método permite que un problema no lineal pueda transformarse en un problema lineal. Se pueden realizar varios tipos de hiperplanos de separación utilizando un núcleo (*kernel*), como el polinomial, el lineal, el sigmoide y la función de base radial (RBF, *radial basis functions*).

  La minimización de riesgos puede aplicarse a los casos que no encajan en la arquitectura tradicional de SVM, como los problemas de datos ausentes o datos no etiquetados [18, 45, 59].

Generalmente, las SVM son capaces de generar resultados con buena precisión, especialmente en conjuntos de datos denominados "limpios", es decir, datos que han recibido un preprocesamiento para eliminar inconsistencias en los propios datos, valores nulos o valores duplicados. Además, es un buen método para trabajar con conjuntos de datos de alta dimensión, cuando el número de dimensiones es mayor que el número de las muestras. Sin embargo, para conjuntos de datos grandes, con mucho ruido o de clases superpuestas, puede ser más eficaz. Además, con conjuntos de datos más grandes, el tiempo de aprendizaje puede ser alto. [43].

- Los modelos de árbol de decisión son modelos precisos, estables y fáciles de interpretar. Su construcción se basa en reglas de decisión que se representan en forma de árbol. Estos modelos pueden dar lugar a relaciones no lineales para la resolución de problemas. Los árboles de decisión y los bosques aleatorios (RF, *random forest*) son los modelos más notables porque son más precisos y elaborados. Su capacidad de predicción es mayor que en otros modelos debido a estas características, pero en cambio, su rendimiento es bajo. Los algoritmos más utilizados para construir árboles de decisión son los CART (árboles de clasificación y regresión), los ID3 (dicotomizador iterativo) y los denominados CHAID (detectores de interacción automática Chi-cuadrado) [5, 18, 45].

  El algoritmo de árbol de decisión clasifica los datos para lograr el propósito de la detección. El árbol se genera a partir de los datos del conjunto de aprendizaje, de tal forma que si no puede ofrecer una clasificación correcta de todos los objetos, entonces se seleccionan algunas excepciones y se añaden al conjunto de datos aprendizaje. Esto se repite hasta que se ha tomado un conjunto de decisiones correcto.

- Las redes bayesianas son modelos gráficos probabilísticos que se utilizan para describir y analizar distribuciones multivariantes. Las variables pueden ser continuas

o discretas; sin embargo, cuando todas las variables son discretas, la anotación se representa como una serie de sumas y productos. En la representación gráfica de una BN, los nodos representan una variable o estado observable, y los bordes simbolizan las dependencias condicionales entre los nodos. La BN se ha utilizado en diferentes áreas, por ejemplo, en el sistema Microsoft Windows, en el control de misiones de la NASA y en aplicaciones de bioinformática.

- El modelo del vecino más cercano a *k* (donde *k* representa el número de elementos mas cercano) se puede utilizar tanto para problemas de regresión como de clasificación. Debido a su simplicidad, eficacia e intuición del concepto, este modelo puede utilizarse para identificar los vecinos más cercanos para un conjunto de datos dados, basado en una medida de distancia [37, 70]. La suposición es que los elementos similares están más cerca. La idea de cercanía es una medida de distancia, que puede ser una simple distancia euclídea entre dos puntos. En este caso, la decisión de clasificación puede estar influenciada por la sensibilidad de *k*, especialmente en pequeños conjuntos de datos con valores atípicos. Existen numerosas familias de medidas de distancia, y se pueden destacar las siguientes: Minkowski, Producto Interior, Acorde Cuadrado, Entropía de Shannon y Vicisitude [1].

- El modelo oculto de Markov es un modelo probabilístico estocástico de eventos discretos que incluye una variación de la cadena de Markov, es decir, de la cadena de estados o eventos vinculados, donde el siguiente estado depende solo del estado actual del sistema. El HMM se utiliza para analizar características u observaciones con el objetivo de predecir secuencias de estados más probables, donde los estados ocultos representan un atributo no observado del proceso. El HMM se ha utilizado para resolver problemas de análisis financiero, secuenciación genética, procesamiento de imágenes y procesamiento de lenguaje natural [8, 45].

3.2.2. Aprendizaje no supervisado

El aprendizaje no supervisado no dispone de un conjunto de datos de entrenamiento. Se presentan algunos datos sin etiquetar, y el propio modelo debe aprender de ellos, y entonces predecir los futuros resultados [71]. Este tipo de modelo de aprendizaje es el más apropiado cuando el problema requiere una gran cantidad de datos sin etiquetar. El aprendizaje no supervisado tiene como objetivo encontrar patrones ocultos o estructuras específicas en los datos. Se utiliza para extraer inferencias de conjuntos de datos que consisten en datos de entrada sin etiquetar respuestas.

Este modelo de aprendizaje utiliza la reducción de la dimensionalidad mediante, por ejemplo, el análisis de componentes principales (PCA, *Principal component analysis*) y técnicas de agrupación (como *k*-means, Fuzzy *c*-means y jerárquicas) para desarrollar modelos predictivos. Un ejemplo de la aplicación del modelo de ML no supervisado es la detección y clasificación de correos *spam*. Se incluye a continuación una descripción de estos algoritmos:

- El análisis de componentes principales es un procedimiento de reducción de dimensión. Este método estadístico es útil cuando hay un gran número de variables, donde cada variable tiene más o menos importancia. Este procedimiento se utiliza para asignar un conjunto de variables interrelacionadas a un conjunto más pequeño de variables no correlacionadas linealmente, al tiempo que representa la mayor varianza posible en el conjunto de datos original [65]. Algunos ejemplos de aplicaciones de este método se encuentran en la extracción de características [46], las ciencias sociales, la medicina y el estudio del genoma humano [92].

- El algoritmo *k*-means es un algoritmo de agrupación; esta técnica consiste en separar *n* objetos de datos en *k* clústeres, donde cada objeto de datos es agrupado en un clúster de acuerdo a la distancia mas cercana. El usuario define previamente el número de clústeres. Generalmente, este algoritmo utiliza la distancia euclídea para

calcular la distancia entre dos puntos. Este algoritmo tiene diversas aplicaciones, una de ellas es que puede ser utilizado en sistemas de detección de intrusos (IDS), para generar firmas de bases de datos con alta calidad y mejorar así la detección de intrusos [51].

El objetivo del algoritmo *k*-means [39], es dividir *M* puntos en *N* dimensiones en *k* clústeres, para que la suma de cuadrados dentro del clúster se minimice. Donde, la suma de los cuadrados del grupo (*WCSS*, within-cluster sums of squares), se encuentra la suma de la distancia de cada observación en un grupo a su centroide [73]. Se busca clasificar un conjunto de observaciones $x = \{x_1, x_2, ..., x_n\}$ y el conjunto de clústeres $C = \{C_1, C_2, ..., C_k\}$, utilizando la distancia

$$\sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2, \tag{3.1}$$

donde μ_i es la media en el clúster C_i.

- El algoritmo de agrupamiento difuso (*Fuzzy C*-means) es un algoritmo que selecciona aleatoriamente el número de clústeres; luego, a cada objeto se le asigna pertenencia a un clúster. Este proceso se revisa continuamente para minimizar la distancia y el grado de pertenencia al clúster [96].
- La agrupación jerárquica se utiliza para dividir los objetos en grupos similares. Este método puede clasificarse en dos categorías: agrupación divisiva y aglomerativa. En la agrupación divisiva, los objetos (punto de datos) se consideran como un clústeres y luego se dividen en clústeres más pequeños. En la agrupación aglomerativa, cada objeto se considera como un elemento individual, que se agrupará añadiéndose a un clúster [3].

Capítulo 4

Detección de APT

En los últimos años, el volumen de datos generados por los sistemas de información ha aumentado considerablemente, lo que ha llevado a que el malware y los ataques a las redes informáticas sean más difíciles de detectar. Para detectar un ataque deben analizarse los datos en el menor tiempo posible, lo que ha llevado a proponer varios enfoques para resolver este problema, como la detección basada en el análisis dinámico [83], el entorno de trabajo basado en el análisis del contexto [36], la detección de servidores C&C basada en el acceso independiente [93], la detección de ataques usando la información contextual [7], y la detección a través del seguimiento del flujo de información [12]. Además, se ha comenzado a utilizar técnicas de aprendizaje automático para mejorar la tasa de verdaderos positivos en la detección de ataques APT [72]; consecuentemente, se detallan algunos de estos enfoques propuestos.

4.1. Aprendizaje automático aplicado a la ciberseguridad para la detección de APT

Cada vez son más frecuentes los ataques masivos y dirigidos, que pueden causar daños a los usuarios o a las organizaciones, tales como la pérdida de información crítica. Se han desarrollado diferentes enfoques para prevenir o minimizar el riesgo de ataques de APT, algunos de los cuales están basados en el aprendizaje automático.

Cualquier medida de prevención requiere una gran capacidad de análisis y respuesta en el menor tiempo posible, debido a la rápida evolución de las amenazas actuales, resultando las técnicas de aprendizaje automático un instrumento útil en el ámbito de la ciberseguridad. Por esta razón, se han creado instrumentos automatizados para tratar de garantizar la ciberseguridad de un sistema. Por ejemplo, se pueden crear modelos de comportamiento del tráfico de la red para detectar actividades anómalas, reducir el número de falsos positivos en las alarmas y detectar las amenazas en tiempo real [38]. Sin embargo, el aprendizaje automático también puede utilizarse para ataques como el envío de correos electrónicos fraudulentos o el descifrado de contraseñas. Las aplicaciones del aprendizaje automático en la ciberseguridad pueden clasificarse de la siguiente manera:

- Detección: son las herramientas que permiten detectar comportamientos anómalos para generar alertas en tiempo real, y facilitar la toma de decisiones.
- Protección: son formas de detectar vulnerabilidades para instalar correcciones de seguridad automáticamente.
- Predicción: son diferentes técnicas y algoritmos para predecir ataques y desarrollar técnicas anti-malware.
- Terminación: es la eliminación automática de la amenaza.

Algunos mecanismos que se han considerado para detectar los ataques de APT son los siguientes: (1) la observación de patrones de alerta inusuales para detectar malware con reconocimiento de carga maliciosa, componentes conocidos, y actividades de control remoto; (2) la vigilancia del tráfico saliente sospechoso en la red, que puede mostrar parámetros significativos como ordenadores infectados, servidores C&C, y filtración de datos; (3) la supervisión del tráfico interno sospechoso en la red, que podría revelar una escalada de privilegios, movimientos laterales y propagación de malware.

Se incluyen a continuación algunas de las aplicaciones de ciberseguridad que utilizan técnicas de aprendizaje automático:

- Detección de *spam* y *phishing*: el *spam* es un correo electrónico que no ha sido solicitado; normalmente, proviene de remitentes desconocidos y su finalidad es publicitaria o comercial, por lo que es esencial distinguirlo de los correos electrónicos legítimos. El *phishing* es uno de los vectores de ataque más utilizados, donde se establece un punto de entrada entre el atacante y la red de una empresa. La ingeniería social se utiliza para engañar a la víctima para que visite un sitio fraudulento con el fin de robar sus credenciales. La detección del *phishing* es cada vez más difícil debido a las avanzadas estrategias de evasión que utilizan los atacantes, como las técnicas de redirecciones abiertas de URL para evitar los filtros de *spam* [66, 69]. Se han propuesto diferentes técnicas de clasificación ML que pueden ayudar a detectar correo *spam*; para ello, es necesaria la clasificación de correos auténticos y fraudulentos, de acuerdo a criterios como el asunto del mensaje, el remitente y enlaces maliciosos en el cuerpo del mensaje; esto permite que el algoritmo aprenda a clasificar los correos electrónicos utilizando el conjunto de datos de entrenamiento. Los autores en [11] propusieron una técnica de puntuación para detectar correos electrónicos dirigidos auténticos usando

una combinación de varias características; además, se ha creado un sistema de detección práctico, desplegable y en tiempo real para estos ataques.

- Detección de malware: el malware moderno es capaz de crear archivos ejecutables que pueden causar daños a los sistemas de una red o robar información sin el permiso de los usuarios. Por lo general, el malware utiliza comunicaciones con uno o varios servidores C&C a través de direcciones IP o URL generadas aleatoriamente. Por esta razón, la creación de *blacklists* es un método ineficiente. Se han utilizado algoritmos de aprendizaje automático para detectar direcciones IP maliciosas; algunos estudios propuestos para la detección de malware con técnicas de ML se discuten en [59]. Los autores en [54] presentaron una propuesta para detectar servidores C&C utilizados en los ataques de APT, que consiste en la observación de patrones de comunicación específicos dentro de la navegación web, con el fin de identificar y detectar el malware utilizado. Otro enfoque para la detección de malware se ha detallado en [98], el objetivo de este trabajo es la detección de malware basada en el análisis del tráfico DNS y el tráfico malicioso mediante monitorización de los paquetes en el punto de salida de la red.

- Detección de intrusos: este método permite vigilar el tráfico de la red para analizar los flujos de datos y buscar comportamientos inusuales. Se utilizan para ello, sistemas de detección de intrusos y sistemas de prevención de intrusos (IPS). La detección de intrusos puede dividirse en detección de uso indebido y detección de anomalías. Esta última utiliza técnicas de modelización de la red e identificación del comportamiento anormal del flujo de datos en la red. La detección de uso indebido, en cambio, utiliza técnicas basadas en firmas sobre los ataques conocidos para detectar posibles ataques. En el trabajo [15], los autores revisan las técnicas de aprendizaje automático usadas para estos métodos de detección. Los autores en [9] proponen la detección de movimiento lateral basado

en anomalías en sesiones maliciosas de protocolo de escritorio remoto (RDP) en los sistemas operativos de Windows; aprovechando los registros de eventos del sistema, se evaluaron varias técnicas de aprendizaje automático supervisado para clasificar las sesiones RDP y detectar el inicio de sesiones maliciosas.

4.2. Detección de APT utilizando aprendizaje automático

En la actualidad, existen diferentes enfoques teóricos que buscan resolver los problemas de la detección de ataques de APT; estos enfoques tienen en cuenta el ciclo de vida de un ataque de APT y utilizan el aprendizaje automático como principal herramienta de detección.

Las tácticas, técnicas y procedimientos (TTP) consisten en todas aquellas herramientas y procesos que son utilizados como vector de ataque; un atacante suele utilizar TTP similares en cada uno de los ataques que lleva a cabo, permitiendo trazar una secuencia de comportamiento, es decir, el *modus operandi*.

En estos enfoques se han descrito algunas de las TTP que utilizan los atacantes, como el *spear-phishing* y el *watering-hole*; no obstante, existen otras técnicas que facilitan la persistencia en los sistemas como los troyanos de acceso remoto y el control de sesiones. Es importante señalar que no todos los trabajos han considerado el uso de las TTP. A continuación, se realiza un análisis de las ventajas y desventajas, así como de las TTP utilizadas en los enfoques detallados en la Tabla 4.1.

Tabla 4.1 Comparación de los enfoques de detección de ataques de APT basados en el aprendizaje automático (ML).

Trabajo	Algoritmo	Enfoque	Detalle del enfoque	Ciclo de vida APT utilizado	Precisión en detección
Ghafir et al. [34]	DT, SVM, *k*-NN and Ensamble learning	MLAPT	Fases: Detección de amenazas; correlación de alertas; predicción de ataques	Modelo de 6 etapas	81,8 %
Sharma et al. [75]	Genetic programming, classification and regression tree, dynamic bayesian game model and SVM.	DFA-AD	Fase: Tráfico de red; evento de correlación; servicio de votación	No se especifica	98,5 %
Siddiqui et al. [79]	*k*-NN and correlation fractal dimension.	Fractal-based anomaly.	Pasos: Captura combinada de paquetes (archivos pcap); extracción de vectores de características; eliminación del ruido; clasificación de anomalías con algoritmos ML	No se especifica	93,58 % (FD), 92,83 % (*k*-NN)
Shenwen et al. [77]	*k*-NN	Detection based on Big Data	Fases: Recuperar; reutilizar; revisar; retener	No se especifica	No se especifica
Bai et al. [9]	LR, GNB, DT, RF y LB	RDP-based LM detection	Pasos: Preprocesamiento del conjunto de datos; definición de métricas; aplicación de técnicas de ML; comparación de resultados	Solo 1 etapa	99,99 % (LB)
Chu et al. [18]	PCA, SVM, NB, DT y MLP	Early discovery of APT attack	Pasos: Preprocesamiento del conjunto de datos; reducción de la dimensión; clasificador	No se especifica	97,22 % (SVM)
Zhang et al. [97]	Fuzzy clustering	APT attack scenarios	Pasos: Preprocesamiento de datos; clasificación de eventos de ataque; agrupamiento difuso; minería de escenarios de ataque	IKC modelo de 4 etapas	No se especifica

Un sistema basado en el aprendizaje automático llamado MLAPT fue presentado en [34]. Este modelo fue creado para detectar los ataques de APT a través de alertas tempranas que son analizadas por algoritmos ML. Estas alertas se crearon a partir de un entorno de correlación entre varios módulos de detección. MLAPT se basa en el análisis de un ciclo de vida de un ataque de APT de seis fases.

Por su parte, el entorno de trabajo de MLAPT funciona en tres fases:

- Detección de amenazas: se escanea el tráfico de la red por ocho módulos de detección para encontrar las técnicas utilizadas por los ataques de APT. Esta fase da como resultado una serie de alertas que se denominan eventos.
- Correlación de alertas: los eventos generados por los módulos de detección están correlacionados, y la salida puede ser unas alertas de dos tipos diferentes.
- Predicción de ataque: se utiliza un módulo de predicción basado en el aprendizaje automático para detectar técnicas que se utilizan en los ataques de APT.

En el caso del entorno de trabajo MLAPT, se han seleccionado ocho tipos de ataque que son detectados por diferentes módulos del entorno, para luego, correlacionar las alertas generadas por estos módulos, aplicando técnicas de aprendizaje automático, y determinar si realmente es un ataque de APT.

Los módulos de este enfoque comprenden desde diseccionar ficheros ejecutables o detectar ficheros a través de firmas, hasta la detección de distintos protocolos de red (nombres de dominio, direcciones IP, certificados SSL maliciosos, nombres de domino flux), por medio de alertas de escaneo o conexiones salientes que utilizan la red TOR.

Las ventajas del modelo MLAPT son que se han identificado ocho de los ataques más comunes, que utiliza aprendizaje automático y además, identifica de manera temprana si estos ataques están relacionados con un ataque de APT. Las desventajas de este modelo son que los módulos de detección, en su mayoría, consideran listas negras y

detección de ficheros por firmas (*hash* y *fingerprint*), que pueden ser manipulados por los atacantes. Una desventaja de este modelo, es que se ha limitado el numero de ataques; por lo que un ataque desconocido, podría generar fallos en la correlación del modelo.

Una arquitectura de entorno distribuido para la detección de ataques de APT llamada DFA-AD, se describe en [75]. Este trabajo clasifica los eventos en un entorno de trabajo distribuido y la correlación entre ellos para detectar las técnicas utilizadas en los ataques de APT. La detección de intrusos se realiza en un entorno distribuido en el módulo de plataforma de confianza.

La arquitectura DFA-AD se ha diseñado en las siguientes tres fases:

- Tráfico de la red: el flujo de tráfico se recoge, se procesa y se analiza por algún método de reconocimiento utilizando algoritmos de aprendizaje automático.
- Evento de correlación: a través de reglas específicas dadas por un administrador, los eventos generados en la fase anterior se recogen para ser evaluados.
- Servicio de votación: se analiza la información anterior y se genera una alerta si se detecta un ataque de APT.

En la arquitectura del entorno de trabajo distribuido DFA-AD se han considerado dos vectores de ataque *spear-phishing* y *watering-hole*; la detección de estos vectores y sus posibles interacciones en el tráfico de red dependen de la identificación previa de los diferentes métodos de clasificación de aprendizaje automático, dando como resultado los eventos relacionados generados entre estos vectores de ataque.

Las ventajas del modelo DFA-AD son que utiliza cuatro clasificadores de aprendizaje automático para la detección de anomalías que generan eventos; estos eventos son correlacionados y enviados a un servicio de toma de decisiones para determinar si la alerta corresponde a un ataque de APT. Como desventajas se puede decir que la

detección de anomalías en la red está limitada al tráfico de red que es analizado por los clasificadores. Además, este entorno de trabajo solo considera los vectores de ataque de una parte del ciclo de vida de un ataque de APT, que podría ser la etapa inicial del ataque.

Otro modelo creado para detectar ataques de APT, incluye un mecanismo de clasificación de anomalías basado en fractales [79]. Este método ha utilizado el *k*-NN y la dimensión fractal de correlación (FD) como algoritmos de clasificación de anomalías para probar el conjunto de datos y la comparación de los resultados. En el primer paso, los autores combinaron dos conjuntos de datos con tráfico de red normal y paquetes de tráfico de ataque de APT. A continuación, extrajeron las características del vector mediante el análisis de los datos de sesión del TCP (Protocolo de Control de Transmisión). Posteriormente, se eliminó el ruido del conjunto de datos, y el conjunto de datos resultante se utilizó en el algoritmo de clasificación de anomalías para detectar un ataque. Por último, los autores demostraron que el algoritmo basado en la distancia euclídea es menos eficaz que el basado en la dimensión fractal, obteniendo mejores resultados con este último.

En este enfoque se ha considerado una combinación de tráfico anómalo y tráfico normal de red, para los cuales se extrae la información de la cabecera del protocolo de control de transmisión con el fin de determinar características de un ataque de APT.

Como ventajas de este modelo, se puede decir que se realiza un pre-procesamiento de los datos para eliminar los que sean innecesarios, dentro del conjunto de datos originado de la combinación previa; además se identifican y etiquetan diferentes tipos de ataque, y se utilizan clasificadores de aprendizaje automático para entrenar el modelo. Alguna de las desventajas de este enfoque es que solo se basa en la información que ofrece la combinación del conjunto de datos. Además, no se consideran los vectores de ataque ni se establece un ciclo de vida.

