Institución Universitaria

Detección de Amenazas Persistentes Avanzadas mediante la aplicación de cadenas de Markov sobre tráfico de red

Daniel Mauricio Ramos Ríos

Instituto Tecnológico Metropolitano

Facultad de Ingenierías

Medellín, Colombia

2019

Detección de Amenazas Persistentes Avanzadas mediante la aplicación de cadenas de Markov sobre tráfico de red

Daniel Mauricio Ramos Ríos

Trabajo de investigación presentado como requisito parcial para optar al título de:
Magíster en Seguridad Informática

Director (a):
Mg. Miguel Mariano Manosalva Pinedo
Magíster en Seguridad Informática

Línea de Investigación:
Análisis de Malware

Instituto Tecnológico Metropolitano
Facultad de Ingenierías
Medellín, Colombia
2019

Hay suficiente fuerza dentro de ti para superar cualquier cosa en la vida.

Lailah Gifty Akita

Agradecimientos

A mi familia gracias por su apoyo incondicional en este proceso.
A mi madre por su paciencia y presencia en los momentos más difíciles.
A mi asesor de tesis por confiar en mi trabajo, por su disposición para ayudarme, por su paciencia y comprensión.
A todos los docentes de la maestría por sus enseñanzas.
A Kelly por su amistad, por enseñarme y acompañarme durante la maestría y por guiarme en el desarrollo de esta tesis.
A Samir por sus consejos y por creer siempre en mí.

Resumen

Las Amenazas Persistentes Avanzadas, también conocidas como APT por sus siglas en inglés, son un tipo de ataque sofisticado caracterizado por seis fases que van desde la obtención de información básica para el acceso a los sistemas que se desean vulnerar hasta la exfiltración de datos mediante la conexión cifrada a servidores maliciosos.

En este trabajo de investigación se abordan este tipo de amenazas con el objetivo de evaluar su comportamiento en una red LAN mediante la aplicación de cadenas de Markov. Para ello, se captura el tráfico de red de un sistema conformado por un conjunto de máquinas virtuales, las cuales fueron infectadas con cepas de virus utilizados en campañas de APT. Se analizaron los patrones de tráfico y se extrajeron las características relacionadas con la actividad maliciosa. Luego de comprobar que el comportamiento del tráfico no depende del pasado, se modeló mediante cadenas de Markov para hallar las matrices de transición de estados que describen su comportamiento.

Al analizar los datos, se pueden observar patrones de infección caracterizados principalmente por conexiones a servidores FTP maliciosos, consulta de dominios maliciosos y paquetes TCP de longitud elevada, que sugieren descarga de archivos adicionales y extracción de información de las máquinas infectadas.

Palabras clave: Amenazas Persistentes Avanzadas (APT), cadenas de Markov, comportamiento, detección, tráfico de red.

Abstract

Advanced Persistent Threats, also known as APT, are a sophisticated type of ciber attack characterized by six infection patterns, in which basic information is extracted to gain access to targeted systems, to finally steal data establishing connections to external malicious servers.

In this research work these types of threats are addressed in order to evaluate their behavior in a LAN network through the application of Markov chains. To make this possible, network traffic is captured from a simulated system with virtual machines infected with tools used in APT campaigns. The traffic patterns are analyzed to extract the features related with the malicious activity. After verifying that the behavior of the traffic does not depend on the past, it is modeled using Markov chains to find the transition matrix that describe its behavior.

The analysis provide accurate information about infection patterns, related with connections to malicious FTP servers, queries to malicious domains and TCP packets of high length, which suggest downloading additional files and extracting information from infected machines.

Keywords: Advanced Persistent Threat, Markov Chain, Behavior, Detection, Network Traffic.

Contenido

Pág.

Lista de figuras

Lista de tablas

Lista de Símbolos y abreviaturas

Símbolos

Símbolo	Término	Unidad SI	Definición
$F_X(x;t_i)$	Función de distribución de probabilidad	1	Ecuación (1.3)
$F_{X_1,\dots,X_n}(x_1,\dots,x_n;t_1,\dots,t_n)$	Función de distribución de probabilidad N-dimensional	1	Ecuación (1.4)
$P[X(t) \leq x \mid X(t_n) \leq x_n,\dots,X(t_0) \leq x_0]$	Propiedad Markoviana de procesos estocásticos	1	Ecuación (1.5)
PX_t	Probabilidades condicionales	1	Ecuación (1.6)
${P_{ij}}^{(n)}$	Probabilidades de transición de n pasos	1	Ecuación (1.8)
$P^{(0)}$	Distribución de probabilidad del estado inicial (vector)	1	Ecuación (1.9)
X_0	Distribución de probabilidad del estado inicial	1	Ecuación (1.9)
X_n	Distribución de probabilidad en la etapa n-ésima		$P(X_n = i)$,
$P^{(n)}$	Distribución de probabilidad del estado inicial (vector)	1	$({p_1}^{(n)},\dots,{p_i}^{(n)},\dots,{p_k}^{(n)})$
${p_i}^{(n)}$	Probabilidad de que en la etapa *n* la cadena de Markov se encuentre en el estado i	1	$P(X_n = i)$,
${p_{ij}}^{(n)}$	Probabilidad de transición en n etapas/Ecuación de Chapman-Kolmogorov.	1	Ecuación (1.10)
${f_{ij}}^{(n)}$	Probabilidad de primera pasada	1	Ecuación (1.14)
f_{ij}	Probabilidad de pasada	1	Ecuación (1.15)

Abreviaturas

Abreviatura	Término
AH	Authentication Header
APT	Advanced Persistent Threat
ARP	Address Resolution Protocol
ATCTDS	Automated Temporal Correlation Traffic Detection System
BROWSER	Browser Protocol
C&C	Command and Control
CLI	Command-line Interface
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CSV	Comma-Separated Values
CTMC	Continuous Time Markov Chains

DA	Directorio Activo
DDoS	Distributed Denial-of-Service
DLP	Data Loss Prevention
DNC	Democratic National Committee
DNS	Domain Name System
DTMC	Discrete Time Markov Chains
EIGRP	Enhanced Interior Gateway Routing Protocol
ELM	Extreme Learning Machine
ESP	Encapsulating Security Payload
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IGMP	Internet Group Management Protocol
IGRP	Interior Gateway Routing Protocol
IP	Internet Protocol
IPS	Intrusion Prevention System
L2TP	Layer 2 Tunneling Protocol
LAN	Local Area Network
LLMNR	Link-Local Multicast Name Resolution
MAC	Media Access Control
MitM	Man In the Middle
MLAPT	Machine-Learning-based APT
NBNS	Servicio de Nombres NetBIOS
NIDS	Network Intrusion Detection System
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OTAN	Organización del Tratado del Atlántico Norte
PCAP	Packet Capture Data File
PKI	Public Key Infrastructure
PKIX	Public-Key Infrastructure (X.509)
SIEM	Security Information and Event Management
SNN	Stabilized Nearest Neighbor
SSDP	Simple Service Discovery Protocol
SSL	Secure Sockets Layer
STUN	Session Traversal Utilities for NAT
TAP	Terminal Access Point
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TPM	Transition Probability Matrix

TTL	Time To Live
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network
XML	Extensible Markup Language

Introducción

Las Amenazas Persistentes Avanzadas, también conocidas por sus siglas en inglés, APT (*Advanced Persistent Threat*), son un tipo de ataque que se ha presentado cada vez con más frecuencia en los últimos años en grandes empresas, organismos e instituciones. No es un ataque donde se puede ver afectado un usuario común, sino que perjudica a grandes sectores de la industria. Su impacto se debe a que es ejecutado por grupos de atacantes con un nivel alto de experticia, con infraestructura avanzada, posiblemente financiados por gobiernos o grandes estructuras criminales, y que persiguen objetivos militares, políticos, financieros e industriales.

El campo de estudio de las APT se ha ido ampliando recientemente, lo cual ha permitido que se identifique un patrón de infección caracterizado por seis fases que van desde la obtención de información básica para el acceso a los sistemas que se desean vulnerar hasta la exfiltración de datos mediante la conexión cifrada a servidores maliciosos externos. Este tema ha sido estudiado hace algunos años (desde el 2008 aproximadamente), aunque fue en el 2015 cuando generó un interés mayor, luego de que la compañía Sony Pictures revelara un ataque en el cual se habría presentado una fuga de más de 100 TB de información [1].

Algunas fuentes [2, 3, 4] plantean un escenario futuro donde este tipo de amenazas son mucho más comunes y dinámicas y, por lo tanto, las áreas de seguridad deberán estar preparadas para afrontar cambios importantes en cuanto a:

- incremento del número de ataques dirigidos y el mercado relacionado con estos;
- aumento del número de atacantes y grupos con motivación política dispuestos a financiar tales ataques;
- evolución de los atacantes en términos de infraestructura para hacer más difícil encontrarlos, rastrearlos e investigarlos;
- uso de herramientas que permiten modificar programas desarrollados por otros;
- desarrollo de técnicas para engañar a los investigadores por medio de inserción de pistas falsas en sus herramientas para confundir a las autoridades o llegar a personas equivocadas; y
- ataques altamente destructivos, donde no solo se produzca fuga de información o denegación de servicios, sino que también afecten la integridad de los datos, modificándolos, corrompiéndolos o cifrándolos. [2, 3, 4]

Las APT están diseñadas para propósitos puntuales como el espionaje industrial o el robo de propiedad intelectual e información confidencial, y se diferencian de los demás ataques por tres características puntuales: (1) están dirigidas a un objetivo específico, (2)

son ejecutadas por equipos organizados que utilizan herramientas y técnicas de evasión sofisticadas y (3) pueden permanecer por mucho tiempo dentro de un sistema.

La infección por APT se puede dar por varias vías; los atacantes, por ejemplo, pueden hacer uso de diversas herramientas que van desde la ingeniería social hasta el malware adaptativo, razón por la cual su detección es muy complicada. Actualmente, no existe una herramienta 100 % efectiva en la detección de APT, por lo cual se han desarrollado diversos estudios para complementar los esquemas de seguridad frente a estas.

En este orden de ideas, el problema que se delimitó en este proyecto de investigación es la dificultad para detectar las APT en sistemas de información al momento en que ya han infectado un sistema. En este sentido, el objetivo general de este trabajo fue evaluar el comportamiento de una red LAN tras la infección por Amenazas Persistentes Avanzadas mediante la aplicación de cadenas de Markov. Para dar cumplimiento a este, se definieron los siguientes objetivos específicos: (1) establecer un mecanismo para el pre-procesamiento de datos recolectados, (2) caracterizar los elementos del tráfico de red que reflejan patrones de infección por APT, (3) representar el tráfico de red mediante cadenas de Markov y (4) demostrar la eficacia de la evaluación realizada mediante la implementación de reglas de red en un IDS.

Lo anterior pone de manifiesto una trayectoria investigativa que aún hoy no se detiene y plantea retos futuros. Por esta razón, y como justificación, fue de vital importancia el estudio de este tipo de amenazas en el campo de la seguridad informática, teniendo en cuenta que este trabajo presenta tres impactos: académico, porque constituye una investigación novedosa en el marco de la reciente incorporación al Instituto Tecnológico Metropolitano (ITM) de la Maestría en Seguridad Informática; científico, dado que el producto resultante aporta al crecimiento de la producción científica del ITM; y tecnológico porque los resultados de este estudio sirven como base para que otros equipos de investigadores desarrollen herramientas para la detección y contención de APT. Estas herramientas podrían ser de gran valor comercial y económico para las empresas, debido a que mejorarían sus mecanismos de seguridad, al tiempo que mantendrían segura su información relevante.

Para desarrollar este proyecto de investigación, se llevó a cabo una metodología basada en el uso de métodos estadísticos sobre tráfico de red, desde los cuales se diagnosticaron y explicaron los patrones asociados con esta clase de amenazas. Para lograrlo, se plantearon cuatro fases, cada una correspondiente a un objetivo específico, así:

1. Recolección y pre-procesamiento de datos: se recopiló el tráfico de red de un sistema virtualizado, infectado por APT, y se estructuraron los datos para obtener como resultado las capturas de tráfico presentadas en tablas.
2. Descripción de patrones de infección: se caracterizaron los elementos del tráfico de red que revelan patrones de infección por APT y se obtuvo el listado de direcciones IP y dominios maliciosos asociados.
3. Análisis de datos: se modeló el comportamiento de la red mediante cadenas de Markov para detectar anomalías. Como resultado, se obtuvieron las cadenas de Markov de las amenazas estudiadas y las características asociadas con la infección.
4. Implementación de reglas de red en IDS: se recogieron las observaciones del modelo estadístico para identificar patrones asociados con infecciones por APT.

A partir de este proceso metodológico, se buscó demostrar la siguiente hipótesis: “En fase de explotación el proceso de detección de amenazas persistentes avanzadas (APT) puede mejorarse, al aplicar cadenas de Markov para el análisis del comportamiento del tráfico de red”.

Es importante aclarar que este trabajo es un proyecto de investigación debido a que con su desarrollo se busca generar teorías y crear conocimiento a partir de un estado del arte y una base metodológica.

Como alcance y limitaciones; por un lado, se incluye el estudio del comportamiento de amenazas utilizadas por grupos APT en campañas conocidas en años recientes. Así, los resultados obtenidos son de tipo correlacional, en el sentido de que se analiza la relación entre variables para predecir un comportamiento futuro, y de tipo explicativo, en tanto se analizan las causas de los eventos estudiados y se explican las condiciones en las que se manifiestan. Por otro lado, dentro de las limitaciones se encontró que el estudio no se realiza sobre una red real. El laboratorio utilizado no está conformado por múltiples estaciones que permitan simular un sistema de mayor complejidad. Adicionalmente, no se cuenta con los recursos suficientes para analizar la cantidad y variabilidad de los datos capturados.

Este trabajo aporta al campo de estudio de la seguridad informática, debido a que proporciona un enfoque para entender el comportamiento de APT desde una perspectiva estadística primordialmente, complementándose con un análisis de las variables involucradas en el proceso de comunicación en redes. Su aplicación reside en la detección ágil de este tipo de amenazas cuando ya están en el sistema, teniendo en cuenta la información de la red.

Finalmente, la estructura del trabajo se divide en tres apartados: el primero corresponde al marco teórico y estado del arte, donde el lector podrá conocer los conceptos principales de este trabajo y los estudios relacionados con aportaciones al campo de la seguridad informática y a las APT. Luego se expone la metodología, en la cual se describe el paso a paso del desarrollo del proyecto y el método a implementar: cadenas de Markov sobre tráfico de red. Por último, se presentan los resultados, los cuales responden a los objetivos de investigación planteados, así como las conclusiones y recomendaciones en donde se plantea el trabajo futuro.

1. Marco Teórico y Estado del Arte

1.1 Estado del arte

Las APT se han estudiado desde varias perspectivas. El primer enfoque en el tema consistió en analizar los casos existentes para luego profundizar en otros aspectos como sus modelos, la correlación de alertas, la reconstrucción de escenarios y la predicción de ataques.

La caracterización de las fases de infección por APT mencionadas anteriormente es uno de los productos resultantes de las investigaciones realizadas en el campo. A partir de allí se han propuesto nuevas soluciones para cada una de las fases. Esta investigación en particular se enfoca en la fase tres de la infección por APT (explotación), donde el malware ya ha sido instalado en la red objetivo. Los estudios más relevantes realizados en esta fase se resumen en los párrafos siguientes.

El análisis de dominios maliciosos es uno de los enfoques más comunes que se aplican en la detección de APT, debido a que proporciona información para identificar las comunicaciones de los servidores de Comando y Control (C&C) en este tipo de ataques.

En el estudio presentado por Manggalanny y Ramli [5] se propuso un marco de investigación dividido en tres fases: recopilación de datos a partir del sistema de nombres de dominio (DNS), procesamiento de datos y comparación con amenazas existentes. Esta investigación se centró en el tráfico del puerto 53 UDP, que es utilizado por el DNS para responder a una solicitud de consulta de dominio desde la aplicación de usuario.

Los datos se capturaron a través de dispositivos TAP (dispositivos que simulan a otros en la capa de enlace de red), se conectaron a un analizador de paquetes y se filtraron con una lista predefinida de varias fuentes, por ejemplo, dominios e IP incluidas en listas negras, bases de datos de *phishing* y anti *spam*, etc. En este caso se utilizó un modelo de agrupación de Vecinos Cercanos Compartidos (Stabilized Nearest Neighbor, SNN), el cual es un método de agrupación que permite segregar varios tipos de comportamiento de tráfico anómalo con actividades o patrones similares en diferentes grupos únicos.

El resultado del análisis de tráfico con este método mostró aumentos significativos en la detección exitosa de tráfico anómalo, al indicar una presencia de actividad DNS maliciosa muy específica del malware utilizado por APT.

Por su parte, el enfoque de Shi, Chen y Li [6] se centró también en las comunicaciones con el C&C mediante el DNS como columna vertebral para implementar la infección de malware y la extracción de datos. Estos autores proponen una metodología basada en el

aprendizaje automático para detectar nombres de dominio de malware a partir del uso del Extreme Learning Machine (ELM), una red neuronal moderna con alta precisión y velocidad de aprendizaje rápida. En este caso se aplicó ELM para clasificar los nombres de dominio según las características extraídas de múltiples recursos: longitud del dominio, número de caracteres consecutivos en el dominio, entropía del dominio (medida del desorden o caos en la estructura de un dominio), número de direcciones IP y países relacionados con el dominio, valor promedio y desviación estándar de TTL (Time To Live), tiempo de vida del dominio y tiempo activo del dominio.

El experimento reveló que el método de detección planteado tiene una alta tasa de detección y precisión de más del 95 % con una velocidad de aprendizaje rápida.

En este punto cabe destacar que las comunicaciones con el servidor de Comando y Control (C&C) son claves para identificar ataques asociados con APT y, por tanto, se han analizado desde otras perspectivas, como la que proponen Debatty, Mees y Gilon [7], cuya investigación se centró en la detección de APT que utilizan el protocolo HTTP para establecer un canal de comunicación con su servidor C&C. A partir de la premisa de que las APT ingenuas realizan conexiones a su servidor de Comando y Control a intervalos de tiempo fijos o variables, plantean los autores que la actividad maliciosa es relativamente fácil de detectar en los registros del proxy.

En este estudio se utilizaron los datos recopilados en una red real, a los cuales se les aplicó la teoría de grafos para obtener un árbol de peticiones HTTP. Los resultados mostraron que el rendimiento del sistema planteado es comparable al de los antivirus disponibles actualmente, aunque estos utilizan firmas para detectar malware conocido, mientras que este algoritmo solo realizó el análisis de comportamiento para detectar nuevos ataques indocumentados, por tanto, no se basan en un conocimiento previo de los ataques.

Retomando el análisis de DNS, Oprea, Li, Yen, Chin, y Alrwais [8] utilizaron registros de DNS y proxy en el estudio *Detection of Early-Stage Enterprise Infection by Mining Large-Scale Log Data*. Con estos logs se obtuvo un algoritmo de propagación de amenazas que permitiera identificar hosts infectados y dominios maliciosos. En este estudio se propuso un marco gráfico-teórico basado en la propagación de creencias para identificar dominios relacionados con infecciones de malware en etapa temprana. Primero se restringió la atención al tráfico de dominios no visitados previamente por ningún usuario en la organización —probablemente asociados con actividad sospechosa—, para luego asignarles una puntuación. Los resultados demostraron que el algoritmo utilizado detecta actividades sospechosas omitidas por las soluciones de seguridad convencionales.

Por otra parte, algunos estudios se han enfocado en el análisis del Directorio Activo (DA) debido a la importancia para los atacantes. A través de este sistema es posible obtener privilegios y permanecer indetectable en la red comprometida.

En 2016, Wu, Lee, Wei, Hsieh y Lai [9] utilizaron los logs del Directorio Activo de Windows para identificar actividad anormal y patrones de infección por APT. Su investigación se centró en el análisis secuencial de logs del DA y el proxy para construir patrones gráficos que resumieran los comportamientos secuenciales de la capa de aplicación del usuario mediante modelos estadísticos, específicamente cadenas de Markov. Se supervisaron las desviaciones con respecto a los comportamientos normales

y los resultados mostraron que el método puede mejorar el proceso de monitoreo además de que las diferencias entre patrones benignos y sospechosos podrían ser organizadas como inteligencia de amenaza para describir posibles escenarios de ataque.

Una investigación más reciente de Matsuda, Fujimoto y Mitsunaga [10] se centró en el procesamiento de la actividad de ataques que utilizan el Directorio Activo (DA) para disfrazarse como administradores de dominio con el uso de cuentas legítimas, un comportamiento típico de APT, donde se propuso un método basado en el Aprendizaje Automático No Supervisado para detectar ataques que utilizan este tipo de cuentas. En este caso se aplicó el método sobre el registro de eventos de Windows para detectar comportamiento anómalo asociado con APT.

Teniendo en cuenta que el Aprendizaje Automático No Supervisado no requiere información de entrenamiento (información que indique si los registros de eventos corresponden un ataque o un comportamiento normal), permite detectar actividades maliciosas mediante la identificación de procesos que normalmente no se utilizan en las operaciones diarias del sistema, incluso si los atacantes se aprovechan de procesos legítimos. El método, además, ayuda a reducir los costos de operación, debido a que no requiere de listas blancas y negras para su funcionamiento.

El Aprendizaje Automático es una técnica utilizada en otros estudios relacionados con detección de APT. Ghafir *et al.* [11] proponen un sistema llamado MLAPT para detectar los ataques de APT mediante aprendizaje automático. El MLAPT se ejecuta en tres fases: la primera de ellas es la detección de amenazas asociadas con APT (archivos ejecutables disfrazados, hash de archivos maliciosos, nombres de dominio maliciosos, direcciones IP maliciosas, certificados SSL maliciosos, escaneos, entre otras), cuyo resultado son las alertas o eventos activados por las amenazas individuales.

En la segunda fase se realiza la correlación de alertas para vincular los resultados de la fase de detección e identificar las actividades relacionadas a un solo escenario de APT. En la última fase se diseña e implementa un módulo de predicción basado en aprendizaje automático con la correlación obtenida, para determinar la probabilidad de que se desarrolle un ataque de APT completo a partir de las alertas tempranas. MLAPT se evaluó experimentalmente y se determinó que el sistema presentado es capaz de predecir APT en sus primeros pasos con una precisión de 84,8 %.

Los árboles de decisión son otro método de predicción que ha sido utilizado en el campo de la seguridad para detectar las APT. En el estudio *An Efficient Classification Model for Detecting Advanced Persistent Threat*, Chandran, Poroli y Poornachandran [12] proponen un modelo matemático fundamentado en árboles de decisión (llamado *Random Forest Model*) para analizar de forma dinámica los sistemas infectados por APT. En este caso se analizan el uso de la memoria y la CPU, los puertos abiertos y el número de archivos del directorio System32. El modelo matemático propuesto describe un método para la detección de APT con menos falsos positivos y negativos.

En Moon, Im, Kim y Park [13] se propuso también un sistema de detección de intrusos que analiza el comportamiento de códigos maliciosos mediante el uso de árboles de decisión. El sistema, llamado DTB-IDS crea un árbol de decisión para mapear el

comportamiento de una amenaza en la red y en el sistema, y con base en este clasifica el código malicioso.

DTB-IDS consta de los siguientes seis módulos: reportero, administrador de comportamiento del sistema, administrador de comportamiento de la red, módulo de detección de intrusos, basado en el árbol de decisiones, administrador de registros y administrador de almacenamiento. El enfoque aplicado permite detectar ataques de APT que cambian constantemente después de la intrusión en un sistema, además, puede detectar la posibilidad en la intrusión inicial y minimizar el impacto del daño al responder rápidamente a los ataques APT.

En el estudio *Nazca: Detecting malware distribution in large-scale networks*, Invernizzi *et al.* [14] analizan las peticiones HTTP relacionadas con descargas e instalaciones de malware para realizar un gráfico de conexión global entre malware, hosts, cuentas y dominios involucrados en la distribución de malware. Los autores realizan un proceso de tres pasos para detectar malware en descargas: primero, identifican y extraen los metadatos de las peticiones HTTP; luego, identifican las peticiones que exhiben características anómalas para, finalmente, etiquetarlas como candidatas sospechosas. Con este enfoque es posible encontrar actividad maliciosa que reduzca posibles falsos positivos y centre la atención en los eventos de infección más significativos.

La correlación de eventos se ha utilizado también en el campo. Lu, Chen, Zhuo y Zhang [15] presentaron un sistema automatizado de detección de APT mediante la correlación temporal de tráfico, llamado ATCTDS. En este caso, los investigadores se concentraron en filtrar el tráfico de las cargas útiles APT —es decir, el malware— con la premisa de que cuando se comunican con los servidores de C&C lo hacen a intervalos de tiempo fijos, lo cual tiene una fuerte correlación con el tipo de carga maliciosa. El análisis incluyó también el tiempo de envío del tráfico combinado con el tamaño promedio del paquete de subida, que se encontró que era diferente entre los tráficos normales y los tráficos asociados con APT.

