UNIVERSIDAD DE GUANAJUATO.

CAMPUS IRAPUATO – SALAMANCA

DIVISIÓN DE INGENIERÍAS

"SISTEMA DE DETECCION DE INTRUSOS CON TECNICAS DE MINERIA DE DATOS."

TESIS

QUE PARA OBTENER EL

RECONOCIMIENTO DE TRABAJO

LAUREADO

PRESENTA:

EMMANUEL GARCÍA ARIAS

DIRECTOR:

M.I. JOSÉ MANUEL MATADAMAS

HERNÁNDEZ

SALAMANCA, GTO 2024

Agradecimientos

En primer lugar, quiero expresar mi más sincero agradecimiento a mis grandes amigos, quienes han sido un pilar fundamental en este proceso. A **Brandon Marqués Salazar**, por su invaluable apoyo en la resolución de dudas de código y optimización. A **José Daniel Hernández Gutiérrez**, por los consejos invaluables sobre algoritmos y sus interminables pláticas multinacionales que enriquecieron el desarrollo de esta tesis. A **Isaac Mejía Flores**, **Martín Trejo Beltrán**, y **Georgia Gómez Tagle**, quienes me alentaron y apoyaron en los momentos de desesperación; su ejemplo y amistad han sido una fuente constante de inspiración. Aunque sé que somos amigos, realmente los considero como hermanos.

A mi familia, por su inquebrantable fe en mí y por estar siempre al pendiente durante este proceso. Agradezco profundamente su apoyo, que incluso incluyó momentos para despejar mi mente y ayudarme a seguir adelante.

Mi más sincero agradecimiento al **Doctor Daniel Alberto Razo Medina**, por ser un excelente Coordinador de Carrera y por su apoyo en el término de mi carrera.

A mis profesores, en especial al **Maestro José Manuel Matadamas Hernández**, por ser mi asesor de tesis y por dedicarse siempre a discutir temas relacionados con ciberseguridad. Sin su apoyo, esta tesis no hubiera sido posible.

También quiero agradecer al **Doctor Fernando Enrique Correa**, por aclarar mis dudas en diversas áreas, especialmente con su profundo conocimiento en sistemas Linux.

Mi gratitud también se extiende a la **Profesora Valentina Ugarte Ortiz**, cuyo reto en la materia de Minería de Datos fue el catalizador para la idea de esta tesis, y al **Maestro Edgar Iván Zepeda Arjona**, cuya orientación en la materia de Sistemas de Información me permitió estructurar el programa implementado, además de su gran amistad de ambos.

A la comunidad de **BeeDevelopers**, en particular a **Jashiel Robles Lecona**, **María Fernanda Paredes Amador**, y **Daniel Najera Flores**, por darme la oportunidad de compartir mis conocimientos en ciberseguridad con la comunidad estudiantil, lo que impulsó la idea de esta tesis.

A mis compañeros y amigos, **José Daniel Hernández Gutiérrez**, **Juan Antonio Hernández Aldaco**, y **Yennifer Olivares Rodríguez**, quienes me permitieron instruirlos en el ámbito de la seguridad informática. Aunque ya no tenga contacto con algunos de ustedes, espero que esos días de servicio social hayan sido útiles, y me han inspirado el amor por enseñar y compartir mi conocimiento.

A los **Doctores Roberto Rojas Laguna**, **Julián Moisés Estudillo Ayala**, **Juan Carlos Hernández García**, **Juan Manuel Sierra Hernández**, y al grupo organizado IEEE Photonics UG Student Chapter por la oportunidad de desarrollarme en el área de óptica durante el XXVIII Verano de la Ciencia en el proyecto "Desarrollo de software didáctico para estudiar aplicaciones de ingeniería con señales ópticas de amplio espectro" y de antemano agradecer a la institución CONAHCYT por permitirme desarrollarme en el Proyecto CONAHCYT No A1-S-33363 CB/2017-2018. ¡Muchas gracias por su apoyo y por iniciar mi camino en la investigación!