Un sistema de detección de ataque de APT basado en el proceso de arquitectura de datos grandes fue propuesto en [77]. Este modelo utiliza algoritmos de ML como *k*-NN en conjuntos con grandes volúmenes de datos sobre datos de red, registros de sistema e información de seguridad. Este sistema se dividió en cuatro pasos:

- Arquitectura del sistema APT: Se reunió un sistema de datos e información de la red para ser analizado.
- Tecnología de procesamiento de Big data: se utilizó un clúster de Hadoop (permite procesar grandes conjuntos de datos) para mejorar el análisis de ataques APT.
- Tecnología de análisis APT: La detección de ataques maliciosos se detectó a partir de vulnerabilidades y conexiones sospechosas con comportamientos anómalos.
- Algoritmo de detección de ataque de APT: Este método utilizó la herramienta Mahout, capaz de procesar big data y el algoritmo *k*-NN para la detección. Este modelo se dividió en cuatro fases: recuperar, reutilizar, revisar y retener.

La ventaja de este enfoque es que se cuenta con una gran cantidad de datos que permite entrenar el modelo dando buen resultado en el entorno de pruebas para la detección de ataques de APT conocidos y desconocidos. Las desventajas este modelo son que se utiliza un solo algoritmo de aprendizaje automático y no se presenta un ciclo de vida de ataque de APT; tampoco se presenta el conjunto de datos entrenado para recrear el trabajo.

Un enfoque basado en las anomalías para la detección de sesiones RDP maliciosas fue detallado en [9]. Este modelo propone sesiones RDP como un método de intrusión utilizado en la fase de movimiento lateral del ciclo de vida de un ataque de APT. Se utilizaron los registros del ordenador y de la red para identificar eventos anómalos que pudieran coincidir con las huellas digitales de un ataque de APT. Para este propósito, se utilizaron dos conjuntos de datos reales, que se dividieron en cinco tipos diferentes

de registros: autenticación, proceso, flujo, DNS y cambios en el registro del sistema obtenidos de una prueba de *Red Team*. Estos conjuntos de datos se evaluaron con las siguientes técnicas de ML: *logistic regression*, *gaussian-naive bayes*, *desicion tree*, *random forest* y *logitBoost*. Los autores concluyeron que el algoritmo *logitBoost* es el más efectivo para la detección de anomalías en las sesiones de RDP.

Las conexiones a través del protocolo de escritorio remoto son otro tipo de TTP que pueden ser utilizado durante un ataque. Este protocolo es un método utilizado para moverse lateralmente a través de la red y así conseguir con éxito acceso a ordenadores no autorizados.

La ventaja de este modelo basado en RDP es que se enfoca en la fase de movimiento lateral del ciclo de vida de un ataque de APT. Sin embargo, como desventaja, solo se consideran las conexiones remotas del sistema operativo Windows.

Un método para detectar ataques de APT que simula escenarios de ataque sobre los registros de seguridad del IDS ha sido propuesto en [97]. Este método utiliza el modelo de intrusión en cadena (IKC) de cuatro fases: recogida de información, intrusión, expansión latente y robo de información. Las acciones se clasificaron de acuerdo con el propósito de cada una de las fases del modelo IKC. Estos eventos fueron entonces correlacionados con los registros del IDS, usando agrupaciones difusas para formar la cadena de ataque. Finalmente, este modelo crea escenarios que sirven como guía para la detección y defensa de estos ataques dirigidos.

En resumen, este enfoque tiene como objetivo construir escenarios basados en los registros de seguridad de un IDS para detectar los ataques de APT a partir de la correlación de los registros del sistema IDS, utilizando algoritmos de agrupaciones difusas.

Las ventajas de este modelo son que se utiliza un ciclo de vida de un ataque de APT basado en el modelo de cuatro fases IKC; además, se buscan similitudes en los

eventos como el tiempo y las direcciones IP, que permitan facilitar la detección de los ataques de APT, finalmente, los escenarios de los ataques de APT se construyen automáticamente.

La desventaja de este enfoque es que solo se utilizan algoritmos de agrupaciones difusas y no se realizan comparaciones con otros algoritmos para evaluar otros posibles escenarios.

Un sistema de detección de ataque de APT que permite el descubrimiento temprano del ataque ha sido detallado en [18]. Este modelo utiliza cuatro algoritmos de aprendizaje automático (SVM, DT, NB (Naive Bayes) y perceptrón multicapa (MLP)) para la detección temprana de cuatro tipos de anomalías.

Estas anomalías utilizadas como TTP son la denegación de servicio, que consiste en la creación de un gran número de paquetes que hace imposible el correcto funcionamiento de un sistema informático o de la red; *probe* que consiste en recolectar información acerca de una víctima remota; y R2L permite una conexión remota no autorizada donde el atacante puede tomar el control del ordenador remoto y ganar acceso local para moverse lateralmente.

Por último, U2R donde el atacante obtiene acceso no autorizado como usuario local del ordenador con privilegios de administrador. El atacante logra iniciar sesión en un ordenador usando una cuenta sin privilegios de acceso como administrador a través de la explotación de las vulnerabilidades del sistema.

La ventaja de este enfoque es que se hace una comparación entre algoritmos para evaluar los resultados; además, la correlación de variables ha sido analizada con el algoritmo PCA. Sin embargo, la limitación de cuatro tipos de ataque puede reducir la detección de un ataque de APT cuando son utilizados otros vectores de ataque.

En resumen, estos modelos propuestos basados en el aprendizaje automático para la detección de ataques de APT han utilizado en su mayoría el algoritmo *k*-NN y, además,

se han utilizados variantes de los algoritmos SVM, DT y NB, para evaluar la precisión de detección. Cabe resaltar, que estos algoritmos han obtenido buenos resultados, por lo que son opciones validadas para ser tomadas en cuenta en nuestro estudio.

Otra característica destacable de estos modelos es que tres de ellos han considerado la detección de ataques de APT a través del ciclo de vida del ataque, donde uno de ellos se ha enfocado en la fase de movimiento lateral y otros dos modelos han seleccionado las TTP de cada una de las etapas del ciclo de vida que han sido descritas.

Una limitación que tienen estos modelos es que se contemplan pocas o ninguna de las tácticas, técnicas y procedimientos que podrían ser detectadas, y en la mayoría de los casos se utilizan las más comunes, sin embargo, la evolución de los vectores de ataque es constante y se busca siempre mantener la persistencia en los sistemas objetivo, por lo que es posible que los modelos queden obsoletos rápidamente.

La ventaja de las TTP descritas en algunos de estos enfoques es que han permitido identificar patrones de comportamiento anómalo en la red. Por ello, es importante considerar las TTP para la detección de ataques de APT; es importante señalar que estas TTP no son utilizadas todas en el mismo ataque o de la misma manera, ya que pueden ser combinadas con otros tipos de vectores de ataque.

No obstante, el MITRE ha creado una matriz de agrupación para las TTP, basado en un ciclo de vida de un ataque en once etapas; esta lista es extensa y se actualiza conforme se van dando los reportes sobre los ataques de APT.

4.3. Otros enfoques propuestos para detectar APT

Existen otros enfoques que no contemplan un ciclo de vida del ataque o solo tienen en cuenta el uso del aprendizaje automático para la detección de los ataques de APT. A continuación, se describen algunos de estos enfoques propuestos:

SPuNge: Los autores en [10], presentan un prototipo de detección llamado SPuNge, este enfoque trabaja con los datos recopilados de ordenadores que se encuentran en una red. Este prototipo esta dividido en dos fases principales, en la primera fase se analizan las direcciones URL que los usuarios visitan a través del navegador de Internet utilizando el protocolo HTTPS y el protocolo HTTP; esto se lleva cabo porque es posible que algún ordenador se encuentre infectado con algún malware. A continuación, se identifican otros ordenadores que presentan un comportamiento de red similar. Este sistema no funciona en tiempo real, la detección depende de las conexiones de una o varias URLs, sean maliciosas o no. Es decir, se utilizan técnicas de agrupamiento y correlación para procesar la información recopilada para su posterior análisis y determinar finalmente posibles ataques dirigidos.

***Data Leakage Prevention*:** Este enfoque se encuentra basado en la prevención de fuga de datos (por su siglas en inglés, DLP) y se centra en la detección de la última fase de un ataque de APT, que es la extracción de datos. Se utiliza un algoritmo DLP para procesar el tráfico de datos y detectar fugas de datos, generando una huella (*fingerprint*) que coincida con las características de la fuga de datos. El sistema propuesto, utiliza sensores CCI (en inglés *Cyber Counter intelligence*), para ubicar la trayectoria de los datos filtrados. Este enfoque detecta únicamente un paso del ciclo de vida de un ataque de APT, la extracción de datos. Además, la detección en tiempo real se ve restringida a la información que contengan los sensores CCI. De igual forma, no existe garantía de que dichos sensores puedan proporcionar la información necesaria sobre las *fingerprints* de los datos filtrados [80].

TerminAPTor: Este enfoque destaca el uso de vínculos del rastro del ataque dejado por los atacantes en sistemas monitorizados durante las diferentes etapas de una campaña; para la detección de un ataque de APT se propone el seguimiento del flujo

de información (IFT, *Information Flow Tracking*). Se evaluaron solo dos escenarios de ataques de APT y se demostró que esta propuesta de detección, necesita mejorar los resultados de los falsos positivos [13].

Capítulo 5

Análisis del ciclo de vida de un ataque de APT

El análisis del ciclo de vida de un ataque de APT es fundamental para comprender cómo funcionan estos ataques, así como para identificar las técnicas maliciosas más utilizadas. Las campañas de ataques de APT utilizan múltiples herramientas de evasión para no ser detectadas. En los últimos años se han propuesto diferentes ciclos de vida organizados en etapas, que a su vez están compuestas por técnicas, métodos y herramientas que se utilizan para realizar un ataque dirigido. El número de etapas de un ciclo de vida difiere según el enfoque propuesto; así, un ciclo de vida puede organizarse en tres etapas [89], cuatro etapas [97], incluso hasta once etapas [84].

A continuación, se realiza un análisis de los diferentes ciclos de vida, en los que se describen brevemente las acciones que se pueden llevar a cabo en cada etapa. Es importante resaltar que los nombres de las etapas se muestran en el idioma original (en inglés) para evitar posibles confusiones entre los modelos analizados.

5.1. Modelos de ataque de tres etapas

Los autores en [89] han propuesto un ciclo de vida que consta de tres etapas, y que se basa en el resultado de analizar los diferentes métodos y técnicas utilizados en 22 campañas de ataques de APT. Cada una de las etapas contempla al menos tres de las siguientes características o técnicas que se utilizan para llevar a cabo el ataque:

1. *Initial compromise*: en esta etapa, los atacantes intentan acceder al objetivo; las técnicas más utilizadas en esta fase son el *spear-phishing*, el *watering-hole*, los ataques del lado del servidor (explotar las vulnerabilidades de los servidores o robar las credenciales por fuerza bruta), y los medios de almacenamiento infectados (memorias USB y discos ópticos comprometidos).

2. *Lateral Movement*: los atacantes intentan comprometer otros servicios en el sistema o red. El objetivo es robar credenciales legítimas que les permitan persistir en el sistema. Algunas de las técnicas utilizadas son las herramientas propias del sistema operativo Windows, por ejemplo, RDP, PsExec y Powershell; y explotar vulnerabilidades de día cero.

3. *Command and Control*: cuando el sistema ha sido comprometido es necesario establecer una conexión externa para extraer los datos. Los atacantes utilizan servicios como HTTP, HTTPS o FTP; también pueden utilizar herramientas como las de conexión remota, por ejemplo, VNC (*Virtual Network Computing*) o RDP.

5.2. Modelos de ataque de cuatro etapas

Uno es de los modelos de cuatro etapas que identifica los comportamientos y propósitos de un ataque de APT es el IKC (*Intrusion Kill Chain*) [97]. El modelo IKC incluye las siguientes etapas:

1. *Information collection*: en esta fase se realiza un reconocimiento inicial de la red, utilizando herramientas de escaneo o ingeniería social.

2. *Intrusion*: en esta fase se utilizan técnicas de *spear-phising*, archivos maliciosos adjuntos o puertas traseras (*backdoors*) para obtener permisos de acceso o archivos adjuntos maliciosos en el correo electrónico.

3. *Latent expansion*: el atacante busca mantener el control para obtener datos que le permitan continuar con la expansión dentro de la red.

4. *Information theft*: el atacante establece una conexión a un servidor donde pueda transferir los datos robados. Se pueden utilizar técnicas de cifrado para camuflar los datos extraídos.

Otro enfoque del ciclo de vida de cuatro fases ha sido detallado en [93]. En este enfoque, las fases se describen de la siguiente manera:

1. *Initial Compromise*: las técnicas utilizadas son de *spear-phising* e ingeniería social.

2. *C&C*: se establece un canal de comunicación entre un servidor comprometido y el objetivo.

3. *Lateral Movement*: los atacantes buscan recolectar información interna y moverse entre varios ordenadores con vulnerabilidades críticas.

4. *Attack achievemnt*: el ataque se ha completado y comienza el robo de información confidencial.

5.3. Modelos de ataque de cinco etapas

En el trabajo [74], se propuso un modelo para analizar el ciclo de vida de un ataque de APT organizado en cinco etapas y denominado *Attack Chain* (AC). Se describen a continuación las etapas de este modelo de ataque:

1. *Delivery*: el *spear-phising* se utiliza para enviar correos electrónicos dirigidos a destinatarios dentro de la red.
2. *Exploit*: se explotan las vulnerabilidades de los servicios, así como los sistemas y las aplicaciones.
3. *Installation*: en esta fase es posible la instalación de malware en los sistemas objetivo, por ejemplo, RAT (*remote access tool*).
4. *Command and Control*: el atacante tiene acceso remoto a un ordenador o servidor comprometido.
5. *Actions*: Las acciones que se llevan a cabo consisten en ganar acceso a otros ordenadores o servidores de la misma red para extraer información confidencial.

Los autores en [42] describen otro modelo diferente con las siguientes cinco etapas:

1. *Recconaisanse*: el objetivo es seleccionado, se busca toda la información publica relacionada al objetivo, que se encuentra publicada en Internet.
2. *Incursion*: el atacante obtiene acceso a la red a través de credenciales robadas con técnicas como *SQL injection* o con la utilización de malware.
3. *Discovery*: el atacante busca datos confidenciales en el sistema.
4. *Capture*: el atacante instala un *rootkit* no detectable para obtener datos confidenciales durante un largo período de tiempo.
5. *Exfiltration*: los datos recogidos son enviados a los servidores C&C.

5.4. Modelos de ataque de seis etapas

Los autores en [17] han adoptado un modelo de seis etapas basado en el concepto ya mencionado de *Intrusion Kill Chain*. Este modelo organiza las fases de la siguiente manera:

1. *Reconnaissase and weaponization*: es una fase de preparación para estudiar y recolectar información técnica de la organización objetivo. Algunas técnicas utilizadas son ingeniería social y *Open Source Intelligence* (OSINT).

2. *Delivery*: los atacantes envían *exploits* personalizados a los objetivos de manera directa o indirecta, por ejemplo, una técnica directa puede ser a través del *spear-phishing* y de manera indirecta a través del *watering-hole* attack.

3. *Initial intrusion*: la información obtenida en la fase anterior como credenciales validas, permiten a los atacantes ganar acceso en el objetivo, ejecutar código malicioso y explotar vulnerabilidades.

4. *Command and control*: los atacantes establecen un mecanismo para tomar el control de los ordenadores comprometidos, para esto los atacantes crean sitios en redes sociales, redes anónimas en TOR o utilizan herramientas de acceso remoto.

5. *Lateral movement*: cuando los atacantes han establecido una conexión a sus servidores C&C, se desplazan por la red de la organización en busca de información útil para obtener acceso a otros sistemas y comprometerlos.

6. *Data exfiltration*: los atacantes envían a sus servidores información confidencial de manera cifrada.

Los autores en [35, 88] propusieron un modelo de ciclo de vida de seis etapas para describir un ataque de APT. Este modelo enfatiza que los atacantes deben engañar a

una persona para que ejecute el malware y explotar cualquier vulnerabilidad de día cero. Luego, los atacantes acceden a la red corporativa desde el ordenador comprometido y ejecutan un ciclo de maniobras difíciles de alcanzar para lograr sus objetivos finales. Las seis etapas de este ciclo de vida son las siguientes:

1. *Information Gathering*: el objetivo de esta etapa es reunir información sobre la estructura de la organización a través de perfiles de redes sociales públicas.

2. *Point of entry*: la ingeniería social, el *spear-phishing* y la explotación de día cero son las técnicas más utilizadas por el atacante para engañar a la víctima y acceder al ordenador.

3. *Command and Control server* : el atacante establece un canal de comunicación desde el ordenador comprometido hasta el servidor C&C para mantener la conexión. El protocolo de cifrado SSL es el método que se utiliza normalmente para enviar el tráfico al servidor C&C.

4. *Lateral movement*: el atacante puede moverse a través de la red para encontrar un ordenador vulnerable cuando ha conseguido el acceso.

5. *Data of interest*: se identifica la información confidencial en los ordenadores y servidores.

6. *External server*: los datos de interés se transmiten a los servidores C&C de los atacantes.

5.5. Modelos de ataque de siete etapas

La compañía Lockheed Martin propuso un ciclo de vida de siete etapas llamado *Cyber Kill Chain* (CKC) [56]. Este modelo busca entender cómo funciona un ataque

para fortalecer la comprensión de las tácticas, técnicas y procedimientos utilizados por los atacantes. Estas etapas se describen a continuación:

1. *Reconnaissance*: el atacante realiza un reconocimiento preliminar de la red de la organización, utilizando técnicas de *spear-phising*, escaneo de puertos e ingeniería social.

2. *Weaponization*: el atacante construye un *payload* que es enviado a la víctima. Usualmente consiste en un *exploit* con un troyano de acceso remoto.

3. *Delivery*: el *payload* (componente del malware que ejecuta una actividad maliciosa) creado se envía a la víctima a través un correo electrónico, sitios web o un dispositivo de almacenamiento.

4. *Explotation*: el atacante ejecuta un *exploit* que ha sido enviado a la víctima.

5. *Installation*: un troyano o un RAT se instala cuando el atacante obtiene acceso al sistema.

6. *Command and control*: el software de acceso remoto se conecta al servidor C&C del atacante.

7. *Actions and objetives*: el atacante realiza una exfiltración de datos que compromete la integridad y la disponibilidad de los mismos. Esta etapa puede durar semanas, meses o incluso años.

Otro modelo de ataque de siete etapas se presentó en [91] y consta de las siguientes etapas:

1. *Research*: los atacantes buscan información sobre la víctima disponible públicamente.

2. *Preparation*: los atacantes crean un ataque inicial para explotar las vulnerabilidades utilizando el escaneo de la red, para crear *exploits* personalizados.

3. *Intrusion*: los atacantes lanzan el primer ataque, que usualmente consiste en *spear-phising*.

4. *Conquering the network*: las herramientas de acceso remoto o puertas traseras para controlar el sistema, son utilizadas cuando el atacante ha comprometido al menos a un ordenador.

5. *Hiding presence*: el atacante busca permanecer oculto en la red durante mucho tiempo. El ataque puede tener períodos de inactividad.

6. *Gathering data*: el atacante busca datos de interés y los enmascara como tráfico legítimo para extraerlos lentamente.

7. *Maintaining access*: el atacante puede modificar o crear *exploits*, herramientas de acceso remoto y servidores C&C, para obtener un acceso prolongado a la red.

5.6. Modelos de ataque de ocho etapas

Mandiant (ahora FireEye), propuso un modelo de ocho etapas después de analizar la campaña llevada a cabo por el grupo APT1 [58], que está organizado de la siguiente manera:

1. *Initial recon*: reconocimiento inicial del objetivo.

2. *Initial compromise*: se describen los métodos utilizados para la primera intrusión del objetivo, por ejemplo, *spear-phishing*.

3. *Establish foothold*: consiste en asegurar el control del objetivo desde fuera de la red, por ejemplo, con servidores de C&C.

4. *Escalate privileges*: el atacante busca credenciales que le permitan acceder a más recursos dentro del sistema.

5. *Internal recon*: en esta etapa, el atacante recoge toda la información posible sobre la víctima.

6. *Move laterally*: el atacante puede conectarse y compartir recursos utilizando credenciales legítimas.

7. *Maintain presence:* el atacante realiza acciones para permanecer durante un período prolongado dentro de la red sin ser detectado.

8. *Complete mission*: los atacantes han utilizado métodos de compresión de ficheros para enviar la información de interés a los servidores de C&C.

5.7. Modelos de ataque de once etapas

El análisis de tácticas del entorno de trabajo ATT&CK, muestra las distintas etapas de un ataque en el que un actor trabaja para lograr una intrusión en el objetivo estratégico. La matriz de técnicas de ATT&CK se describe la siguiente manera:

1. *Initial access*: consiste en el contacto inicial con el objetivo para buscar al paciente cero.

2. *Persistence*: el atacante busca acceder durante mucho tiempo al objetivo.

3. Privilege escalation: para obtener privilegios en la red es necesario instalar un malware o acceder a datos confidenciales.

4. *Discovery*: consiste en obtener información relevante del objetivo, como la ubicación del sistema o nombres de usuario.

5. *Lateral movement*: se refiere a la forma en que el atacante se mueve dentro de la red para buscar información o servicios importantes y vulnerables.
6. *Collection*: recopilación de información relevante para el atacante.
7. *Exfiltration*: el atacante extrae los datos recogidos.

Las siguientes etapas logran el objetivo del ataque, y pueden ser ejecutadas en paralelo con las siete etapas anteriores.