En el estudio se propuso también un filtro novedoso basado en las características de flujo, que puede ayudar a filtrar más del 70 % del tráfico en el entorno real, lo que puede mejorar en gran medida la eficiencia de detección. Así pues, la evaluación realizada demostró que el sistema puede detectar comportamientos de un amplio espectro de ataques con alta precisión y bajas tasas de falsos positivos.

Sharma, Moon, Moon y Park [16] propusieron una arquitectura distribuida para la detección de APT, denominada DFA-AD. A diferencia de otros enfoques, la técnica DFA-AD para detectar ataques APT se basa en varios clasificadores paralelos, que catalogan los eventos en un entorno distribuido y realizan correlación de los mismos.

En este sistema se recopilan los paquetes de tráfico de red con el uso de diferentes técnicas de recolección de datos, y se envían directamente al módulo de análisis. El proceso de funcionamiento se divide en tres fases diferentes: en la primera, cuando el hipervisor recibe los nuevos paquetes, envía copias duplicadas de cada paquete a los cuatro clasificadores diferentes, los cuales procesarán cada paquete de forma independiente y generarán eventos. Una vez que los clasificadores completen su proceso, todos los eventos generados se enviarán para el proceso en la siguiente fase.

En la segunda fase, todos los eventos generados se procesan en paralelo con los patrones de búsqueda, crean clases uniformes, hipótesis y finalmente evalúan las reglas. Esta fase es conocida como correlación de eventos. En la última fase, el sistema decide si hay algún ataque o no mediante la correlación recibida y genera la alarma de forma individual. Como resultado, el sistema llega a saber si hay alguna actividad o ataque malicioso. Los resultados de la evaluación muestran que el enfoque propuesto logra una mayor efectividad y precisión.

Los procesos de Markov se han utilizado en la detección de amenazas comunes [17]. Sin embargo, algunos académicos [9] han aplicado su teoría en la detección de APT. Un método similar, denominado *AD*^{2}, utiliza también series de tiempo y cadenas de Markov para detectar amenazas en los registros del directorio activo [18]. Las características de estos dos métodos ofrecen un buen rendimiento computacional; no obstante, tienen una baja precisión que muestra que puede ser limitada la detección de anomalías basada solo en registros del directorio activo.

Los estudios mencionados anteriormente se resumen en la Tabla 1-I. Como se puede apreciar, el análisis de logs y registros del sistema representa una buena alternativa para la detección de APT, razón por la cual se estudiaron en esta propuesta de investigación. Teniendo en cuenta los antecedentes de los investigadores, se utilizaron las cadenas de Markov, debido a que se ha demostrado que son eficaces para detectar patrones anómalos.

TABLA 1-I
Compilación de estudios del estado del arte

Método \ Información analizada	**Proxy Logs**	**DA Logs**	**DNS Logs**	**Uso de memoria**	**Uso de CPU**	**Puertos abiertos**	**Archivos System32**	**Peticiones HTTP**	**Tráfico de red**
Stabilized Nearest Neighbor (SNN)			[5]						
Extreme Learning Machine (EML)			[6]						
Teoría de grafos								[7]	
Algoritmo de propagación de creencias	[8]		[8]						
Cadenas de Markov	[9]	[9, 18]							
Aprendizaje Automático/Correlación de eventos	[11]	[10]	[11]					[11, 14, 15]	[11, 16]
Árboles de decisión			[13]	[12]	[12]	[12]	[12]	[13]	[13]

Nota: Se resumen de estados del arte relacionados con el tema de investigación.

1.2 Marco teórico

1.2.1 Amenazas Persistentes Avanzadas (APT)

Las Amenazas Persistentes Avanzadas, también conocidas por sus siglas en inglés, APT (*Advanced Persistent Threat*) se definen como la capacidad que tiene un adversario para utilizar un amplio conocimiento y herramientas que le permiten vulnerar sistemas por medio de múltiples vectores de ataque, con los cuales busca robar información o afectar misiones, programas u organizaciones [19]. Como se puede apreciar, esta definición describe el perfil y el objetivo por parte de un atacante que ejecuta un ciber ataque con características especiales.

El nombre atribuido a este tipo de amenazas engloba su verdadera naturaleza, puesto que son utilizadas por los atacantes para ocasionar daño, interrumpir servicios o robar información privilegiada. Son persistentes porque logran perdurar en los sistemas por mucho tiempo sin ser detectados. Finalmente, son avanzadas porque son ejecutadas por atacantes organizados, con altos conocimientos en seguridad de la información, con capacidad de hacer uso de malware ya existente para ejecutar sus ataques y con habilidades para explotar vulnerabilidades para las cuales no se tienen parches de seguridad (0-day).

Las APT inicialmente estaban dirigidas a objetivos políticos y militares, pero luego migraron hacia objetivos industriales y empresariales para obtener ganancias financieras. En el 2013, los sectores más atacados por APT fueron el educativo, financiero y tecnológico; otros sectores como el gubernamental, químico, telecomunicaciones, consultoría, energía, salud y aeroespacial también se vieron afectados [20].

Cada amenaza de este tipo se caracteriza también por ser específica para cada objetivo. Sin embargo, las APT tienen unas fases en común que se diferencian por las herramientas utilizadas para pasar de una a otra [21].

Las fases en mención se resumen a continuación:

- **Reconocimiento:** en esta fase, los atacantes obtienen la información del objetivo, realizan un perfilamiento del personal y los servicios de red, identifican puertos abiertos y dispositivos utilizados para proteger el perímetro de la red, información de clientes y proveedores, etc. Para hacerlo, utilizan algunas técnicas entre las que se destaca la ingeniería social.
- **Acceso:** en esta fase, los atacantes preparan la herramienta para acceder al sistema, bien sea mediante un troyano, una URL maliciosa o un correo electrónico. Este último es uno de los vectores de ataque más común.
- **Explotación:** una vez se haya descargado e instalado el malware, el atacante establece una conexión con el centro de mando y control (C&C) desde donde analiza la red interna, obtiene credenciales de acceso, nombres de red, correos electrónicos, etc.
- **Operación:** en este punto de la infección por APT los atacantes localizan la información objetivo, identifican usuarios clave y escalan permisos en la red para acceder a la información sensible. Esta etapa requiere de un trabajo de

persistencia por parte de los atacantes, así como de pericia para moverse de manera transversal en los sistemas sin ser detectado.
- **Recolección de información:** en esta fase los atacantes acceden a la información que requieren para empaquetarla, comprimirla y cifrarla.
- **Exfiltración:** finalmente, los atacantes establecen canales seguros y cifrados a servidores externos para extraer la información obtenida. [22, 23, 24]

Algunas de las APT que han sido estudiadas en los últimos años son las siguientes:

- **Operación Aurora**, detectada en febrero de 2010, la cual estuvo activa desde junio hasta diciembre de 2009 [25].
- **Flame y Stuxnet**, ambas detectadas en el año 2010 (mayo y junio, respectivamente) [25].
- **RSA Breach**, detectada en 2011 [26].
- **Operación SnowMan**, identificada en 2014 [26].
- Y algunas otras como **Glassrat** [27] y **Moonlight** [28] detectadas en 2015 y 2016 respectivamente.

Actualmente, existen algunas técnicas de defensa para este tipo de ataques. Sin embargo, no hay una solución única que brinde protección frente a las APT. Sumado a esto, las herramientas existentes deben adaptarse a su contexto particular, debido a que cada una tiene un comportamiento diferente, como se mencionó anteriormente. Algunas de las técnicas de defensa son [26]:

- entrenamiento en seguridad para prevenir ataques de ingeniería social (uno de los mecanismos utilizados por atacantes en la fase de reconocimiento);
- mecanismos de defensa tradicionales para identificar accesos no autorizados y vectores de infección, entre los que se encuentran los antivirus, firewalls, IPS/IDS, SIEM (Security Information and Event Management), entre otros;
- detección avanzada de malware mediante el análisis comportamental de amenazas;
- detección anómala de eventos;
- herramientas para la prevención de pérdida de datos (DLP), y
- ciber inteligencia.

En este punto cabe resaltar que, pese a la existencia de soluciones como las ya mencionadas, el hecho de que no sean únicas las hace débiles. En este sentido, este tipo de amenazas son creadas específicamente para la organización objetivo y utilizan firmas únicas y difíciles de correlacionar con las de ataques conocidos, por lo cual no son fácilmente detectables por muchos antimalware e IPS existentes en el mercado. Además, los ataques de APT habitualmente se desarrollan en largos periodos de tiempo y no de forma periódica, por lo tanto, es complicado correlacionar alertas basándose en los datos de fechas u horas. Adicionalmente, el tráfico de datos suele ser cifrado y utiliza técnicas que ocultan su ilegitimidad.

Sumado a lo anterior, el hecho de que cada año surgen nuevas variantes de APT en el mundo, como se puede apreciar en la Figura 1-1 (en 2016 se reportaron alrededor de 90

nuevos casos —Github—). Por esta razón, el tema sigue siendo vigente en el ámbito investigativo.

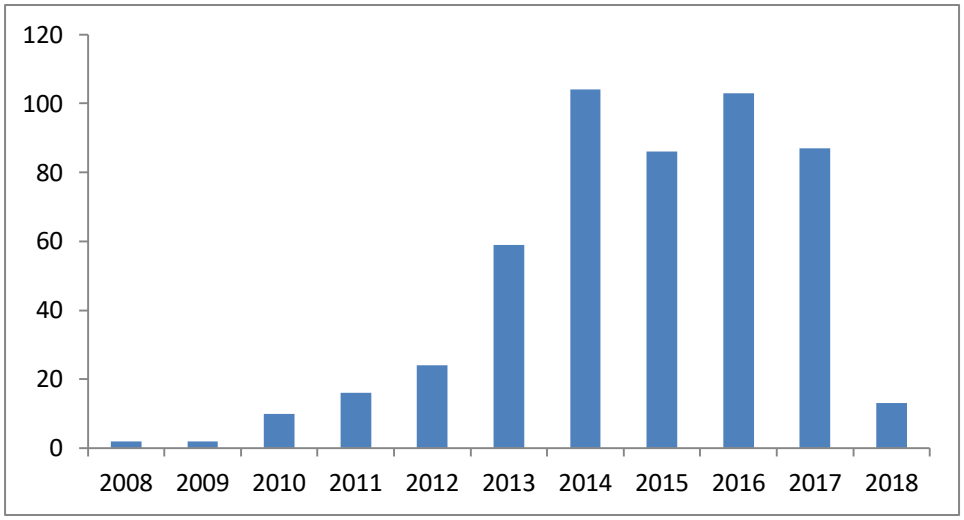


Fig. 1-1. Actividad de APT reportadas por año. [29]

1.2.2 Redes informáticas

En términos generales, una red informática es un conjunto de dispositivos que se encuentran conectados entre sí a través de diferentes elementos como cables, fibras ópticas, enlaces inalámbricos u otros [30], que permiten realizar tareas diversas a medida que se añaden periféricos, como impresoras, escáneres, módems y demás.

Cada una de las computadoras conectadas a la red se conocen como “nodos”, y su conexión con otros equipos hace que se forme una red informática en la que todas las computadoras que la componen puedan comunicarse entre sí para realizar intercambio de datos y compartir recursos y servicios.

Para que una red informática funcione se requiere de una serie de componentes claves que permiten la compartición de dispositivos, funciones y características. Mientras más grande sea una red, mayor cantidad de elementos necesitará para que funcione de manera eficiente. En este sentido, los principales elementos de una red son servidores (equipos que suministran información a clientes), clientes (equipos que acceden a la red), medios de transmisión (instalaciones utilizadas para interconectar los equipos), datos compartidos, tarjetas de red, enrutadores y switches [31]. Todos estos componentes interactúan entre sí para facilitar la compartición de datos a través de la red.

Ahora bien, el sistema básico de comunicación de datos se compone de cinco elementos:

1. **Mensaje:** es la información que se transmite a través de la red, el cual puede ser texto, números, imágenes, audio y video o cualquier combinación de estos.

2. **Transmisor:** es el dispositivo que envía el mensaje de datos. Puede ser una computadora, una estación de trabajo, un teléfono, una cámara de video o cualquier otro dispositivo.
3. **Receptor:** es el dispositivo que recibe el mensaje.
4. **Medio:** es el camino físico por donde viaja un mensaje.
5. **Protocolo:** es el conjunto de reglas que gobierna la comunicación de datos. Representa un acuerdo entre los dispositivos que se comunican.

- **Estructura de los paquetes de red**

El tráfico de red circula en forma de paquetes —conocidos también como datagramas— que consisten en datos encapsulados con información sobre su transporte (como la dirección IP de destino) [32].

Los routers analizan y modifican los datos contenidos en los datagramas para que puedan viajar de un lugar a otro en la red. La Figura 1-2 ilustra el formato de cabecera TCP:

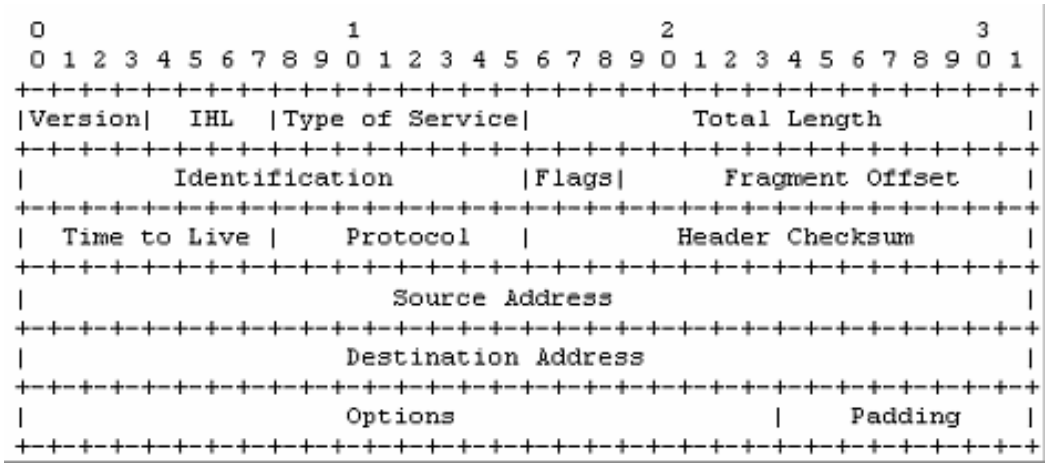


Fig. 1-2. Formato de cabecera TCP. [33]

- **Versión:** tiene una longitud de 4 bits que indica la versión del formato de la cabecera IP. Si es 4 (0100 en binario), corresponde a la versión IPv4, y si el valor es 6, se trata de la versión IPv6.

- **IHL:** longitud de la cabecera en palabras de 32 bits. Este campo indica en qué punto o bit termina la cabecera del datagrama. Se requiere este campo, debido a que el datagrama es de longitud variable.

- **ToS:** tipo de servicio respecto a la fiabilidad, velocidad, retardo y seguridad deseados. Tiene un tamaño de 8 bits, y es utilizado en algunas redes para ofrecer prioridad de servicio.

Los valores de bits para el ToS son:

- Bits 0 - 2 Prioridad.
- Bit 3 0 = Demora normal, 1 = Baja demora.
- Bit 4 0 = Rendimiento normal, 1 = Alto rendimiento.
- Bit 5 0 = Fiabilidad normal, 1 = Alta fiabilidad.
- Bits 6 - 7 Reservado para uso futuro.

- Los bits de prioridad están relacionados con la procedencia de los mensajes e indica el nivel de urgencia de los mismos.

- **Total length:** longitud total del datagrama medida en octetos. Incluye los datos encapsulados, cabecera y datos.

- **Identification:** número de identificación único de cada datagrama que permitirá el ensamblaje posterior al ser dividido en fragmentos más pequeños. Tiene una longitud 16 bits.

- **Flags:** campo de 3 bits usado como indicadores de control en caso de desfragmentación.
- **Fragment Offset:** es la posición del fragmento dentro del datagrama en caso de fragmentación.

- **TTL (Time to Live):** tiene una longitud de 8 bits. Impide que un paquete esté indefinidamente viajando por la red e indica el máximo número de enrutadores que un paquete puede atravesar.

- **Protocol:** se refiere al protocolo de siguiente nivel al que debe entregarse el paquete. Los valores para los protocolos pueden ser: 1 ICMP, 2 IGMP, 6 TCP, 9 IGRP, 17 UDP, 47 GRE, 50 ESP, 51 AH, 88 EIGRP, 89 OSPF, 115 L2TP.

- **Header Checksum:** CRC o Suma de Control de la Cabecera. Tiene una longitud 16 bits. Es la suma de comprobación de errores de la cabecera del datagrama.

- **Source Address:** dirección IP de origen.

- **Destination Address:** dirección IP de destino.

Los campos *Options* y *Padding* son de longitud variable con información opcional para el datagrama.

El entendimiento de la estructura de los paquetes de red será de gran utilidad para los análisis de tráfico que se hicieron más adelante.

1.2.3 Cadenas de Markov

Antes de exponer este tema, cabe aclarar que la primera parte de este apartado se apoyó en B. Bylina y J. Bylina [34] y Hillier y Gerald [35].

El modelamiento de redes es una herramienta útil para entender su comportamiento. Este permite identificar congestiones y cuellos de botella, la probabilidad de pérdida de paquetes, además de probar topologías, arquitecturas y configuraciones de manera que se puedan optimizar los recursos [34]. En general, los resultados de estos análisis permiten diseñar mecanismos de control de red en aspectos como control de admisión, control de flujo, control de congestión, control de memoria, asignación de recursos (ancho de banda en los enlaces y memoria en buffers de transmisión), caché dinámico, enrutamiento dinámico, etc.

Las redes de comunicaciones dependen de muchos factores relacionados entre sí, por lo tanto, pueden ser representadas como estaciones de servicio con colas y clientes que viajan a través de ellas, donde las estaciones son los componentes de la red como switches y enrutadores, y los clientes son representados por paquetes. Las colas son dichos paquetes que esperan en los dispositivos para ser enviados. Como la demanda es de naturaleza estadística, es posible entonces representarla mediante un proceso estadístico adecuado.

Los modelos de cola logran describir el comportamiento real de las redes y permiten un análisis de eficiencia relativamente fácil. Estos modelos se pueden resolver de varias maneras matemáticas, como el análisis del valor medio, los modelos markovianos, la aproximación de la difusión, el cálculo de la red, la aproximación del flujo de fluidos, entre otros.

Las cadenas de Markov son procesos estocásticos o modelos estadísticos que permiten examinar el estado de un sistema en un periodo de tiempo. Un proceso estocástico $X(t)$ es un conjunto de variables aleatorias, definidas en el mismo espacio de probabilidad e indexadas por t [35]. Para cada $t = t_i$ se define una función de distribución (ver Ecuación (1.1)):

$$F_X(x; t_i) = P[X(t_i) \leq x] \tag{1.1}$$

Los valores de las variables forman el espacio de estados del proceso, que puede ser continuo o discreto, y este último caso se denomina cadena. Las dependencias estadísticas entre $X(t_i)$ para diferentes t_i se pueden describir mediante una función de distribución n-dimensional (ver Ecuación (1.2)).

$$F_{X_1,\dots,X_n}(x_1, \dots, x_n; t_1, \dots, t_n) = P[X(t_i) \leq x_1, \dots, X(t_n) \leq x_n] \tag{1.2}$$

Los procesos de Markov son una clase especial de procesos estocásticos debido a que cumplen con la propiedad markoviana (ver Ecuación (1.3)):

$$P[X(t) \leq x \mid X(t_n) \leq x_n, \dots, X(t_0) \leq x_0] = P[X(t) \leq x \mid X(t_n) \leq x_n] \tag{1.3}$$

Lo anterior quiere decir que la probabilidad condicional de cualquier evento futuro, dado cualquier evento pasado y el estado actual, es independiente del evento pasado y solo depende del estado actual del proceso.

El espacio de estado de una cadena de Markov generalmente se mapea a un subconjunto del conjunto de números naturales. El parámetro t puede pertenecer a un

conjunto continuo o discreto. En el primer caso, se tiene una cadena de Markov de tiempo continuo (CTMC) y en el segundo una de tiempo discreto (DTMC). El sistema modelado y la cadena de Markov que lo representa adoptan exactamente un estado en cualquier momento t. La evolución del sistema modelado se representa mediante transiciones entre estados en dicha cadena.

Las probabilidades condicionales $PX_{t+1} = j\ X_t = i$ para una cadena de Markov se llaman probabilidades de transición (de un paso) si para cada i y j (ver Ecuación (1.4)):

$$PX_{t+1} = j\ X_t = i\ =\ PX_1 = 1\ X_0 = i\ \text{, para toda } t = 1, 2, \ldots. \qquad (1.4)$$

Entonces se dice que las probabilidades de transición (de un paso) son estacionarias. Así, tener probabilidades de transición estacionarias implica que no cambian con el tiempo. La existencia de estas probabilidades también implica que, para cada para cada i y j y n $(n = 0,1,2, \ldots,)$ (ver Ecuación (1.5)):

$$PX_{t+n} = j\ X_t = i\ =\ PX_n = j\ X_0 = i \qquad (1.5)$$

Para toda $t = 1, 2, \ldots.$ estas probabilidades condicionales se llaman probabilidades de transición de *n* pasos.

Así, las probabilidades de transición de n pasos ${P_{ij}}^{(n)}$son simplemente la probabilidad condicional de que el sistema se encuentre en el estado j justo después de n pasos (unidades de tiempo), dado que comenzó en el estado i en cualquier tiempo t. Cuando n = 1, observe que ${P_{ij}}^{(1)} = {P_{ij}}^{(2)}$.

Como las ${P_{ij}}^{(n)}$ son probabilidades condicionales, deben ser no negativas y, como el proceso debe hacer una transición a algún estado, deben satisfacer las propiedades (ver Ecuaciones (1.6 y 1.7)):

$${P_{ij}}^{(n)} \geq 0, \text{ para toda } i\ y\ j;\ n = 0, 1, 2, \ldots. \qquad (1.6)$$

$$\sum_{j=0}^{M} {p_{ij}}^{(n)} = 1, \text{ para toda } i;\ n = 0, 1, 2, \ldots. \qquad (1.7)$$

Una notación conveniente para representar las probabilidades de transición de n pasos es la forma matricial (ver Ecuación (1.8)):

$$P^{(n)} = \begin{bmatrix} {p_{00}}^n & \cdots & {p_{01}}^n & \cdots & {p_{0M}}^n \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ {p_{10}}^n & \cdots & {p_{11}}^n & \ldots & {p_{1M}}^n \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ {p_{M0}}^n & \ldots & {p_{M1}}^n & \ldots & {p_{MM}}^n \end{bmatrix} \qquad (1.8)$$

Se puede observar que la probabilidad de transición en un renglón y columna dados es para la transición del estado en ese renglón al estado en la columna.

Se dice que el estado j es accesible desde el estado i si ${P_{ij}}^{(n)} \geq 0$ para alguna $n \geq 0$ teniendo en cuenta que ${P_{ij}}^{(n)} \geq 0$ es solo la probabilidad condicional de llegar al estado j

después de n pasos, si el sistema está en el estado i. Entonces, que el estado j sea accesible desde el estado i significa que es posible que el sistema llegue eventualmenteal estado j si comienza en el estado i. En general, una condición suficiente para que todos los estados sean accesibles es que exista un valor de n para el que ${P_{ij}}^{(n)} \geq 0$ para todo i y j.

La cadena de Markov basada en el tráfico de red es entonces una secuencia de todos los posibles sucesos que pueden ocurrir en la red, por ejemplo, podría mostrar cuánto es el tráfico de red (en Kb/s) en un en un momento dado.

A continuación, se describen algunos elementos importantes en la cadena de Markov, tomados de Prada [36]:

- **Distribución inicial**

Una cadena de Markov está completamente determinada por la distribución de probabilidad del estado inicial X_0 y por las probabilidades de transición p_{ij}. La distribución inicial de la cadena se expresa en forma vectorial, de manera que cada entrada indica la probabilidad de que la cadena se encuentre en el estado i en el instante inicial. De esta forma, se conoce el punto de partida del proceso [36]. El vector se expresa así (ver Ecuación (1.9)):

$$P^{(0)} = ({p_1}^{(0)}, \dots, {p_i}^{(0)}, \dots, {p_k}^{(0)}), \qquad (1.9)$$

Donde ${p_1}^{(0)} = P(X_0 = i)$, $i \in E$, y E es el conjunto de posibles valores que puede tomar el proceso en cada una de sus etapas. Además, la distribución inicial cumple: ${p_1}^{(0)} \geq 0$ y $\sum_{i \in E} {p_1}^{(0)} = 1$

- **Distribución de probabilidad en la etapa n-ésima**

Una vez definida la cadena de Markov en tiempo discreto, se puede obtener la distribución de la cadena en la etapa n- ésima X_n (distribución marginal).

Si se denota la probabilidad de que en la etapa *n* la cadena de Markov se encuentre en el estado i por ${p_i}^{(n)} = P(X_n = i)$, para cada etapa se tiene un vector $P^{(n)} = ({p_1}^{(n)}, \dots, {p_i}^{(n)}, \dots, {p_k}^{(n)})$ que representa la probabilidad de que la cadena se encuentre en cada uno de los posibles estados en la etapa *n*. Para obtener esta distribución se calcula previamente la matriz de probabilidades de transición en *n* etapas.