A la **Licenciada Ana María Robles Razo**, por siempre alentarme a seguir mis sueños, incluso cuando parecían difíciles, y por dejarme la frase que solo ella puede completar: "Se aclimata o se acli...". ¡Gracias por ser mi madre cuando más necesité mi genética maternal!

Finalmente, y de manera muy especial, a **Jenaro García Díaz**, por haberme dado la vida y por ser la persona que me ha acompañado en este camino. Tu genética ha sido la mejor herencia y, sin ti, no sería quien soy hoy. Quiero que sepas que no solo este logro, sino todos los futuros, también serán tuyos. Siempre llevaré tu frase "Siempre hay que mejorar a la Raza" en mi corazón. ¡Gracias por ser mi padre, por tu gran apoyo, tu tiempo y tu vida a mi lado! Siempre te presumiré con tus errores y aciertos.

Índice

Tabla de ilustraciones

Abstract

Currently, most existing intrusion detection systems (IDS) are based on virus signatures registered in CVE[1], which presents a high probability of failure in specific network situations, such as new vulnerabilities not recorded in these CVEs. Therefore, the proposed IDS studies the network to find anomalies and evaluate them against the parameters detected and compared in each network record, thus resolving this conflict.

The proposed Intrusion Detection System is divided into two main functional sections:

IDS Zero-Day Algorithm, which aims to analyze network traffic for computer viruses, ensuring that the analyzed packets have not been improperly manipulated by internal personnel or external agents known as "hackers". This will result in a database that will be continuously updated as the IDS base algorithm is used, called "Zero Day".

IDS Base Algorithm, responsible for analyzing network traffic and comparing it with the database obtained in the IDS zero-day algorithm. It should be noted that this is a brief interpretation of these algorithms, which will be addressed in detail later.

When employing this program in a company, various files have been obtained as a result, providing detailed information about the network. At the time, anomaly-based intrusion detection alerts have been presented as a result of analyzing said network, thus fulfilling the objective.

[1] **CVE**: Common Vulnerabilities and Exposures (CVE) is a publicly available list of information security vulnerabilities and exposures.

Resumen

En la actualidad la mayoría de los detectores de intrusos actuales (IDS) son basados en firmas de virus registrados en CVE[2] por lo que presentan una gran probabilidad de fallo ante situaciones específicas de cada red, como lo son nuevas vulnerabilidades no registradas en dichos CVE´s, por lo que el IDS propuesto estudia la red para encontrar anomalías y evaluarlas entre los parámetros detectados y comparados en cada registro de la red resolviendo dicho conflicto.

El Sistema de Detección de Intrusos propuesto se divide en dos secciones funcionales principales:

Algoritmo de día cero del IDS, que tiene como objetivo analizar el tráfico de red en busca de virus informáticos, asegurando que los paquetes analizados no hayan sido manipulados indebidamente por personal interno o agentes externos conocidos como "hackers". De esta manera, se obtendrá una base de datos que se actualizará continuamente a medida que el algoritmo base del IDS se utilice, llamado “Dia Cero”.

Algoritmo base del IDS, encargado de analizar el tráfico de red y compararlo con la base de datos obtenida en el algoritmo de día cero del IDS. Cabe destacar que esta es una breve interpretación de estos algoritmos, los cuales se abordarán en detalle más adelante.

Al emplear dicho programa en una empresa se ha logrado obtener como resultado diversos archivos que proporcionan información detallada de la red y en su momento se ha presentado las alertas de detección de intrusos basado en anomalías como resultado al analizar ducha red así de esta manera cumpliendo con el objetivo.

[2] CVE: Los puntos vulnerables y las exposiciones comunes (CVE) conforman una lista de las fallas de seguridad informática que está disponible al público.