8. *Execution*: la ejecución de malware a través de conexiones remotas que se llevan a cabo entre la etapa de acceso inicial y el movimiento lateral.
9. *Defence evasion*: consiste en no ser detectado por los mecanismos de defensa y detección, por ejemplo, el cortafuegos o los registros del sistema.
10. *Credential access*: se refiere a acceder al sistema comprometido con credenciales válidas.
11. *Command and control*: consiste en crear un canal C&C para comunicar los servidores del atacante con los sistemas comprometidos del objetivo.

Los ciclos de vida propuestos tienen similitudes en los métodos y técnicas utilizados por los atacantes en cada etapa. Por consiguiente, una etapa de un ciclo de vida puede dividirse en diferentes pasos, para explicar con más detalle cómo funciona un ataque de APT. Por esta razón, los investigadores pueden seleccionar uno o varios ciclos de vida para adaptarlo a su trabajo, también, pueden tomar un ciclo propuesto como base para crear un nuevo modelo de ataque. Cada ataque de APT tiene características únicas, y se pueden describir utilizando diferentes enfoques similares.

5.8. Comparación de los modelos de ataques de APT

Una comparación de los modelos de ciclos de vida de los diferentes ataques de APT se ha realizado en la Tabla 5.1. Se puede observar que los ciclos con la misma cantidad de etapas tienen diferentes maneras de explicar el comportamiento de un ataque de APT, como es el caso de los ciclos de ataque de cuatro, cinco, seis y siete etapas, de los cuales se han descrito más de un enfoque.

Las bases de algunos de estos enfoques propuestos son los modelos IKC, CKC y AC. El CKC es un modelo muy conocido y que ha sido utilizado como base para el ciclo de vida de siete etapas; por otro lado, el modelo IKC se ha utilizado como base para los ciclos de vida de cuatro y seis etapas; y ciclo de vida de cinco etapas se ha utilizado como base del modelo de AC.

Un ciclo de tres etapas puede describir los pasos que sigue un ataque de APT, de manera similar a un ciclo de vida de cinco etapas, e incluso hasta de once etapas. Por esta razón, se han agrupado los ciclos de vida que presentan etapas con características similares; por ejemplo, la etapa inicial del ciclo de vida de tres etapas puede ser similar a las etapas de acceso inicial, persistencia y escalada de privilegios del modelo de once etapas.

Otro punto a destacar es que algunos autores indican que la conexión C&C se realiza antes de iniciar el escaneo de la red. No obstante, estos mismo autores sitúan esta etapa al final del ciclo de vida cuando se extraen los datos. El ciclo de vida de once etapas indica que hay etapas que pueden desarrollarse en paralelo con las principales etapas del ciclo, para mantener la persistencia en el objetivo y extraer información confidencial cuando ha sido hallada.

Los ciclos de vida revisados coinciden en que los primeros pasos del ataque son el estudio y el análisis del objetivo. Luego, se produce una explotación de las vulnerabilidades para comprometer a uno o más ordenadores, dentro de la red del objetivo. Por último, la extracción de los datos hacia un servidor C&C se realiza de manera sigilosa por los atacantes. El ciclo de vida resultante del análisis de los ataques realizados por el grupo APT1, describe la limpieza de huellas como una etapa final, que luego de ser ejecutada, es posible que la organización no detecte que ha sido atacada.

Es importante señalar que los ciclos de vida son propuestos para dar una idea de cómo funciona un ataque de APT, sin embargo, cada atacante puede llevar a cabo las etapas en cualquier orden y utilizar las TTP que se adaptan para cumplir sus objetivos. A continuación, se realiza un análisis de estos ciclos de vida.

El modelo de ataque de tres etapas tiene la ventaja de describir de forma breve como funciona un ataque de APT. En este caso se han analizado al menos 20 campañas y se han descrito por etapas las TTP que han resultado del análisis realizado.

Para el modelo de cuatro etapas, se han analizado dos propuestas, el primero de ellos está basado en el modelo IKC. La diferencia entre estas dos propuestas, es que el primer modelo describe una fase de reconocimiento inicial de red, a partir de técnicas de escaneo. El segundo modelo, ha divido la etapa de expansión por la red del primer modelo, en dos etapas; una para la conexión con los servidores C&C y otra para el movimiento lateral.

En el modelo de cinco etapas se han descrito dos propuestas, la primera propuesta está basada en el modelo AC. Las primeras dos etapas de estos modelos son similares, en ellas se estudia el objetivo y se realiza la primera intrusión. La diferencia entre estos dos modelos, se enfoca principalmente en el momento que se realiza la conexión a los servidores de C&C; en el primer modelo esta conexión se lleva a cabo antes de realizar el movimiento lateral en la red. Sin embargo, en el segundo enfoque, la conexión a los

servidores C&C se lleva a cabo luego que han sido recolectados los datos confidenciales por un largo periodo de tiempo.

Para el modelo de seis etapas hay dos propuestas descritas, el primer modelo está basado en IKC. Estos modelos presentan las etapas del ataque descritas de manera similar, la diferencia está en los nombres de las etapas y algunas TTP utilizadas por cada uno de los modelos. Por ejemplo, en la última etapa, que es el envío de la información a los servidores C&C, en el primer modelo se ha llamado *external server* y en el segundo modelo *data exfiltration*.

Para el modelo de siete etapas, también hay dos propuestas descritas; el primer modelo está basado en CKC. La diferencia entre estos modelos es que en el primero se describe una etapa para establecer la presencia oculta o persistencia durante mucho tiempo, destacando que puede haber periodos de inactividad; mientras que el segundo modelo se detalla el uso de un troyano de acceso remoto para mantener dicha persistencia.

El modelo de ocho etapas consiste en el análisis de los ataques realizados por el grupo APT1. Este modelo es el más conocido en la literatura para la descripción de los modelos de ataque de APT. Este enfoque describe que un ataque puede permanecer oculto durante un largo periodo sin ser detectado y que los datos pueden ser enviado de manera comprimida a los servidores de C&C.

Por último, el modelo de once etapas es el más detallado de los modelos existentes, el cual describe de manera precisa los pasos que sigue un ataque de APT. La principal diferencia con respecto al resto de modelos, es que se han propuesto cuatro etapas que pueden ejecutarse en paralelo junto a las siete etapas principales; estas etapas paralelas describen la interacción constante con los servidores C&C del atacante.

En resumen, los ciclos de vida pueden describir de diversas maneras la forma en que opera un ataque de APT. Estos ciclos pueden adaptarse a las necesidades de los

atacantes y de los investigadores para explicar los pasos realizados durante el ataque. Por esta razón, es posible utilizar un ciclo de vida para describir un modelo que permita detectar un ataque de APT.

En este caso, se han estudiado los diferentes ciclos de vida propuestos por otros autores, para determinar un ciclo que pueda resumir de manera concreta los pasos de un ataque de APT, en el menor número de etapas posibles, y que, a la vez, permita identificar las técnicas, tácticas y procedimientos utilizados en cada una de estas etapas.

Tabla 5.1 Comparación entre los diferentes enfoques propuestos para conocer el ciclo de vida de un ataque de APT.

3 etapas [89]	**4 etapas [97]**	**4 etapas [93]**	**5 etapas [42]**	**5 etapas [74]**	**6 etapas [35]**	**6 etapas [17]**	**7 etapas [91]**	**7 etapas [56]**	**8 etapas [58]**	**11 etapas [84]**
Initial compromise	Information collection	Initial compromise	Reconnaissance	Delivery	Intelligence gatherin	Reconnaissance and weaponization	Research	Reconnaissance	Initial recon	Initial access
	Intrusion phase		Incursion		Initial compromise	Delivery	Preparation	Weaponization	Initial compromise	Persistence
						Initial intrusion	Intrusion	Delivery		Privilege escalation
Lateral movement	Lateral expansion	C&C	Discovery	Exploit	C&C	C&C	Conquering network	Exploitation	Establish foothold	Discovery
		Lateral movement	Capture	Installation	Lateral movement	Lateral movement	Hiding presence	Installation	Escalate privileges	Lateral movement
					Assets/data discovery				Internal recon	
									Move laterally	
Command and control	Information theft phase	Attack achievement	Ex-filtration	C&C	Data ex-filtration	Data ex-filtration	Gathering data	C&C	Maintain presence	Collection
				Actions			Maintaining access	Actions on objective	Complete mission	Exfiltration
										Stages executed in parallel: Execution, Defence evasion, Credential access, and Command & control

Capítulo 6

Modelo propuesto

Las amenazas persistentes avanzadas han sido objeto de estudio con la finalidad de encontrar una forma de describir su comportamiento e identificar las posibles técnicas que permitan su detección de manera temprana, eficiente y en tiempo real; para ello, se han propuesto diferentes enfoques que se basan en el ciclo de vida de un ataque de APT.

Como ya se ha mencionado a lo largo de este trabajo, los ataques de APT son ataques dirigidos que funcionan de manera sigilosa, por lo que pueden comprometer un sistema durante un largo periodo de tiempo y ser difíciles de detectar.

No obstante, el comportamiento de un ataque clasificado como APT es diferente de acuerdo a las acciones que realiza el atacante y el tipo de organización que sufre el ataque; por lo que puede resultar compleja la identificación de las tácticas, técnicas y procedimientos utilizados por los atacantes para cumplir su objetivo.

Tal como se describe en la Sección 2.4.1, los atacantes suelen ser grupos delictivos organizados o actores gubernamentales cuyo objetivo son las infraestructuras críticas, los diferentes sectores industriales y las organizaciones gubernamentales. Por lo general, estos objetivos tienen una infraestructura tecnológica robusta y cuentan con medidas de seguridad avanzada que les permiten proteger los activos tecnológicos, como datos

confidenciales, procesos y servicios industriales. Sin embargo, los ataques de APT son capaces de evadir las medidas de seguridad y causar una intrusión dentro de la organización.

Los enfoques propuestos por diferentes autores para realizar el análisis del comportamiento y la detección de un ataque de APT han seleccionado aquellas TTP más comunes, por ejemplo, conexiones remotas, registros del sistema, alertas IDS y vulnerabilidades de día cero (ver Sección 4).

Consecuentemente, con la finalidad de facilitar la detección de un ataque de APT de manera temprana y eficiente, el modelo propuesto se basa en el ciclo de vida de un ataque de APT, donde cada etapa de este ciclo incluye algunas de las TTP utilizadas frecuentemente por los atacantes. Estas TTP se han seleccionado de la matriz ATT&CK *Entreprise* de MITRE [84], donde los autores propusieron un listado de TTP organizado en un ciclo de vida de once etapas (ver Sección 5.7). Además, el modelo propuesto consta de módulos de detección que podrían generar alertas cuando se identifique que se está ejecutando un ataque de APT. En la Figura 6.1 se incluye un esquema del modelo propuesto que se detallará en las siguientes secciones de este capítulo.

En el modelo planteado se considera que el ciclo de vida de un ataque de APT puede tener acciones pasivas, activas y recurrentes, realizadas por los atacantes, que pueden ir desde ataques de ingeniería social hasta ataques específicos, como el acceso no autorizado a servidores dentro de la red del objetivo.

Aquellas acciones que no modifican los datos ni interfieren en la transmisión de la información dentro de la red se consideran acciones pasivas; por ejemplo, las técnicas de escaneo de puertos. Por otro lado, las acciones que modifican los datos, eliminan información o cambian el flujo de paquetes de red se consideran acciones activas; como es el caso de los ataques de denegación de servicio distribuido (DDoS). Adicionalmente,

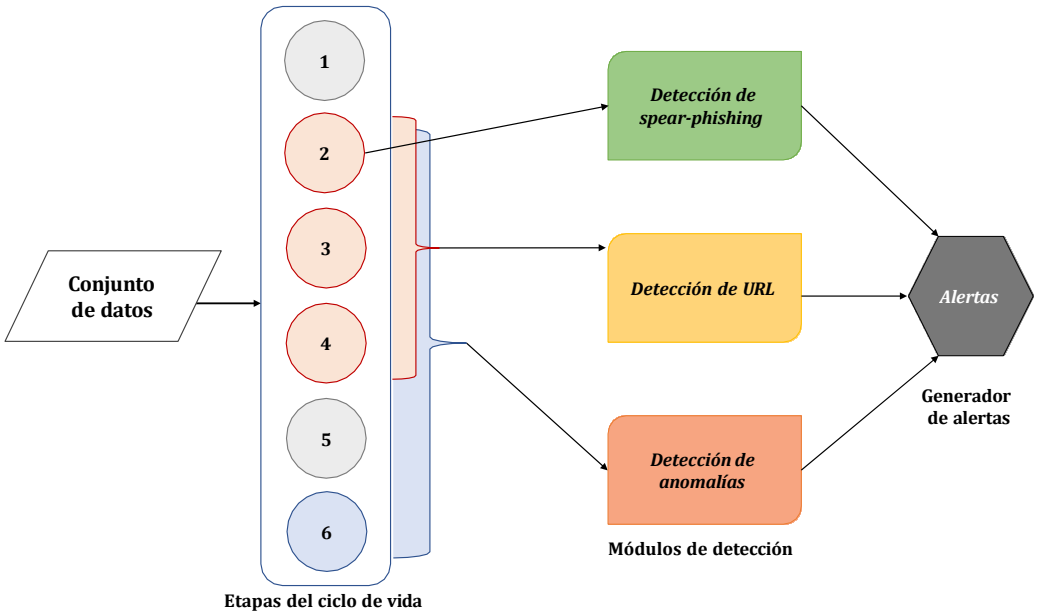


Figura 6.1 Esquema del modelo propuesto.

las acciones que se realizan a lo largo de todo el periodo activo del ataque pueden considerarse como acciones recurrentes, como la evasión de sistemas de defensa.

El ciclo de vida de este modelo se ha organizado en seis etapas que describen los pasos seguidos en un ataque de APT: dos etapas pasivas en las que las acciones que se llevan a cabo no suelen ser identificadas como un ataque de APT real; tres etapas activas del ciclo de vida que incluyen las TTP seleccionadas de la matriz de MITRE y detallan las posibles medidas para mitigar el ataque; y finalmente, una etapa recurrente donde las acciones se ejecutan paralelamente con las acciones de las etapas activas.

En las dos etapas pasivas del ciclo de vida se presentan los pasos que el atacante utiliza para conocer su objetivo, mediante el uso de herramientas de exploración y obtención de información pública relacionada con la organización; además, una vez terminado el ataque, el atacante procede a la eliminación del posible rastro que ha dejado dentro de la red, específicamente en los sistemas de recolección de registros y programas instalados que han sido utilizados durante la ejecución del ataque.

Las tres etapas activas del ciclo de vida están compuestas por una serie de pasos que el atacante utiliza para acceder a la red de la organización; durante estas etapas, se utilizan diferentes técnicas de intrusión para identificar las vulnerabilidades que puedan estar presentes dentro de los dispositivos comprometidos.

Una vez dentro de la red, el atacante intentará moverse de manera sigilosa utilizando accesos legítimos para acceder a directorios y servicios que contengan información confidencial; además, buscará la forma de mantener la persistencia en la red a través de la manipulación de cuentas de acceso.

Por último, la etapa recurrente de este ciclo de vida, es la que se encarga de identificar y conocer la forma de evadir los sistemas de defensa con los que cuenta la organización. Cabe señalar que esta última etapa, está presente durante todo el periodo de ejecución del ataque.

Para la detección de un ataque de APT, el modelo propuesto consta de tres módulos de detección para identificar los correos dirigidos, las URL maliciosas y las anomalías en la red, utilizando técnicas y algoritmos de aprendizaje automático. Estas técnicas y algoritmos de ML proporcionan una solución para el análisis de grandes cantidades de datos, como alertas de IDS, registros o conexiones remotas no autorizadas; el análisis de estos datos puede ayudar a los administradores de TI a identificar comportamientos anómalos en la red, que pueden estar asociados a la utilización indebida de los recursos informáticos, malware instalado en un ordenador de la red o un ataque de APT.

6.1. Escenario de propagación

Para comprender el comportamiento de un ataque de APT es primordial analizar dicho ataque a partir de los elementos involucrados en el proceso; estos elementos deben analizarse en detalle antes de llevar a cabo el ataque, aunque alguno de ellos pueda modificarse durante la ejecución del mismo. A continuación, se describen de forma

resumida las características de un ataque de APT, que se han descrito en capítulos anteriores de esta memoria (ver Figura 6.2):

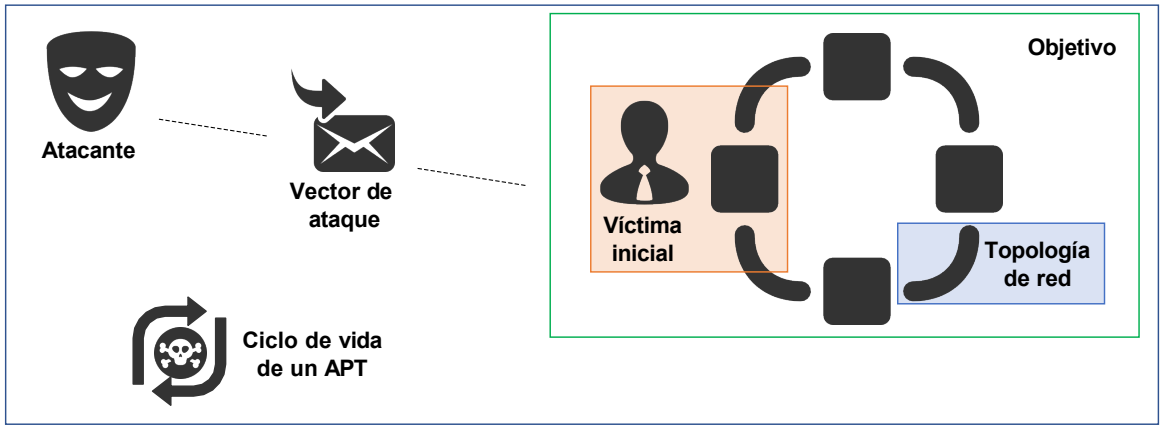


Figura 6.2 Esquema de un ataque de APT con sus características.

1. El atacante: también conocido como actor, está formado por un grupo de individuos que se encargan de llevar a cabo el ataque. Estos actores pueden ser de dos tipos, actores gubernamentales y grupos delictivos organizados.

2. El objetivo: suele ser una organización gubernamental o privada que posee información confidencial, a la que los atacantes quieren tener acceso. El objetivo puede ser una infraestructura crítica, una organización gubernamental o el sector industrial.

3. Víctima inicial: puede ser un colaborador o un ordenador de la red de la organización objetivo. Para llegar a la víctima inicial, los atacantes utilizan vectores de ataque que definen a partir de las vulnerabilidades o debilidades que han sido encontradas en las víctimas.

4. Vector de ataque: son las tácticas, técnicas y procedimientos utilizados por el atacante durante todo el periodo de actividad del ataque. Cada ataque de APT

adapta sus vectores de ataque al objetivo de forma personalizada, y es posible que sean modificados en tiempo real para ajustarse a las circunstancias del momento.

5. Topología de red: la red de cada objetivo está organizada de diferentes formas, puede contener redes de área local, redes de área local virtuales, cortafuegos o sistemas de detección y prevención de intrusos.

6. Ciclo de vida: está compuesto por las etapas que contempla un ataque de APT, existen diferentes modelos con un número diferente de etapas, desde tres hasta once etapas, esto ayuda a comprender el funcionamiento de estos ataques.

La descripción de un ataque de APT puede realizarse a partir de diferentes puntos de vista; en este caso, se describe el ataque de APT a partir del comportamiento del malware para crear una conexión a servidores de comando y control; además, se consideran las características del ataque, es decir, el atacante, objetivo, víctima inicial, vector de ataque, topología de red y el ciclo de vida; finalmente, se identifican las etapas que serán analizadas en el ciclo de vida del modelo propuesto. En la Figura 6.3 se muestra el escenario de propagación que se detalla a continuación.

Inicialmente, se desconoce quién es el atacante que esta ejecutando el ataque de APT, por lo que no se puede predecir su comportamiento. Los atacantes suelen ocultar su ubicación para engañar a los analistas de ciberseguridad, simulando una ubicación falsa, ya que la ubicación del atacante puede estar próxima a la dirección de la organización o encontrarse en algún lugar remoto del mundo. Por consiguiente, el descubrimiento del objetivo, suele diferenciarse de acuerdo al comportamiento del atacante y puede llevarse a cabo desde una ubicación falsa.

El malware utilizado por los atacantes llega a la víctima a través de un vector de ataque, que está compuesto por un conjunto de herramientas de explotación. En este escenario de propagación, el malware que ha llegado a la víctima inicial ha comprometido

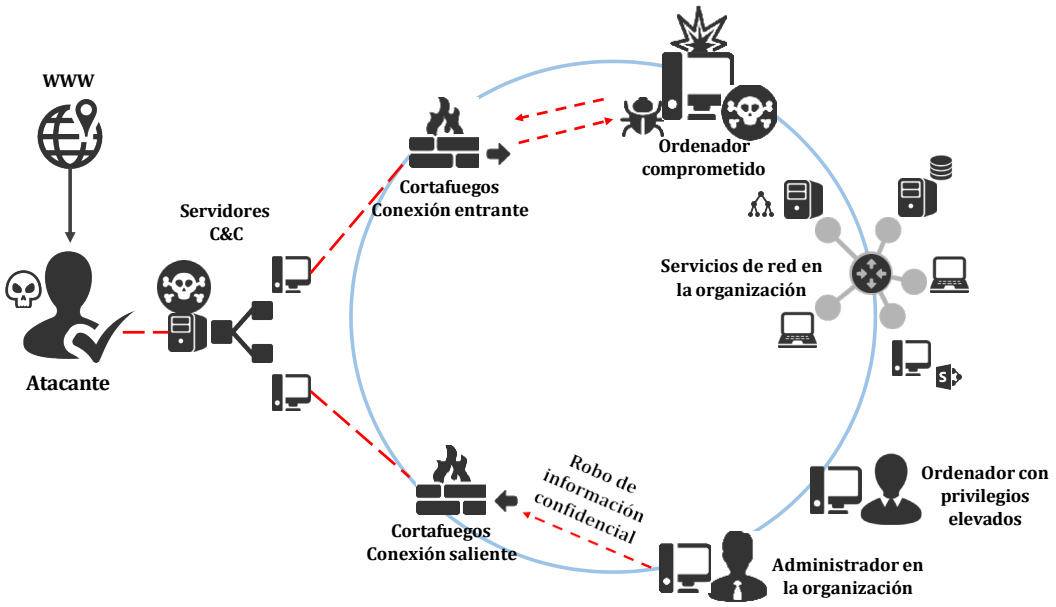


Figura 6.3 Escenario de propagación comúnmente utilizado en un ataque de APT.

al menos un ordenador de la red de la organización, permitiendo establecer la conexión a los servidores de comando y control, a través de herramientas como las redes Tor, para evadir los sistemas de defensa.