Partiendo de que se conoce la probabilidad de transición en una etapa dada, el paso siguiente es obtener la probabilidad de transición en *n* etapas, ${p_{ij}}^{n} = P(X_{m+n} = j | X_m = i), \forall n, m \in \mathbb{N}; i, j \in E,$ calculando la potencia n-ésima de la matriz de transición P a partir de la ecuación de Chapman-Kolmogorov para cadenas de Markov en tiempo discreto (ver Ecuación (1.10)):

$${p_{ij}}^{(n)} = \sum_{g \in E} {p_{ig}}^{(m)} {p_{gj}}^{(n)}, \forall n, m \in \mathbb{N}\ i, j \in E \qquad (1.10)$$

Intuitivamente, para pasar del estado i al j en $(m+n)$ etapas, se debe transitar por un estado g en m etapas y después ir desde g hasta j en las n etapas restantes. La condición de Markov implica que las dos partes de la transición de i a j son independientes, por lo que se puede escribir ${p_{ij}}^{(m+n)}$ como el producto de las probabilidades de transición y, por lo tanto, $P^{(m+n)} = P^{(m)}P^{(n)}$, donde $P^{(m+n)}$ es la matriz de transición de la etapa $(m+n)$ y tiene como elementos $(P^{(m+n)})$.

Tomando n = 1 en la ecuación anterior, se obtiene (ver Ecuación (1.11)):

$${p_{ij}}^{(m+1)} = \sum_{g \in E} {p_{ig}}^{(m)} p_{gj}, \forall n, m \in \mathbb{N}\ i, j \in E \qquad (1.11)$$

Se tiene entonces que $P^{(1)} = P, P^{(2)} = P^{(1)}P^{(1)} = P^{(2)}$, y así sucesivamente hasta que se obtiene $P^{(n)} = P^{(n-1)}P = P^{(n-2)}P^{(2)} = \cdots P^{(0)}P^{(n)}$, y con esta fórmula se tiene la distribución de la cadena en la etapa *n*. Por lo tanto, la probabilidad de transición en *n* etapas, ${p_{ij}}^{n} = P(X_{m+n} = j | X_m = i)$, se obtiene del elemento (i, j) de la n-ésima potencia de la matriz de transición P; $\forall n, m \in \mathbb{N}\ i, j \in E$.

- **Orden de una cadena de Markov**

El orden de una cadena de Markov establece el número de estados anteriores de los cuales depende la probabilidad de un estado en un instante determinado del proceso. Así, dado $E = \{E_1, \ldots E_k\}$, en una cadena de primer orden se tiene (ver Ecuación (1.12)):

$$P(X_{n+1} = i_{n+1} | X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n) = P(X_{n+1} = i_{n+1} | X_n = i_n) \qquad (1.12)$$

Y en una cadena de orden dos se tiene (ver Ecuación (1.13)):

$$P(X_{n+1} = i_{n+1} | X_0 = i_0, \ldots, X_{n-1} = i_{n-1} = X_n = i_n) = P(X_{n+1} = i_{n+1} | X_{n-1} = i_{n-1}, X_n = i_n) \qquad (1.13)$$

Para $i \in E\ y\ n \in \mathbb{N}$, análogamente, se definirá una cadena de Markov de orden mayor que dos.

- **Clasificación de estados**

Los estados de una cadena de Markov pueden clasificarse de acuerdo con las transiciones permitidas entre estos y las probabilidades de pasar de un estado a otro. Para realizar dicha clasificación, es necesario definir los conceptos de probabilidad y tiempos de primera pasada:

Probabilidad de primera pasada: es la probabilidad de que, empezando en i, la cadena pase por primera vez por el estado j en la etapa *n*. Se denota por (ver Ecuación (1.14)):

$${f_{ij}}^{(n)} = P(X_n = j, X_r \neq j, \forall r < n | X_0 = i) \qquad (1.14)$$

Probabilidad de pasada: se trata de la probabilidad de que la cadena llegue alguna vez al estado j, partiendo del estado i (ver Ecuación (1.15)):

$$f_{ij} = \sum_{n=1}^{\infty} {f_{ij}}^{(n)} = P(\exists n;\ X_n = j |\ X_0 = i) \qquad (1.15)$$

Tiempo de primera pasada: es el número de etapa en que la cadena llega por primera vez al estado j cuando parte del estado i. Se determina por la siguiente variable aleatoria:

$N_{ij} = \{de\ la\ primera\ etapa\ en\ la\ cual\ la\ cadena\ está\ en\ j\ partiendo\ de\ i\}$. Se describe entonces (ver Ecuación (1.16)):

$$f_{ij}{}^{(n)} = P(N_{ij} = n), f_{ij} = \sum_{n=1}^{\infty} f_{ij}{}^{(n)} = P(N_{ij} < \infty) \qquad (1.16)$$

Se establece la siguiente relación entre las probabilidades de transición en *n* etapas, $p_{ij}{}^{n}$, y las probabilidades de primera pasada, $f_{ij}{}^{(n)}$ (ver Ecuación (1.17)):

$$p_{ij}{}^{n} = f_{ij}{}^{(1)} p_{ij}{}^{(n-1)} + f_{ij}{}^{(2)} p_{ij}{}^{(n-2)} + \cdots + f_{ij}{}^{(n-1)} p_{jj} + f_{ij}{}^{(n)} \qquad (1.17)$$

Así, los estados de una cadena de Markov en tiempo discreto pueden clasificarse en:

- **Estado recurrente:** un estado $j \in E$ se dice que es recurrente si es seguro que la cadena va a volver al estado j una vez que ya ha llegado a él en alguna etapa, es decir, $f_{ij} = 1$.
- **Estado transitorio:** se dice que un estado $j \in E$ es transitorio si no es recurrente, es decir, $f_{ij} < 1$. En este caso se espera un número finito de visitas a dicho estado.
- **Estado que comunica con otro estado:** se dice que un estado $i \in E$ comunica con otro estado $j \in E$, $f_{ij} > 0$, es decir, existe la posibilidad de que la cadena llegue al estado j partiendo del estado i.
- **Estado que intercomunican:** dos estados $i, j \in E$ intercomunican si i comunica con j y j comunica con i.
- **Estado efímero:** se dice que un estado $j \in E$ es efímero si $p_{ij} = 0$, $\forall i \in E$, esto significa que no se puede llegar a él desde ningún otro, solo se puede salir desde él hacia cualquier otro.
- **Estado absorbente:** un estado $j \in E$ es absorbente si es imposible abandonarlo, es decir, $p_{ij} = 1$. Una vez se alcanza, la cadena solo puede mantenerse en él.
- **Estado periódico:** un estado $j \in E$ se dice que es periódico con periodo $l > 1$ si l es el menor número de tal manera que todas las secuencias de transiciones que parten del estado i y regresan al estado i tienen una longitud múltiplo de l. Si un estado no es periódico se llama aperiódico.

El estado del arte muestra que las Amenazas Persistentes Avanzadas son un tema de estudio relevante en el ámbito de la seguridad informática debido a su alto impacto en organizaciones y al incremento en el número de campañas que se presentan año a año, explicado por el beneficio que trae para atacantes. Los estudios muestran también que el análisis estadístico aplicado a la detección de amenazas informáticas es de gran aporte por su adaptabilidad y efectividad, convirtiéndose así en una herramienta útil para estudiar el comportamiento de APT.

2. Metodología

En este apartado se hace referencia a la metodología aplicada y a las herramientas utilizadas para desarrollar este trabajo de grado y alcanzar los objetivos propuestos. La metodología se divide en enfoque, método, instrumentos de recolección de información y procedimiento de análisis y se sintetiza en la Tabla 2-I. El detalle se desarrolla a lo largo del capítulo.

TABLA 2-I
Resumen de metodología

Fase	Objetivo	Herramienta utilizada	Resultado esperado/Entregable
1. Mecanismo para el pre-procesamiento de datos	Obtener tráfico de red del sistema infectado por APT y estructurar los datos en forma de tabla para facilitar su análisis	VMWare; D-ITG, PackeTH; Wireshark; Excel	Tabla en Excel con el tráfico de red capturado
2. Elementos del tráfico de red que permiten identificar patrones de infección por APT	Caracterizar los elementos del tráfico de red que revelan patrones de infección por APT	Wireshark; Excel	Listado de protocolos con mayor número de paquetes TCP y su frecuencia dentro de la data. Descripción de los patrones de comunicación observados. Análisis de datos atípicos. Listado de direcciones IP y dominios maliciosos
3. Representación del tráfico de red de un sistema mediante cadenas de Markov	Modelar el comportamiento de la red utilizando cadenas de Markov para detectar anomalías	Excel	Cadenas de Markov con las probabilidades de transición del sistema
4. Eficacia de la evaluación realizada	Diseñar e implementar reglas de red que permiten identificar patrones asociados con infecciones por APT en un sistema de detección de intrusos (IDS)	Snort	Listado de reglas para el IDS

Nota: Cuadro resumen de la metodología aplicada en el trabajo de investigación. Se muestra la fase de la metodología con su objetivo, las herramientas que se utilizaron y su resultado esperado.

2.1 Enfoque

El enfoque de investigación empleado fue el cuantitativo, puesto que a través de la recolección de información proveniente de una red de área local se analizó el

comportamiento del tráfico normal y anormal; es decir, el tráfico afectado por la actividad de una APT, lo cual permitió identificar patrones de comportamiento asociados con este tipo de amenazas. El enfoque cuantitativo facilita el análisis y la descripción de la realidad al contemplar el sistema con toda su complejidad. Al respecto, Hernández, Fernández y Baptista [37] destacan que este enfoque posibilita emplear instrumentos cerrados, registros estadísticos y otros métodos estandarizados y de medición para el análisis del fenómeno.

Precisamente, la técnica empleada para la recolección información fue la simulación de tráfico en ambientes virtuales controlados. En este caso se utilizaron el software de virtualización VMWare, generadores de paquetes como D-ITG y PaqueTH y el analizador de protocolos Wireshark.

Igualmente, para el análisis de la información recolectada se utilizaron métodos de medición, específicamente las cadenas de Markov, así como estadística descriptiva en el entorno estadístico R. Los resultados obtenidos se resumen en tablas y gráficos. Además, se describen y analizan datos objetivamente.

2.2 Método

El método utilizado en este trabajo de investigación fue el analítico. En general, este método consiste en la descomposición del objeto de estudio en sus partes o elementos para observar sus causas, su naturaleza y sus efectos. Así pues, el método analítico es apropiado en esta investigación debido a que permite conocer en profundidad el objeto de estudio, con lo cual se puede explicar y comprender mejor su comportamiento.

Para utilizar este método se siguieron sistemáticamente los siguientes pasos:

- **Observación:** en este paso se recopiló la información que fue objeto de estudio, es decir, el tráfico de red mediante la utilización de diversos instrumentos. El estudio de muestras y la experimentación es una parte vital en esta investigación —y en general del método analítico—, por tanto, se puso especial atención en los mecanismos de obtención para no afectar los resultados.
- **Descripción:** en este paso se da una idea general de lo que se observó en los datos de tráfico recolectados. La descripción es importante porque permite obtener las características del tráfico de red afectado por APT, sus propiedades y atributos, lo cual aportó información útil sobre el objeto de estudio con la mayor cantidad de detalles posibles. Aquí se describieron los protocolos, número y tamaño de paquetes que intervienen en la comunicación, direcciones IP y contenido de los mensajes.
- **Segmentación del fenómeno**: en este punto se trató de descomponer las partes del tráfico analizado a partir de un examen de sus partes para visualizar desde varios puntos de vista el comportamiento del sistema bajo una APT.
- **Examen crítico:** en este paso se buscaron respuestas lógicas a las características identificadas en el tráfico y se interpretaron las observaciones de

forma clara y concisa. Se hizo uso de elementos estadísticos como lo son las cadenas de Markov para explicar con cifras el comportamiento del sistema. Finalmente, las respuestas obtenidas fueron utilizadas posteriormente para diseñar estrategias de detección.

No se consideraron aisladamente las partes del objeto de estudio, es decir, se tuvieron en cuenta todos los elementos del tráfico y sus relaciones para no formar ideas inexactas y erróneas.

Es importante destacar que este método está abierto a la incorporación de nuevos conocimientos y procedimientos con el fin de asegurar un mejor acercamiento a la verdad. Igualmente, cabe señalar que el método aplicado no es exploratorio, ya que este se utiliza en investigaciones inéditas, es decir cuando no hay estado del arte u otros referentes para hablar del tema.

2.3 Instrumentos de recolección de información

Para el desarrollo de este trabajo de investigación se usaron las siguientes herramientas:

2.3.1 Herramienta para virtualización de ambientes: VMWare WorkStation Player 14

Es la aplicación que se utilizó para la virtualización de escritorios. Asimismo, el montaje de las máquinas del laboratorio se realizó en esta aplicación.

VMware Workstation Player es una aplicación de escritorio estándar que permite instalar sistemas operativos nuevos y ejecutarlos como máquinas virtuales en una ventana por separado. Este software ofrece funcionalidades que permiten crear y configurar las máquinas virtuales y acceder a los dispositivos conectados a la PC.

Se selecciona esta herramienta considerando los siguientes aspectos:

- Es un software de virtualización simple y potente. Es una de las soluciones más maduras y estables para la virtualización de escritorios. Se ejecuta sobre un sistema operativo como una máquina virtual sin afectar su entorno de escritorio principal.
- La herramienta tiene capacidades de aislamiento que permiten explorar las amenazas en un entorno “real” sin interferir con el escritorio host. Cuenta con diferentes configuraciones de privacidad, de red y herramientas para mantener el host seguro y protegido mientras navega en línea.
- Está disponible de forma gratuita para uso no comercial y es una de las mejores herramientas de virtualización del mercado [38].

2.3.2 Herramienta para la generación de tráfico

Para seleccionar la herramienta de generación de tráfico se consideraron varias opciones, teniendo en cuenta que la calidad del tráfico simulado depende de la calidad de la herramienta de simulación en sí.

Las herramientas seleccionadas pueden dar lugar al tráfico de red similar, es decir, al tráfico ajustado a un sistema real, ya que permite interrelacionar los tres parámetros relevantes desde la perspectiva de la red: el tiempo de aparición, su longitud en bytes y su dirección de destino.

Las características consideradas en la selección de la herramienta de generación de tráfico lícito se resumen en la Tabla 2-I:

TABLA 2-II
Comparación de herramientas para la generación de tráfico lícito

Aspecto	D-ITG	PACKETH	OSTINATO	TOMAHAWK
Gratis	Sí	Sí	No	Sí
OpenSource	Sí	Sí	Sí	Sí
Privilegios	Usuario	Usuario	Usuario	Usuario
Plataforma soportada	Linux/Windows	Linux, Mac OS X, Windows	Windows, Mac OS X, Linux, FreeBSD	RedHat (Linux)
Protocolo de red	IPv4, IPv6, ICMPv4, ICMPv6	ARP, IPv4, IPv6	ARP, IPv4, IPv6, IP-in-IP	IP
Protocolo de transporte	TCP, UDP	TCP, UDP, ICMPv4, ICMPv6	TCP, UDP, ICMPv4, ICMPv6, IGMP, MLD	TCP
Resultados reportados	Retardo, jitter y throughput	Creación de paquetes Ethernet	Creación de streams	Rendimiento de los IPS
Interfaz de usuario	Consola	GUI, CLI	GUI	Consola
Número de fuentes donde se citan	151	19	181	461

Nota: Cuadro comparativo de herramientas para la generación de tráfico lícito [48]

Luego de evaluar los parámetros, se seleccionaron las aplicaciones D-ITG (versión 2.8.1) y PackeTH (versión 2.0), debido a que:

- la distribución del tiempo entre llegadas en el flujo generado es constante en el intervalo de tiempo de simulación;
- el flujo de tráfico generado puede ser estacionario; es decir, sus propiedades estadísticas no cambian significativamente con el tiempo;
- soportan múltiples protocolos útiles para el análisis; y
- se soportan en las plataformas utilizadas en el laboratorio.

2.3.3 Herramienta para captura de tráfico: Wireshark

Para la interceptación y captura del tráfico analizado se utilizó Wireshark (versión 2.6.5), herramienta que convierte la captura en un formato legible y de fácil interpretación.

Además, permitió identificar el tráfico que está cruzando la red, su frecuencia y la latencia que hay entre ciertos saltos.

Se selecciona esta herramienta debido a que admite una gran cantidad de protocolos de red, facilita el análisis de identidades IP y permite aplicar filtros para reducir el volumen de tráfico que captura.

2.3.4 Herramienta para la obtención de cadenas de Markov y para realizar análisis estadísticos

Actualmente, existen múltiples herramientas en el mercado que incluyen librerías y funcionalidades para obtener y analizar cadenas de Markov, entre las cuales se encuentran:

- Markov (https://www.itemsoft.com/markov.html).
- RAM Commander's Markov (https://aldservice.com/Reliability-Products/markov.html).
- Muninn (http://www.muninn.org/).
- QHQsv++ (https://www.wiwi.hu-berlin.de/de/professuren/vwl/oe/software/qhq).
- MATLAB (https://www.mathworks.com/products/matlab.html).
- R (https://www.r-project.org/).

Algunas de las principales funcionalidades de estas herramientas son:

- Construir diagramas de Markov.
- Obtener matrices de estados y transiciones.
- Obtener modelos de misión gradual.
- Realizar análisis de estado estacionario.
- Analizar modelos de transición de tiempo discreto y continuo.
- Definir estados y grupos de estados.
- Realizar análisis de disponibilidad y confiabilidad basado en el tiempo.

Luego de analizar las herramientas disponibles, se seleccionó la herramienta R para realizar análisis estadísticos, debido a que es una de las más completas y robustas que existen, es de uso libre, incluye librerías útiles para obtener cadenas de Markov y cuenta con amplia documentación sobre su uso.

Cabe mencionar que se utilizó Excel versión 2010 para apoyar el análisis de datos y facilitar la obtención de frecuencias, esto debido a que la estructura de los datos obtenidos tiene forma de tabla y, por tanto, pueden ser procesados de forma intuitiva en el software.

2.3.5 Herramienta para creación de reglas de red: Snort

Para la creación de reglas de red se usó Snort, un Sistema de detección de intrusos basado en red (NIDS) que implementa un motor de detección de ataques y barrido de puertos para registrar, alertar y responder ante cualquier anomalía previamente definida

como patrones de ataques, intrusiones, análisis de protocolos y puertos, etc., todo esto en tiempo real.

Se empleó Snort debido a que es uno de los IDS más usados, dispone de una gran cantidad de filtros o patrones predefinidos, está disponible de forma gratuita y funciona bajo plataformas Windows y UNIX/Linux. Este sistema implementa un lenguaje de creación de reglas flexibles, potente y sencillo y puede funcionar también como sniffer para ver el tráfico en la red y registrar paquetes para su posterior análisis.

2.4 Procedimiento de análisis

Las amenazas objeto de esta investigación fueron seleccionadas a partir de diversos criterios. Inicialmente, se realizó una caracterización de las APT con impacto significativo en las industrias y que, actualmente, continúan activas. En la Tabla 2-II se mencionan los grupos relevantes, el país al que pertenece el grupo atacante, el año de descubrimiento de sus operaciones, el objetivo del atacante, las técnicas y herramientas utilizadas en sus campañas y nombres alternativos —si tienen—. Es importante tener en cuenta que la APT no es una amenaza como tal, sino un grupo de atacantes con herramientas de hackeo especializadas; por tanto, el enfoque de selección se centra en la herramienta utilizada por dichos grupos.

TABLA 2-III
Principales actores APT y sus técnicas

Nombre	País	Año	Objetivo	Técnicas	Herramientas utilizadas	Otros nombres
APT28	Rusia	2017	Georgia, países y militares de Europa oriental, Organización del Tratado del Atlántico Norte (OTAN) y otras organizaciones de seguridad y empresas de defensa europeas.	Descargadores y puertas traseras. Utiliza el cifrado RSA para proteger los archivos y la información robada que se transfiere de la red de la víctima al controlador.	CHOPSTICK, SOURFACE, EVILTOS	Tsar Team
APT19	China	2017	Sector legal y de inversión	Phishing, explotación de vulnerabilidades de Microsoft Windows, uso de documentos de Microsoft Excel (XLSM) habilitados para macros.	BEACON, COBALTSTRIKE	Codoso Team
APT29	Rusia	2016	Gobiernos de Europa occidental, grupos de política exterior y otras organizaciones similares.	Transmisión de comandos y extracción de datos de redes comprometidas a través de redes sociales e imágenes con datos ocultos y cifrados.	HAMMERTOSS, TDISCOVER, UPLOADER, FAREIT	Cozy Bear, Cozy Duke
APT16	China	2015	Organizaciones japonesas y taiwanesas de alta tecnología, servicios gubernamentales, medios y servicios financieros.	*Spear phishing* con documentos señuelo.	IRONHALO, ELMER	
APT34	Irán	2014	Sector financiero, gubernamental, energético, químico y de telecomunicaciones.	Explotación de vulnerabilidades de Microsoft	POWBAT, POWRUNER, BONDUPDATER, URSNIF.	
APT32	Vietnam	2014	Sector de manufactura, productos de consumo, consultoría y hotelería.	*Spear phishing* través de Gmail, ingeniería social.	SOUNDBITE, WINDSHIELD, PHOREAL, BEACON, KOMPROGO	OceanLotus Group
APT17	China	2014	Gobierno de EE. UU., firmas de abogados internacionales y compañías de tecnología de la información.	Creación de perfiles y publicación en foros para incrustar variantes del malware utilizado.	BLACKCOFFEE	Tailgator Team, Deputy Dog
APT12	China	2014	Periodistas, gobierno, defensa de la base industrial.	Entrega de documentos de explotación mediante correos electrónicos de phishing desde cuentas válidas comprometidas.	RIPTIDE, HIGHTIDE, THREBYTE, WATERSPOUT	Calc Team, IXESHE, DynCalc
APT33	Irán	2013	Industria de la aviación	*Spear phishing* avanzado desde dominios conocidos, publicidad de video juegos	SHAPESHIFT, DROPSHOT, TURNEDUP, NANOCORE, NETWIRE, ALFA Shell	
APT1	China	2013	Tecnología de la información,	*Spear phishing* con archivos adjuntos	TROJAN.ECLTYS,	Unit 61398,

			aeroespacial, administración pública, satélites y telecomunicaciones, investigación y consultoría, energía, transporte, construcción y manufactura, servicios de ingeniería, electrónica, organizaciones internacionales, servicios legales de medios, publicidad y entretenimiento, navegación, productos químicos, servicios financieros, alimentación y agricultura, sanidad, metales y minería, educación.	maliciosos, enviado desde cuentas de correo de personas reales. Uso de puertas traseras disponibles públicamente y personalizadas.	BACKDOOR.BARKIOFORK, BACKDOOR.WAKEMINAP, TROJAN.DOWNBOT, BACKDOOR.DALBOT, BACKDOOR.REVIRD, TROJAN.BADNAME, BACKDOOR.WUALESS	Comment Crew
APT18	China	2010	Sector aeroespacial y defensa, construcción e ingeniería, educación, salud y biotecnología, alta tecnología, telecomunicaciones, transporte	Explotaciones de vulnerabilidades zero-day	Gh0st RAT	Wekby
APT3	China	2010	Aeroespacial y defensa, construcción e ingeniería, alta tecnología, telecomunicaciones, transporte	*Phishing* tradicional, spam, explotación de vulnerabilidades sin parche de Adobe Flash Player	SHOTPUT, COOKIECUTTER, SOGU	UPS Team, Gothic Panda, Buckeye
APT 10	China	2009	Contratistas de defensa, entidades gubernamentales y de salud.	*Spear phishing* avanzado de dominios conocidos, publicidad de video juegos	Haymaker backdoor, Scanbox, and the Bug Juice backdoor, Sogu, Poison Ivy, PlugX	Potassium, Red Apollo
APT10	China	2009	Empresas de construcción e ingeniería, aeroespaciales y de telecomunicaciones. Gobiernos de Estados Unidos, Europa y Japón	Phishing tradicional, acceso a las redes de las víctimas a través de proveedores de servicios gestionados.	HAYMAKER, SNUGRIDE, BUGJUICE, QUASARRAT	Menupass Team
Turla	Rusia	2008	Agencias gubernamentales	PDF exploits, descargas falsas de Flash Player, ataques de phishing dirigidos, ingeniería social, y explotación de vulnerabilidades zero-day.	Snake rootkit (Urbororos rootkit), Mini Duke, CosmicDuke	Waterbug, Krypton, Wipbot
APT30	China	2005	Miembros de la Asociación de Naciones del Asia Sudoriental (ASEAN por sus siglas en inglés)	Utiliza descargadores, puertas traseras, un controlador central y otros componentes diseñados para infectar unidades extraíbles y redes air-gap para robar datos.	SHIPSHAPE, SPACESHIP, FLASHFLOOD	

Nota: Grupos APT relevantes con las técnicas y herramientas que han utilizado [39, 40]

Luego de analizar los grupos APT y las herramientas utilizadas en sus campañas, se seleccionaron para el estudio tres amenazas: CosmicDuke, Fareit y URNISF, teniendo en cuenta los siguientes aspectos:

- disponibilidad de la cepa para propósitos de investigación;
- nivel de complejidad de la amenaza; y
- características de comportamiento y conocimiento existente de la amenaza.