Cuando el atacante ha comprometido a su objetivo y ha logrado establecer una conexión con sus servidores, el siguiente paso a realizar es la intrusión interna; este paso consiste en escalar privilegios y acceder a otros ordenadores y servidores de la organización para buscar información que sea de interés para el atacante.

Posteriormente, el atacante establece canales para la extracción de los datos; estos datos pueden enviarse a través de archivos comprimidos utilizando paquetes que parecen reales en el tráfico de red. Este proceso suele realizarse de manera minuciosa y lentamente para evitar llamar la atención de los administradores de la red.

Finalmente, cuando el atacante considera que ha obtenido la información deseada, o cree que puede haber sospechas de que se está realizando una intrusión en la red, es

posible que realice la eliminación de todas las huellas que ha podido dejar durante el ataque.

6.2. Etapas del ciclo de vida propuesto

Algunos ataques de APT pueden permanecer sin ser detectados en una o más etapas de su ciclo de vida durante la ejecución del ataque; por ello, se propone una solución diferente, por etapas, del ciclo de vida, que permita la detección del ataque de APT, que va desde el inicio hasta el final del ataque, con el fin de identificar las TTP del ataque mediante la utilización de técnicas y algoritmos de aprendizaje automático.

Consecuentemente, en este modelo, las TTP juegan un papel importante para la detección de un ataque de APT; por ello, la identificación de las principales TTP se detallará en cada una de las etapas del ciclo de vida. En este caso, es importante identificar los recursos informáticos de los cuales el atacante puede extraer información dentro o fuera de la red de la organización, como los servidores, la arquitectura de la red, los sistemas operativos y los programas instalados.

Como se muestra en la Figura 6.4, las etapas del ciclo de vida están divididas en acciones pasivas (etapas 1 y 5), activas (etapas 2, 3 y 4) y una etapa recurrente (etapa 6), las cuales se describen a continuación:

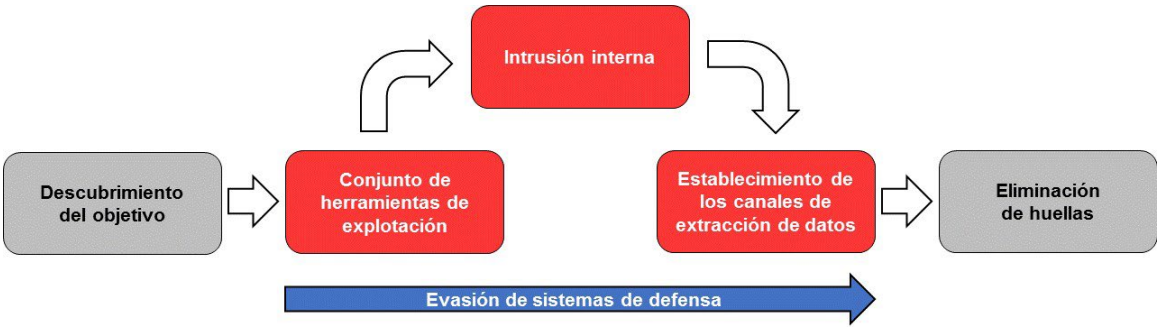


Figura 6.4 Esquema del ciclo de vida del modelo propuesto.

1. Descubrimiento del objetivo: esta etapa consiste en la exploración pasiva de los recursos informáticos de la red de la organización, con el fin de obtener detalles aproximados de la infraestructura informática, como los servicios web, puertos de red abiertos y vulnerables, la topología de la red y los servidores internos. Para obtener estos datos de la organización, el atacante puede utilizar técnicas de escaneo de puertos, búsqueda de servicios indexados en Internet (cámaras de vigilancia en la web, servidores o sistemas SCADA), perfiles públicos en las redes sociales de los empleados y herramientas de reconocimiento OSINT (*Open-source intelligence*).

   Estos tipos de técnicas de reconocimiento son difíciles de detectar mediante el uso de ML, ya que se realizan normalmente de forma pasiva. Una técnica de ataque pasiva, no modifica ni interfiere con la comunicación desde dentro o fuera de la red de la organización, lo cual permite la escucha o monitorización de la información transmitida por la red. La información disponible públicamente en Internet puede ser recopilada para su venta en la red profunda (*Deep web*); estas técnicas, pueden requerir el uso de múltiples herramientas especializadas durante un largo período.

   En concreto, el uso de ML en esta primera etapa del ciclo de vida no ayudaría a detectar de manera temprana un ataque de APT. Sin embargo, algunos algoritmos de ML de clasificación, como, SVM o NB pueden procesar datos a partir de los ficheros de registro generados por el sistema; esto puede representar una mejor visualización de la información de la actividad dentro de la red, y así determinar si un atacante está realizando algún tipo de reconocimiento en la red interna, utilizando herramientas como Nmap [64] o escaneo desde fuera de la red, con Shodan [78] o Spiderfoot [81].

   Para prevenir que un atacante obtenga información no autorizada, se recomienda cerrar los puertos no utilizados, utilizar cortafuegos, IDS y asegurar las conexiones

virtuales privadas, crear políticas de contraseñas, y la concienciar a los usuarios de la organización.

2. Conjunto de herramientas de explotación: esta etapa consiste en acceder a la red objetivo, a través de las vulnerabilidades detectadas en la etapa anterior o engañando a un empleado de la organización.

   El proceso se inicia con la elaboración de un método para alcanzar el objetivo del ataque. Para ello, el atacante utiliza técnicas como el envío de correos electrónicos dirigidos, conocido como *spear-phishing*, credenciales comprometidas o la replicación de malware a través de memorias USB.

   Luego, el atacante explota la vulnerabilidad detectada mediante el uso de *scripts* o pequeños programas para obtener una *webshell*, y ejecuta líneas de comando que le permitan mantenerse dentro de la red; o mediante la ejecución de archivos maliciosos por parte de los usuarios, que al mismo tiempo, este archivo explota cualquier servicio remoto desde fuera de la red de la organización, como el protocolo RDP.

   Otras TTP que pueden ser utilizadas en esta etapa son las descargas no autorizadas, el uso de credenciales comprometidas, la replicación de malware por medio de almacenamiento y la ejecución de comandos a través de diferentes plataformas; con estas técnicas, se busca que haya una forma de interactuar con los sistemas informáticos y los usuarios para comprometer los ordenadores de la red.

   Para evitar que un empleado sea persuadido para ser parte de un ataque, se recomienda evitar el uso de dispositivos personales dentro de la red de la organización, con políticas como BYOD (*bring your own device*), y evitar así la apertura de archivos sospechosos en caso de duda. Para ello, es importante realizar jornadas

de concienciación sobre los peligros en Internet con todos los colaboradores de la organización y evitar que se conviertan en víctimas de los atacantes.

Las técnicas de ML permiten crear soluciones automatizadas para detectar posibles ataques en esta etapa temprana. Por ejemplo, se puede crear un módulo para escanear correos electrónicos en busca de enlaces o archivos maliciosos.

Otra solución sería escanear el tráfico de la red en busca de paquetes de conexión remota desde servidores no autorizados, o analizar los registros para detectar actividad anómala dentro de la red, además de una gestión controlada y diaria de las actualizaciones de software y de seguridad del sistema operativo.

La implementación de estas soluciones de ML requiere de dos conjuntos de datos de entrenamiento, uno con el flujo normal del tráfico de la red de la organización, y otro conjunto de datos con flujo de datos anómalo. Se debe elegir el algoritmo de aprendizaje automático que proporcione los mejores resultados en cuanto a la detección de estas anomalías mencionadas. Finalmente, las pruebas deben realizarse en un ambiente controlado para evitar cualquier incidente que impacte contra de la red de la organización.

Los algoritmos de ML que han dado mejores resultados para la detección de estas anomalías han sido *k*-NN y SVM. Además, durante el entrenamiento inicial o el reentrenamiento del algoritmo de ML, se pueden añadir conjuntos de datos de entrenamiento con otros tipos de técnicas de ataque para mejorar el grado de detección del modelo.

3. Intrusión interna: en esta etapa el atacante busca comprometer un ordenador dentro de la red de la organización; una vez comprometido, el siguiente objetivo es elevar los privilegios de las cuentas de acceso para acceder a los directorios que contengan información confidencial y crítica de la organización. Para esto, el

atacante debe ser capaz de mantener la persistencia durante un período prolongado utilizando diferentes TTP; esta etapa es la más larga del ciclo de vida.

En esta etapa, el atacante utiliza una combinación de TTP para comprometer los ordenadores previamente identificados como vulnerables, como la exploración de la red (*network sniffing*), autenticación forzada (*forced authentication*) y la obtención de las credenciales de acceso de sistemas operativos (*OS credential dumping*); estas TTP le permitirán moverse lateralmente dentro de la red.

La persistencia dentro de la red se puede mantener a través de accesos redundantes, manipulación de cuentas o una *webshell*. Por otro lado, la obtención de las credenciales de acceso, puede lograrse mediante la implementación de técnicas de fuerza bruta, manipulación de cuentas, autenticación forzada, interceptación de autenticación de doble factor o la obtención de credenciales.

Consecuentemente, las soluciones para detectar un ataque en esta etapa consisten en utilizar técnicas de ML, como, *k*-means, NB, y SVM, para el análisis de los registros generados por los sistemas de IDS/IPS, a través de la identificación de patrones de ataques que ayuden a detectar un ataque de APT (por ejemplo, accesos fallidos a los servicios SSH, FTP o telnet), o el análisis de los registros del sistema (por ejemplo, instalaciones de programas no autorizados, directorios y archivos con nombres codificados u ordenadores desconocidos en la red).

4. Establecimiento de los canales de extracción de datos: esta etapa consiste en establecer una conexión con el servidor C&C del atacante, para acceder a toda la información recopilada de los ordenadores de la red de la organización. Normalmente, estos datos son enviados a través de canales de comunicación de red, en archivos comprimidos y cifrados; consecuentemente, para evitar ser descubierto, el atacante limita la cantidad y el tamaño de los ficheros de los datos extraídos.

Los datos recopilados suelen enviarse en paquetes pequeños durante las horas de menor uso del ancho de banda de la red, como las noches o fines de semana; además, el atacante puede usar técnicas de *fast-flux* para realizar las conexiones salientes desde la red de la organización. Otra opción, es almacenar los datos recopilados en un ordenador dentro de la red de la organización, para ser enviados en pequeños paquetes al servidor de C&C cuando el objetivo de la misión es completado.

Algunas TTP utilizadas para recopilar datos son la recolección automatizada, correo electrónico y *man-in-the-browser*. La extracción de datos puede ser automatizada y por diferentes medios (por ejemplo, mediante protocolos alternativos, usando la red o dispositivos físicos). Las herramientas utilizadas en los servidores de C&C son protocolos de generación de dominios, herramientas de acceso remoto y cifrado en varias capas.

Como solución para la detección del envío de datos a los servidores de C&C, se pueden utilizar técnicas de ML, como los algoritmos *k*-NN y *k*-means para buscar ordenadores con datos cifrados, conexiones con direcciones IP y DNS aleatorias, y flujos de datos cifrados a servidores desconocidos y no autorizados.

5. Eliminación de huellas: esta etapa se lleva a cabo al finalizar el ataque; es decir, cuando el atacante ha completado su misión. El objetivo de esta etapa es eliminar todos los posibles rastros del ataque en la red de la organización y los sistemas comprometidos, estos rastros pueden encontrase en registros, archivos comprimidos, software instalado o malware.

   Si el atacante ha logrado ejecutar esta etapa, es posible que la organización no pueda detectar que ha sido comprometida y atacada por un ataque de APT; por lo que sería difícil comprobar la cantidad de información que ha sido extraída

por el atacante y el tiempo que ha permanecido oculto dentro de la red de la organización.

En esta etapa del ciclo de vida, el atacante puede ejecutar ataques de denegación de servicio dentro de la red para causar una distracción a los administradores de la red, luego se encargará de no dejar rastro en los sistemas de la organización. Además, otras TTP utilizadas en esta etapa son la destrucción y manipulación de datos e información.

Esta etapa puede ser difícil de detectar, puesto que la eliminación o sobre-escritura de archivos es muy común durante la actividad diaria de un sistema informático. Sin embargo, las copias de seguridad pueden ser de gran ayuda para recuperar ficheros dañados; en este caso, la copia de seguridad de los ficheros podría ayudar a identificar las huellas del atacante.

Por este motivo, las copias de seguridad deben ser uno de los activos más valorados dentro de una organización. Para llevar a cabo este proceso, se deben establecer políticas de seguridad, como detallar cuándo, dónde y cómo se realizarán estas copias de seguridad, definir qué información será almacenada, y lo más importante, establecer dónde se almacenará, puesto que debe almacenarse fuera de la organización y, si es posible, en más de un lugar con diferentes ubicaciones desconocidas para los atacantes.

Para evitar que un atacante llegue a esta etapa, lo recomendable es detectar de forma temprana el ataque y detenerlo en alguna de las etapas descritas anteriormente, con el fin de evitar posibles daños perjudiciales en la organización; por lo que, en esta etapa no se utilizan técnicas de ML.

6. Evasión de los sistemas de defensa: esta etapa es otro de los pasos esenciales que realiza el atacante, que consiste en la evasión de los sistemas de defensa dentro

de la red, por ejemplo, el IDS, el IPS y el cortafuegos; básicamente, esta evasión puede realizarse a través de conexiones *proxy* y la ofuscación de ficheros.

No obstante, la evasión de los sistemas de defensa se lleva a cabo a lo largo de todas las etapas del ciclo de vida del ataque, con la finalidad de mantener la persistencia en las conexiones a los servidores C&C de los atacantes; por lo que esta etapa se considera como recurrente.

En esta última etapa del ciclo de vida, los atacantes por medio de las TTP buscan mantener la persistencia dentro de la red y de los ordenadores comprometidos. Principalmente, estas TTP buscan ofuscar información, ocultar artefactos y explotar cualquier sistema de defensa. Otras TTP que se pueden mencionar son los *rootkits* y la utilización de *template injection*.

Los sistemas de defensa son cada vez más sofisticados y capaces de evitar conexiones maliciosas en la mayoría de los casos. Sin embargo, el atacante busca evitar ser detectado utilizando credenciales de acceso robadas, explotando alguna vulnerabilidad de los sistemas de defensa, o incluso aprovechando los periodos en que los sistemas pueden quedar obsoletos o sin actualizaciones por parte del fabricante.

Uno de los principales objetivos de esta etapa sería identificar las conexiones frecuentes a destinos desconocidos y bloquear esas conexiones. Por otro lado, se deben mantener actualizados los sistemas de defensa y, en caso de ser posible, utilizar más de uno, tanto sistemas a nivel de red como sistemas a nivel de dispositivos.

En la mayoría de los ataques de APT se utilizan ficheros infectados para ejecutar y descargar piezas de malware, esto se puede utilizar para evadir los sistemas de defensa. La identificación y el análisis de estos ficheros (normalmente de ofimática)

puede ser de gran ayuda para combatir un ataque de APT dentro de esta etapa del ciclo de vida.

Como solución, para la identificación de conexiones de red desconocidas se pueden utilizar algoritmos de ML como *k*-NN y ANN que pueden facilitar la búsqueda de direcciones URL que presenten comportamiento anómalo.

En resumen, se ha propuesto un ciclo de vida de seis etapas; en la Figura 6.5 se puede observar las TTP que fundamentalmente son utilizadas por los atacantes en cada etapa de este ciclo de vida y que han sido analizadas para detectar un ataque de APT en el menor tiempo posible. Sin embargo, es importante señalar que estas TTP no son las únicas utilizadas por los atacantes, puesto que pueden servirse de otras que permitan la detección temprana.

Además, una misma TTP puede ser ejecutada de diversas formas, es decir, puede cambiar el orden de ejecución debido a la utilización de programas maliciosos o líneas de comando ejecutadas mediante *scripts*, con los que se lleva a cabo el ataque.

Considerando todo esto, es necesario identificar las técnicas y algoritmos de ML que mejor se adapten para la detección de las TTP en cada etapa del ciclo de vida del modelo propuesto, ya que esto ayudaría a combatir este tipo de amenazas.

6.3. Módulos de detección de un ataque de APT

Se proponen tres módulos de detección basados en el análisis previo de las TTP identificadas; además, el funcionamiento de estos módulos está relacionado con el ciclo de vida propuesto para detectar ataques de APT. En este caso, se detalla cómo se pueden aplicar los algoritmos de aprendizaje automático supervisado para la detección de *spear-phishing*, URL maliciosas y anomalías en la red.

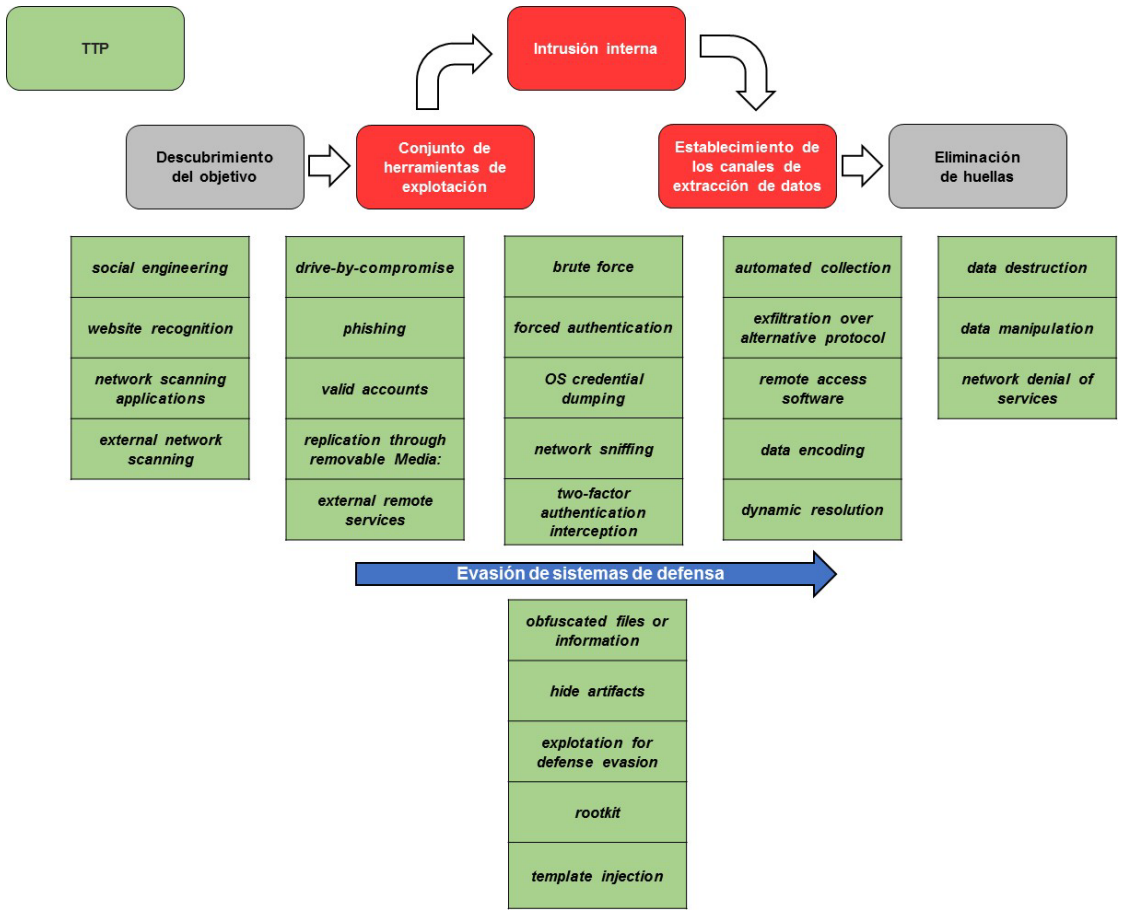


Figura 6.5 TTP utilizadas en las diferentes etapas del ciclo de vida.

En cada uno de los tres módulos de detección que se describen a continuación, se mencionan los algoritmos que pueden aplicarse para la detección de un ataque de APT. Estos algoritmos de ML han sido seleccionados porque han demostrado su utilidad en situaciones similares; además, presentan las siguientes ventajas:

- El algoritmo *logistic regression* puede ser fácil de entrenar, soporta un gran número de características y tiene un alto grado de escalabilidad, debido a la simplicidad de su función de puntuación.
- El algoritmo *decision tree* es rápido y preciso, ya que puede manejar un gran conjunto de datos; los patrones generados son fácil de entender y en comparación a otros enfoques, este algoritmo toma decisiones de forma similar a los humanos.
- El algoritmo *support vector machine* es relativamente eficiente en el uso de memoria con un conjunto de datos pequeño; es más efectivo en los casos en que la dimensión es mayor al número de muestras y cuando hay un claro margen de separación entre las clases.
- El algoritmo *artificial neural network* tiene la capacidad de aprender de eventos no lineales y complejos; logra altas tasas de cálculo a través de elementos de procesamientos simples, y también aprende muy bien de las relaciones ocultas en los datos, sin imponer relaciones fijas.
- El algoritmo *k-NN* es un gran algoritmo de clasificación y también puede ser usado para regresión; resulta fácil de implementar, inicialmente solo se necesita el valor de k y la función de distancia; por otro lado, al no requerir entrenamiento antes de predecir, se le pueden añadir nuevos datos sin afectar a la precisión del algoritmo.
- El algoritmo *naive bayes* funciona de forma muy eficiente, es rápido, simple y fácil de implementar; requiere una pequeña muestra de los datos de entrenamiento

para estimar los datos de prueba; además, utiliza muy pocos recursos, por lo que es computacionalmente bueno en comparación con otros algoritmos.

- Algunas de las ventajas del algoritmo *k*-means es que es fácil de implementar y entender; es un algoritmo que realiza cálculos bastante rápidos y suele adaptarse bien a grandes conjuntos de datos; es considerado un algoritmo lineal debido a que *k* y *t* son pequeños.

Los tres módulos de detección propuestos están unidos y trabajarán a la par del modelo propuesto de ciclo de vida de un ataque de APT, esto ayudará a identificar los diferentes patrones que se puedan generar en cada una de las etapas, a partir de la ejecución de las TTP por parte del atacante.

En la primera etapa activa del ciclo de vida propuesto, llamada conjunto de herramientas de explotación, se utilizará el módulo de detección basado en *spear-phishing*. En las etapas activas del ciclo de vida propuesto conocidas como conjunto de herramientas de explotación, intrusión interna y establecimiento de canales de extracción de los datos, se usará el módulo de detección de URL. Por último, el módulo basado en la detección de anomalías será utilizado en todas las etapas activas del modelo y en la etapa recurrente, nombrada evasión de sistemas de defensa.

6.3.1. Módulo de detección de *spear-phishing*

Hoy en día, el correo electrónico juega un papel importante en el funcionamiento de las organizaciones, puesto que es el medio de comunicación digital más utilizado; es, además, una comunicación bidireccional que reduce costes.

Por ello, los atacantes buscan crear formas fraudulentas de engañar a sus víctimas, ya sea ofreciéndoles una oferta de un producto y a su vez solicitando datos personales (lo que se conoce como *phishing*); o enviándoles un correo de manera dirigida, haciéndose

pasar por un contacto de confianza, que incluye adjunto un fichero infectado, como es el caso del *spear-phishing*.

Tradicionalmente, los métodos para detectar correos fraudulentos utilizan filtros basados en clasificación, que funcionan a base de palabras clave sospechosas; comparado con un correo dirigido que aparenta haber sido enviado por un contacto de confianza, el método tradicional no resulta efectivo, incluso dichos correos están diseñados para pasar desapercibidos ante las soluciones de antivirus.

Además, los ficheros adjuntos en los correos dirigidos suelen estar infectados para dar paso al atacante sin ser detectado; estos ficheros maliciosos pueden ser, documentos PDF, documentos OLE de Microsoft Office o documentos en Open XML, entre otros.

Este módulo de detección consiste en clasificar e identificar correos electrónicos fraudulentos y dirigidos que han sido enviados por los atacantes. Estos correos han pasado del filtro de correos *spam* a los buzones de entrada de cualquier miembro de la organización.

Considerando esto en el modelo propuesto, se plantea un módulo de detección de *spear-phishing*. La cantidad de correos que puede estar en el tráfico de la red en una organización es particularmente alto, por ello, al ser esto un problema de clasificación, los algoritmos de aprendizaje automático *decision tree*, *logistic regression* o *naive bayes*, pueden ser de ayuda para la detección de este tipo de TTP.

La utilización de un conjunto de datos con una buena cantidad de características relevantes y datos etiquetados permitirá modelar y analizar el tráfico de red para detectar ataques de *spear-phishing*. De forma general, las características fundamentales de un conjunto de datos adecuado serían la dirección IP, la longitud de la URL, el certificado SSL de seguridad del servidor de correos, objetos incrustados en el correo (*iframes*) y la redirección hacia otros dominios web.

6.3.2. Módulo de detección de URL maliciosas

Actualmente, un gran número de personas pasa muchas horas del día utilizando Internet, ya sea en la vida laboral o personal; esto representa que tienen a su alcance una gran cantidad de contenido por medio de las direcciones URL a las que pueden acceder; sin embargo, están expuestos y pueden ser engañados al ingresar a un enlace malicioso, por no saber identificar correctamente una URL legítima de una maliciosa.

Los atacantes suelen crear URL maliciosas para que parezcan legítimas con el fin de que la víctima visite un sitio web infectado o fraudulento, de manera directa o indirecta. Existen diversas maneras de que un atacante distribuya una URL maliciosa, por ejemplo, insertándola en imágenes gratuitas, en productos falsificados o sitios web que imitan ser legítimos, como los sitios web de entidades bancarias.

Tradicionalmente, los métodos de listas negras de direcciones URL funcionan para detectar direcciones URL maliciosas; no obstante, estas listas son estáticas y contienen una gran cantidad de direcciones identificadas previamente como maliciosas.

Este módulo de detección consiste en identificar las direcciones utilizadas por los atacantes, como nombres de domino web, conexiones a bases de datos o direcciones IP, servidores de comando y control que utilizan direcciones *fast-flux* como se muestra en la Figura 6.6, entre otras.

En el caso de los ataques de APT, las direcciones URL maliciosas son utilizadas por las diferentes TTP dentro de cada una de las etapas del ciclo de vida propuesto; dichas URL se podrían clasificar de la siguiente manera:

1. *Phishing URL*: esta acción se lleva a cabo a través de correos electrónicos o mensajería instantánea SMS.

2. *Drive-by-download URL*: se produce cuando el usuario hace clic en un enlace e infecta su ordenador a través de la descarga de un fichero ejecutable.

3. *Command and control URL*: este tipo de URL puede ser clasificada como maliciosa cuando se detecta que un malware se conecta a través de servidores de comando y control remotos.

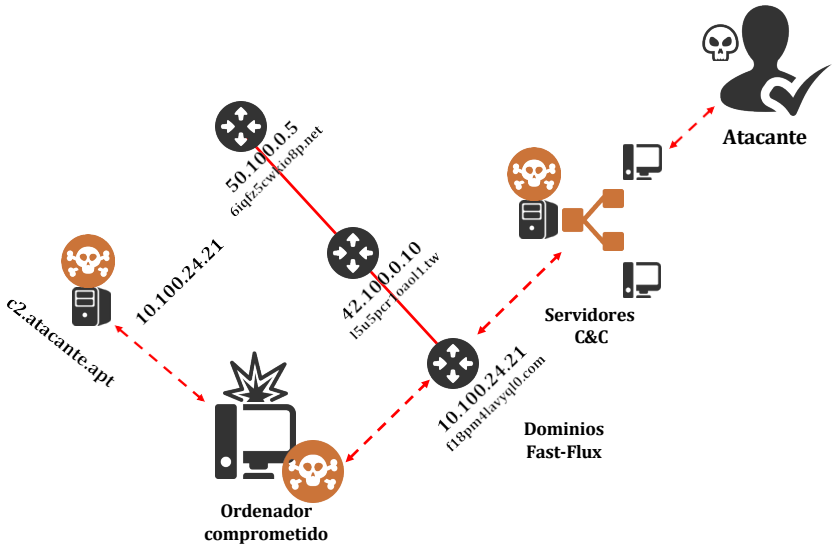


Figura 6.6 Ejemplo del funcionamiento de los servidores de C&C utilizando *fast-flux*.

Con el uso del análisis heurístico podemos obtener los parámetros necesarios para el funcionamiento del módulo de detección de URL; estos parámetros estarán basados en reglas para identificar si una URL es potencialmente maliciosa, para luego detectar sitios web maliciosos. En este caso, se necesitará un conjunto de datos previamente etiquetado para extraer las características más relevantes como las características léxicas basadas en contenidos web y en la popularidad del sitio. Se pueden extraer diferentes componentes de la URL, como la dirección, el nombre del ordenador, la ruta, el recuento de componentes HTML o los identificadores únicos.

Dado que la identificación de URL maliciosas es un problema de clasificación, el aprendizaje automático proporciona diferentes algoritmos para resolver este tipo de problema, como *logistic regression*, *decision tree* o *support vector machine*.

6.3.3. Módulo de detección de anomalías

Para cumplir su objetivo, un ataque de APT puede comprometer una red informática de diversas formas, se pueden mencionar la escalada de privilegios, la obtención de credenciales, la utilización de malware o los ataques de denegación de servicio.

El módulo de detección se utiliza para detectar cualquier anomalía dentro de la red de la organización, con el fin de detener los movimientos laterales u otras etapas descritas en el ciclo de vida propuesto.

Consecuentemente, para detectar anomalías de la red, se deben identificar patrones maliciosos que no corresponden al comportamiento habitual en la red; no solo basta con implementar herramientas de detección de intrusos, si no que se trata de buscar soluciones con otro tipo de técnicas.

Además de la detección de patrones anómalos en la red, se puede obtener otro tipo de detección con este módulo, como el movimiento lateral en un ataque de APT; ya que, como se ha descrito en la Sección 6.2, el movimiento lateral forma parte fundamental del ciclo de vida de un ataque de APT.

Teniendo en cuenta que la propagación un malware puede dejar un rastro en el tráfico de la red y el gran volumen de datos que se pueden generar a lo largo de una jornada laboral, los algoritmos de ML como *k*-means, *k*-NN o NB podrían ser de utilidad para analizar el tráfico anómalo en la red.

6.4. Escenarios de implementación

La implementación del modelo para la detección de ataques de APT propuesto en este estudio comienza con la configuración de un entorno controlado para la creación de un conjunto de datos que, posteriormente, será clasificado de acuerdo a las etapas del ciclo de vida propuesto y, finalmente, los módulos de detección generarán alertas

que serán analizadas por los administradores de red, como se muestra en la Figura 6.7; esta implementación tiene una serie de requisitos que se pueden dividir en requisitos de los datos o lógicos y requisitos en la infraestructura o físicos.

El requisito a nivel lógico es el más importante, en este caso los datos deben ser recopilados utilizando información crítica que se obtiene a lo largo de un ataque de APT, incluyendo los diferentes tipos de TTP y el flujo de datos que corresponden al tráfico normal de una red.

El segundo de estos requisitos es la infraestructura, compuesta por los equipos informáticos y los programas que permiten recopilar, analizar y generar alertas de manera temprana en el momento en que se está ejecutando un ataque de APT; a continuación, se detallan estos dos requisitos.

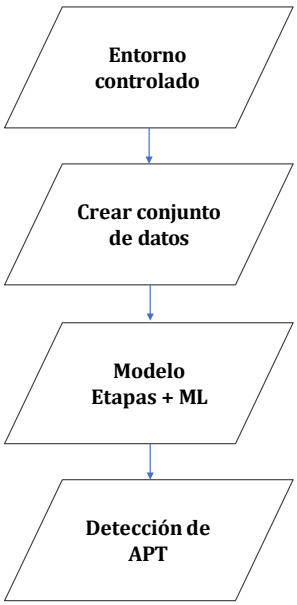


Figura 6.7 Esquema del escenario de implementación.

6.4.1. Requisitos de los datos

En este escenario de implementación, los datos están compuestos por los elementos clave que contienen información crítica de un ataque de APT, tales como las comunicaciones entre el ordenador comprometido y los servidores de comando y control, las anomalías presentadas en la red y los correos maliciosos; generalmente, las TTP que se utilizan en un ataque pueden ser identificadas mediante estos elementos.

En la actualidad, existe una amplia variedad de conjuntos de datos que recopilan información sobre distintos ataques a una red, muchos de estos conjuntos son elaborados para un trabajo específico y no son de carácter público; existen otra serie de datos de carácter público que están basados en el comportamiento del malware en la red y, otros de uso específico que contienen o identifican un ataque de APT.

Recientemente, se ha publicado un estudio que presenta un conjunto de datos que trata de brindar una solución a este tema; en el trabajo de Tork Ladani [50] se propone la utilización de un conjunto de datos semi-real a partir de un conjunto de datos libre de ataques de APT y otro conjunto de datos extraído de diferentes escenarios de ataques de APT del mundo real.

El resultado del trabajo de Ladani es un conjunto de datos con siete campos: fecha y hora de la alerta, riesgo de la alerta (clasificado en bajo, medio y alto), dirección de origen y puerto de origen (dirección y puerto desde donde se han enviado los datos), dirección de destino y puerto de destino (dirección y puerto donde se han recibido los datos), y la etapa del ciclo de vida (utiliza el modelo de cuatro etapas IKC).

Algunas de las ventajas que ofrece este conjunto de datos, es que ha sido publicado recientemente y es un conjunto de datos etiquetado y basado en el tráfico de red; además, contiene una gran cantidad de registros que clasifican los ataques de APT por riesgo y por etapa del ciclo de vida en la que tiene lugar el ataque.

El conjunto de datos de Ladani presenta como desventaja la clasificación que hace del riesgo de los ataques, ya que solo se considera riesgo bajo, medio y alto, lo que puede generar confusión debido a que una fase del ciclo puede tener estos tres niveles de riesgo a la vez.

No obstante, este conjunto de datos podría ser mejorado añadiendo algunas características, como el etiquetado de las alertas por TTP, el etiquetado del tráfico normal o identificando el inicio y fin de la ejecución de las TTP.

El estudio propuesto en [62] presenta como resultado la construcción de un conjunto de datos específico para la detección de ataques de APT, llamado DAPT2020 (*Dataset for Advanced Persistent Threats*); los autores se han basado en la idea de que los conjuntos de datos actuales son muy genéricos, presentan muchas limitaciones y no se hace la correcta diferenciación entre los comportamientos normal y anómalo, que son importantes para distinguir un ataque de APT.

Entre las principales ventajas del conjunto DAPT2020 cabe destacar que proporciona datos que están disponibles en un repositorio de uso público y, además, dicho conjunto de datos tiene en cuenta varios aspectos reales sobre el comportamiento de un ataque de APT. También se han realizado distintas comparaciones y se han contrastado los resultados con respecto a otros conjuntos de datos específicos para la detección de intrusos. Finalmente, en este trabajo se ha considerado el uso del aprendizaje automático semi-supervisado, para construir una aproximación del comportamiento de una red e identificar diferentes anomalías, que son utilizadas para detectar un ataque de APT en distintas etapas del ciclo de vida.

El conjunto de datos DAPT2020 está especialmente diseñado para la detección de anomalías en la red, lo que supone una desventaja, ya que la ejecución de un ataque de APT no siempre se realiza a través de la red; existen casos en los que no se realizan conexiones a los servidores de los atacantes por largos periodos de tiempo, debido a que

la amenaza se instala en un ordenador que recopila toda la información. Esto podría mejorarse incluyendo otros tipos de métodos de detección de ataques de APT.

Un gran número de muestras de malware de diferentes grupos de APT se encuentran alojadas en una plataforma de desarrollo, que a su vez es un repositorio de uso público [21]; estas muestras, han sido utilizadas para generar un conjunto de datos, que busca evaluar los diferentes técnicas de ML para la atribución de ataques de APT.

El grupo de muestras de malware está formado por aproximadamente 3.500 muestras que han sido relacionadas con al menos 12 grupos de APT; las características del conjunto de datos están agrupadas por país, grupo de APT, familia, número de solicitud de descarga y número de veces que ha sido descargado.

Además, las muestras han sido nombradas y agrupadas de acuerdo al grupo de APT al que pertenecen, a través de su valor *hash* SHA-256, que resulta ser un identificador único para cada muestra de malware, que coincide con los informes publicados por distintos proveedores de seguridad.

Como ventajas de este grupo de muestras de malware, cabe destacar que permite analizar el malware de manera individual, estudiar las características que comparten diferentes grupos APT y, conocer cómo operan estos actores. Es importante, que al analizar muestras de malware se configure y verifique el entorno de prueba controlado, para evitar que el malware se propague deliberadamente. No obstante, la desventaja es que no se especifican los diferentes algoritmos de aprendizaje automático que se utilizan para la atribución de ataques de APT.

Los conjuntos de datos suelen estar relacionados con la detección de anomalías a partir del tráfico de red; además, se pueden crear otros conjuntos de datos que permitan la detección de malware a partir de las conexiones URL y la identificación de *spear-phishing*.

El principal problema de obtener datos para evaluar y analizar los ataques de APT, mediante el uso de ML es la falta de disponibilidad de conjuntos de datos especializados, ya que normalmente las muestras de malware se encuentran disponibles solo para los analistas SOC, o en repositorios de la red oscura; esto se debe a su alta volatilidad y al nivel de propagación que tienen en los sistemas no seguros, o que no cuentan con las medidas necesarias para realizar un análisis de manera segura.

El proceso de creación de este tipo de conjuntos busca que los datos sean etiquetados por tipo de ataque, esto conlleva un arduo trabajo debido a la cantidad de registros que se pueden generar; consecuentemente, para la creación de un conjunto de datos de entrenamiento, es necesario la ejecución de muestras de malware de ataques de APT dentro de un espacio controlado o *sandbox*, con el objetivo de recopilar los datos necesarios que serán utilizados para identificar patrones de comportamiento; esto debe llevarse a cabo utilizando sistemas de monitorización de seguridad.

El preprocesamiento de los datos juega un papel importante en la creación de cualquier conjunto de datos, más aún en un conjunto para detectar los ataques de APT, ya que si no se normaliza la información obtenida o no está debidamente etiquetada ni depurada, podrían generarse problemas de interpretación por parte de los algoritmos de ML; esto afecta la precisión de la detección, dando como resultado valores no concluyentes.

Cuando un conjunto de datos ha sido bien etiquetado, se podrán obtener mejores resultados y disminuir el tiempo del preprocesamiento; esto aportaría diferentes valores y resultados a través de los algoritmos de ML. Además, se podrá conocer el comportamiento de las TTP mediante las diversas métricas que se utilicen para evaluar el modelo.

Generalmente, el fichero que se genera a partir del conjunto de datos es un fichero de texto plano sin formato, lo que permite que los datos sean compatibles con la

mayoría de los software especializados para el manejo de grandes cantidades de datos, en diferentes entornos y sistemas operativos.

Finalmente, los datos obtenidos aportarán las características necesarias para etiquetar el tipo de ataque y generar un nuevo conjunto de datos.

6.4.2. Requisitos de la infraestructura

La infraestructura tecnológica para la implementación del modelo propuesto en este trabajo debe considerar el hardware y el software necesarios para alcanzar los siguientes objetivos: recopilar datos a partir de las muestras de malware para ataques de APT, crear y actualizar los conjuntos de datos y, ejecutar el modelo propuesto para la detección de los ataques de APT, a través de los algoritmos de aprendizaje automático. Se describen a continuación las características de una infraestructura basada en estos objetivos:

- Para obtener los datos, se necesita una infraestructura con un entorno semi-real controlado, donde los equipos tengan la mayor similitud posible con una red informática dentro de una organización.
- Los equipos deben simular tener en funcionamiento los servicios que puedan estar disponibles en la red, para luego almacenar las acciones que se ejecutan en dicho entorno. Estas acciones serán almacenadas en un fichero *log* de registro; este fichero debe poder ser analizado por los administradores de la red para buscar posibles anomalías.
- El entorno controlado debe estar situado dentro de la red principal de la organización sin afectar a la funcionalidad de la red; además, este entorno deberá simular ser poco seguro para convertirse en un señuelo para los atacantes.

- La información a la que podría acceder el atacante debe ser ficticia y similar a los datos que maneja la organización dentro de red informática.

Para lograr los objetivos de la infraestructura, se pueden utilizar diferentes herramientas como los *honeypots* y los contenedores *docker* ; estas herramientas utilizan pocos recursos computacionales, por lo que pueden ser ejecutados varios servicios virtuales en un mismo ordenador físico.

Los *honeypots* son un tipo de sistemas utilizados para reforzar el nivel de detección de las amenazas, siendo muy utilizados en los últimos años [6, 41]. De igual forma, los contenedores *docker* permiten separar las aplicaciones empaquetándolas con todas las dependencias y librerías necesarias para su ejecución [16, 25]. Estas herramientas permitirán captar los datos necesarios para la creación de los conjuntos de datos necesarios a partir de muestras de malware.

Posteriormente, una vez completados estos objetivos de la infraestructura se podrán realizar las pruebas para verificar el funcionamiento del modelo propuesto para la detección temprana de un ataque de APT; por lo que es necesario, la configuración de un entorno de pruebas en el que se simule la red informática de una organización; aunque lo ideal sería un entorno de pruebas físico controlado.

Consecuentemente, los módulos de detección podrían iniciar su funcionamiento cuando un correo electrónico es recibido por un usuario de la red, el primer módulo verificaría si el correo es fraudulento, a través de la búsqueda de patrones léxicos etiquetados para conocer la autenticidad del correo.

Si no se ha generado una alerta en este momento, la siguiente verificación se haría a través del módulo de detección de URL, el cual busca patrones etiquetados que den a conocer si una URL es utilizada para establecer una comunicación con los servidores C&C de los atacantes.

Si en este caso tampoco se ha generado una alerta, la última verificación se haría a través del módulo de detección de anomalías; este módulo identifica las anomalías basadas en red de las TTP en las etapas activas y recurrente del ciclo de vida.

Las alertas generadas por los módulos deben ser analizadas para comprobar que verdaderamente se trata de un ataque de APT, esto se puede realizar a través de un sistema semi-autónomo, donde un algoritmo de red neuronal junto a un árbol de decisiones analiza las alertas, y comunica el resultado al administrador del sistema. Finalmente, es el administrador del sistema quien debe aplicar las políticas de seguridad de acuerdo al tipo de ataque que se identifique.

6.5. Análisis experimental

Se ha realizado un análisis experimental con la finalidad de analizar el nivel de predicción de diferentes algoritmos de aprendizaje automático supervisado, para la detección de anomalías de red en los ataques de APT. En este caso, se han utilizado cinco algoritmos de ML para analizar un conjunto de datos diseñado para detectar ataques de APT.