Se selecciona la amenaza CosmicDuke por ser una herramienta utilizada por el grupo de atacantes conocido como Turla, con actividad conocida desde 2008. CosmicDuke es un híbrido entre las herramientas Cosmu y Mini Duke, utilizadas por el grupo y descubiertas en marzo de 2011 y en noviembre de 2012, respectivamente [41].

CosmicDuke tiene los mismos mecanismos de Cosmu para lograr la persistencia en el sistema. Generalmente, crea una tarea programada e instala un servicio de Windows. La tarea programada normalmente se denomina *Watchmon*, la cual ejecuta el malware al iniciar sistema. Por su parte, el servicio tiene el nombre *javamtsup*, que abre un identificador para el proceso explorer.exe, duplica su token de proceso, lee la ruta del malware real en el registro y lo inicia utilizando el token duplicado.

El malware tiene como objetivo robar contraseñas de aplicaciones como Skype, Google Talk, Google Chrome, Internet Explorer, Firefox, Thunderbird, Ŝ Bat email client, Outlook, Google Desktop, credenciales de Windows y WLAN. También cuenta con procesos para registrar pulsaciones de teclado (*keylogger*), tomar capturas de pantalla, robar datos del portapapeles, archivos y certificados PKI.

En la Tabla 2-III se resumen las características generales del malware CosmicDuke.

TABLA 2-IV
Características malware CosmicDuke

Nombre	CosmicDuke
Año	2010
Objetivo	La OTAN y agencias gubernamentales europeas.
Método de infección	Ingeniería social y exploits (objeto Flash malicioso embebido en archivos PDF). Hace uso de vulnerabilidad CVE-2011-0611 en productos Acrobat.
Propósito	Robo de información mediante keylogger; capturas de pantalla; robo de datos del portapapeles; robo de archivos; robo de certificados PKI y claves privadas asociadas; robo de nombres de usuario y contraseñas de navegadores: mensajería instantánea y clientes de correo electrónico; robo de contraseñas WLAN; y robo hashes de contraseña de Windows.
Transmisión de datos	Enviada a servidores externos a través de FTP.

Nota: Características relevantes de las amenazas estudiadas (año de descubrimiento, organizaciones afectadas, métodos utilizados por los grupos atacantes, propósito de la amenaza y mecanismo de exfiltración de información) [41].

Otro malware seleccionado para estudiar en esta investigación es Fareit. Como se aprecia en la Tabla 2-IV, esta herramienta fue utilizada por el grupo conocido como Cozy Bear en el ataque al Comité Nacional Demócrata de Estados Unidos (DNC) en el año

2016. Este malware, al ser un ladrón de contraseñas, es comúnmente utilizado en las primeras etapas de los ataques por APT [42], razón por la cual se selecciona para este estudio.

El malware de Fareit generalmente se entrega como la carga útil de otro malware a través de mensajes de correo electrónico no deseado o servidores DNS maliciosos.El objetivo principal de este malware es robar datos de usuario tales como cookies de navegador y credenciales de acceso para clientes FTP, correo electrónico y servicios de almacenamiento en la nube [43]. La información robada se envía a un servidor remoto.

Algunas variantes de Fareit descargan otros malware, mientras que otros permiten que un atacante remoto use el sistema afectado para participar en los ataques de Denegación de Servicio Distribuido (DDoS). En la Tabla 2-IV se resumen las características de la herramienta:

TABLA 2-V
Características malware Fareit

Nombre	Fareit
Año	2014
Objetivo	Gobiernos de Europa occidental, grupos de política exterior y otras organizaciones similares
Método de infección	Se entrega a través de archivos maliciosos adjuntos en correos electrónicos. Los usuarios reciben un correo electrónico con un archivo PDF o un documento de Word con código malicioso y que explota Windows PowerShell.
Propósito	Es utilizado por atacantes para descargar otros malware como ZeuS / ZBOT en sistemas infectados. Sus variantes suelen robar nombres de usuario y contraseñas, almacenados en navegadores web, además de robar credenciales de correo electrónico y credenciales FTP.
Transmisión de datos	La información es cifrada y enviada a servidores externos.

Nota: Características relevantes de las amenazas estudiadas (año de descubrimiento, organizaciones afectadas, métodos utilizados por los grupos atacantes, propósito de la amenaza y mecanismo de exfiltración de información) [42, 43, 44]

La tercera amenaza que se estudia en este trabajo de investigación es Ursnif, la cual es utilizada también en etapas tempranas de infección por APT. Ursnif es un caballo de Troya que roba información en los sistemas comprometidos.

El troyano puede enviarse a través de correo electrónico no deseado y malicioso que contiene un enlace a un archivo comprimido (zip). El archivo zip contiene un archivo JavaScript que, al ejecutarse, descarga y ejecuta el malware.

El troyano es capaz de realizar inyecciones web y robar credenciales bancarias de las estaciones comprometidas. Se selecciona esta amenaza debido a su comportamiento particular que le permite lograr una persistencia dentro del sistema mediante el uso de PowerShell, lo cual hace que no necesite archivos para infectar hosts y, por tanto, le sea fácil esconderse de las soluciones antimalware actuales [45]. En la Tabla 2-V se resumen las características de este malware:

TABLA 2-VI
Características malware Urnisf

Nombre	Ursnif
Año	2014
Objetivo	Sector bancario en EE. UU., Brasil, Asia y Europa.
Método de infección	La infección por URSNIF se presenta cuando el usuario abre un archivo adjunto malicioso que llega a través de correo electrónico no deseado.
Propósito	Ursnif roba información del sistema, roba credenciales bancarias y credenciales de acceso. Roba datos de correo electrónico e intercepta datos y formularios web de los siguientes navegadores: Chrome, Internet Explorer, Thunderbird y Firefox.
Transmisión de datos	La información es cifrada y enviada a servidores externos.

Nota: Características relevantes de las amenazas estudiadas (año de descubrimiento, organizaciones afectadas, métodos utilizados por los grupos atacantes, propósito de la amenaza y mecanismo de exfiltración de información) [46]

Las actividades necesarias para el desarrollo del proyecto de investigación se agrupan en cinco fases, descritas a continuación.

2.4.1 Fase I: Mecanismo para el pre-procesamiento de datos

La primera fase del procedimiento consistió en obtener los datos que fueron objeto de análisis. Para hacerlo se ambientó un laboratorio conformado por tres máquinas virtuales conectadas entre sí, cuyas características se describen en el Anexo A.

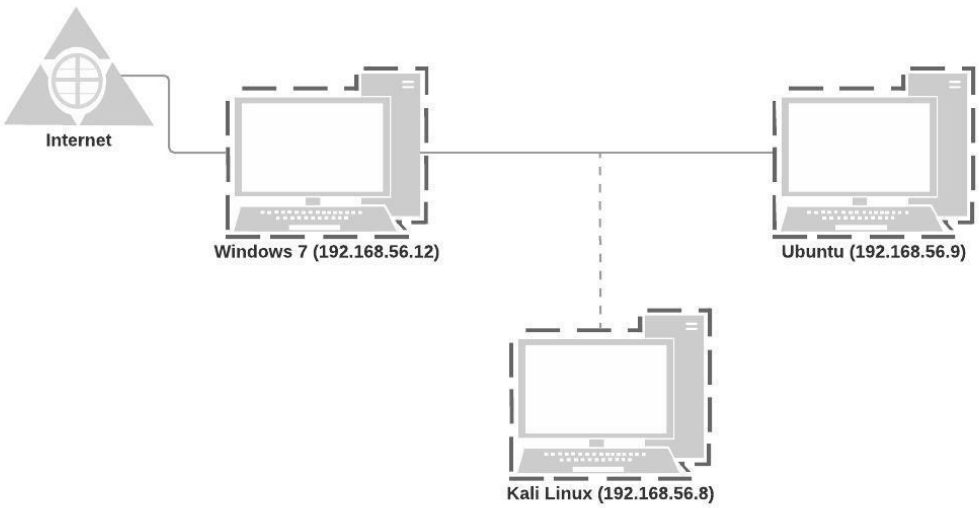


Fig. 2-1. Diagrama de red del laboratorio.

Como se aprecia en la Figura 2-1, el sistema está compuesto por una estación Windows, una estación Linux y una estación Kali que funciona como sniffer y captura los paquetes que viajan por la red. Para capturar el tráfico se simula un ataque de hombre en el medio (MitM), tal como se describe en el Anexo B.

El tráfico objeto de análisis se genera mediante el uso de la herramienta instalada en el laboratorio para la automatización de pruebas de red y recolectado mediante Wireshark en archivos PCAP.

Las capturas de tráfico pueden verse en Wireshark, como se muestra en la Fig. 2-2:

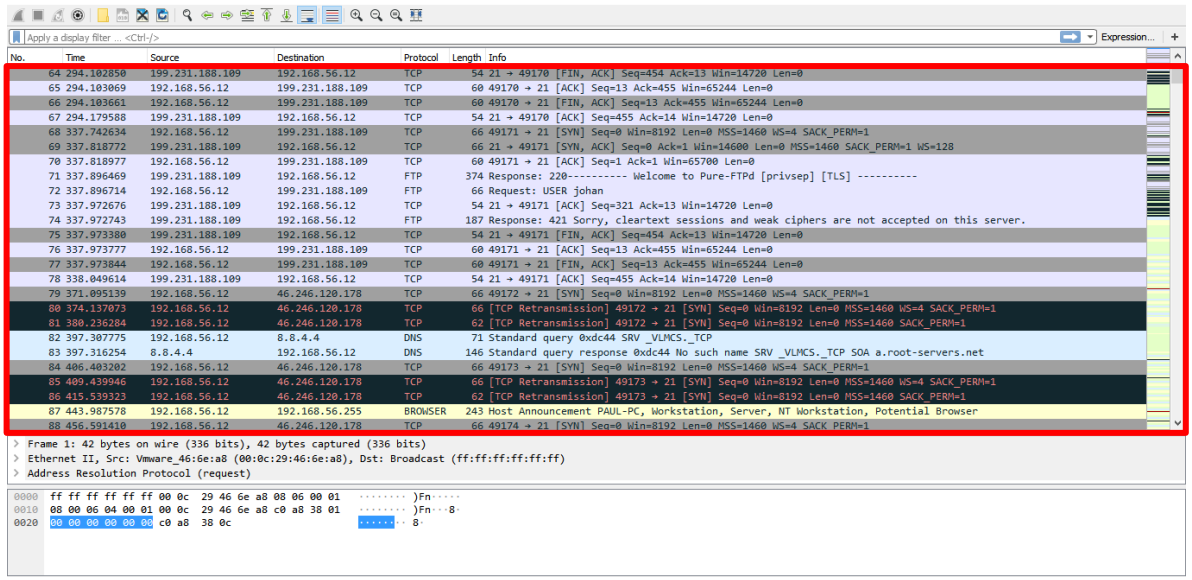


Fig. 2-2. Capturas de tráfico en Wireshark.

En la sección resaltada se observan todos los paquetes que se están capturando en tiempo real. Para interpretar correctamente los datos proporcionados en esta sección (tipo de protocolo, números de secuencia, flags, marcas de tiempo, puertos, etc.) es importante entender la estructura de los paquetes de red (ver marco teórico).

En las secciones inferiores se observan las capas de los paquetes seleccionados y su respectivo formato hexadecimal, tal y como fue capturado por la tarjeta de red. Cabe resaltar que el tráfico capturado se realiza antes y después de la infección por las versiones de malware mencionadas en la sección 2.3: Instrumentos de recolección de información (ver Tabla 2-II).

Seguidamente, se desarrolló el mecanismo de pre-procesamiento de datos que facilitó la construcción del modelo estadístico para el tráfico de red, tal como se ha establecido en el primer objetivo del trabajo de investigación.

Los datos obtenidos requieren una estructura claramente definida que permita hacer una selección de las características apropiadas para el análisis. Así, los datos fueron tratados de la siguiente manera:

1. Como se mencionó anteriormente, los datos recolectados fueron almacenados en archivos con extensión PCAP. Para cada amenaza estudiada se obtuvieron múltiples muestras de tráfico. Con la ayuda de Wireshark se guardó la información en formato CSV (*Comma-Separated Values*) para representar los datos en forma de tabla —en las que las columnas se separan por comas y las filas por saltos de línea—.
2. Los archivos en formato CSV son abiertos y consolidados en Excel, donde se separaron los campos por coma, obteniéndose una tabla en la que las columnas representan el número (consecutivo), unidad de tiempo, fuente, destino, protocolo, longitud e información del paquete, tal como se observan los paquetes en Wireshark (ver Figura 2-3).
3. Finalmente, se separaron los campos por comas para obtener los datos estructurados en forma de tabla (ver Figura 2-4). Los campos o variables obtenidas son los siguientes: No (secuencia del paquete), Time (unidad de tiempo), Source (dirección IP fuente), Destination (dirección IP de destino), Protocol (protocolo del paquete), Length (longitud del paquete en bytes) e Info (información adicional del paquete). En este punto se consideró el riesgo de pérdida de integridad de la información, debido a la eliminación de información útil para el análisis o manipulación errada de los datos. Para evitar el riesgo se hicieron validaciones de los datos en Excel con los PCAP en Wireshark.
4. Para la construcción de la serie de tiempo que permita ver el comportamiento del sistema en un momento dado, se ordenaron los datos por el campo unidad de tiempo, la cual fue la variable explicativa (o independiente) del experimento. La longitud del paquete fue en este caso la variable de respuesta (o dependiente).

```
135,"588.855317","192.168.56.12","46.246.120.178","TCP","62","[TCP Retransmission] 49180 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1"
136,"591.105336","192.168.56.12","199.231.188.109","TCP","66","49181 > 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1"
137,"591.181409","199.231.188.109","192.168.56.12","TCP","66","21 > 49181 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=128"
138,"591.181570","192.168.56.12","199.231.188.109","TCP","60","49181 > 21 [ACK] Seq=1 Ack=1 Win=65700 Len=0"
139,"591.261124","199.231.188.109","192.168.56.12","FTP","374","Response: 220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------"
140,"591.261352","192.168.56.12","199.231.188.109","FTP","68","Request: USER madonna"
141,"591.337307","199.231.188.109","192.168.56.12","TCP","54","21 > 49181 [ACK] Seq=321 Ack=15 Win=14720 Len=0"
142,"591.337350","199.231.188.109","192.168.56.12","FTP","187","Response: 421 Sorry, cleartext sessions and weak ciphers are not accepted on this server."
143,"591.338546","199.231.188.109","192.168.56.12","TCP","54","21 > 49181 [FIN, ACK] Seq=454 Ack=15 Win=14720 Len=0"
144,"591.338956","192.168.56.12","199.231.188.109","TCP","60","49181 > 21 [ACK] Seq=15 Ack=455 Win=65244 Len=0"
145,"591.339100","192.168.56.12","199.231.188.109","TCP","60","49181 > 21 [FIN, ACK] Seq=15 Ack=455 Win=65244 Len=0"
146,"591.339852","192.168.56.12","199.231.188.109","TCP","66","49182 > 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1"
147,"591.415268","199.231.188.109","192.168.56.12","TCP","54","21 > 49181 [ACK] Seq=455 Ack=16 Win=14720 Len=0"
148,"591.415751","199.231.188.109","192.168.56.12","TCP","66","21 > 49182 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=128"
149,"591.415846","192.168.56.12","199.231.188.109","TCP","60","49182 > 21 [ACK] Seq=1 Ack=1 Win=65700 Len=0"
150,"591.492965","199.231.188.109","192.168.56.12","FTP","374","Response: 220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------"
151,"591.493122","192.168.56.12","199.231.188.109","FTP","68","Request: USER madonna"
152,"591.569191","199.231.188.109","192.168.56.12","TCP","54","21 > 49182 [ACK] Seq=321 Ack=15 Win=14720 Len=0"
153,"591.569219","199.231.188.109","192.168.56.12","FTP","187","Response: 421 Sorry, cleartext sessions and weak ciphers are not accepted on this server."
154,"591.569894","199.231.188.109","192.168.56.12","TCP","54","21 > 49182 [FIN, ACK] Seq=454 Ack=15 Win=14720 Len=0"
155,"591.569969","192.168.56.12","199.231.188.109","TCP","60","49182 > 21 [ACK] Seq=15 Ack=455 Win=65244 Len=0"
156,"591.570037","192.168.56.12","199.231.188.109","TCP","60","49182 > 21 [FIN, ACK] Seq=15 Ack=455 Win=65244 Len=0"
157,"591.645633","199.231.188.109","192.168.56.12","TCP","54","21 > 49182 [ACK] Seq=455 Ack=16 Win=14720 Len=0"
158,"600.861326","192.168.56.12","46.246.120.178","TCP","66","49183 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1"
159,"603.862506","192.168.56.12","46.246.120.178","TCP","66","[TCP Retransmission] 49183 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1"
```

Fig. 2-3. Estructura del archivo .CSV.

No.	Time	Source	Destination	Protocol	Length	Info
244	753,529124	192.168.56.12	46.246.120.178	TCP	66	[TCP Retransmission] 49201 > 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
245	758,271594	192.168.56.12	46.246.120.178	TCP	62	[TCP Retransmission] 49200 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
246	759,535106	192.168.56.12	46.246.120.178	TCP	62	[TCP Retransmission] 49201 > 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
247	778,292293	192.168.56.12	46.246.120.178	TCP	66	49202 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
248	781,297151	192.168.56.12	46.246.120.178	TCP	66	[TCP Retransmission] 49202 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
249	787,303154	192.168.56.12	46.246.120.178	TCP	62	[TCP Retransmission] 49202 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
250	809,321836	192.168.56.12	46.246.120.178	TCP	66	49203 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
251	812,325685	192.168.56.12	46.246.120.178	TCP	66	[TCP Retransmission] 49203 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
252	818,331592	192.168.56.12	46.246.120.178	TCP	62	[TCP Retransmission] 49203 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
253	839,095445	192.168.56.12	224.0.0.252	LLMNR	64	Standard query 0x1add A wpad
254	839,096714	192.168.56.12	224.0.0.252	LLMNR	64	Standard query 0xc867 A wpad
255	839,205314	192.168.56.12	224.0.0.252	LLMNR	64	Standard query 0xc867 A wpad
256	839,205633	192.168.56.12	224.0.0.252	LLMNR	64	Standard query 0x1add A wpad
257	839,410892	192.168.56.12	192.168.56.255	NBNS	92	Name query NB WPAD<00>
258	839,41238	192.168.56.12	192.168.56.255	NBNS	92	Name query NB WPAD<00>
259	840,171964	192.168.56.12	192.168.56.255	NBNS	92	Name query NB WPAD<00>
260	840,172131	192.168.56.12	192.168.56.255	NBNS	92	Name query NB WPAD<00>
261	840,936151	192.168.56.12	192.168.56.255	NBNS	92	Name query NB WPAD<00>
262	840,93622	192.168.56.12	192.168.56.255	NBNS	92	Name query NB WPAD<00>
263	841,706631	192.168.56.12	199.231.188.109	TCP	66	49204 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
264	841,708402	192.168.56.12	199.231.188.109	TCP	66	49205 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
265	841,782838	199.231.188.109	192.168.56.12	TCP	66	80 > 49204 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=128
266	841,78334	192.168.56.12	199.231.188.109	TCP	60	49204 > 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
267	841,784862	199.231.188.109	192.168.56.12	TCP	66	80 > 49205 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=128

Fig. 2-4. Datos estructurados en forma de tabla.

Otros riesgos que se consideraron en esta fase, sin materialización, fueron los siguientes:

- Vulnerabilidad de la información frente a terceros o hackers por mal diseño experimental. Para mitigar este riesgo se aisló la red del laboratorio y se instalaron controles de seguridad en el escritorio host como antivirus y firewall.
- Rezago o insuficiencia de los recursos (hardware y sistemas de información). Para esto se utilizó una máquina con amplia capacidad en hardware.
- Daño o pérdida de activos de información por actos mal intencionados u ocurrencia de incidente grave. En este caso se utilizó un equipo de uso experimental; por tanto, se aceptó el riesgo.

2.4.2 Fase II: Elementos del tráfico de red que permiten identificar patrones de infección por APT

Se seleccionó la longitud de los paquetes como variable de respuesta, debido a que la variación en este campo puede indicar la presencia de comportamiento anómalo en el sistema. Se tomó como variable explicativa la unidad de tiempo, dado que tiene influencia sobre la longitud de los paquetes —se desea identificar variaciones en el tamaño de los paquetes asociadas a comportamiento anómalo—.

Una vez se ha estructurado la información en forma de tabla, se caracterizaron los elementos del tráfico de red que revelan patrones de infección por APT, para lo cual se tuvo en cuenta el comportamiento de las amenazas estudiadas. En este punto de la metodología se dio cumplimiento al segundo objetivo de la investigación.

CosmicDuke, Fareit y URSNIF tienen un propósito en común que es el robo de información y, para ello, necesitan establecer conexiones con servidores externos a la red; por tanto, es relevante monitorearlas.

Un patrón claro de infección por APT es una conexión del equipo infectado hacia un dominio desconocido. Para identificarlas se revisan en las capturas los paquetes DNS desde la máquina infectada, así como los protocolos FTP y HTTP que también indican conexiones externas para filtrar los datos procesados por el campo protocolo.

De estos paquetes se extrajeron las direcciones IP (fuente y destino) y dominios consultados, y se analizaron uno a uno con en el sitio web VirusTotal, el cual proporciona un motor de análisis de archivos, URL, hashes y direcciones IP, apoyado en 55 antivirus y 69 motores de detección en línea.

Otro patrón puede observarse en el tamaño de los paquetes de salida, que pueden revelar una fuga de información del sistema.

Los resultados de esta fase se apoyaron también en otros análisis estadísticos de los datos como, por ejemplo, la media, las desviaciones estándar, la tendencia en el comportamiento, la estacionariedad y la estacionalidad, entre otros.

2.4.3 Fase III: Representación del tráfico de red de un sistema mediante cadenas de Markov

Los datos pre-procesados de la etapa anterior se analizaron mediante la aplicación de cadenas de Markov para modelar el comportamiento de la red, con el propósito de detectar anomalías.

Como se mencionó anteriormente, los paquetes analizados son los del protocolo TCP, los cuales se filtran en Excel por el campo Protocol y se colocan en un archivo separado. Los pasos seguidos para construir la cadena fueron:

1. Inicialmente, se identificaron los estados del sistema, es decir, E (el conjunto de valores que puede tomar el proceso en cada una de sus etapas). En este caso los estados son los posibles tamaños que pueden tener los paquetes capturados.
2. Se calculó la frecuencia absoluta de cada estado dentro de la data como el número de paquetes de cada longitud entre el total de paquetes TCP de la muestra. Suponiendo que el total de paquetes analizados es *n* (tamaño de la muestra) y la frecuencia de cada estado n_i, la frecuencia absoluta estará dada entonces por la razón n_i/n. Esta frecuencia se utiliza para calcular la probabilidad condicional de cada estado en la matriz de transición en los pasos siguientes.
3. Para conformar la matriz de transición se identificaron los estados desde los cuales se puede acceder en un estado en particular, es decir, en un barrido de los datos desde el inicio hasta el final. Para cada longitud de paquete (cada estado) se determina si es posible que a partir de este se pueda pasar a otro de tamaño diferente y con qué frecuencia (denotada por n_{ij}). Así, se obtiene una matriz inicial con la siguiente estructura (ver Matriz 2.1):

$$\begin{bmatrix} n_{11} & \cdots & n_{12} & \cdots & n_{1j} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ n_{21} & \cdots & n_{22} & \dots & n_{2j} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ n_{i1} & \dots & n_{i2} & \dots & n_{ij} \end{bmatrix}, \text{ para toda } i,j \in E \qquad (2.1)$$

4. Una vez se tienen las frecuencias anteriores, se calculó la probabilidad conjunta (la probabilidad de que sucedan los dos eventos al mismo tiempo), para lo cual se dividió cada entrada de la matriz anterior por el número total de paquetes de la muestra, es decir:

$$n_{ij} \Big/ \sum_{i,j \in E} n_{ij}$$

5. Finalmente se calculó la probabilidad condicional como la probabilidad conjunta entre la frecuencia absoluta de cada estado dentro de la data, hallada en el paso dos. El resultado es la probabilidad de ocurrencia de los estados futuros del sistema analizado y con esto se construyó una matriz de estados de transición, donde cada entrada representa la probabilidad de pasar de un estado a otro.

2.4.4 Fase IV: Eficacia de la evaluación realizada

Con base en los resultados obtenidos en la fase anterior, se diseñan e implementan reglas de red que permiten identificar patrones asociados con infecciones por APT en un sistema de detección de intrusos (IDS). En esta fase se realizan las siguientes actividades:

1. Instalación y configuración del IDS

El IDS es instalado y configurado en el equipo de laboratorio, específicamente en la máquina Kali, y se siguen la documentación oficial del software (https://www.snort.org/documents#OfficialDocumentation).

2. Diseño de las reglas de red [47]

Esta actividad consiste en la elaboración del código con la regla de red. Una regla Snort se compone de dos partes, un encabezado y las opciones de regla. A continuación, se describen los campos que aplican en cada una de ellas:

- **Encabezado de regla**

Acciones: es el primer atributo visible de la regla. Permite ejecutar un evento sobre el paquete capturado. Puede tener los siguientes tres valores:

- Alert: genera una alerta usando el método de alerta seleccionado y genera un log del paquete.
- Log: genera un log del paquete.
- Pass: ignora el paquete.

Protocolos: indican el protocolo al cual se aplicará la regla. En Snort se pueden definir reglas sobre los protocolos TCP, UDP e ICMP.

Direcciones IP y puertos: indican la dirección o rango de direcciones de origen y destino relacionadas con los paquetes capturados. El primer conjunto de direcciones y puertos declarados en la regla corresponden al origen del paquete, mientras que las direcciones y puertos declarados después del símbolo de dirección de la regla corresponden a la dirección destino.

Operador de dirección: es un símbolo que indica la dirección del tráfico que aplicará la regla (“->” unidireccional, “<>” bidireccional).

- **Opciones de regla**

Msg: es una cadena que indica la descripción del mensaje que debe imprimir.

Reference: indica una referencia externa a sistemas de identificación de ataques. Algunos de los sistemas externos soportados por Snort son:

- bugtraq (http://www.securityfocus.com/bid/)
- cve (http://cve.mitre.org/cgi-bin/cvename.cgi?name=)
- nessus (http://cgi.nessus.org/plugins/dump.php3?id=)
- arachnids (http://www.whitehats.com/info/IDS)
- mcafee (http://vil.nai.com/vil/content/v _)
- osvdb (http://osvdb.org/show/osvdb/)

Estas referencias se encuentran dentro del directorio de instalación de Snort en el fichero “reference.config”.

Gid: indica el id que identifica el componente de Snort que ha generado el evento asociado a la regla. Se encuentran ubicados en el fichero gen-msg.map.

Rev: indica la revisión de la regla de forma única, normalmente se utiliza con el atributo sid.

Sid: permite identificar de forma única las reglas de Snort. Los siguientes son los rangos manejados por Snort los posibles SID.

- Menores de 100: reservados para uso futuro.
- Entre 100 y 999.999: Incluidos en la distribución de Snort.
- Mayor o igual a 1.000.000: usados para reglas locales.

Classtype: permite categorizar la regla para clasificar el ataque y permite mejorar la organización de los eventos que Snort produce durante su ejecución.

Priority: se utiliza para sobrescribir el valor de la prioridad definida en el atributo classtype de la regla. Se trata de un valor entero entre 1 y 4 que hace que la prioridad declarada en el classtype sea sobrescrita.

Metadata: proporciona información adicional para el procesador de reglas de Snort. Los valores declarados tienen significados para Snort y hacen que el comportamiento de la regla cambie en función de los siguientes valores (Tabla 2-VI):

TABLA 2-VII
Valores para el campo metadata

Clave	Descripción	Valor de Formato
engine	“shared”	Indica una regla de librería compartida
soid	gid|sid	Regla de librería compartida para generadores y sid
service	“http”	Identificador de servicio basado en objetivo

Nota: Valores que puede tener el campo metadata en la regla de Snort [47].

3. Implementación de regla de red

Las reglas diseñadas se incluyen en el repositorio de reglas de Snort, ubicadas en el directorio /etc/snort/rules/sites.rules y se inicia la captura de paquetes con el comando snort –A console –c snort.conf –i eth0.

4. Captura y análisis de tráfico

Finalmente, se utiliza el tráfico de red capturado, además de otros comandos para probar las reglas creadas.

La metodología aplicada en este trabajo de investigación se sintetiza en la Tabla 2-VII:

El método analítico aplicado en esta investigación con el enfoque cuantitativo permite estudiar y comprender el comportamiento de las Amenazas Persistentes Avanzadas en un sistema simulado, mediante la descomposición del fenómeno en sus partes y el análisis de la información que genera (en este caso tráfico de red). En este capítulo se explicó cada fase de la metodología, su resultado esperado y cómo cada uno de ellos da cumplimiento a los objetivos planteados, los cuáles culminan en la obtención de reglas de red que permiten identificar comportamiento anómalo asociado a una infección por APT y se apoyan en el modelo estadístico obtenido a partir del sistema estudiado.

3. Resultados

3.1 Mecanismo de pre-procesamiento de los datos capturados

Luego de aplicar el mecanismo de pre-procesamiento de los datos capturados de acuerdo con los pasos descritos en la metodología, se obtienen los datos estructurados en forma de tabla. Esta estructura permitió identificar los protocolos de red activados durante el proceso de comunicación del sistema, las direcciones IP involucradas y el tamaño de los paquetes.

Dentro del tráfico capturado para todas las amenazas, se observaron paquetes de 12 protocolos de comunicación diferentes, los cuales se inspeccionaron teniendo en cuenta su propósito y el número de paquetes en cada uno de ellos. Los protocolos en mención son los siguientes (Tabla 3-I):

TABLA 3-I
Protocolos observados

Protocolo	Nombre	Objetivo
ARP	Address Resolution Protocol	Conocer la dirección física (MAC) de una tarjeta de interfaz de red correspondiente a una dirección IP (Internet Protocol).
BROWSER	Browser Protocol	Registra los nombres SMB (CIFS) o de NetBIOS de una red, los almacena y los comparte para los demás nodos de la red
DNS	Domain Name System	Traducir direcciones IP en nombres de dominio
FTP	File Transfer Protocol	Permite el intercambio de archivos entre equipos remotos
HTTP	Hypertext Transfer Protocol	Permite realizar una petición de datos y recursos, para el intercambio de información en la Web
ICMP	Internet Control Message Protocol	Control y notificación de errores del Protocolo de Internet (IP).
LLMNR	Link-Local Multicast Name Resolution	Resuelve los nombres de los sistemas informáticos cercanos, si la red no tiene un servidor de Sistema de nombres de dominio (DNS)

NBNS	Servicio de Nombres NetBIOS	Mantiene un registro central de todos los participantes en una red. Esto permite una consulta rápida para establecer si un nombre solicitado ya está reservado. NBNS simplifica y acelera el proceso de registro de un nuevo nodo de red.
NTP	Network Time Protocol	Sincronizar los relojes de los sistemas informáticos a través del enrutamiento de paquetes en redes con latencia variable
SSDP	Simple Service Discovery Protocol	Sirve para la búsqueda de dispositivos UPnP (Universal Plug and Play) en una red
TCP	Transmission Control Protocol/Internet	Permite establecer conexiones e intercambiar datos garantizando la entrega de datos.
TLS	Transport Layer Security	Permite establecer una conexión segura por medio de un canal cifrado entre el cliente y servidor.

Nota: Protocolos observados en los datos capturados. Describe el protocolo, su nombre y para qué se sirve dentro de la red.

El número de paquetes recolectados para cada amenaza estudiada se resume a continuación en las Tablas 3-II, 3-III y 3-IV:

TABLA 3-II
Número de paquetes capturados CosmicDuke

Protocolo	Número de paquetes	%
NBNS	17.645	28,2 %
TCP	15.704	25,1 %
HTTP	13.125	21,0 %
LLMNR	11.770	18,8 %
ARP	2.473	4,0 %
FTP	822	1,3 %
DNS	560	0,9 %
SSDP	180	0,3 %
BROWSER	105	0,2 %
ICMP	92	0,1 %
NTP	48	0,1 %
Total	**62.524**	

Nota: Protocolos capturados, cantidad de paquetes y proporción de paquetes respecto al total.

TABLA 3-III
Número de paquetes Fareit

Protocolo	Número de paquetes	%
TCP	1.318	50,1 %
DNS	546	20,7 %
HTTP	310	11,8 %
NBNS	282	10,7 %
LLMNR	112	4,3 %
ICMP	49	1,9 %
BROWSER	14	0,5 %
ARP	2	0,1 %
Total	**2.633**	

Nota: Protocolos capturados, cantidad de paquetes y proporción de paquetes respecto al total.

TABLA 3-IV
Número de paquetes URSNIF

Protocolo	Número de paquetes	%
TCP	154.184	79,8 %
TLSv1	14.288	7,4 %
TLSv1.2	7.847	4,1 %
HTTP	5.130	2,7 %
ARP	3.673	1,9 %
NBNS	3.110	1,6 %
DNS	2.244	1,2 %
TLSv1.1	1.188	0,6 %
SSDP	552	0,3 %
LLMNR	484	0,3 %
OCSP	175	0,1 %
BROWSER	159	0,1 %
ICMP	141	0,1 %
STUN	13	0,0%
MP4	11	0,0%
PKIX-CRL	10	0,0%
SSLv2	2	0,0%
HTTP/XML	1	0,0%
Total	**193.212**	79,8%

Nota: Protocolos capturados, cantidad de paquetes y proporción de paquetes respecto al total.

El número de paquetes capturados revela el tráfico predominante y los roles que desempeñaron durante la operación del sistema. En las fuentes investigadas no se

identificaron criterios para definir el tamaño de una muestra que permita obtener un modelo estadístico ajustado; sin embargo, la muestra obtenida es suficientemente grande como para detectar patrones de tráfico asociados con APT.

3.2 Caracterización de elementos de tráfico de red

Los datos capturados contienen paquetes de longitud completa, recolectados durante un periodo de varias semanas. Para caracterizar los elementos de tráfico se realizó una inspección de la data con el fin de determinar si las direcciones IP de destino y los protocolos permiten detectar amenazas en la red.

Como se puede observar en las Tablas 3-II, 3-III y 3-IV, uno de los protocolos con mayor número de paquetes es el TCP, por lo cual se seleccionó para analizar su comportamiento mediante las cadenas Markov. En particular, se estudia el tamaño de los paquetes con relación al tiempo.

Al graficar la variación del tamaño de los paquetes se obtienen gráficos de correlación automática que muestran picos persistentes con varios armónicos (Figuras 3-1, 3-2 y 3-3):

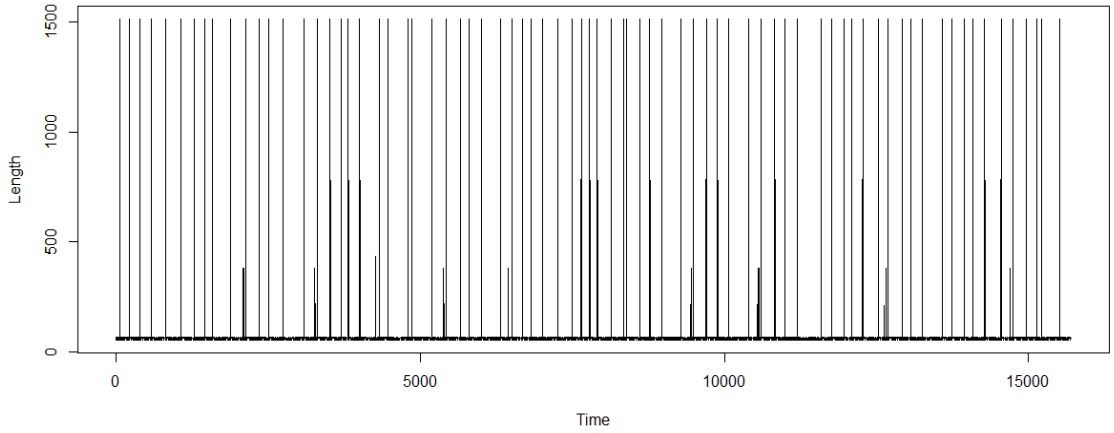


Fig. 3-1. Serie de tiempo paquetes TCP CosmicDuke.

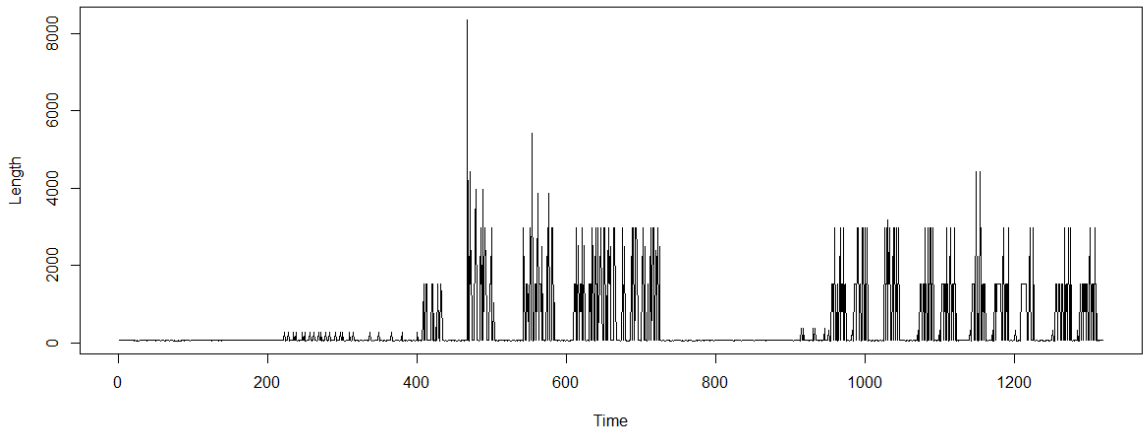


Fig. 3-2. Serie de tiempo paquetes TCP Fareit.

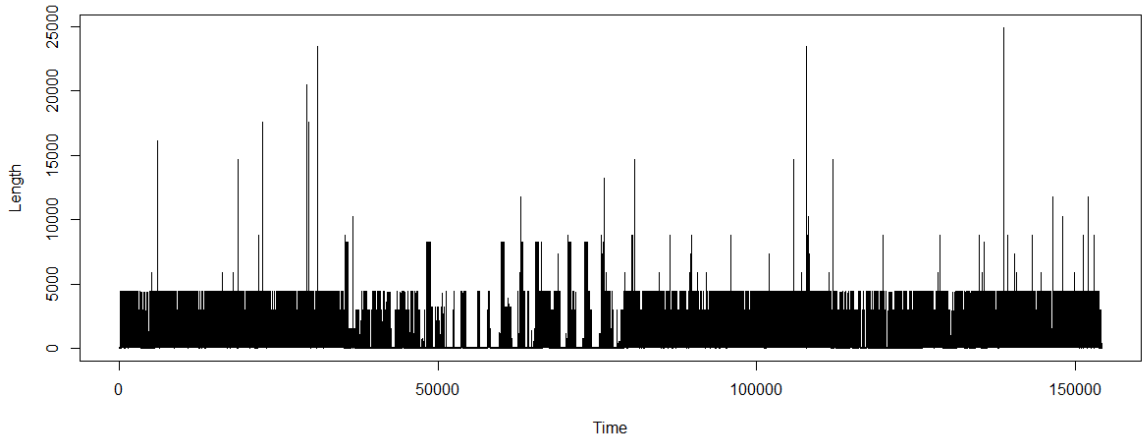


Fig. 3-3. Serie de tiempo paquetes TCP URSNIF.

La gráfica de serie de tiempo para la amenaza CosmicDuke (Figura 3-1) muestra un patrón de comunicación caracterizado por una gran cantidad de paquetes de longitud inferior a 100 bytes, con paquetes de un tamaño superior (1.500 bytes) cada cierto periodo de tiempo.

En el caso de Fareit, la Figura 3-2 muestra un patrón de comunicación armónico, donde se presenta un crecimiento del tamaño de paquetes (hasta alrededor de 3.000 bytes), seguido de un decrecimiento (paquetes con tamaño inferior a 60 bytes).

Para el caso de URSNIF (Figura 3-3) la serie de tiempo muestra una concentración de paquetes de 5.000 bytes de longitud con algunos paquetes de longitud mayor, sin un patrón definido.

Se realizaron gráficas de cajas (boxplot) para buscar indicadores de datos inusuales o no normales. Los valores atípicos, que son valores de datos que están muy alejados de otros valores de datos, pueden afectar fuertemente sus resultados.

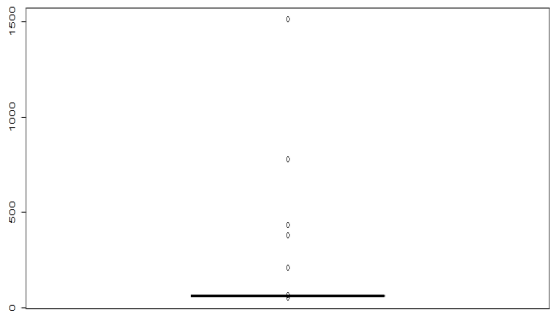


Fig. 3-4. Boxplot CosmicDuke.

El boxplot obtenido para los paquetes recolectados con la amenaza CosmicDuke (Figura 3-4) muestra que el tamaño se concentra de manera casi homogénea; sin embargo, existen algunos datos atípicos de paquetes con longitud superior a 60 bytes. Como se observa en la Tabla 3-V, alrededor del 50 % de los paquetes tienen una longitud de 60 bytes, mientras que los paquetes con mayor tamaño (1.514 bytes) solo tienen una frecuencia del 0,5 %.

TABLA 3-V
Frecuencia de tamaño de paquetes TCP para la amenaza CosmicDuke

Tamaño de paquete(bytes)	Cantidad de paquetes	%
60	7.782	49,6 %
66	3.724	23,7 %
54	2.735	17,4 %
62	1.371	8,7 %
1.514	77	0,5 %
380	11	0,1 %

Nota: Longitud de paquetes observados, frecuencia dentro de la data y su proporción sobre el total de paquetes.

La gráfica de caja para los paquetes TCP con la amenaza Fareit (Figura 3-5) revela también datos atípicos con una frecuencia mayor. Nuevamente, se observa una concentración de paquetes con tamaños pequeños en bytes: el 81 % de ellos tiene una longitud entre 54 y 63 bytes, de acuerdo con la Tabla 3-VI, así el 19 % restante presenta un tamaño superior.

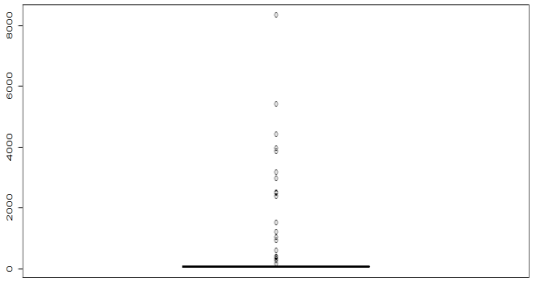


Fig. 3-5. Boxplot Fareit.

TABLA 3-VI
Frecuencia de tamaño de paquetes TCP para la amenaza Fareit

Tamaño de paquete (bytes)	Cantidad de paquetes	%
60	449	34,1 %
66	308	23,4 %
54	215	16,3 %
1514	118	9,0 %
62	81	6,1 %
2974	66	5,0 %
277	11	0,8 %
274	11	0,8 %

Nota: Longitud de paquetes observados, frecuencia dentro de la data y su proporción sobre el total de paquetes.

El boxplot obtenido con los datos generados por la amenaza URSNIF (Figura 3-6) muestra una variabilidad mayor de los datos —lo cual es coherente con el tamaño de la muestra—. La mediana de los datos es similar a la de las demás muestras; es decir, cercana a 60 bytes, y los datos atípicos tienen una longitud superior a los 7.000 bytes (denotada por la línea superior de la caja). En este caso se observaron 608 tamaños de paquete diferentes; sin embargo, el 97 % se concentra en 8 longitudes, tal como se observa en la Tabla 3-VII.

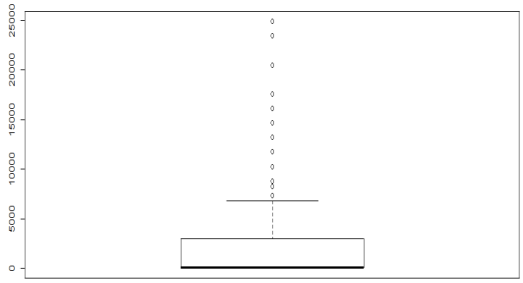


Fig. 3-6. Boxplot URSNIF.

TABLA 3-VII
Frecuencia de tamaño de paquetes TCP para la amenaza URSNIF

Tamaño de paquete (bytes)	Cantidad de paquetes	%
60	56.925	37 %
2974	50.058	32 %
66	15.509	10 %
54	13.071	8 %
1514	10.046	7 %
4434	1.931	1 %
74	1.379	1 %
90	815	1 %

Nota: Longitud de paquetes observados, frecuencia dentro de la data y su proporción sobre el total de paquetes.

Así, el análisis anterior sugiere que los paquetes con mayor longitud son atípicos y, por lo tanto, pueden estar asociados con actividad anómala en el sistema.

Para complementar el análisis se realiza una inspección de otros protocolos de red involucrados en la comunicación y que son especialmente críticos, como el FTP y el DNS, que permiten transmitir archivos y consultar dominios en Internet, respectivamente.

Como se observa en la Tabla 3-II, dentro de los datos recopilados con la amenaza CosmicDuke se encuentran paquetes FTP. Al observar el detalle de las conexiones, se identifica la dirección IP 199.231.188.109 (Figura 3-7), la cual corresponde a una dirección maliciosa de acuerdo con el sitio VirusTotal. Los paquetes HTTP capturados también muestran interacción con esta dirección IP.

No se identifican conexiones FTP con las amenazas Fareit y URSNIF. Del mismo modo, la inspección de conexiones HTTP para estas amenazas no muestra comportamiento anómalo.

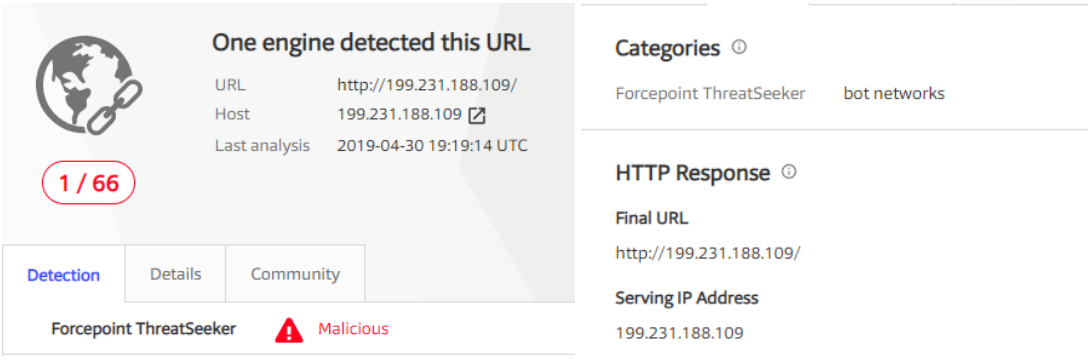


Fig. 3-7. Resultados de análisis de dirección IP.

Para los dominios consultados se extrajeron del detalle los paquetes DNS y se analizaron con el sitio web VirusTotal. CosmicDuke realiza consultas a seis dominios, de los cuales ninguno es malicioso.

Fareit realiza consulta a 39 dominios de los cuales 18 son maliciosos; es decir, el 46 %, y URSNIF consulta 557 dominios, de los cuales 18 son maliciosos; es decir, el 14 %. Estos resultados se resumen en las Tablas 3-VIII y 3-IX.

TABLA 3-VIII
Dominios consultados por Fareit

Clasificación	Dominios consultados	%
Malicioso	18	46 %
No malicioso	21	54 %
Total	**39**	

Nota: Tipo de dominio, cantidad de dominios consultados por tipo y su proporción dentro de la data

TABLA 3-IX
Dominios consultados por URSNIF

Clasificación	Dominios consultados	%
No malicioso	464	83 %
Malicioso	80	14 %
Sospechoso	13	2 %
Total	**557**	

Nota: Tipo de dominio, cantidad de dominios consultados por tipo y su proporción dentro de la data

En conclusión, los elementos que representan patrones de infección por APT son:

1. Paquetes TCP de grandes longitudes. Si bien no es un indicador certero de comportamiento malicioso en un sistema, puede ser un signo de alerta especialmente cuando se dan de forma periódica y de manera aislada; es decir, no soportada por una actividad normal (por ejemplo, cuando se reproduce video).

2. Conexiones a servidores FTP maliciosos. Como se ha mencionado anteriormente, un comportamiento común de las APT son las conexiones a servidores remotos, ya sea para descargar nuevo código malicioso o para extraer información del sistema atacado. Los datos analizados revelan claramente este comportamiento en el sistema.
3. Conexiones a DNS maliciosos. Mediante estas conexiones se descargan archivos maliciosos o se establece contacto con servidores de atacantes.

3.3 Cadenas de Markov

Inicialmente se analizaron los datos recolectados de la amenaza CosmicDuke. Al agrupar los datos por longitud de paquete, se observó que el espacio de estados es $E = \{54, 60, 66, 62, 210, 380, 434, 778, 1.514\}$ con k = 9; es decir, que dentro de los datos existen paquetes TCP de nueve longitudes diferentes. La frecuencia de los estados en la data recopilada se muestra a continuación (Tabla 3-X):

TABLA 3-X
Frecuencia absoluta de los estados para CosmicDuke

Estado	Frecuencia	Proporción
60	7.782	0,495543
66	3.724	0,237137
54	2.735	0,174159
62	1.371	0,087303
1514	77	0,004903
380	11	0,000700
210	2	0,000127
434	1	0,000064
778	1	0,000064
Total	**15.704**	

Nota: Estados de la Cadena de Markov (longitudes de paquetes), cantidad de paquetes por estado y su proporción dentro de la data.