Los algoritmos de ML que han sido seleccionados son *k*-NN, NB, LR, DT y ANN, debido a que han sido evaluados por otros autores para la detección de anomalías de red en ataques de malware con una alta precisión. Además, se ha empleado la herramienta KNIME para el análisis de los datos, que integra varios componentes denominados nodos, para el aprendizaje automático por medio de una interfaz gráfica.

El conjunto de datos utilizado en este experimento es DAPT2020, que está formado por datos etiquetados, para los que se ha definido un etiquetado múltiple organizado de acuerdo al tipo de TTP, y a su vez ha sido asociado a una etapa del ciclo de vida de un ataque APT; adicionalmente, este conjunto de datos cuenta con una etiqueta para el tráfico de red normal.

El experimento está organizado por tareas, la primera tarea comienza con el preprocesamiento de los datos, con la finalidad de obtener un conjunto de datos normalizado; en la siguiente tarea se realiza un análisis exploratorio de los datos, para determinar las variables que serán utilizadas posteriormente; en la última tarea, se ejecutan los algoritmos de ML previamente seleccionados para comprobar su capacidad de clasificación para el conjunto de datos DAPT2020. A continuación, se describen cada una de estas tareas y, por último, se presentan los resultados obtenidos.

6.5.1. Tarea 1: Preprocesamiento

El preprocesamiento del conjunto de datos DAPT2020 se ha realizado con el fin obtener un conjunto de datos normalizado, sin datos nulos y erróneos, que puedan afectar a los resultados durante la tarea de clasificación donde se utilizan los algoritmos de ML. Para esto, se han ejecutado los siguientes pasos detallados en la Figura 6.8:

1. Cargar los datos de los 24 ficheros que componen el conjunto de datos DAPT2020; en este caso, se ha utilizado el nodo "*CSV Reader*" para establecer la ruta de origen del fichero en formato csv y leer cada uno de ellos como una tabla en el espacio de trabajo de KNIME.

2. Concatenación de los datos de los ficheros que han sido previamente cargados; para lo cual se ha utilizado el nodo "*Concatenate*", donde se han organizado los ficheros en dos grupos de doce ficheros cada uno para facilitar el manejo de los datos, posteriormente son concatenados estos dos grupos para formar un solo conjunto de datos con el que se seguirá trabajando.

3. Se inicia la limpieza de los datos con la eliminación de la primera columna del conjunto de datos, ya que la información que contenía estaba duplicada, es decir,

que funcionaba como un identificador creado a partir de la concatenación de los datos de las siguientes cinco columnas.

4. Normalización o estandarización de los datos; es decir, establecimiento del mismo peso o importancia para cada una de las variables, con el fin de mejorar la precisión del modelo. En este caso, se ha utilizado el nodo "*Normalizer*".

5. Creación de varias reglas utilizando el nodo "*Rule Engine*", tanto en la columna de los tipos de ataques como en las etapas del ciclo de vida, donde se han modificado las etiquetas que hacen referencia al tráfico benigno y normal para unificar el etiquetado. Además, se ha creado una nueva columna donde se ha asignado un etiquetado binario, uniendo las etiquetas de las diferentes TTP en una única etiqueta denominada anomalías, dejando la etiqueta de benigno para el tráfico normal.

6. Reemplazo de los datos perdidos utilizando el nodo "*Missing Value*"; los valores que se han asignado para reemplazar los valores nulos son los siguientes: en el caso de los datos tipo cadena (*string*) se asignó el valor más frecuente; para los enteros (*integer*) se ha establecido un valor fijo en cero, puesto que las columnas de este tipo de dato se corresponden con los puertos de comunicación de red; y para los datos tipo doble (*double*) se reemplazó con el valor promedio.

7. Guardado del conjunto de datos procesado con el nodo "*CSV Writer*".

El volumen del conjunto de datos procesado está formado por un total de 322.072 filas y 85 columnas, donde las primeras seis columnas muestran la información del origen y destino de los datos y la fecha y hora del evento. Las siguientes columnas muestran información relacionada con paquete de datos y las últimas tres columnas corresponden al etiquetado.

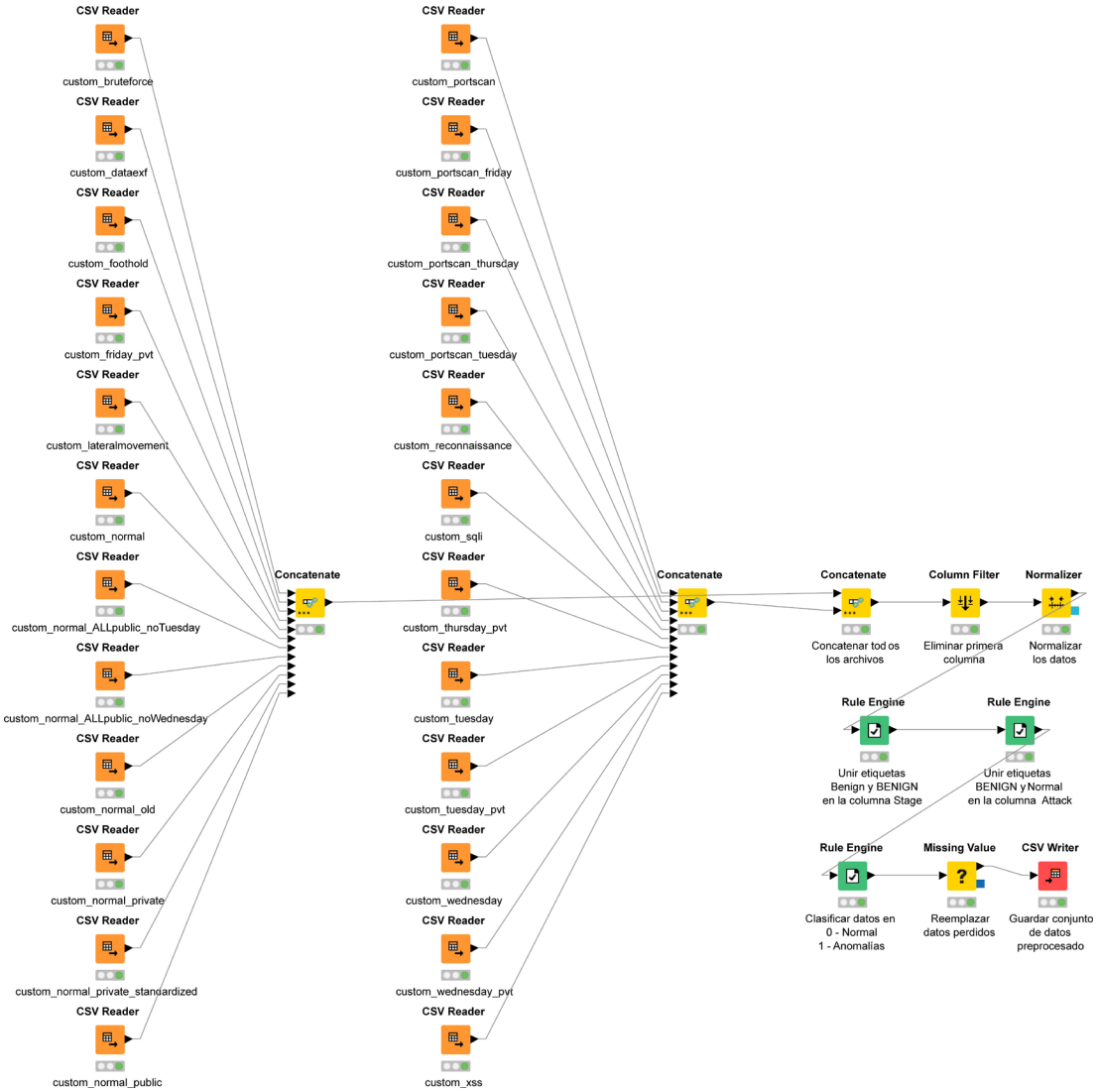


Figura 6.8 Preprocesamiento del conjunto de datos DAPT2020.

6.5.2. Tarea 2: Análisis exploratorio

Durante el análisis exploratorio se ha verificado la correlación lineal de las variables, con la finalidad de eliminar aquellas variables que no presenten una correlación con otras variables y, por lo tanto no son significativas para el análisis de los datos. Se ha utilizado el nodo "*Linear Correlation*" para esta implementación; este nodo ofrece un resultado detallado de la matriz de correlación que se muestra en la Figura 6.9.

Consecuentemente, de las 85 variables iniciales se eliminaron 12 variables que no presentaban correlación con el resto de variables. Se ha podido observar, en la matriz de correlación, que las variables que presentan una correlación fuerte e inversa, se encuentran identificadas con un tono rojizo; la correlación fuerte y directa se distingue con el tono azulado; y la correlación débil se representa en color blanco.

Por lo tanto, se ha obtenido una matriz de correlación con diferentes correlaciones; en el caso de las variables "*Fwd Header Length*" (es decir, el total de bytes utilizados para las cabeceras en la dirección de envió) y "*Total Fwd Packets*" (total de paquetes en la dirección de envío) tienen una correlación fuerte e inversa; las variables "*Flow Duration*" (duración del flujo en microsegundos) y "*Down/Up Ratio*" (porcentaje de carga y descarga) presentan una correlación fuerte y directa; sin embargo, las variables "*Flow Bytes/s*" (número de flujo de bytes por segundo) y "*Bwd Packets/s*" (número de paquetes de retorno por segundo) muestran una correlación débil.

6.5.3. Tarea 3: Clasificación

El objetivo de esta tarea ha sido analizar el conjunto de datos DAPT2020 a través de diferentes algoritmos de ML, y determinar cuáles de estos algoritmos presentan mayor precisión en la detección de anomalías de red en ataques de APT.

La partición de los datos que se ha asignado a los algoritmos de clasificación ha sido de 70 % para los datos de entrenamiento y de 30 % para los datos de prueba; ya

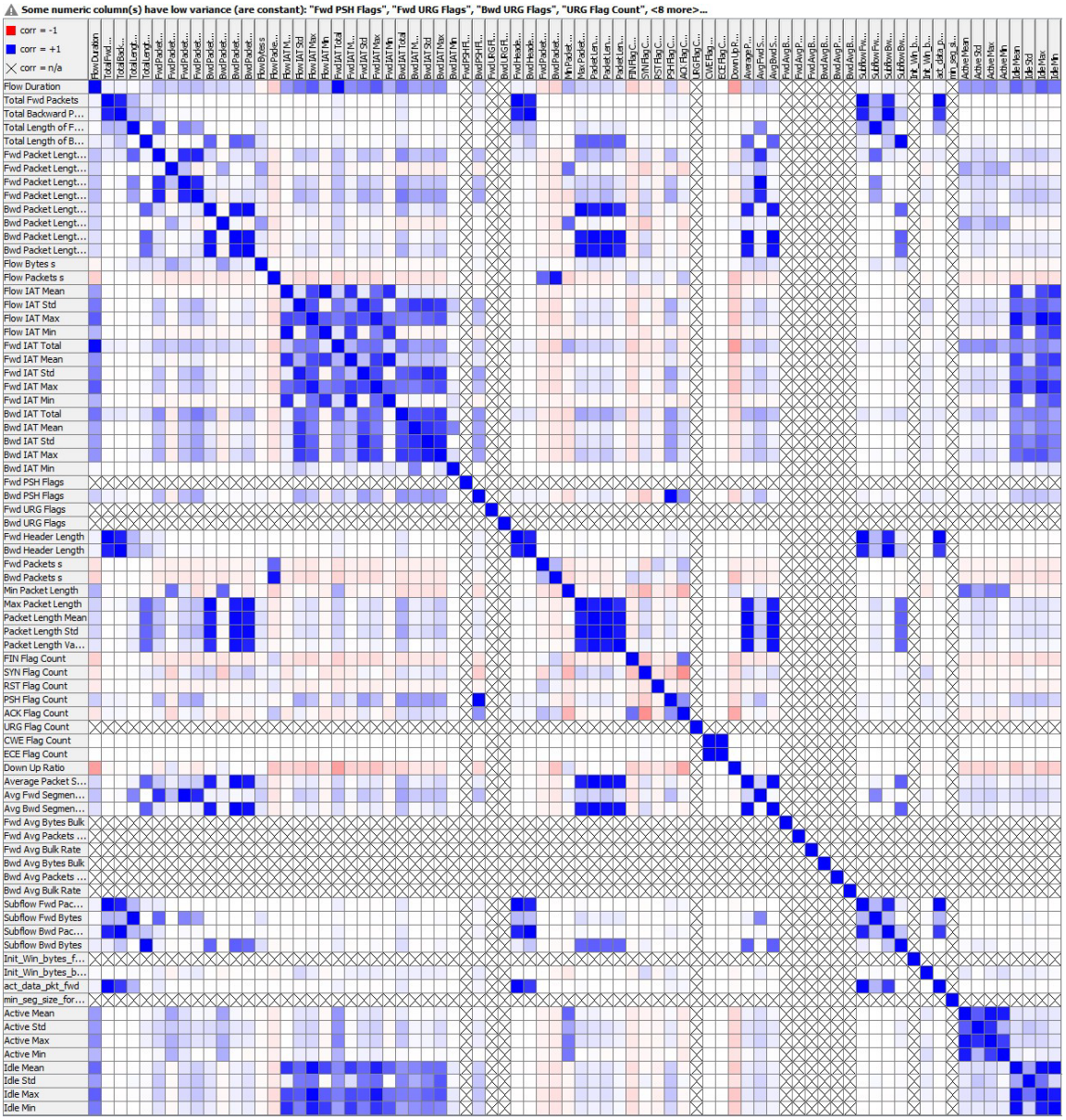


Figura 6.9 Correlación lineal del conjunto de datos DAPT2020.

que estos valores son considerados como una buena proporción entrenamiento/pruebas en aprendizaje automático.

El etiquetado múltiple consiste en la agrupación de los datos en varias clases; en este caso, se cuenta con nueve clases que corresponden a los tipos de TTP utilizados en un ataque de APT (*Account Bruteforce*, *Data Exfiltration*, *SQL Injection*, *Directory Bruteforce*, *Account Discovery*, *CSRF*, *Malware Download* y *Network Scan*) (en adelante, estas TTP serán llamadas tipos de ataque), y el tráfico normal (*Benign*).

Otra forma de etiquetado, conocido como etiquetado binario, consiste en asignar a todas las TTP una etiqueta de tráfico con anomalías (*Anomaly*) y otra etiqueta diferente para el tráfico normal (*Benign*).

En este experimento se han utilizado ambos tipos de etiquetado para el análisis con los diferentes algoritmos de ML.

6.5.3.1. Etiquetado múltiple

La implementación de los algoritmos de ML utilizando el etiquetado múltiple se ha llevado a cabo siguiendo los pasos que se describen en Figura 6.10:

1. Carga del conjunto de datos procesado.
2. Filtrado de aquellas columnas que no son significativas para el modelo.
3. Partición de los datos en entrenamiento y prueba.
4. Entrenamiento y predicción con los diferentes algoritmos de ML utilizando los siguientes parámetros:
    - ***k*-NN**: se ha utilizado el nodo "*k Nearest Neighboor*", que realiza los procesos tanto de entrenamiento como de predicción. Además, se ha definido el número de vecinos cercanos $k = 5$, después de haber realizado varias pruebas con

Figura 6.10 Etiquetado múltiple del conjunto de datos DAPT2020.

diferentes valores para k, siendo el valor 5 el que mejor resultados ha obtenido.

- **NB**: se ha utilizado el nodo "*Naive Bayes Learner*" para realizar el proceso de entrenamiento. Para la clasificación se ha seleccionado la columna de los tipos de ataque y el resto de los parámetros se han configurado con los valores predefinidos; es decir: la probabilidad tiene el valor 0, 0001, la desviación estándar mínima es 0, 0001, el umbral de la desviación estándar es 0 y el número máximo del valor nominal único por atributo es 20.

  Además, para la predicción se ha utilizado el nodo "*Naive Bayes Predictor*"; para este nodo y los siguientes nodos de predicción de los algoritmos, se ha activado la opción de cambiar el nombre de la columna donde se almacenan los datos de la predicción.

- **LR**: en este caso, se ha utilizado el nodo "*Logistic Regresion Learner*". Se ha seleccionado en la columna de los tipos de ataque la categoría de referencia *Benign*, se ha utilizado el solucionador gradiente medio estocástico (SGD, *stochastic average gradient*); se ha asignado el número máximo de iteraciones (*epochs*) en 1.000 y el parámetro $\epsilon = \cdot 10^{-5}$ que determina la convergencia del modelo; la estrategia del ritmo de aprendizaje es fija con un intervalo de 0, 1 y se ha definido la distribución de Gauss con una varianza de 0, 1. También, se ha utilizado el nodo "*Logistic Regresion Predictor*" para el proceso de predicción.

- **DT**: en el entrenamiento de los datos se ha utilizado el nodo "*Decision Tree Learner*". Para la columna de clasificación se ha seleccionado la columna de los tipos de ataque, la medida de calidad es el índice de Gini (mide el grado de desigualdad en una distribución), no se ha seleccionado ningún método

de poda, el número mínimo de registros por nodo es 2 y el número de hilos es 8.

Para el proceso de predicción, se ha utilizado el nodo "*Decision Tree Predictor*", el parámetro del número máximo de patrones almacenados por hilo debe ser igual al total de filas de los datos de prueba.

- **ANN**: el proceso de entrenamiento se ha realizado utilizando el nodo "*RProp MLP Learner*", donde el número máximo de iteraciones es 5, el número de capas ocultas es 1, el número de neuronas ocultas por capa es 10 y para la clasificación se ha definido la columna de los tipos de ataques. Además, se ha utilizado el nodo "*MultiLayerPerceptron Predictor*" para el proceso de predicción.

5. La puntuación de cada modelo se obtiene utilizando el nodo "*Scorer*", donde se define la primera columna para el tipo de ataque y la segunda columna para la predicción de cada modelo; como resultado se obtiene una matriz de confusión donde se puede observar el rendimiento de cada algoritmo de ML supervisado y, las estadísticas de precisión de cada uno de ellos.

6.5.3.2. Etiquetado binario

En el caso del etiquetado binario, se han seguido los mismos pasos que en el etiquetado múltiple para la implementación de los algoritmos de ML, utilizando los mismo parámetros, tal como se muestra en la Figura 6.11, con el fin de comparar los resultados del etiquetado múltiple con el etiquetado binario.

Además de esto, ha sido necesario activar la opción para añadir las columnas que contienen la distribución de las clases normalizadas, es decir, una columna con la probabilidad de que los datos de una fila sean considerados normales, y otra columna con la probabilidad de que los datos sean considerados como anomalía.

Posteriormente, se ha realizado el proceso para unir dichas columnas de las probabilidades de cada modelo junto a la columna que contiene el resultado de la predicción en formato cadena; cada uno de estos resultados debe coincidir con la fila que ha sido evaluada con cada uno de los algoritmos de ML. Otro valor que se ha añadido es la columna del etiquetado binario que contiene el tipo de clasificación original; para todo esto se ha utilizado en tres ocasiones el nodo "*Joiner*", debido a que las uniones se realizan entre dos tablas.

Estos valores probabilísticos serán utilizados para generar una gráfica comparativa; sin embargo, es necesario realizar un paso nuevo para asignar un nombre a las columnas que contengan las probabilidades de tipo normal para evitar confusiones, ya que el nombre por defecto de estas columnas es muy similar entre ellos. En esta ocasión, se ha utilizado el nodo "*Column Rename*".

Finalmente, se ha generado la curva ROC para cada uno de los algoritmos de ML; esta curva ayuda a determinar el rendimiento de los algoritmos de ML para medir y comparar los resultados de este experimento de manera gráfica.

Los parámetros necesarios para configurar el nodo "*ROC Curve*" son los siguientes: en primera instancia se debe escoger la columna que contiene el etiquetado binario, posteriormente, seleccionar la clase que será utilizada como positiva para la curva, que en este caso es la clase normal. Además, para calcular el espacio debajo de la curva de cada algoritmo, se deben agregar las columnas con las probabilidades asociadas a la misma clase.