Teniendo en cuenta el número de estados, la matriz de transición debe tener una estructura, como se muestra a continuación (Tabla 3-XI):

TABLA 3-XI
Estructura de la matriz de transición para CosmicDuke

	1	2	3	4	5	6	7	8	9
1	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}	p_{19}
2	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	p_{26}	p_{27}	p_{28}	p_{29}
3	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}	p_{38}	p_{39}
4	p_{41}	p_{42}	p_{43}	p_{44}	p_{45}	p_{46}	p_{47}	p_{48}	p_{49}
5	p_{51}	p_{52}	p_{43}	p_{54}	p_{55}	p_{56}	p_{57}	p_{58}	p_{59}
6	p_{61}	p_{62}	p_{53}	p_{64}	p_{65}	p_{66}	p_{67}	p_{68}	p_{69}
7	p_{71}	p_{72}	p_{73}	p_{74}	p_{75}	p_{76}	p_{77}	p_{78}	p_{79}
8	p_{81}	p_{82}	p_{83}	p_{84}	p_{85}	p_{86}	p_{87}	p_{88}	p_{89}
9	p_{91}	p_{92}	p_{93}	p_{94}	p_{95}	p_{96}	p_{97}	p_{98}	p_{99}

Nota: Estructura general de la matriz de transición de estados para una Cadena de Markov.

Mediante un barrido de los datos se identificaron los estados desde los cuales se pudo acceder y calcular la frecuencia, obteniéndose la siguiente matriz (Tabla 3-XII):

TABLA 3-XII
Matriz de frecuencias transicionales para CosmicDuke

	60	66	54	62	1.514	380	210	434	778	**Total**
60	4.165	1.620	1.449	532	15	1	0	0	0	**7.782**
66	1.519	1.351	401	435	4	10	1	1	1	**3.723**
54	1.544	307	712	117	55	0	0	0	0	**2.735**
62	549	439	93	287	3	0	0	0	0	**1.371**
1.514	2	1	74	0	0	0	0	0	0	**77**
380	1	4	6	0	0	0	0	0	0	**11**
210	0	1	0	0	0	0	1	0	0	**2**
434	1	0	0	0	0	0	0	0	0	**1**
778	0	1	0	0	0	0	0	0	0	**1**
										15.703

Nota: Número de paquetes de longitud i (columnas) que se encuentran luego de un paquete con longitud j (filas)

Como se puede observar en la matriz anterior, una vez el sistema está en un estado cualquiera puede pasar a otro sin depender del estado inmediatamente anterior; por tanto, se comprueba la condición de Markov que afirma que, una vez conocido el presente, el futuro no depende del pasado, en consecuencia, el comportamiento obedece a una cadena de Markov.

Con la matriz de frecuencias transicionales, se obtiene la siguiente matriz de probabilidad conjunta (Tabla 3-XIII):

TABLA 3-XIII
Matriz de probabilidad conjunta para CosmicDuke

	60	66	54	62	1.514	380	210	434	778
60	0,26524	0,10317	0,09228	0,03388	0,00096	0,00006	0	0	0
66	0,09673	0,08603	0,02554	0,0277	0,00025	0,00064	0,00006	0,00006	0,00006
54	0,09833	0,01955	0,04534	0,00745	0,0035	0	0	0	0
62	0,03496	0,02796	0,00592	0,01828	0,00019	0	0	0	0
1.514	0,00013	0,00006	0,00471	0	0	0	0	0	0
380	0,00006	0,00025	0,00038	0	0	0	0	0	0
210	0	0,00006	0	0	0	0	0,00006	0	0
434	0,00006	0	0	0	0	0	0	0	0
778	0	0,00006	0	0	0	0	0	0	0

Nota: Probabilidad de que el sistema pase de un estado j (columnas) dado que estuvo en un estado j (filas).

Teniendo en cuenta las frecuencias calculadas se obtiene finalmente la matriz de transición de estados para la amenaza CosmicDuke:

TABLA 3-XIV
Matriz de transición de estados para CosmicDuke

	60	66	54	62	1.514	380	210	434	778
60	0,53524	0,20819	0,18621	0,06837	0,00193	0,00013	0	0	0
66	0,40792	0,36281	0,10769	0,11682	0,00107	0,00269	0,00027	0,00027	0,00027
54	0,56457	0,11226	0,26035	0,04278	0,02011	0	0	0	0
62	0,40046	0,32022	0,06784	0,20935	0,00219	0	0	0	0
1.514	0,02598	0,01299	0,96110	0	0	0	0	0	0
380	0,09091	0,36366	0,54549	0	0	0	0	0	0
210	0	0,50003	0	0	0	0	0,50003	0	0
434	1,00006	0	0	0	0	0	0	0	0
778	0	1,00006	0	0	0	0	0	0	0

Nota: Probabilidades de pasar de un estado i (filas) a un estado j (columnas).

Las probabilidades de transición muestran que el sistema tiende a comunicarse con paquetes de longitud pequeña. Cuando en un momento dado un paquete TCP de longitud pequeña se encuentra viajando a través de la red es mayormente probable que otro paquete de longitud pequeña se presente en el siguiente espacio de tiempo. Los paquetes de mayor tamaño se presentan con menor frecuencia dentro de la comunicación. Al analizar nuevamente las capturas, se observó que los paquetes TCP de mayor longitud se encuentran asociados con peticiones con direcciones IP maliciosas. Por ejemplo, los paquetes de 1.514 bytes muestran una comunicación con la IP

199.231.188.109, mientras que los paquetes de 434, 210 y 380 se encuentran asociados con la IP 52.18.63.80. Ambas direcciones son sospechosas de actividad maliciosa, de acuerdo con el sitio web VirusTotal. En la Tabla 3-XV se resumen el número de archivos que comunican, las URL alojadas y los archivos descargados desde dichas direcciones.

TABLA 3-XV
Direcciones IP asociadas a la comunicación con paquetes TCP de mayor tamaño para CosmicDuke

Dirección IP	Archivos que comunican	URL alojadas	Archivos descargables
199.231.188.109	59	28	0
52.18.63.80	6	38	0

Nota: dirección IP maliciosa, archivos maliciosos asociados, número de URL que apuntan a dicha IP y archivos descargables asociados.

Para obtener la cadena de Markov asociada con la amenaza Fareit, se identificó el espacio de estados a partir de los datos capturados, que en este caso es E = {8347, 5427, 4434, 3967, 3956, 3858, 3170, 2974, 2507, 2496, 2398, 1514, 1227, 1047, 1036, 938, 599, 410, 380, 336, 277, 274, 165, 66, 63, 62, 60, 56, 55, 54} con k = 30, lo que indica que dentro de los datos existen paquetes TCP de 30 longitudes diferentes. Debido a que el espacio de estados es grande, el cálculo de la matriz de transiciones es más extenso; por lo tanto, se optó por trabajar con una matriz de k = 9 (siguiendo la estructura de la Tabla 3-I hallada para la amenaza CosmicDuke). El espacio de estados se reduce entonces teniendo en cuenta el rango entre el valor mínimo y el máximo de la serie, y se divide entre el número de estados deseados, en este caso 9, obteniéndose así los siguientes rangos:

TABLA 3-XVI
Frecuencia para los estados para Fareit

Estado	Rango	Frecuencia	Proporción
1	0 - 975 Bytes	1.104	0,837633
2	976 - 1896 Bytes	125	0,094841
3	1897 - 2817 Bytes	13	0,009863
4	2818 - 3738 Bytes	67	0,050835
5	3739 - 4659 Bytes	7	0,005311
6	4660 - 5580 Bytes	1	0,000759
7	5581 - 6501 Bytes	0	0
8	6502 - 7422 Bytes	0	0
9	7423 - 8347 Bytes	1	0,000759
		1.318	

Nota: Estados con los que se construirá la matriz de transición de estados, rangos que limitan los tamaños de paquetes agrupados, número de paquetes dentro del rango y proporción respecto al total.

La matriz de frecuencias transicionales, así como la matriz de probabilidad conjunta se muestran en las Tablas 3-XVII y 3-XVIII:

TABLA 3-XVII
Matriz de frecuencias transicionales para Fareit

	1	2	3	4	5	6	7	8	9	**Total**
1	979	53	13	50	6	1	0	0	1	**1.103**
2	71	47	0	6	1	0	0	0	0	**125**
3	9	3	0	1	0	0	0	0	0	**13**
4	37	21	0	9	0	0	0	0	0	**67**
5	5	1	0	1	0	0	0	0	0	**7**
6	1	0	0	0	0	0	0	0	0	**1**
7	0	0	0	0	0	0	0	0	0	**0**
8	0	0	0	0	0	0	0	0	0	**0**
9	1	0	0	0	0	0	0	0	0	**1**
										1.317

Nota: Número de paquetes de longitud i (columnas) que se encuentran luego de un paquete con longitud j (filas)

TABLA 3-XVIII
Matriz de probabilidad conjunta para Fareit

	1	2	3	4	5	6	7	8	9
1	0,74336	0,04024	0,00987	0,03797	0,00456	0,00076	0	0	0,00076
2	0,05391	0,03569	0	0,00456	0,00076	0	0	0	0
3	0,00683	0,00228	0	0,00076	0	0	0	0	0
4	0,02809	0,01595	0	0,00683	0	0	0	0	0
5	0,00380	0,00076	0	0,00076	0	0	0	0	0
6	0,00076	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0,00076	0	0	0	0	0	0	0	0

Nota: Probabilidad de que el sistema pase de un estado j (columnas) dado que estuvo en un estado j (filas).

Con las matrices anteriores finalmente se obtuvo la matriz de transición de estados para Fareit (Tabla 3-XIX):

TABLA 3-XIX
Matriz de transición de estados para Fareit

	1	2	3	4	5	6	7	8	9
1	0,88745	0,04804	0,01178	0,04532	0,00544	0,00091	0	0	0,00091
2	0,56843	0,37628	0	0,04804	0,00801	0	0	0	0
3	0,69286	0,23095	0	0,07698	0	0	0	0	0
4	0,55265	0,31367	0	0,13443	0	0	0	0	0
5	0,71484	0,14297	0	0,14297	0	0	0	0	0
6	1,00040	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	1,00040	0	0	0	0	0	0	0	0

Nota: Probabilidades de pasar de un estado i (filas) a un estado j (columnas).

Nuevamente, se observó un patrón de comunicación, donde los estados mayores (o paquetes de mayor longitud) tienen una probabilidad de ocurrencia menor, y que también se presentan debido a la comunicación con direcciones IP asociadas con actividad maliciosa, para este caso en particular, cuatro direcciones IP.

Direcciones IP involucradas:

TABLA 3-XX
Direcciones IP asociadas con la comunicación con paquetes TCP de mayor tamaño para Fareit

Dirección IP	Archivos que comunican	URL alojadas	Archivos descargables
217.160.223.131	4	104	0
52.72.88.241	24	0	0
66.96.163.236	0	11	0
194.204.43.84	0	36	1

Nota: dirección IP maliciosa, archivos maliciosos asociados, número de URL que apuntan a dicha IP y archivos descargables asociados.

Para obtener la cadena de Markov de la amenaza URSNIF se realizó el mismo análisis similar sobre los datos capturados. En este caso, los paquetes tienen longitudes de mayor variabilidad, por lo tanto, el espacio de estados es mucho mayor que para la amenaza Fareit (se identifican 608 estados en total). Al igual que para Fareit, el espacio de estados se reduce a nueve, obteniéndose los rangos de la Tabla 3-XXI. Vale la pena precisar que a menor número de estados la diferencia entre los rangos para cada estado aumenta, lo que permite obtener un pronóstico con menor exactitud.

TABLA 3-XXI
Frecuencia para los estados para URSNIF

Estado	Rango	Frecuencia	Proporción
1	54 – 2812 Bytes	100.315	0,650625
2	2813 – 5571 Bytes	53.162	0,344796
3	5572 – 8329 Bytes	659	0,004274
4	8330 – 11088 Bytes	31	0,000201
5	11089 – 13847 Bytes	5	0,000032
6	13848 – 16606Bytes	5	0,000032
7	16607 – 19364 Bytes	2	0,000013
8	19365 – 22123 Bytes	1	0,000006
9	22124 – 24882Bytes	3	0,000019
		154.183	

Nota: Estados con los que se construirá la matriz de transición de estados, rangos que limitan los tamaños de paquetes agrupados, número de paquetes dentro del rango y proporción respecto al total.

La matriz de frecuencias transicionales se muestra seguidamente (Tabla 3-XXII). Dichas frecuencias describen que los paquetes de longitud inferior a 5.000 bytes se presentan de forma más seguida dentro de la data.

TABLA 3-XXII
Matriz de frecuencias transicionales para URSNIF

	1	2	3	4	5	6	7	8	9	
1	69.100	30.989	199	14	4	4	1	1	3	**100.315**
2	30.970	22.156	24	11	1	0	0	0	0	**53.162**
3	209	12	436	1	0	1	0	0	0	**659**
4	25	3	0	3	0	0	0	0	0	**31**
5	4	0	0	0	0	0	1	0	0	**5**
6	4	0	0	1	0	0	0	0	0	**5**
7	0	1	0	1	0	0	0	0	0	**2**
8	1	0	0	0	0	0	0	0	0	**1**
9	2	1	0	0	0	0	0	0	0	**3**
										154.183

Nota: Número de paquetes de longitud i (columnas) que se encuentran luego de un paquete con longitud j (filas)

La matriz de probabilidad conjunta, así como la matriz de transición de estados (cadena de Markov) para esta última amenaza analizada se muestran seguidamente (Tablas 3-XXIII y 3-24).

TABLA 3-XXIII
Matriz de probabilidad conjunta para URSNIF

	1	2	3	4	5	6	7	8	9
1	0,448169	0,200988	0,001291	0,000091	0,000026	0,000026	0,000006	0,000006	0,000019
2	0,200865	0,143699	0,000156	0,000071	0,000006	0	0	0	0
3	0,001356	0,000078	0,002828	0,000006	0	0,000006	0	0	0
4	0,000162	0,000019	0	0,000019	0	0	0	0	0
5	0,000026	0	0	0	0	0	0,000006	0	0
6	0,000026	0	0	0,000006	0	0	0	0	0
7	0	0,000006	0	0,000006	0	0	0	0	0
8	0,000006	0	0	0	0	0	0	0	0
9	0,000013	0,000006	0	0	0	0	0	0	0

Nota: Probabilidad de que el sistema pase de un estado j (columnas) dado que estuvo en un estado j (filas).

TABLA 3-XXIV
Matriz de transición de estados para URSNIF

	1	2	3	4	5	6	7	8	9
1	0,68883	0,30892	0,00198	0,00014	0,00004	0,00004	0,00001	0,00001	0,00003
2	0,58256	0,41677	0,00045	0,00021	0,00002	0	0	0	0
3	0,31715	0,01821	0,66161	0,00152	0	0,00152	0	0	0
4	0,80646	0,09677	0	0,09677	0	0	0	0	0
5	0,80001	0	0	0	0	0	0,20000	0	0
6	0,80001	0	0	0,20000	0	0	0	0	0
7	0	0,50000	0	0,50000	0	0	0	0	0
8	1,00001	0	0	0	0	0	0	0	0
9	0,66667	0,33334	0	0	0	0	0	0	0

Nota: Probabilidades de pasar de un estado i (filas) a un estado j (columnas).

En este caso los paquetes de mayor longitud muestran comunicación con 15 direcciones IP maliciosas (Tabla 3-XXV):

TABLA 3-XXV
Direcciones IP asociadas con la comunicación con paquetes TCP de mayor tamaño para URSNIF

Dirección IP	Archivos que comunican	URL alojadas	Archivos que descarga
95.211.210.34	112	14	1
54.230.15.223	88	0	1
88.208.60.173	136	102	0
192.133.141.11	21	69	39
162.213.10.82	0	41	3
2.16.4.185	14	0	0
88.208.60.169	12	33	6
5.149.249.101	6	10	2
50.7.184.162	128	16	6
54.230.15.118	9	0	0
54.230.15.101	79	0	0
54.208.23.129	0	100	114
54.230.15.66	94	0	0
198.232.125.32	0	100	63
192.133.141.12	3	102	2

Nota: dirección IP maliciosa, archivos maliciosos asociados, número de URL que apuntan a dicha IP y archivos descargables asociados.

3.4 Reglas de Snort

Las reglas diseñadas a partir de los resultados obtenidos se observan en el Anexo D. Estas reglas incluyen los dominios maliciosos a los cuales realizan peticiones las amenazas estudiadas, las conexiones FTP sospechosas y las direcciones IP asociadas con los paquetes de mayor longitud, que, como ya se comprobó, tienen una probabilidad baja de aparición dentro de tráfico y están relacionados con actividad maliciosa.

Las reglas se pueden ver en el repositorio de Snort de la siguiente forma (Figura 3-8):

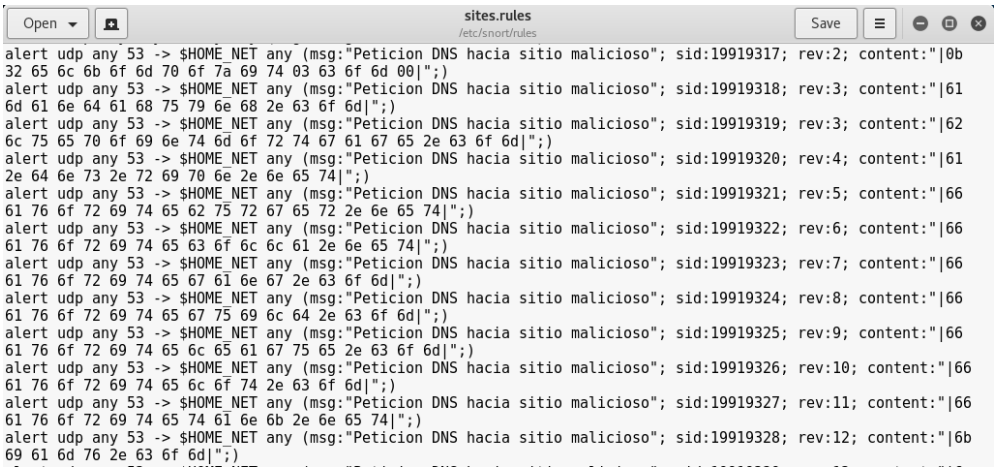


Fig. 3-8. Repositorio de reglas personalizadas.

Una vez insertadas las reglas, se puso en funcionamiento la captura de paquetes en Snort. Como se aprecia en la Figura 3-9 se inicializa el IDS junto con sus procesadores para comenzar a analizar el tráfico que pasa por la interfaz de red eth0. Las reglas parametrizadas generan alertas cuando encuentran patrones asociados a las APT estudiadas.

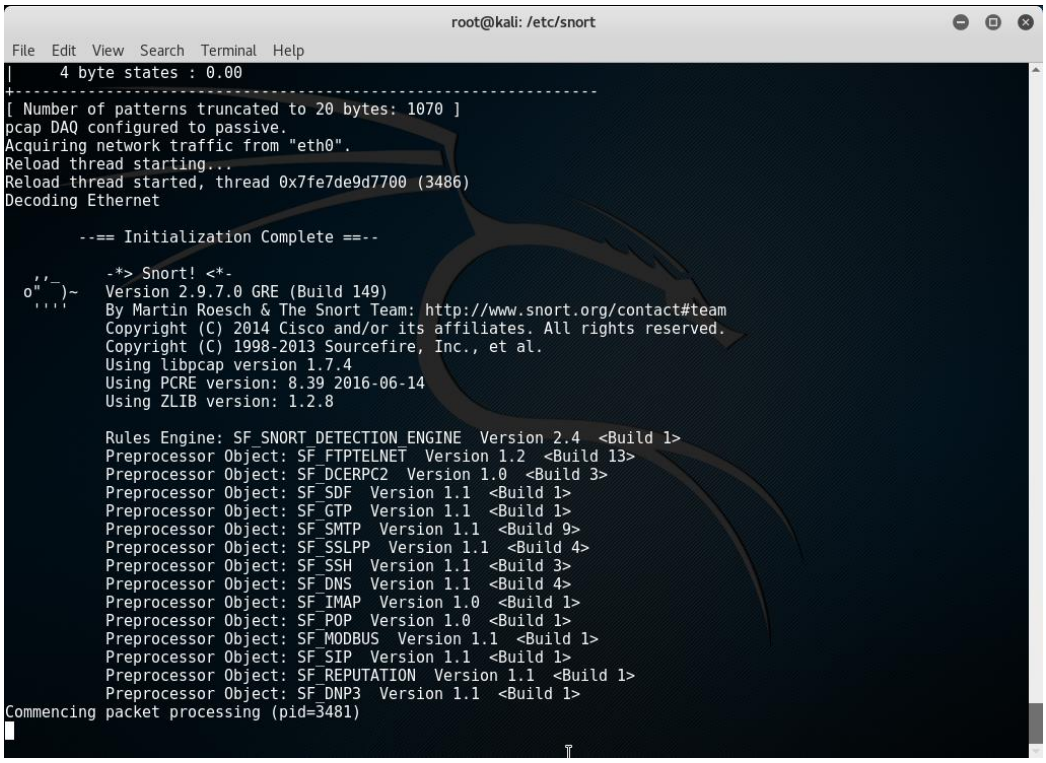


Fig. 3-9. Captura de paquetes en Snort.

En términos generales la metodología aplicada permitió identificar conexiones DNS y FTP a servidores maliciosos, así como paquetes de grandes longitudes que pueden estar asociados con una campaña de APT. Esta actividad pudo ser aislada y traducida a un conjunto de reglas de red que permitieron identificar la actividad sospechosa en un sistema comprometido.

4. Conclusiones y recomendaciones

4.1 Conclusiones

El tráfico de red se puede considerar como un proceso estocástico, dado que involucra variables aleatorias, donde sus posibles valores son sus estados característicos. El análisis de paquetes TCP en una red se valoró como una cadena de Markov, al definir el tamaño de los mismos como los estados por los que pasa un sistema dentro de una red y las probabilidades de que este se encuentre en un estado determinado a partir del estado anterior a medida que transcurre el tiempo.

En el caso de las APT estudiadas, se identificaron elementos o variables que permitieron la toma de decisiones a partir del tráfico capturado, relacionadas con el número de transiciones. Igualmente, se identificaron las probabilidades de terminar en un estado partiendo de un algún estado en particular, lo cual posibilitó de alguna forma pronosticar en términos de probabilidades el estado de este subsistema en el futuro. Los resultados obtenidos muestran que un estado normal del sistema se da generalmente cuando los paquetes TCP son de longitud pequeña. Los paquetes de longitud grande se presentan cuando el sistema se comunica con sistemas externos asociados con actividad maliciosa, y su tamaño se explica en que la herramienta utilizada por la APT necesita de estas conexiones para obtener código malicioso adicional o para sustraer la información objetivo.

El enfoque utilizado en este trabajo de investigación permitió obtener un modelo estadístico ajustado al comportamiento de una red infectada por una APT, con el cual se pueden hacer proyecciones del comportamiento futuro de un sistema. Esta proyección podría ser utilizada como insumo en herramientas de detección de amenazas para mejorar su eficacia y reducir el número de variables requeridas para sus análisis.

La evaluación de otros modelos estadísticos para realizar análisis y pronosticar el comportamiento de un sistema se puede complementar favorablemente con el campo de investigación relacionado con las APT. Los modelos de medias móviles, modelos ARIMA y SARIMA, modelos de Markov Ocultos (HMM), entre otros, han sido aplicados en diversos campos de la computación con resultados prometedores, por lo tanto, su aplicación sobre tráfico de red para detectar APT en estados tempranos o tardíos de infección puede resultar igualmente satisfactoria. Una comparación de estos modelos con el de cadenas de Markov aplicado en este trabajo de investigación, podría arrojar un modelo de mejor ajuste para los datos analizados.

Con el análisis realizado se extrajeron características relevantes en el tráfico de red que son útiles para la construcción de reglas en un Sistema de Detección de Intrusos (IDS), como lo son dominios maliciosos y direcciones IP asociadas con actividad sospechosa. Con el desarrollo de este trabajo de investigación se generaron un conjunto de reglas que permiten alertar sobre actividad maliciosa asociada a herramientas utilizadas por grupos APT en sus campañas. Cabe notar que el procedimiento aplicado no bloquea la actividad maliciosa, sino que alerta al administrador de la red para que actúe en casos de infección potencial por APT.

Luego de la aplicación de la metodología, se logró cumplir con cada uno de los objetivos propuestos. Así, en el primer objetivo específico se obtuvieron los datos estructurados asociados con tres APT. Para el segundo objetivo se obtuvieron también 111 dominios y un servidor FTP malicioso, así como el tamaño de paquetes relacionados con actividad maliciosa. Respecto al tercer objetivo, igualmente, se obtuvieron las matrices de transición de estados con las probabilidades que describen el comportamiento del sistema en un momento dado y que permiten identificar desviaciones. Finalmente, en cuanto al objetivo específico cuatro se alcanzó un conjunto de reglas que alertan sobre posibles conexiones a los dominios ya mencionados, así como a los paquetes relacionados con las APT estudiadas.

4.2 Recomendaciones

Como trabajo futuro se recomienda recopilar datos de un sistema de mayor complejidad, desde el cual se genere tráfico de red más diverso (por ejemplo, tráfico de voz, audio y video). Estas variaciones pueden derivar en un comportamiento caracterizado por paquetes de muchos otros protocolos y de mayor longitud, mayor número de conexiones externas a la red, lo que podría invalidar el modelo estudiado en esta investigación.

El cálculo de la matriz de transición de estados para un conjunto de observaciones, al ser un proceso repetitivo, puede ser fácilmente automatizado mediante un desarrollo simple en cualquier lenguaje de programación, incluyendo Visual Basic para macros de Excel. Este desarrollo puede ser de gran valor para identificar los estados de una matriz, así como sus frecuencias absolutas y condicionales, que a su vez son utilizadas para obtener la matriz. Dentro de las fuentes consultadas no se encontró un procedimiento automatizado para realizar dichos cálculos; asimismo, muchos de los casos de estudio de cadenas de Markov partían de probabilidades conocidas u homogéneas.

Las reglas de red pueden ser implementadas en un Sistema de Prevención de Intrusos (IPS) para que actúe de forma proactiva y bloquee la actividad maliciosa detectada. Las alertas de un IDS como Snort deben ser complementadas con herramientas adicionales como antivirus y firewalls que permitan ejecutar acciones concretas sobre la amenaza identificada.

A. Anexo A: Características del laboratorio

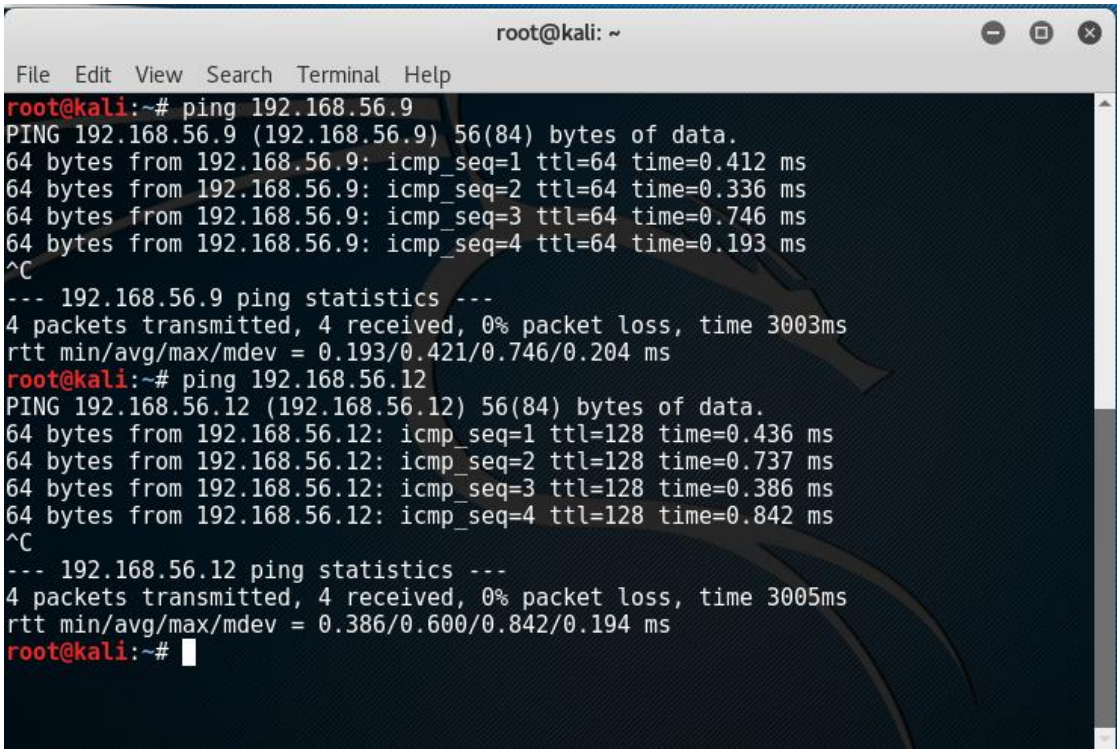
Se instalaron tres máquinas virtuales en el software de virtualización VMWare (VMWare Player). Los sistemas operativos seleccionados para las pruebas son Windows 7 y Ubuntu. Las características de las máquinas instaladas se detallan en la siguiente tabla:

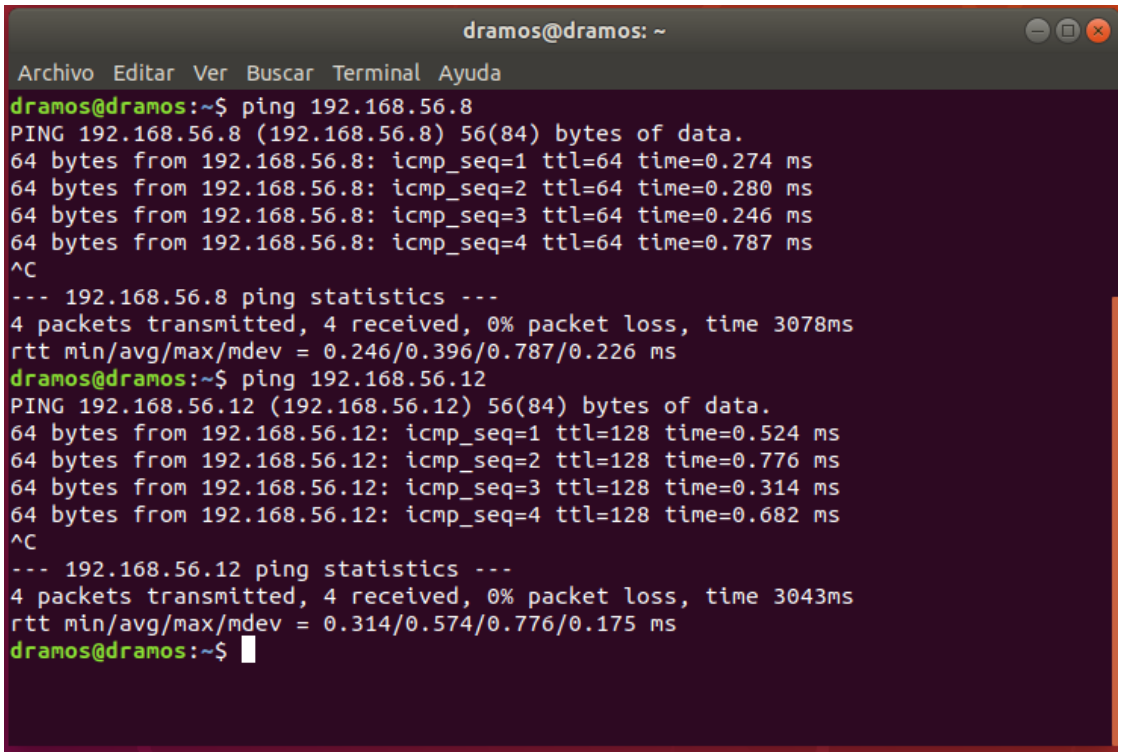
Aspecto	Kali Linux	Windows 7	Ubuntu
Versión	Kali-rolling	Professional	18.04.1 desktop
Memoria base (RAM)	2048 MB	2048 MB	2048 MB
Procesadores	1	1	1
Disco duro	40 GB	60 GB	30 GB
Modo Tarjeta de red	Red interna	Red interna	Red interna

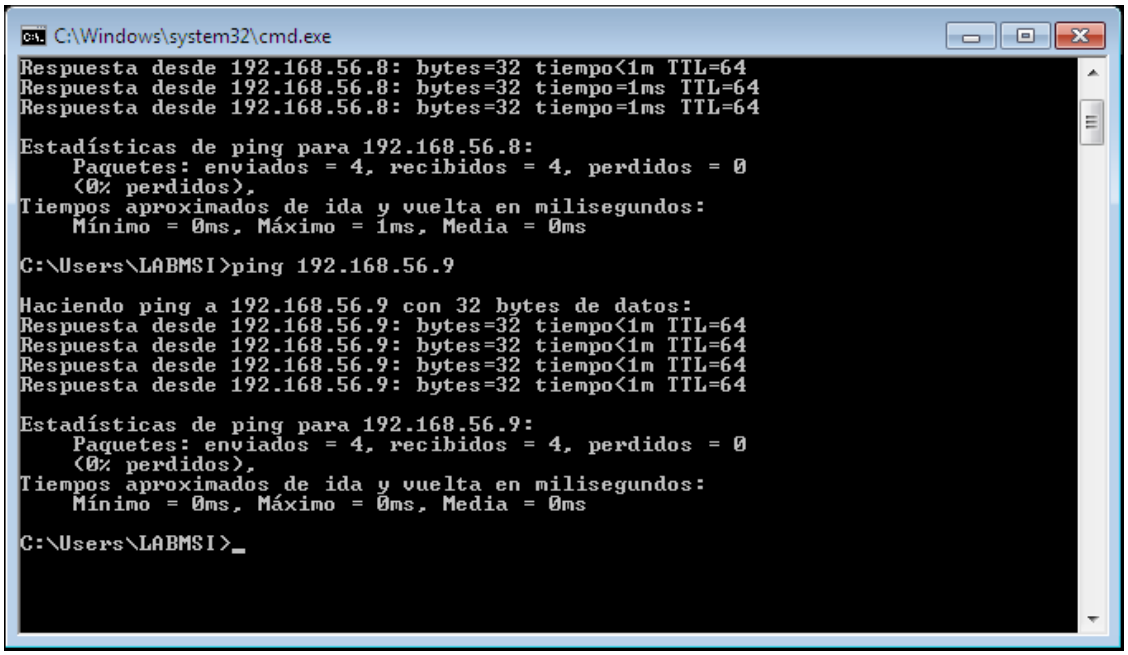
Para establecer conexión entre las máquinas se asignaron direcciones IP estáticas a las interfaces de red y se deshabilitaron las interfaces innecesarias.

Estación	Dirección IP asignada
Kali Linux	192.168.56.8
Ubuntu	192.168.56.9
Windows 7	192.168.56.12

Se realizaron pruebas de conexión entre las máquinas para validar la comunicación entre las mismas. Como se puede apreciar a continuación, las máquinas transmiten y reciben paquetes de datos con normalidad (se prueba mediante el comando ping):


root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ping 192.168.56.9
PING 192.168.56.9 (192.168.56.9) 56(84) bytes of data.
64 bytes from 192.168.56.9: icmp_seq=1 ttl=64 time=0.412 ms
64 bytes from 192.168.56.9: icmp_seq=2 ttl=64 time=0.336 ms
64 bytes from 192.168.56.9: icmp_seq=3 ttl=64 time=0.746 ms
64 bytes from 192.168.56.9: icmp_seq=4 ttl=64 time=0.193 ms
^C
--- 192.168.56.9 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.193/0.421/0.746/0.204 ms
root@kali:~# ping 192.168.56.12
PING 192.168.56.12 (192.168.56.12) 56(84) bytes of data.
64 bytes from 192.168.56.12: icmp_seq=1 ttl=128 time=0.436 ms
64 bytes from 192.168.56.12: icmp_seq=2 ttl=128 time=0.737 ms
64 bytes from 192.168.56.12: icmp_seq=3 ttl=128 time=0.386 ms
64 bytes from 192.168.56.12: icmp_seq=4 ttl=128 time=0.842 ms
^C
--- 192.168.56.12 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.386/0.600/0.842/0.194 ms
root@kali:~#


dramos@dramos: ~
Archivo Editar Ver Buscar Terminal Ayuda
dramos@dramos:~$ ping 192.168.56.8
PING 192.168.56.8 (192.168.56.8) 56(84) bytes of data.
64 bytes from 192.168.56.8: icmp_seq=1 ttl=64 time=0.274 ms
64 bytes from 192.168.56.8: icmp_seq=2 ttl=64 time=0.280 ms
64 bytes from 192.168.56.8: icmp_seq=3 ttl=64 time=0.246 ms
64 bytes from 192.168.56.8: icmp_seq=4 ttl=64 time=0.787 ms
^C
--- 192.168.56.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3078ms
rtt min/avg/max/mdev = 0.246/0.396/0.787/0.226 ms
dramos@dramos:~$ ping 192.168.56.12
PING 192.168.56.12 (192.168.56.12) 56(84) bytes of data.
64 bytes from 192.168.56.12: icmp_seq=1 ttl=128 time=0.524 ms
64 bytes from 192.168.56.12: icmp_seq=2 ttl=128 time=0.776 ms
64 bytes from 192.168.56.12: icmp_seq=3 ttl=128 time=0.314 ms
64 bytes from 192.168.56.12: icmp_seq=4 ttl=128 time=0.682 ms
^C
--- 192.168.56.12 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3043ms
rtt min/avg/max/mdev = 0.314/0.574/0.776/0.175 ms
dramos@dramos:~$


C:\Windows\system32\cmd.exe
Respuesta desde 192.168.56.8: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.56.8: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.56.8: bytes=32 tiempo=1ms TTL=64
Estadísticas de ping para 192.168.56.8:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 1ms, Media = 0ms
C:\Users\LABMSI>ping 192.168.56.9
Haciendo ping a 192.168.56.9 con 32 bytes de datos:
Respuesta desde 192.168.56.9: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.56.9: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.56.9: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.56.9: bytes=32 tiempo<1m TTL=64
Estadísticas de ping para 192.168.56.9:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 0ms, Media = 0ms
C:\Users\LABMSI>

B. Anexo B: Ataque Man in the Middle para la captura de paquetes

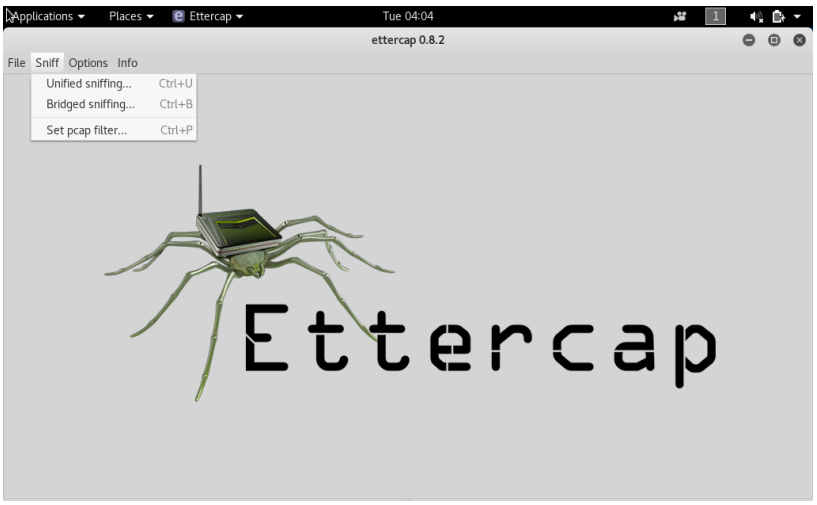
Para la captura de datos se simula un ataque Man In the Middle (MitM), el cual es un método en el que un atacante se interpone entre dos víctimas para captar paquetes o modificar la comunicación.

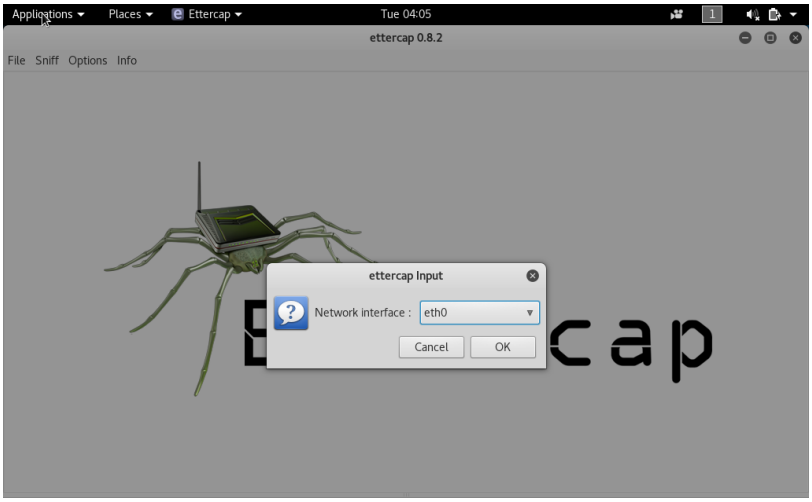
Es importante resaltar que este tipo de ataque se realiza debido a que no se cuenta con un suiche en el laboratorio que permita ver el tráfico de toda la red. Este método puede ser ofensivo; sin embargo, se utiliza porque se trata de un entorno no crítico, donde solo se busca interceptar el tráfico entre varias máquinas.

Lo que se busca con este ataque es que el equipo que se desea monitorear envíe todas las tramas a la estación Kali, donde se estará ejecutando el Wireshark. El proceso se lleva a cabo contaminando la caché de los equipos involucrados con una asociación IP/MAC falsa.

A continuación, se describen los pasos seguidos para la simulación del ataque:

1. En la estación Kali se abre la aplicación Ettercap (en el menú *Application -> Kali Linux -> Sniffing/Spoofing -> Network sniffers -> ettercap-graphical*).
2. En el menu *Sniff*, se da clic en *Unified Sniffing*, para ingresar a las opciones de sniffing y se selecciona la interface de red.





3. Se listan los hosts disponibles en la red para realizar el ataque. Se da clic en el menú *Hosts* y *Scan for Hosts*. Ettercap identifica los equipos y despliega las opciones disponibles junto con la cantidad de equipos encontrados. Para listar los hosts, se hace clic en el menú *Hosts -> Hosts List*.
4. En esta lista aparecen los equipos con su IP y MAC respectiva. Se seleccionan los equipos que serán objetivos del ataque y se añaden con las opciones *Add to Target*.
5. Con los targets seleccionados, se hace clic sobre el menú *mitm -> ARP poisoning* y se activa la opción para monitorear las conexiones remotas. Con esto intercepta la comunicación de los equipos en la red, la cual se encuentra redirigida hacia Kali. Desde Wireshark se puede observar el tráfico que fluye a través de la red.

C. Anexo C: Dominios maliciosos detectados

Dominios maliciosos asociados a amenaza Fareit

Dominio	Clasificación
2elkompozit.com	Malicioso
3.amandahuynh.com	Malicioso
3.bluepointmortgage.com	Malicioso
a.dns.ripn.net	Malicioso
favoriteburger.net	Malicioso
favoritecolla.net	Malicioso
favoritegang.com	Malicioso
favoriteguild.com	Malicioso
favoriteleague.com	Malicioso
favoritelot.com	Malicioso
favoritetank.net	Malicioso
kiamv.com	Malicioso
linerline.info	Malicioso
linertweet.com	Malicioso
martvarauto.ee	Malicioso
thiagoantonio.com.br	Malicioso
www.cleanmax.com.br	Malicioso
www.martvarauto.ee	Malicioso

Dominios maliciosos asociados a amenaza URSNIF

Dominio	Clasificación
1txg5964.ru	Malicioso
2cxan.trackvoluum.com	Malicioso
3uorg03dxfy.ru	Malicioso
ad.sxp.smartclip.net	Malicioso
adadvisor.net	Sospechoso
admin.nsatc.net	Malicioso
ads.stickyadstv.com	Malicioso
cdnrep.reimage.com	Malicioso
client.updsoft.net	Malicioso
cm.g.doubleclick.net	Malicioso
cmi.ironbeast.io	Malicioso
connect.facebook.net	Sospechoso
counter.yadro.ru	Malicioso

counter99.com	Malicioso
d.adroll.com	Malicioso
d.turn.com	Malicioso
d11m2p9mpffp32.cloudfront.net	Malicioso
d1vh0xkmncek4z.cloudfront.net	Malicioso
data6.saveserpnow.com	Malicioso
dbz0abtfpkod2.cloudfront.net	Malicioso
dns1.registrar-servers.com	Malicioso
download-file.ru	Malicioso
dpm.demdex.net	Malicioso
dt.adsafeprotected.com	Malicioso
dt.scanscout.com	Malicioso
dxedge-prod-lb-404808087.eu-central-1.elb.amazonaws.com	Sospechoso
e.nexac.com	Malicioso
feed.sugarpulse.com	Malicioso
fw.adsafeprotected.com	Malicioso
g.symcd.com	Malicioso
getmywebshield.org	Malicioso
gmt-max.org	Malicioso
ib.anycast.adnxs.com	Sospechoso
icontent.us	Malicioso
idsync.rlcdn.com	Malicioso
ih.adscale.de	Malicioso
install-apps.com	Malicioso
l.facebook.com	Sospechoso
livestatscounter.com	Malicioso
loadm.exelator.com	Malicioso
m.addthisedge.com	Malicioso
map.media6degrees.com	Malicioso
n46gd0nenr1az.ru	Malicioso
numbercounters.com	Malicioso
nyctradersacademy.com	Malicioso
o.ss2.us	Malicioso
ocsp.entrust.net	Malicioso
pix04.revsci.net	Malicioso
pixel.advertising.com	Malicioso
pixel.quantserve.com	Malicioso
pixel.tapad.com	Malicioso
pki.google.com	Malicioso
post.securestudies.com	Malicioso
prod.d.ssl.global.fastlylb.net	Malicioso
pug22000c.pubmatic.com	Malicioso
px.owneriq.net	Malicioso
r.turn.com.akadns.net	Malicioso
rcdelivery.integraclick.netdna-cdn.com	Malicioso
reimageplus.com	Malicioso
rtd.tubemogul.com	Sospechoso
s.ytimg.com	Malicioso
s2.symcb.com	Malicioso
s2s.rafotech.com	Malicioso

saveserpnow.com	Malicioso
sb.scorecardresearch.com	Malicioso
sc.iasds01.com	Malicioso
secure-au.imrworldwide.com	Malicioso
securepubads.g.doubleclick.net	Malicioso
ss.symcd.com	Malicioso
star.c10r.facebook.com	Sospechoso
star.facebook.com	Sospechoso
static.doubleclick.net	Malicioso
static.hotjar.com	Malicioso
static.hotjar.netdna-cdn.com	Malicioso
static.squarespace.com	Sospechoso
static.xx.fbcdn.net	Sospechoso
static1.squarespace.com	Malicioso
staticxx.facebook.com	Sospechoso
stats.g.doubleclick.net	Malicioso
sync.search.spotxchange.com	Malicioso
tags.expo9.exponential.com	Malicioso
teredo.ipv6.microsoft.com	Sospechoso
timeforvictory144.ru	Malicioso
trotux.com	Malicioso
turgolde.ru	Malicioso
urlvalidation.com	Malicioso
us-u.openx.net	Malicioso
www.icoway.net	Malicioso
www.reimageplus.com	Malicioso
www.technologietrudeau.com	Malicioso
www.theguardian.com	Sospechoso
www.trotux.com	Malicioso
yt3.ggpht.com	Malicioso