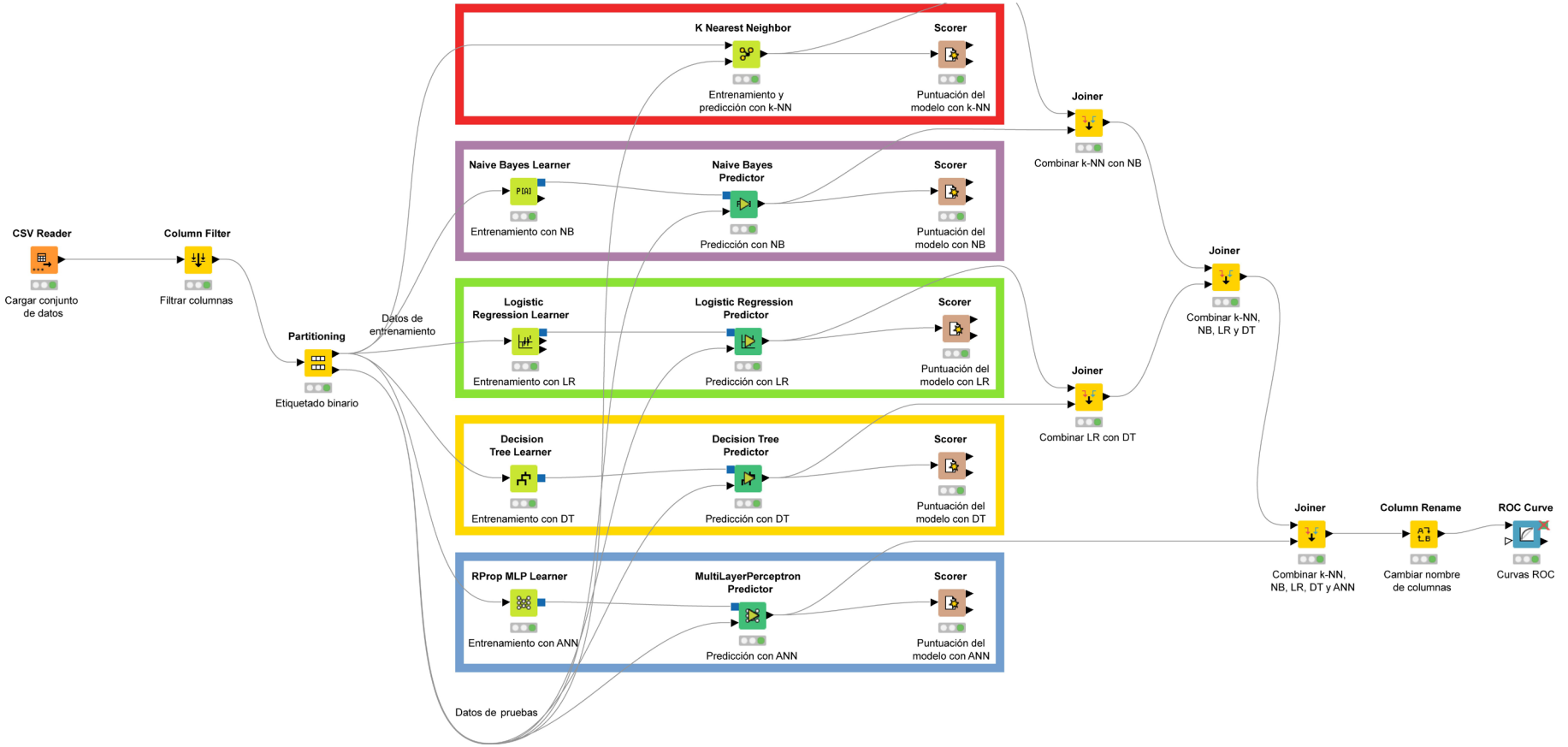


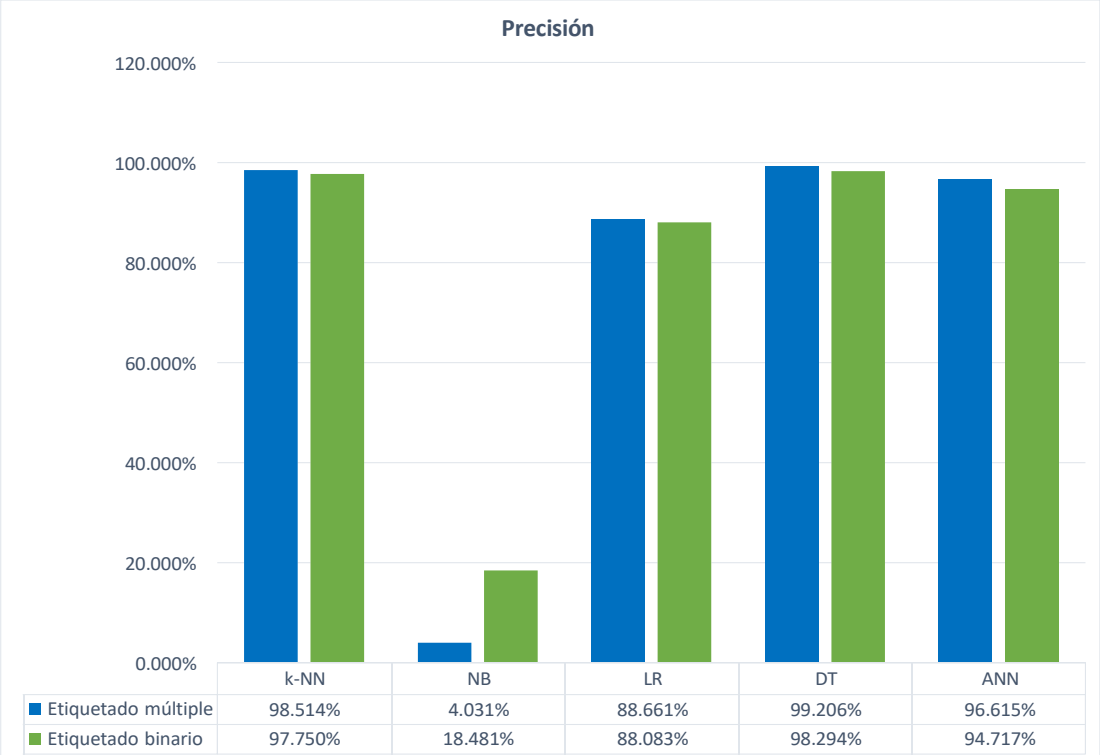
Figura 6.11 Etiquetado binario del conjunto de datos DAPT2020.

6.5.4. Resultados

En esta Sección, se realiza un análisis descriptivo de los resultados obtenidos en cada uno de los modelos de clasificación a partir de la matriz de confusión, las curvas ROC y las estadísticas de precisión.

6.5.4.1. Matriz de confusión

En la Figura 6.12 se muestran los resultados obtenidos de la precisión de los algoritmos de ML, tanto en el etiquetado múltiple como en el etiquetado binario. Estos resultados se han obtenido a partir de la matriz de confusión.



	k-NN	NB	LR	DT	ANN
Etiquetado múltiple	98.514%	4.031%	88.661%	99.206%	96.615%
Etiquetado binario	97.750%	18.481%	88.083%	98.294%	94.717%

Figura 6.12 Resultados de la precisión de los algoritmos de ML.

Los algoritmos DT, *k*-NN y ANN han dado buenos resultados tanto en el etiquetado múltiple como en el binario, con una precisión superior al 94 % para la detección de anomalías en la red. En el caso de ANN, se podrían ajustar los parámetros de las capas intermedias encargadas de procesar la información, es decir, los parámetros del número de capas ocultas y el número de neuronas ocultas por capa para tratar de aumentar la precisión del modelo.

Con el algoritmo LR se ha obtenido una precisión buena, ya que en ambos etiquetados el resultado es cercano al 88 %; en este caso, estos resultados se han obtenido después de realizar un exhaustivo ajuste de los parámetros del algoritmo, como la configuración del número de *epochs*, donde se han realizado pruebas para determinar el número adecuado necesario para el análisis de este conjunto de datos, ya que valores como 100, 200 o 500 *epochs* daban resultados no concluyentes.

El algoritmo NB ha dado resultados no favorables, ya que en el etiquetado múltiple la precisión es muy baja y en el etiquetado binario la precisión es del 18 %; por lo que no sería recomendable utilizar este algoritmo para la detección de anomalías de red.

Por último, se utilizó el algoritmo SVM para realizar este experimento, ya que en otros estudios presenta altos porcentajes de precisión; durante las tareas de este experimento, este algoritmo se ejecutó durante aproximadamente 48 horas para el entrenamiento del modelo. A pesar de utilizar diferentes configuraciones de parámetros y extensiones de otras plataformas como Weka y Spark para KNIME, no se logró que el aprendizaje culminara correctamente.

6.5.4.2. Curvas ROC

Las curvas ROC correspondientes a los algoritmos de ML de este experimento se pueden observar en la Figura 6.13. El área bajo la curva (AUC) es de 0, 98 para el algoritmo *k*-NN, en DT se obtiene el AUC = 0, 986, y en ANN el AUC = 0, 955, por lo

que se puede decir que son curvas muy buenas con un alto rendimiento para cada uno de estos algoritmos de clasificación.

Por otro lado, el algoritmo LR tiene un AUC = 0, 859 y AUC = 0, 727 para NB; por lo tanto, se puede concluir que son curvas regulares donde la clasificación es menos precisa.

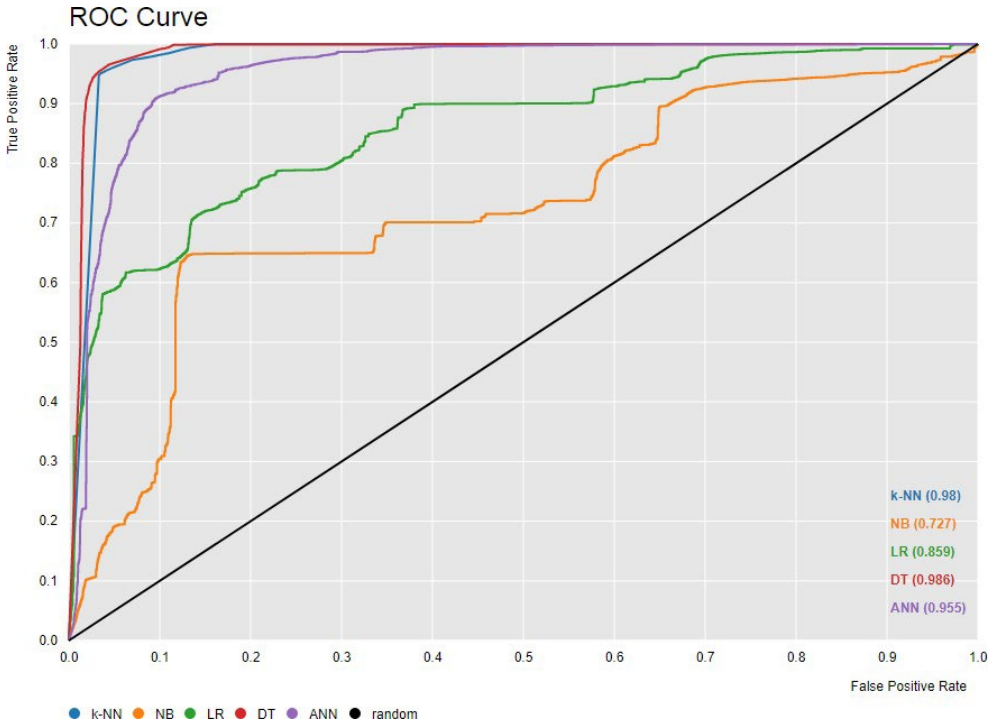


Figura 6.13 Curvas ROC.

6.5.4.3. Estadísticas de precisión

Las estadísticas de precisión brindan información que permiten evaluar la predicción del algoritmo; una de estas estadísticas es el valor de F_1, que permite calcular la precisión media utilizando los valores de exactitud y sensibilidad del modelo, permitiendo conocer el rendimiento de los algoritmos sobre el conjunto de datos.

En la Tabla 6.1 se pueden observar los resultados de las estadísticas de precisión de los algoritmos de ML; en este caso, la predicción media F_1 de los tipos de ataque para los algoritmos de ML se define como buena cuando $F_1 \geq 90$ %, aceptable cuando $75\ \% \leq F_1 < 90\ \%$ y se considera baja cuando $F_1 < 75\ \%$.

Consecuentemente, en el caso del etiquetado múltiple, el valor de F_1 para la predicción de tráfico normal es buena para los algoritmos de *k*-NN, LR, DT y ANN; sin embargo, para el algoritmo de NB, el valor de F_1 es muy bajo.

Para cada uno de los tipos de ataques etiquetados y dependiendo del algoritmo de ML utilizado, se ha obtenido un valor de F_1 diferente. En el caso del algoritmo *k*-NN, con dos tipos de ataques se obtiene una predicción buena, otro de estos tipos de ataque tiene una predicción aceptable y los otros cinco tipos de ataque tienen una predicción baja.

Además, el algoritmo DT ha obtenido resultados similares a *k*-NN como una predicción buena para dos tipos de ataque, una predicción aceptable para otros dos tipos de ataque y una predicción baja para cuatro tipos de ataque.

El algoritmo ANN ha obtenido una predicción aceptable para dos tipos de ataque y una predicción baja para los otros seis tipos de ataque; sin embargo, para los algoritmos NB y LR, el valor de F_1 es bajo para todos los tipos tipos de ataques.

Por otro lado, los resultados del valor de F_1 en el etiquetado binario son diferentes, para los algoritmos *k*-NN y DT la predicción es buena tanto para el tráfico normal como para la predicción de anomalías; en el caso de ANN, la predicción de tráfico normal es buena y la de predicción de anomalías es aceptable; la predicción de tráfico normal en el algoritmo LR es buena, sin embargo, la predicción de anomalías es baja; por último, para NB la predicción es baja tanto para tráfico normal como para la predicción de anomalías.

Una vez analizados estos resultados, se puede concluir que los algoritmos *k*-NN y DT son los que mejores resultados han obtenido, cuando se observan y comparan las diferentes métricas de evaluación de estos algoritmos de ML; además, los resultados con un etiquetado binario son mucho más fiables, ya que pueden ser identificados todos los tipos de ataques, porque han sido agrupados como anomalías; en cambio en el etiquetado múltiple, es posible que algunos tipos de ataques que han sido etiquetados, puedan no ser identificados correctamente y, por tal motivo, el modelo no sería fiable. En cuanto a la curva ROC de estos algoritmos se ha observado un alto rendimiento, de manera que *k*-NN y DT podrían ser utilizados para la detección de ataques de APT. Los resultados del algoritmo ANN pueden considerarse como aceptables, ya que este algoritmo podría ser configurado con otros parámetros que permitan incrementar su nivel de predicción; de igual forma, en el etiquetado binario se han obtenido mejores resultados de predicción con respecto al etiquetado múltiple; en la curva ROC se ha observado un buen rendimiento del algoritmo. Por ello, este algoritmo no se descarta totalmente para realizar detecciones de ataques de APT.

Los algoritmos NB y LR han sidos descartados para la detección de ataques de APT, debido a que los resultados que se han obtenido tanto en precisión, rendimiento y predicción no son concluyentes; de hecho los valores obtenidos son muy bajos con respecto a los valores obtenidos de los otros algoritmos analizados.

Tabla 6.1 Estadísticas de precisión de los algoritmos de ML.

Algoritmos	Medidas	Etiquetado múltiple									Etiquetado binario	
		Benign	Account Bruteforce	Data Exfiltration	SQL Injection	Directory Bruteforce	Account Discovery	CSRF	Malware Download	Network Scan	Benign	Anomaly
k-NN	**Exactitud**	99,050 %	61,111 %	100,000 %	5,263 %	94,732 %	87,277 %	0,000 %	0,000 %	95,169 %	98,067 %	95,340 %
	Sensibilidad	99,480 %	45,205 %	60,000 %	1,613 %	95,434 %	55,382 %	0,000 %	0,000 %	93,813 %	99,378 %	86,661 %
	F_1-valor	99,264 %	51,969 %	75,000 %	2,469 %	95,082 %	67,764 %	0,000 %	0,000 %	94,486 %	98,718 %	90,794 %
NB	**Exactitud**	95,388 %	0,602 %	0,009 %	0,044 %	5,664 %	0,000 %	0,133 %	33,333 %	65,683 %	97,099 %	13,438 %
	Sensibilidad	1,382 %	23,288 %	60,000 %	8,065 %	48,878 %	0,000 %	75,000 %	100,000 %	3,337 %	6,713 %	98,634 %
	F_1-valor	2,725 %	1,174 %	0,019 %	0,088 %	10,152 %	0,000 %	0,266 %	50,000 %	6,351 %	12,558 %	23,653 %
LR	**Exactitud**	88,679 %	0,000 %	0,000 %	0,000 %	0,000 %	70,000 %	0,000 %	0,000 %	88,406 %	88,573 %	68,322 %
	Sensibilidad	99,912 %	0,000 %	0,000 %	0,000 %	0,000 %	0,992 %	0,000 %	0,000 %	8,005 %	99,122 %	12,902 %
	F_1-valor	93,961 %	0,000 %	0,000 %	0,000 %	0,000 %	1,955 %	0,000 %	0,000 %	14,681 %	93,551 %	21,705 %
DT	**Exactitud**	99,388 %	81,081 %	100,000 %	82,759 %	85,039 %	85,039 %	100,000 %	99,249 %	0,000 %	98,360 %	97,798 %
	Sensibilidad	99,927 %	82,192 %	60,000 %	38,710 %	97,893 %	30,595 %	50,000 %	99,100 %	0,000 %	99,707 %	88,674 %
	F_1-valor	99,657 %	81,633 %	75,000 %	52,747 %	97,541 %	45,000 %	66,667 %	99,174 %	0,000 %	99,029 %	93,013 %
ANN	**Exactitud**	97,385 %	73,333 %	89,505 %	90,304 %	0,000 %	0,000 %	0,000 %	0,000 %	0,000 %	95,324 %	89,021 %
	Sensibilidad	98,985 %	15,068 %	87,863 %	82,415 %	0,000 %	0,000 %	0,000 %	0,000 %	0,000 %	98,787 %	66,993 %
	F_1-valor	98,179 %	25,000 %	88,677 %	86,179 %	0,000 %	0,000 %	0,000 %	0,000 %	0,000 %	97,024 %	76,452 %

6.6. Ventajas y desventajas del modelo propuesto

El modelo propuesto para detectar ataques de APT presenta las siguientes ventajas: la definición de un ciclo de vida que explica de manera sencilla el funcionamiento de un ataque de APT y, además, se han identificado las TTP más utilizadas para cada etapa del ciclo de vida.

Consecuentemente, este ciclo de vida ha sido dividido en seis etapas que han sido clasificadas en pasivas, activas y recurrente, según el comportamiento de las TTP involucradas.

Cabe destacar que en este modelo se ha propuesto la creación de módulos de detección, que, mediante la utilización del aprendizaje automático, buscan detectar correos dirigidos, URL maliciosas y anomalías en la red.

Además, se ha planteado un escenario de implementación para realizar pruebas del modelo lo más reales posible; este escenario contempla desde la creación de un conjunto de datos a partir de muestras de malware, hasta la infraestructura de hardware y software.

La identificación de posibles ataques por etapas en este modelo, facilita la detección de los ataques de APT, ayudando a anticipar estos comportamientos anómalos en la red. Es importante recordar que cada organización debe incluir, en su plan de ciberseguridad, las políticas de seguridad que se adapten a su infraestructura, sin olvidar que los usuarios deben ser informados con frecuencia.

Como desventaja, se puede decir que los conjuntos de datos que se necesitan para estudiar estas amenazas, ya sea de manera simulada o en un entorno real controlado, actualmente se encuentran muy limitados, ya que la mayoría de estos datos son de uso exclusivo de los centros de operaciones de seguridad o de los centros de investigación.

Capítulo 7

Conclusiones

Las amenazas persistentes avanzadas fueron identificadas desde hace más de una década; a pesar de los grandes esfuerzos por detectar estas amenazas en tiempo real, aún se desconoce una forma eficiente de detección para contener la propagación de esta amenaza en una red informática. Al mismo tiempo, el panorama de los ciberataques sigue en constante evolución, por lo que los ataques de APT continúan teniendo éxito en su objetivo primordial, que consiste en obtener acceso a los datos críticos y sistemas de información de organizaciones gubernamentales y diferentes sectores industriales.

Un análisis profundo del comportamiento de un ataque de APT permite comprender e identificar el alcance y la intención de estas amenazas; sin embargo, el principal obstáculo de este análisis es la constante evolución de las tácticas, técnicas y procedimientos utilizados para llevar a cabo un ataque de APT, lo que se convierte en una de las principales limitaciones para la detección temprana.

Considerando estas ideas, en este trabajo se ha propuesto un modelo que busca brindar una solución eficiente que disminuya el tiempo de detección, lo que generaría un impacto positivo en el coste de los recursos informáticos y personal técnico a la hora de enfrentarse a una amenaza, como es el caso de un ataque de APT.

Inicialmente, se ha realizado un análisis exhaustivo de los enfoques propuestos por diferentes autores para la detección de ataques de APT; en este sentido, se ha observado que estos enfoques se basan en ciclos de vida de los ataques de APT, además, en ciertos enfoques se identifican algunas de las TTP que están relacionadas a las etapas del ciclo de vida. Sin embargo, la detección de los ataques se lleva a cabo únicamente en una de las etapas del ciclo de vida, donde no siempre corresponde a la primera etapa del ciclo. Por otro lado, los enfoques que utilizan aprendizaje automático lo hacen para buscar una TTP previamente seleccionada en una etapa específica del ciclo de vida, lo que puede limitar una detección temprana del ataque.

Como resultado de este trabajo, se ha propuesto un modelo basado en un ciclo de vida que ha sido cuidadosamente elaborado para describir de forma general el comportamiento de un ataque de APT; este ciclo de vida ha sido organizado en etapas activas, pasivas y recurrente, facilitando la clasificación de las TTP en cada una de estas etapa del ciclo de vida, lo que ha permitido determinar los módulos de detección; estos módulos utilizan algoritmos de aprendizaje automático para clasificar los datos y generar alertas de ataque en el menor tiempo posible.

En el caso de las TTP, uno de los principales problemas es su detección en todas las etapas del ciclo de vida, ya que es posible que algunas de estas TTP logren evadir las medidas de seguridad de algunas etapas y por ende, el ataque avance a la siguiente etapa del ciclo de vida; se debe considerar que algunas de estas TTP se llevan a cabo de manera sigilosa, otras tratan de simular actividad normal en la red, o pueden hacerse notar en un periodo de tiempo definido.

En cuanto al número de módulos de detección se puede decir que se ha buscado la forma de agrupar las características de las TTP en la menor cantidad posible de módulos; se ha identificado que una gran parte de las TTP utilizan conexiones a servidores de comando y control empleando los diferentes tipos de direcciones URL

existentes (como IPv4, IPv6, dominios DNS y enlaces .onion) o técnicas de fast-flux, estas características han sido agrupadas en el módulo de detección de URL maliciosas. De manera similar, se han elaborado los módulos de detección de correos dirigidos y anomalías de red.

Por consiguiente, los módulos de detección propuestos utilizan técnicas de aprendizaje automático para generar alertas que identifican cuando se está produciendo un ataque de APT; para ello, es necesario utilizar un conjunto de datos e identificar los algoritmos adecuados. En este caso, hemos seleccionado el conjunto de datos DAPT2020, ya que es una propuesta interesante que recopila diferentes tipos de anomalías de red clasificadas en etapas de un ciclo de vida, además, se ha determinado utilizar los siguientes algoritmos de aprendizaje automático supervisado, *k nearest neighboor*, *naive bayes*, *logistic regresion*, *decision tree*, *artificial neural network*, *support vector machine* y *k-means*, para evaluar su precisión y rendimiento.