D. Anexo D: Reglas de Snort

Reglas asociadas a DNS maliciosos de Fareit:

alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919317; rev:2; content:"|0b alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919317; rev:2; content:"|0b 32 65 6c 6b 6f 6d 70 6f 7a 69 74 03 63 6f 6d 00|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919318; rev:3; content:"|61 6d 61 6e 64 61 68 75 79 6e 68 2e 63 6f 6d|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919319; rev:3; content:"|62 6c 75 65 70 6f 69 6e 74 6d 6f 72 74 67 61 67 65 2e 63 6f 6d|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919320; rev:4; content:"|61 2e 64 6e 73 2e 72 69 70 6e 2e 6e 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919321; rev:5; content:"|66 61 76 6f 72 69 74 65 62 75 72 67 65 72 2e 6e 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919322; rev:6; content:"|66 61 76 6f 72 69 74 65 63 6f 6c 6c 61 2e 6e 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919323; rev:7; content:"|66 61 76 6f 72 69 74 65 67 61 6e 67 2e 63 6f 6d|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919324; rev:8; content:"|66 61 76 6f 72 69 74 65 67 75 69 6c 64 2e 63 6f 6d|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919325; rev:9; content:"|66 61 76 6f 72 69 74 65 6c 65 61 67 75 65 2e 63 6f 6d|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919326; rev:10; content:"|66 61 76 6f 72 69 74 65 6c 6f 74 2e 63 6f 6d|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919327; rev:11; content:"|66 61 76 6f 72 69 74 65 74 61 6e 6b 2e 6e 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919328; rev:12; content:"|6b 69 61 6d 76 2e 63 6f 6d|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919329; rev:13; content:"|6c 69 6e 65 72 6c 69 6e 65 2e 69 6e 66 6f|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919330; rev:14; content:"|6c 69 6e 65 72 74 77 65 65 74 2e 63 6f 6d|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919331; rev:15; content:"|6d 61 72 74 76 61 72 61 75 74 6f 2e 65 65|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919332; rev:16; content:"|74 68 69 61 67 6f 61 6e 74 6f 6e 69 6f 2e 63 6f 6d 2e 62 72|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919333; rev:17; content:"|77 77 77 2e 63 6c 65 61 6e 6d 61 78 2e 63 6f 6d 2e 62 72|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919334; rev:18; content:"|77 77 77 2e 6d 61 72 74 76 61 72 61 75 74 6f 2e 65 65|";)

alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919335; rev:19; content:"|31 74 78 67 35 39 36 34 2e 72 75|";)

alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919336; rev:20; content:"|32 63 78 61 6E 2E 74 72 61 63 6B 76 6F 6C 75 75 6D 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919337; rev:21; content:"|33 75 6F 72 67 30 33 64 78 66 79 2E 72 75|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919338; rev:22; content:"|61 64 2E 73 78 70 2E 73 6D 61 72 74 63 6C 69 70 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919339; rev:23; content:"|61 64 61 64 76 69 73 6F 72 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919340; rev:24; content:"|61 64 6D 69 6E 2E 6E 73 61 74 63 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919341; rev:25; content:"|61 64 73 2E 73 74 69 63 6B 79 61 64 73 74 76 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919342; rev:26; content:"|63 64 6E 72 65 70 2E 72 65 69 6D 61 67 65 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919343; rev:27; content:"|63 6C 69 65 6E 74 2E 75 70 64 73 6F 66 74 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919344; rev:28; content:"|63 6D 2E 67 2E 64 6F 75 62 6C 65 63 6C 69 63 6B 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919345; rev:29; content:"|63 6D 69 2E 69 72 6F 6E 62 65 61 73 74 2E 69 6F|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919346; rev:30; content:"|63 6F 6E 6E 65 63 74 2E 66 61 63 65 62 6F 6F 6B 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919347; rev:31; content:"|63 6F 75 6E 74 65 72 2E 79 61 64 72 6F 2E 72 75|";)

alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919348; rev:32; content:"|63 6F 75 6E 74 65 72 39 39 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919349; rev:33; content:"|64 2E 61 64 72 6F 6C 6C 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919350; rev:34; content:"|64 2E 74 75 72 6E 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919351; rev:35; content:"|64 31 31 6D 32 70 39 6D 70 66 66 70 33 32 2E 63 6C 6F 75 64 66 72 6F 6E 74 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919353; rev:37; content:"|64 61 74 61 36 2E 73 61 76 65 73 65 72 70 6E 6F 77 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919354; rev:38; content:"|64 62 7A 30 61 62 74 66 70 6B 6F 64 32 2E 63 6C 6F 75 64 66 72 6F 6E 74 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919355; rev:39; content:"|64 6E 73 31 2E 72 65 67 69 73 74 72 61 72 2D 73 65 72 76 65 72 73 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919356; rev:40; content:"|64 6F 77 6E 6C 6F 61 64 2D 66 69 6C 65 2E 72 75|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919357; rev:41; content:"|64 70 6D 2E 64 65 6D 64 65 78 2E 6E 65 74|";)

alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919358; rev:42; content:"|64 74 2E 61 64 73 61 66 65 70 72 6F 74 65 63 74 65 64 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919359; rev:43; content:"|64 74 2E 73 63 61 6E 73 63 6F 75 74 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919360; rev:44; content:"|64 78 65 64 67 65 2D 70 72 6F 64 2D 6C 62 2D 34 30 34 38 30 38 30 38 37 2E 65 75 2D 63 65 6E 74 72 61 6C 2D 31 2E 65 6C 62 2E 61 6D 61 7A 6F 6E 61 77 73 2E 63 6F 6D|";)

```
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919361; rev:45; content:"|65 2E 6E 65 78 61 63 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919362; rev:46; content:"|66 65 65 64 2E 73 75 67 61 72 70 75 6C 73 65 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919363; rev:47; content:"|66 77 2E 61 64 73 61 66 65 70 72 6F 74 65 63 74 65 64 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919364; rev:48; content:"|67 2E 73 79 6D 63 64 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919365; rev:49; content:"|67 65 74 6D 79 77 65 62 73 68 69 65 6C 64 2E 6F 72 67|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919366; rev:50; content:"|67 6D 74 2D 6D 61 78 2E 6F 72 67|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919367; rev:51; content:"|69 62 2E 61 6E 79 63 61 73 74 2E 61 64 6E 78 73 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919368; rev:52; content:"|69 63 6F 6E 74 65 6E 74 2E 75 73|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919369; rev:53; content:"|69 64 73 79 6E 63 2E 72 6C 63 64 6E 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919370; rev:48; content:"|69 68 2E 61 64 73 63 61 6C 65 2E 64 65|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919371; rev:49; content:"|69 6E 73 74 61 6C 6C 2D 61 70 70 73 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919372; rev:50; content:"|6C 2E 66 61 63 65 62 6F 6F 6B 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919373; rev:51; content:"|6C 69 76 65 73 74 61 74 73 63 6F 75 6E 74 65 72 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919374; rev:52; content:"|6C 6F 61 64 6D 2E 65 78 65 6C 61 74 6F 72 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919375; rev:53; content:"|6D 2E 61 64 64 74 68 69 73 65 64 67 65 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919376; rev:54; content:"|6D 61 70 2E 6D 65 64 69 61 36 64 65 67 72 65 65 73 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919377; rev:55; content:"|6E 34 36 67 64 30 6E 65 6E 72 31 61 7A 2E 72 75|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919378; rev:56; content:"|6E 75 6D 62 65 72 63 6F 75 6E 74 65 72 73 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919379; rev:57; content:"|6E 79 63 74 72 61 64 65 72 73 61 63 61 64 65 6D 79 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919380; rev:58; content:"|6F 2E 73 73 32 2E 75 73|";)

alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919381; rev:59; content:"|6F 63 73 70 2E 65 6E 74 72 75 73 74 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919382; rev:60; content:"|70 69 78 30 34 2E 72 65 76 73 63 69 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919383; rev:61; content:"|70 69 78 65 6C 2E 61 64 76 65 72 74 69 73 69 6E 67 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919384; rev:62; content:"|70 69 78 65 6C 2E 71 75 61 6E 74 73 65 72 76 65 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919385; rev:63; content:"|70 69 78 65 6C 2E 74 61 70 61 64 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919386; rev:64; content:"|70 6B 69 2E 67 6F 6F 67 6C 65 2E 63 6F 6D|";)
```

alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919387; rev:65; content:"|70 6F 73 74 2E 73 65 63 75 72 65 73 74 75 64 69 65 73 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919388; rev:66; content:"|70 72 6F 64 2E 64 2E 73 73 6C 2E 67 6C 6F 62 61 6C 2E 66 61 73 74 6C 79 6C 62 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919389; rev:67; content:"|70 75 67 32 32 30 30 30 63 2E 70 75 62 6D 61 74 69 63 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919390; rev:68; content:"|70 78 2E 6F 77 6E 65 72 69 71 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919391; rev:69; content:"|72 2E 74 75 72 6E 2E 63 6F 6D 2E 61 6B 61 64 6E 73 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919392; rev:70; content:"|72 63 64 65 6C 69 76 65 72 79 2E 69 6E 74 65 67 72 61 63 6C 69 63 6B 2E 6E 65 74 64 6E 61 2D 63 64 6E 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919393; rev:71; content:"|72 65 69 6D 61 67 65 70 6C 75 73 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919394; rev:72; content:"|72 74 64 2E 74 75 62 65 6D 6F 67 75 6C 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919395; rev:73; content:"|73 2E 79 74 69 6D 67 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919396; rev:74; content:"|73 32 2E 73 79 6D 63 62 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919397; rev:75; content:"|73 32 73 2E 72 61 66 6F 74 65 63 68 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919398; rev:76; content:"|73 61 76 65 73 65 72 70 6E 6F 77 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919399; rev:77; content:"|73 62 2E 73 63 6F 72 65 63 61 72 64 72 65 73 65 61 72 63 68 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919400; rev:78; content:"|73 63 2E 69 61 73 64 73 30 31 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919401; rev:78; content:"|73 65 63 75 72 65 2D 61 75 2E 69 6D 72 77 6F 72 6C 64 77 69 64 65 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919402; rev:79; content:"|73 65 63 75 72 65 70 75 62 61 64 73 2E 67 2E 64 6F 75 62 6C 65 63 6C 69 63 6B 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919403; rev:80; content:"|73 73 2E 73 79 6D 63 64 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919404; rev:81; content:"|73 74 61 72 2E 63 31 30 72 2E 66 61 63 65 62 6F 6F 6B 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919405; rev:82; content:"|73 74 61 72 2E 66 61 63 65 62 6F 6F 6B 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919406; rev:83; content:"|73 74 61 74 69 63 2E 64 6F 75 62 6C 65 63 6C 69 63 6B 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919407; rev:84; content:"|73 74 61 74 69 63 2E 68 6F 74 6A 61 72 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919408; rev:85; content:"|73 74 61 74 69 63 2E 68 6F 74 6A 61 72 2E 6E 65 74 64 6E 61 2D 63 64 6E 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919409; rev:86; content:"|73 74 61 74 69 63 2E 73 71 75 61 72 65 73 70 61 63 65 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919410; rev:87; content:"|73 74 61 74 69 63 2E 78 78 2E 66 62 63 64 6E 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919411; rev:88; content:"|73 74 61 74 69 63 31 2E 73 71 75 61 72 65 73 70 61 63 65 2E 63 6F 6D|";)

alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919412; rev:89; content:"|73 74 61 74 69 63 78 78 2E 66 61 63 65 62 6F 6F 6B 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919413; rev:90; content:"|73 74 61 74 73 2E 67 2E 64 6F 75 62 6C 65 63 6C 69 63 6B 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919414; rev:91; content:"|73 79 6E 63 2E 73 65 61 72 63 68 2E 73 70 6F 74 78 63 68 61 6E 67 65 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919415; rev:92; content:"|74 61 67 73 2E 65 78 70 6F 39 2E 65 78 70 6F 6E 65 6E 74 69 61 6C 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919416; rev:92; content:"|74 65 72 65 64 6F 2E 69 70 76 36 2E 6D 69 63 72 6F 73 6F 66 74 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919417; rev:93; content:"|74 69 6D 65 66 6F 72 76 69 63 74 6F 72 79 31 34 34 2E 72 75|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919418; rev:94; content:"|74 72 6F 74 75 78 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919419; rev:95; content:"|74 75 72 67 6F 6C 64 65 2E 72 75|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919420; rev:96; content:"|75 72 6C 76 61 6C 69 64 61 74 69 6F 6E 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919421; rev:97; content:"|75 73 2D 75 2E 6F 70 65 6E 78 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919422; rev:98; content:"|77 77 77 2E 69 63 6F 77 61 79 2E 6E 65 74|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919423; rev:99; content:"|77 77 77 2E 72 65 69 6D 61 67 65 70 6C 75 73 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919424; rev:100; content:"|77 77 77 2E 74 65 63 68 6E 6F 6C 6F 67 69 65 74 72 75 64 65 61 75 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919425; rev:101; content:"|77 77 77 2E 74 68 65 67 75 61 72 64 69 61 6E 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919426; rev:102; content:"|77 77 77 2E 74 72 6F 74 75 78 2E 63 6F 6D|";)
alert udp any 53 -> $HOME_NET any (msg:"Peticion DNS hacia sitio malicioso"; sid:19919428; rev:103; content:"|79 74 33 2E 67 67 70 68 74 2E 63 6F 6D|";)

Direcciones IP de mayor tamaño para Fareit:

alert tcp $HOME_NET any <> 199.231.188.109 any (msg:"Conexion a IP maliciosa"; sid:19919430; rev:104;)
alert tcp $HOME_NET any <> 52.18.63.80 any (msg:"Conexion hacia IP maliciosa"; sid:19919431; rev:105;)
alert tcp $HOME_NET any <> 217.160.223.131 any (msg:"Conexión hacia IP maliciosa"; sid:19919432; rev:106;)
alert tcp $HOME_NET any <> 52.72.88.241 any (msg:"Conexión hacia IP maliciosa"; sid:19919433; rev:107;)
alert tcp $HOME_NET any <> 66.96.163.236 any (msg:"Conexión hacia IP maliciosa"; sid:19919434; rev:108;)
alert tcp $HOME_NET any <> 194.204.43.84 any (msg:"Conexión hacia IP maliciosa"; sid:19919435; rev:109;)

Direcciones IP de mayor tamaño para Ursnif:

alert tcp $HOME_NET any <> 192.133.141.12 any (msg:"Conexión hacia IP maliciosa"; sid:19919436; rev:110;)
alert tcp $HOME_NET any <> 198.232.125.32 any (msg:"Conexión hacia IP maliciosa"; sid:19919437; rev:111;)
alert tcp $HOME_NET any <> 54.230.15.66 any (msg:"Conexión hacia IP maliciosa"; sid:19919438; rev:112;)
alert tcp $HOME_NET any <> 54.208.23.129 any (msg:"Conexión hacia IP maliciosa"; sid:19919439; rev:113;)
alert tcp $HOME_NET any <> 54.230.15.101 any (msg:"Conexión hacia IP maliciosa"; sid:19919440; rev:114;)
alert tcp $HOME_NET any <> 54.230.15.118 any (msg:"Conexión hacia IP maliciosa"; sid:19919441; rev:115;)
alert tcp $HOME_NET any <> 50.7.184.162 any (msg:"Conexión hacia IP maliciosa"; sid:19919442; rev:116;)
alert tcp $HOME_NET any <> 5.149.249.101 any (msg:"Conexión hacia IP maliciosa"; sid:19919443; rev:117;)
alert tcp $HOME_NET any <> 88.208.60.169 any (msg:"Conexión hacia IP maliciosa"; sid:19919444; rev:118;)
alert tcp $HOME_NET any <> 2.16.4.185 any (msg:"Conexión hacia IP maliciosa"; sid:19919445; rev:119;)
alert tcp $HOME_NET any <> 162.213.10.82 any (msg:"Conexión hacia IP maliciosa"; sid:19919446; rev:120;)
alert tcp $HOME_NET any <> 192.133.141.11 any (msg:"Conexión hacia IP maliciosa"; sid:19919447; rev:121;)
alert tcp $HOME_NET any <> 88.208.60.173 any (msg:"Conexión hacia IP maliciosa"; sid:19919448; rev:122;)
alert tcp $HOME_NET any <> 54.230.15.223 any (msg:"Conexión hacia IP maliciosa"; sid:19919449; rev:123;)
alert tcp $HOME_NET any <> 95.211.210.34 any (msg:"Conexión hacia IP maliciosa"; sid:19919450; rev:124;)

Intentos de conexión FTP:

alert tcp any any -> $HOME_NET 21 (msg:”Intento de conexión FTP”; sid:19919451; rev:125;)

Bibliografía

[1] M. González, “Si hay un ciberataque que merece ser llamado “de película”, ése es el sufrido por Sony Pictures”, 2014. [En línea]. Disponible: https://www.xataka.com/aplicaciones/si-hay-un-ciberataque-que-merece-ser-llamado-de-pelicula-ese-es-el-sufrido-por-sony-pictures

[2] K. Baumgartner, J. Guerrero-Saade y C. Raiu, “Boletín de seguridad Kaspersky: Predicciones sobre amenazas para el 2018”, 2017. [En línea]. Disponible: https://securelist.lat/boletin-de-seguridad-kaspersky-predicciones-sobre-amenazas-para-el-2018/85748/

[3] T. Magee, “Cybersecurity trends 2019”, 2019. [En línea] Disponible: https://www.computerworlduk.com/security/security-trends-for-2019-3689719/

[4] TrendMicro, “Mapping the Future: Dealing With Pervasive and Persistent Threats”, 2018. [En línea] Disponible: https://www.trendmicro.com/vinfo/us/security/research-and-analysis/predictions/2019

[5] M. S. Manggalanny y K. Ramli, “Combination of DNS traffic analysis: A design to enhance APT detection”, en *3rd International Conference on Science and Technology - Computer (ICST)*, 2017, pp. 171-175.

[6] Y. C. Shi, G. Chen y J. Li, “Malicious domain name detection based on extreme machine learning”, *Neural Processing Letters,* vol. 48, no. 3, pp. 1347-1357, 2018.

[7] T. M. Debatty, W. Mees y T. Gilon, “Graph-based APT detection”, en *International Conference on Military Communications and Information Systems (ICMCIS)*, 2018, pp. 1-8.

[8] A. Oprea, Z. Li, T. F. Yen, S. H. Chin y S. Alrwais, “Detection of Early-Stage Enterprise Infection by Mining Large-Scale Log Data”, en *45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2015, pp. 45-56.

[9] J.-S. Wu, Y.-J. Lee, T.-E. Wei, C.-H. Hsieh y C.-M. Lai, “*ChainSpot*: Mining Service Logs for Cyber Security Threat Detection”, en *IEEE TrustCom/BigDataSE/ISPA*, 2016, pp. 1867-1874.

[10] W. F. Matsuda, M. Fujimoto y T. Mitsunaga, “Detecting APT attacks against Active Directory using Machine Leaning”, en *IEEE Conference on Applications, Information and Network Security (AINS)*, 2018, pp. 60-65.

[11] I. Ghafir, V. Prenosil, R. Hegarty, M. Hammoudeh, L. Han, R. Rabie y F. Aparicio-Navarro, “Detection of advanced persistent threat using machine-learning correlation analysis”, *Future Generation Computer Systems*, no. 89, pp. 349-359, 2018. doi [10.1016/j.future.2018.06.055]

[12] S. Chandran, H. Poroli y P. Poornachandran, “An Efficient Classification Model for Detecting Advanced Persistent Threat”, en *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015, pp. 2001-2009.

[13] D. Moon, H. Im, I. Kim, J. H. Park, “DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks”, *The Journal of supercomputing*, vol. 73, no. 7, pp. 2881-2895, 2017.

[14] L. Invernizzi, S. Miskovic, R. Torres, S. Saha, S.-J. Lee, M. Mellia, C. Kruegel y G. Vigna, “Nazca: Detecting Malware Distribution in Large-Scale Networks”, en *NDSS*, 2014, pp. 23-26.

[15] J. Lu, K. Chen, Z. Zhuo, X. Zhang, “A temporal correlation and traffic analysis approach for APT attacks detection”, *Cluster computing*, pp. 1-12, 2017.

[16] P. K. Sharma, S. Y. Moon, D. Moon y J. H. Park, “DFA-AD: a distributed framework architecture for the detection of advanced persistent threats”, *Cluster Computing*, vol. 20, no. 1, pp. 597-60, 2017.

[17] T. S. Prasad y N. R. Kisore, “Application of Hidden Markov Model for Classifying Metamorphic Virus”. en *IEEE International Advance Computing Conference (IACC)*, 2015, pp. 1201-1206.

[18] C. H. Hsieh, C. M. Lai, C. H. Mao, T. C. Kao y K. C. Lee, “AD2: Anomaly Detection on Active Directory Log Data for Insider Threat Monitoring”, en *International Carnahan Conference on Security Technology (ICCST)*, 2015, pp. 287-292.

[19] R. Ross, *Managing Information Security Risk: Organization, Mission, and Information System View*. Gaithersburg, MD: National Institute of Standards and Technology, 2011. [En línea]. Disponible: https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=908030

[20] FireEye, “Advanced Threat Report 2013”, 2013. [En línea]. Disponible: https://www2.fireeye.com/advanced-threat-report-2013.html

[21] P. Giura y W. Wang, *A Context-Based Detection Framework for Advanced Persistent Threats. International Conference on Cyber Security*. Washington, DC: IEEE Computer Society, 2013.

[22] R. Jasek, M. Kolarik y T. Vymola, “APT detection system using honeypots”, en *Proceedings of the 13th International Conference on Applied Informatics and Communications (AIC'13)*, 2013, pp. 25-29.

[23] M. Ussath, D. Jaeger, F. Cheng y C. Meinel, “Advanced persistent threats: Behind the scenes”, en *2016 Annual Conference on Information Science and Systems (CISS)*, 2016.

[24] J. Vukalović y D. Delija, “Advanced persistent threats-detection and defense”, en *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015.

[25] O. S. Adebayo y N. AbdulAziz, “An intelligence based model for the prevention of advanced cyber-attacks”, en *The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M)*, 2014, pp. 1-5.

[26] P. Chen, L. Desmet y C. Huygens, “A study on advanced persistent threats”, en *IFIP International Conference on Communications and Multimedia Security*, 2014, pp. 63-72.

[27] K. Backman, J. Myers, C. Ahearn, M. Franco y P. Beardmore, “Peering into Glassrat. A Zero Detection Trojan from China”, 2015. [En línea]. Disponible: https://paper.seebug.org/papers/APT/APT_CyberCriminal_Campagin/2015/2015.11.23.PEERING_INTO_GLASSRAT/GlassRAT-final.pdf

[28] C. Doman, “Moonlight – Targeted attacks in the Middle East”, 2016. [En línea]. Disponible: https://blog.vectra.ai/blog/moonlight-middle-east-targeted-attacks

[29] A. Dulaunoy, K. Maxwell, N. Hausrath, R. Scott, A. J. y D. Weinstein, “APT Notes”, 2017. [En línea]. Disponible: https://github.com/kbandla/APTnotes

[30] Tecnología & Informática, “Tipos de redes informáticas. ¿Qué es una red? LAN, WAN, MAN, WLAN, WMAN, WWMAN, SAN, PAN”, 2013. [En línea]. Disponible: https://tecnologia-informatica.com/tipos-de-redes-informaticas-lan-wan-man-wlan-wman-wwman-san-pan/

[31] Universidad Autónoma del Estado de Hidalgo, “Redes de computadoras. Apuntes digitales”, 2009. [En línea]. Disponible: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro35/index.html

[32] C. Villagómez, “Comunidad Informática CCM”, 2017. [En línea]. Disponible: https://es.ccm.net/contents/274-el-protocolo-ip

[33] Seguridad y redes, “Wireshark / Windump. Análisis capturas tráfico red. Interpretación Datagrama IP. (Actualización).”, 2009. [En línea]. Disponible: https://seguridadyredes.wordpress.com/2009/11/05/wireshark-windump-analisis-capturas-trafico-red-interpretacian-datagrama-ip-actualizacian/

[34] B. Bylina y J. Bylina, “Using Markov chains for modelling networks”, *Annales UMCS Informatica*, vol. AI, no. 3, pp. 27-34, 2005.

[35] F. Hillier y L. Gerald, “Introducción a la Investigación de Operaciones”, México: Mc Graw Hill, 2010.

[36] S. Prada A., “Cadenas de Markov en la Investigación del Genoma”, Galicia, 2013.

[37] R. Hernández, C. Fernández y C. Baptista, *Metodología de investigación*, 2a ed. México: McGraw-Hill, 2010.

[38] J. A. Castillo, “Mejores aplicaciones de virtualización para Windows y Linux”, Profesional Review, 2018. [En línea]. Disponible: https://www.profesionalreview.com/2018/11/09/aplicaciones-virtualizacion/

[39] Rapid7, “Top Threat Actors and Their Tactics, Techniques, Tools, and Targets”, 2017. [En línea]. Disponible: https://blog.rapid7.com/2017/05/16/top-threat-actors/

[40] FireEye, “Advanced Persistent Threat Groups Who's who of cyber threat actors”, 2019. [En línea]. Disponible: https://www.fireeye.com/current-threats/apt-groups.html

[41] F-Secure, “COSMICDUKE. Cosmu with a twist of MiniDuke”, 2014. [En línea]. Disponible: https://www.f-secure.com/documents/996508/1030745/cosmicduke_whitepaper.pdf

[42] McAfee Labs, “Threats Report”, 2017. [En línea]. Disponible: https://www.mcafee.com/enterprise/en-us/assets/infographics/infographic-threats-report-jun-2017.pdf

[43] TrendMicro, “FAREIT”, TrendMicro Threat Encyclopedia, 2014. [En línea]. Disponible: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/fareit

[44] Kaspersky, “TROJAN-PSW.WIN32.FAREIT”, Kaspersky Labs, 2015. [En línea]. Disponible: https://threats.kaspersky.com/mx/threat/Trojan-PSW.Win32.Fareit/

[45] S. Gatlan, “New Ursnif Malware Campaign Uses Fileless Infection to Avoid Detection”, Bleeping Computer, 2019. [En línea]. Disponible: https://www.bleepingcomputer.com/news/security/new-ursnif-malware-campaign-uses-fileless-infection-to-avoid-detection/

[46] TrendMicro Blog, “URSNIF, EMOTET, DRIDEX and BitPaymer Gangs Linked by a Similar Loader”, 2018. [En línea]. Disponible: https://blog.trendmicro.com/trendlabs-security-intelligence/ursnif-emotet-dridex-and-bitpaymer-gangs-linked-by-a-similar-loader/

[47] F. de Haro Bermejo, “Detección de intrusiones con Snort” Tesis de Postgrado, Universitat Oberta de Catalunya, España, 2015.
[48] E. Rodríguez M., «Evaluación de herramientas para la generación de tráfico (tesis de maestría),» Universidad politécnica de Madrid, Madrid, 2015.