En virtud de los resultados del análisis exploratorio, hemos llegado a la conclusión de que no todos los algoritmos previamente seleccionados brindan buenos resultados para la detección de anomalías de red; siendo *k*-NN y DT los algoritmos que han obtenido mejores resultados tanto en precisión como en rendimiento, el algoritmo ANN ha dado buenos resultados que pueden ser mejorados al utilizar otros parámetros de configuración; no obstante, los algoritmos NB y LR han demostrado no ser muy eficiente; por último, los algoritmos SVM y *k*-means obtuvieron resultados no concluyentes.

Asimismo, las métricas de precisión y rendimiento son importantes para la evaluación de los algoritmos, como en el caso del algoritmo LR que obtuvo buenos resultados de precisión en la matriz de confusión; sin embargo, cuando fue evaluada la métrica de *F1 Score*, los resultados de predicción eran muy buenos para el tráfico normal, pero bajos para la predicción del tráfico anómalo. Por estos motivos, se deben considerar

varias métricas para la selección de los algoritmos que se utilizarán para la detección de ataques de APT.

Otro punto a destacar, es que los algoritmos de aprendizaje automático supervisado tienen un mejor rendimiento con etiquetado binario en comparación a un etiquetado múltiple; esto debe ser considerado al momento de seleccionar los datos o durante el preprocesamiento de los datos de entrenamiento.

En conclusión, la detección en tiempo real de un ataque de APT no es tarea fácil ni para los administradores de red, ni para los investigadores, ya que una detección basada en aprendizaje automático depende de dos factores fundamentales, es decir, la calidad de los datos y la precisión y rendimiento de los algoritmos.

Teniendo en cuenta que la creación de un conjunto de datos a partir de muestras de malware en un entorno controlado, requiere una configuración extensa y una gran cantidad de recursos y tiempo para etiquetar los datos de manera correcta, es por ello, que actualmente los conjuntos de datos disponibles para realizar experimentos para la detección de estas amenazas es limitado; el etiquetado juega un papel primordial para la creación y el análisis del conjunto de datos, ya que este etiquetado se puede realizar por tipo de TTP, etapa del ciclo de vida o un etiquetado binario.

Por otro lado, los algoritmos de aprendizaje automático tradicionales pueden no ser totalmente eficientes para la detección de un ataque de APT, por lo que, seria recomendable estudiar otros algoritmos de aprendizaje automático especializados en detección, puesto que estos algoritmos pueden haber sido mejorados matemáticamente utilizando como base los algoritmos tradicionales. Otra consideración para seleccionar los algoritmos de detección, podría ser emplear los procesos de AutoML (automated machine learning), aprendizaje no supervisado o deep learning, para analizar conjuntos de datos con etiquetado múltiple o incluso datos no etiquetados.

De manera que, los módulos de detección son una pieza fundamental del modelo, la realización de pruebas con diferentes algoritmos y conjuntos de datos podría conllevar a la identificación de los posibles fallos y debilidades, permitiendo la actualización de estos módulos, ya sea agrupando nuevas características a las existentes o creando nuevos módulos de detección y mejorar así, la detección de estas amenazas.

Finalmente, recodemos que los ataques de APT son muy versátiles y se adaptan muy rápido para engañar a los usuarios y sistemas de información; generalmente, las grandes empresas tardan años en descubrir de manera precisa, la forma de como se ha desarrollado y ejecutado el ataque, a pesar de contar con un gran número de investigadores y recursos.

Para terminar, es posible que las amenazas persistentes avanzadas no desaparezcan, pero con herramientas eficientes para la detección de estos ataques, se puede disminuir las pérdidas de datos críticos y evitar futuras ciberguerras.

Como trabajo futuro, se propone el desarrollo e implementación de un entorno de trabajo basado en el modelo propuesto, que permita realizar diversas simulaciones de ataques de APT, utilizando para ello un espacio de pruebas controlado y seguro; esto permitiría comprobar la eficiencia real de los módulos de detección.

Por otro lado, la actualización de los módulos de detección mediante la optimización de algoritmos de *deep learning* podrían proporcionar mejores resultados de precisión y rendimiento, para la predicción de un ataque de APT en un menor tiempo o incluso en tiempo real.

Bibliografía

[1] Abu Alfeilat, H. A., Hassanat, A. B., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., and Prasath, V. S. (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. *Big Data*, 7(4):221–248.

[2] Adams, C. (2018). Learning the lessons of wannacry. *Computer Fraud & Security*, 2018(9):6 – 9.

[3] Ahuja, R., Chug, A., Gupta, S., Ahuja, P., and Kohli, S. (2020). *Classification and Clustering Algorithms of Machine Learning with their Applications*, pages 225–248. Springer International Publishing, Cham.

[4]Aleroud, A. and Zhou, L. (2017). Phishing environments, techniques, and countermeasures: a survey. *Computers & Security*, 68:160–196.

[5] Alloghani, M., Al-Jumeily, D., Hussain, A., Mustafina, J., Baker, T., and Aljaaf, A. J. (2020). *Implementation of Machine Learning and Data Mining to Improve Cybersecurity and Limit Vulnerabilities to Cyber Attacks*, pages 47–76. Springer International Publishing, Cham.

[6]Alshamrani, A., Myneni, S., Chowdhary, A., and Huang, D. (2019). A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities. *IEEE Commun. Surv. Tutorials*, (8):1–1.

[7]Aparicio-navarro, F. J., Kyriakopoulos, K. G., Ghafir, I., Lambotharan, S., Chambers, J. A., and Technology, F. (2018). Multi-Stage Attack Detection Using Contextual Information. pages 920–925.

[8]Awad, M. and Khanna, R. (2015). *Hidden Markov Model*, pages 81–104. Apress, Berkeley, CA.

[9] Bai, T., Bian, H., Daya, A. A., Salahuddin, M. A., Limam, N., and Boutaba, R. (2019). A Machine Learning Approach for RDP-based Lateral Movement Detection. In *2019 IEEE 44th Conf. Local Comput. Networks*, pages 242–245. IEEE.

[10] Balduzzi, M., Ciangaglini, V., and McArdle, R. (2013). Targeted attacks detection with spunge. In *Eleventh Annual International Conference on Privacy, Security and Trust (PST)*, pages 185–194. IEEE.

[11] Bhadane, A. and Mane, S. B. (2019). Detecting lateral spear phishing attacks in organisations. *IET Information Security*, 13:133–140.

[12] Brogi, G. and Tong, V. V. T. (2016a). TerminAPTor: Highlighting advanced persistent threats through information flow tracking. *2016 8th IFIP Int. Conf. New Technol. Mobil. Secur. NTMS 2016*.

[13] Brogi, G. and Tong, V. V. T. (2016b). Terminaptor: Highlighting advanced persistent threats through information flow tracking. In *8th IFIP International Conference on New Technologies, Mobility and Security*.

[14] BSI (2013). Fokus IT-Sicherheit 2013. Technical report, Bundesamt für Sicherheit in der Informationstechnik.

[15] Buczak, A. L. and Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Commun. Surv. Tutorials*, 18(2):1153–1176.

[16] Bystrov, E. (2017). TensorFlow Object Detection with Docker from scratch.

[17] Chen, P., Desmet, L., and Huygens, C. (2014). A Study on Advanced Persistent Threats. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8735 LNCS, pages 63–72.

[18] Chu, W.-L., Lin, C.-J., and Chang, K.-N. (2019). Detection and Classification of Advanced Persistent Threats and Attacks Using the Support Vector Machine. *Appl. Sci.*, 9(21):4579.

[19] Coopers, P. (2017). Operation cloud hopper. Technical report, PwC UK Cyber security and data privacy.

[20] Cordey, S. (2019). Trend Analysis: The Israeli Unit 8200 – An OSINT-based study. Technical Report December, Center for Security Studies (CSS), ETH Zürich.

[21] Cyber-research (2019). APT Malware Dataset.

[22] CyberMonitor (2020). APT & Cybercriminals Campaign Collection. visitado el 30 de abril de 2020.

[23] da Silva, I. N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L. H. B., and dos Reis Alves, S. F. (2017). *Artificial Neural Networks*. Springer International Publishing.

[24] Dahl, G. E., Dong Yu, Li Deng, and Acero, A. (2012). Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. *IEEE Trans. Audio. Speech. Lang. Processing*, 20(1):30–42.

[25] Docker Inc. (2020). Docker overview. visitado el 1 de Agosto de 2020.

[26] Dua, S. and Du, X. (2011). *Data mining and machine learning in cybersecurity*. Auerbach Publications.

[27] Erkaymaz, O., Ozer, M., and Perc, M. (2017). Performance of small-world feedforward neural networks for the diagnosis of diabetes. *Appl. Math. Comput.*, 311:22–28.

[28] Falliere, N., Murchu, L. O., and Chien, E. (2011). W32. stuxnet dossier. *White Pap. Symantec Corp., Secur. Response*, 5(6):29.

[29] FireEye (2014). APT28: A window into Russia's cyber security. Technical report.

[30] FireEye (2016). Follow the money: DISSECTING THE OPERATIONS OF THE CYBER CRIME GROUP FIN6. Technical Report April.

[31] FireEye (2018). Advanced Persistent Threat Groups. visitado el 20 de marzo de 2019.

[32] FireEye (2019). Double Dragon: APT41, a dual espionage and cyber crime operation. Technical report, FireEye.

[33] Fireeye (2019). M-Trends 2019: Fireeye Mandiant Services Special Report. Technical report.

[34] Ghafir, I., Hammoudeh, M., Prenosil, V., Han, L., Hegarty, R., Rabie, K., and Aparicio-Navarro, F. J. (2018). Detection of advanced persistent threat using machine-learning correlation analysis. *Futur. Gener. Comput. Syst.*, 89:349–359.

[35] Ghafir, I. and Prenosil, V. (2016). Proposed Approach for Targeted Attacks Detection. In *Lect. Notes Electr. Eng.*, volume 362, pages 73–80.

[36] Giura, P. and Wang, W. (2012). A Context-Based Detection Framework for Advanced Persistent Threats. In *2012 Int. Conf. Cyber Secur.*, number SocialInformatics, pages 69–74. IEEE.

[37] Gou, J., Ma, H., Ou, W., Zeng, S., Rao, Y., and Yang, H. (2019). A generalized mean distance-based k-nearest neighbor classifier. *Expert Syst. Appl.*, 115:356–372.

[38] Guan, Z., Bian, L., Shang, T., and Liu, J. (2018). When Machine Learning meets Security Issues: A survey. *2018 IEEE Int. Conf. Intell. Saf. Robot.*, pages 158–165.

[39] Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

[40] Hoefelmeyer, R. and Phillips, T. E. (2011). Malicious code. In *Encyclopedia of Information Assurance*.

[41] Huang, C., Han, J., Zhang, X., and Liu, J. (2019). Automatic Identification of Honeypot Server Using Machine Learning Techniques. *Secur. Commun. Networks*, 2019:1–8.

[42] Jeun, I., Lee, Y., and Won, D. (2012). A Practical Study on Advanced Persistent Threats. In *Commun. Multimed. Secur.*, volume 8735, pages 144–152.

[43] Jing, R. and Zhang, Y. (2010). A view of support vector machines algorithm on classification problems. In *Multimedia Communications (Mediacom), 2010 International Conference on*, pages 13–16. IEEE.

[44] Joseph, A. D., Laskov, P., Roli, F., Tygar, J. D., and Nelson, B. (2013). Machine learning methods for computer security (dagstuhl perspectives workshop 12371). In *Dagstuhl Manifestos*, volume 3. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[45] Joshi, A. V. (2020). *Machine Learning and Artificial Intelligence*, volume 64. Springer International Publishing, Cham.

[46] Joshi, V. B., Raval, M. S., Gupta, D., Rege, P. P., and Parulkar, S. K. (2016). A multiple reversible watermarking technique for fingerprint authentication. *Multimed. Syst.*, 22(3):367–378.

[47] Kasperky Lab (2015). The Duqu 2.0 - Technical Details (V2.1). Technical Report June.

[48] Kaspersky Lab (2019). Targeted Cyberattacks LOGBOOK.

[49] Kaviani, S. and Sohn, I. (2020). Influence of random topology in artificial neural networks: A survey. *ICT Express*.

[50] Khosravi, B. T. L. M. (2020). A semi real dataset of meta-alerts for apt attack detection.

[51] Khraisat, A., Gondal, I., Vamplew, P., and Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1):20.

[52] Krombholz, K., Hobel, H., Huber, M., and Weippl, E. (2015). Advanced social engineering attacks. *Journal of Information Security and applications*, 22:113–122.

[53] Kuhnert, N. (2018). Threat Actor Map. visitado el de 20 abril de 2020.

[54] Lamprakis, P., Dargenio, R., Gugelmann, D., Lenders, V., Happe, M., and Vanbever, L. (2017). Unsupervised Detection of APT C&C Channels using Web Request Graphs. In *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, volume 10327 LNCS, pages 366–387.

[55] Lemay, A., Calvet, J., Menet, F., and Fernandez, J. M. (2018). Survey of publicly available reports on advanced persistent threat actors. *Computers and Security*, 72:26–59.

[56] Lockheed Martin (2009). Cyber Kill Chain.

[57] Malenkovich, S. (2013). ¿ Qué es un rootkit ? visitado el 16 de septiembre de 2019.

[58] Mandiant (2013). APT1 Exposing One of China's Cyber Espionage Units. Technical report.

[59] Martínez Torres, J., Iglesias Comesaña, C., and García-Nieto, P. J. (2019). Review: machine learning techniques applied to cybersecurity. *Int. J. Mach. Learn. Cybern.*, 10(10):2823–2836.

[60] Mitchell, T. (1997). *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill.

[61] Mitnick, K. D. and Simon, W. L. (2011). *The art of deception: Controlling the human element of security*. John Wiley & Sons.

[62] Myneni, S., Chowdhary, A., Sabur, A., Sengupta, S., Agrawal, G., Huang, D., and Kang, M. (2020). DAPT 2020 - Constructing a Benchmark Dataset for Advanced Persistent Threats. Number July, pages 138–163.

[63] NIST (2011). Managing information security risk: Organization, mission, and information system view. *Special Publication 800-39*.

[64] Nmap (2020). Nmap: Network Mapper - Free Security Scanner. visitado el 17 de noviembre de 2020.

[65] Olivieri, A. C. (2018). *Principal Component Analysis*, pages 57–71. Springer International Publishing, Cham.

[66] OWASP (2019). Unvalidated Redirects and Forwards. visitado el 17 de septiembre de 2019.

[67] Paganini, P. (2018). Turla APT group's espionage campaigns now employs Adobe Flash Installer and ingenious social engineering. visitado el 20 de agosto de 2019.

[68] Paganini, P. (2019a). Iran-linked APT33 updates infrastructure following its public disclosure. visitado el 21 de noviembre de 2019.

[69] Paganini, P. (2019b). Phishers continue to abuse Adobe and Google Open Redirects. visitado el 11 de octubre de 2019.

[70] Pan, Y., Pan, Z., Wang, Y., and Wang, W. (2020). A new fast search algorithm for exact k-nearest neighbors based on optimal triangle-inequality-based check strategy. *Knowledge-Based Syst.*, 189:105088.

[71] Portugal, I., Alencar, P., and Cowan, D. (2017). The use of machine learning algorithms in recommender systems: a systematic review. *Expert Systems with Applications*.

[72] Quintero-Bonilla, S. and del Rey, A. M. (2020). Proposed models for advanced persistent threat detection: A review. In *Adv. Intell. Syst. Comput.*, volume 1004, pages 141–148. Springer Verlag.

[73] Ramasubramanian, K. and Singh, A. (2017). Machine learning theory and practices. In *Machine Learning Using R*, pages 219–424. Springer.

[74] Sexton, J., Storlie, C., and Neil, J. (2015). Attack chain detection. *Stat. Anal. Data Min. ASA Data Sci. J.*, 8(5-6):353–363.

[75] Sharma, P. K., Moon, S. Y., Moon, D., and Park, J. H. (2017). DFA-AD: a distributed framework architecture for the detection of advanced persistent threats. *Cluster Comput.*, 20(1):597–609.

[76] Shenwen, L., Yingbo, L., and Xiongjie, D. (2015a). Study and research of apt detection technology based on big data processing architecture. In *5th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 313–316. IEEE.

[77] Shenwen, L., Yingbo, L., and Xiongjie, D. (2015b). Study and research of APT detection technology based on big data processing architecture. In *2015 IEEE 5th Int. Conf. Electron. Inf. Emerg. Commun.*, pages 313–316. IEEE, IEEE.

[78] Shodan (2020). Shodan: search engine for Internet-connected devices. visitado el 17 de noviembre de 2020.

[79] Siddiqui, S., Khan, M. S., Ferens, K., and Kinsner, W. (2016). Detecting Advanced Persistent Threats using Fractal Dimension based Machine Learning Classification. In *Proc. 2016 ACM Int. Work. Secur. Priv. Anal. - IWSPA '16*, IWSPA '16, pages 64–69. ACM Press.

[80] Sigholm, J. and Bang, M. (2013). Towards offensive cyber counterintelligence: Adopting a target-centric view on advanced persistent threats. In *European Intelligence and Security Informatics Conference (EISIC)*, pages 166–171. IEEE.

[81] Spiderfoot (2020). Spiderfoot automates OSINT. visitado el 17 de noviembre de 2020.

[82] Steer, J. (2017). Defending against spear-phishing. *Computer Fraud & Security*, 2017(8):18–20.

[83] Su, Y., Li, M., Tang, C., and Shen, R. (2016). A Framework of APT Detection Based on Dynamic Analysis. In *Proc. 2015 4th Natl. Conf. Electr. Electron. Comput. Eng.*, number Nceece 2015, pages 1047–1053. Atlantis Press.

[84] Swisscom (2019). Targeted Attacks Cyber Security Report 2019. Technical report, Swisscom (Switzerland) Ltd Group Security.

[85] Symantec (2016). Internet security threat report. Technical Report 2.

[86] Tanaka, Y., Akiyama, M., and Goto, A. (2017). Analysis of malware download sites by focusing on time series variation of malware. *Journal of computational science*, 22:301–313.

[87] ThaiCERT (2019). Threat Group Cards: A Threat Actor Encyclopedia. visitado el 24 de junio de 2019.

[88] Trend Micro (2013). The Custom Defense Against Targeted Attacks. Technical report.

[89] Ussath, M., Jaeger, D., Cheng, F., and Meinel, C. (2016). Advanced persistent threats: Behind the scenes. In *Annual Conference on Information Science and Systems (CISS)*, pages 181–186. IEEE.

[90] Valverde Ramírez, M. C., de Campos Velho, H. F., and Ferreira, N. J. (2005). Artificial neural network technique for rainfall forecasting applied to the São Paulo region. *J. Hydrol.*, 301(1-4):146–162.

[91] Vukalovic, J. and Delija, D. (2015). Advanced Persistent Threats - detection and defense. In *2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron.*, pages 1324–1330. IEEE.

[92] Wang, D. and Xu, J. (2020). Principal Component Analysis in the local differential privacy model. *Theor. Comput. Sci.*, 809:296–312.

[93] Wang, X., Zheng, K., Niu, X., Wu, B., and Wu, C. (2016). Detection of command and control in advanced persistent threat based on independent access. In *2016 IEEE Int. Conf. Commun.*, pages 1–6. IEEE.

[94] Wang, Y., Gu, D., Peng, D., Chen, S., and Yang, H. (2012). Stuxnet vulnerabilities analysis of scada systems. In *Network Computing and Information Security*, pages 640–646. Springer.

[95] Yager, R. R., Reformat, M. Z., and Alajlan, N. (2014). *Intelligent Methods for Cyber Warfare*, volume 563. Springer.

[96] Yang, L. and Deng, M. (2010). Based on k-Means and Fuzzy k-Means Algorithm Classification of Precipitation. In *2010 Int. Symp. Comput. Intell. Des.*, volume 1, pages 218–221. IEEE.

[97] Zhang, R., Huo, Y., Liu, J., and Weng, F. (2017). Constructing APT Attack Scenarios Based on Intrusion Kill Chain and Fuzzy Clustering. *Secur. Commun. Networks*, 2017:1–9.

[98] Zhao, G., Xu, K., Xu, L., and Wu, B. (2015). Detecting APT Malware Infections Based on Malicious DNS and Traffic Analysis. 3:1132–1142.

Apéndice A

A.1. Trabajos publicados que apoyan esta tesis

Los siguientes artículos han sido publicados como resultado de la investigación realizada por el candidato a doctor:

- Artículo publicado en revista científica:
  - Quintero-Bonilla, S. y Martín del Rey, A. A New Proposal on the Advanced Persistent Threat: A Survey. *Appl. Sci.* **10**, 3874 (2020).
- Conferencias internacionales:
  - Quintero-Bonilla S., Martín del Rey A. y Queiruga-Dios A. *New Perspectives in the Study of Advanced Persistent Threats* en *Cyber-Physical Multi-Agent Systems. The PAAMS Collection - 15th International Conference, PAAMS 2017* (eds. De la Prieta, F. y col.) **619** (Springer, Cham, Switzerland, 2017), 242-244.
  - Quintero-Bonilla S. y Martín del Rey A. *Proposed Models for Advanced Persistent Threat Detection: A Review* en *Distributed Computing and Artificial Intelligence, 16th International Conference, Special Sessions* (eds.

Herrera-Viedma E., Vale Z., Nielsen P., Martin Del Rey A., Casado Vara R.) **1004** (Springer, Cham, Switzerland, 2019), 141-148.

- Otras publicaciones:

  - Batista, F. K., Martin del Rey, A., Quintero-Bonilla, S. y Queiruga-Dios, A. *A SEIR Model for Computer Virus Spreading Based on Cellular Automata* en *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17* (eds. Pérez García, H., AlfonsoCendón, J., Sánchez González, L., Quintián, H. y Corchado, E.) **649** (Springer, Cham, Switzerland, 2018), 641-650.