Introducción

En la era digital actual, donde las redes de computadoras y los sistemas informáticos son fundamentales para el funcionamiento de empresas, instituciones y hogares, la seguridad cibernética se ha convertido en una prioridad crítica. Los ataques cibernéticos, estos representan una amenaza constante que puede causar daños significativos, desde la pérdida de información valiosa hasta interrupciones en las operaciones y pérdidas financieras.

Uno de los enfoques más efectivos para proteger las redes y sistemas informáticos es la implementación de Sistemas de Detección de Intrusos (IDS, por sus siglas en inglés). Estos sistemas tienen como objetivo identificar y alertar sobre actividades sospechosas o amenazas en tiempo real, permitiendo una respuesta rápida y efectiva para mitigar los riesgos.

Sin embargo, el desarrollo de un IDS eficiente y adaptable a entornos de pequeña y mediana escala plantea desafíos significativos. Estos sistemas deben ser capaces de analizar grandes volúmenes de tráfico de red, reconocer patrones complejos y diferenciar entre actividades legítimas y maliciosas, todo ello de manera precisa y en tiempo real.

Esta tesis se enfoca en abordar estos desafíos mediante la implementación de técnicas innovadoras de minería de datos. El objetivo principal es demostrar la viabilidad de desarrollar un Sistema de Detección de Intrusos aplicable a redes computacionales de pequeña y mediana escala.

Antecedentes

Los sistemas de detección de intrusos (IDS) son herramientas diseñadas para monitorear el tráfico de red y detectar patrones sospechosos que puedan indicar intentos de intrusión o violaciones de seguridad (Zuech, 2015).

Tradicionalmente, los IDS se han basado en enfoques basados en firmas o en anomalías. Los enfoques basados en firmas utilizan patrones predefinidos para detectar ataques conocidos, mientras que los enfoques basados en anomalías analizan el comportamiento del tráfico de red para identificar desviaciones de lo normal (Sharma, 2012). Sin embargo, ambos enfoques presentan limitaciones, como la incapacidad de detectar ataques desconocidos o la alta tasa de falsos positivos (Sultana, 2021).

En este contexto, las técnicas de minería de datos han ganado interés como una alternativa prometedora para mejorar la detección de intrusos (Tsai, 2009). La minería de datos permite analizar grandes cantidades de datos de tráfico de red y descubrir patrones complejos que pueden indicar actividades maliciosas (Sharma, 2012). Algunas técnicas de minería de datos utilizadas en este ámbito incluyen árboles de decisión, redes neuronales, máquinas de vectores de soporte, reglas de asociación y clustering (Sultana, 2021).

Varios estudios han demostrado el potencial de los enfoques basados en minería de datos para mejorar la precisión y la eficiencia de los sistemas IDS (Bhuyan, 2014), (Tsai, 2009), (Zuech, 2015). Sin embargo, aún existen desafíos y áreas de oportunidad, como la necesidad de desarrollar enfoques híbridos que combinen diferentes técnicas de minería de datos, la optimización de los algoritmos para el procesamiento en tiempo real y la adaptación a nuevas amenazas y entornos de red en constante evolución (Sultana, 2021).

Objetivos

Desarrollar un Sistema de Detección de Intrusos (IDS) aplicable a redes computacionales de pequeña y mediana escala. El objetivo principal es explorar la viabilidad de esta solución innovadora para la identificación de amenazas en la red.

El enfoque propuesto consiste en diseñar, crear y probar un código en Python capaz de detectar intrusos en una red utilizando técnicas de minería de datos. Este sistema se basará en los siguientes componentes clave:

- Detección de tráfico de red: El sistema monitoreará y analizará el tráfico de la red para identificar posibles actividades maliciosas.
- Registro de datos descriptivos del tráfico de red: Se recopilarán y almacenarán datos relevantes sobre el tráfico de red, como direcciones IP, puertos, protocolos, entre otros.
- Tratamiento de datos para la detección de anomalías intrusivas: Mediante técnicas de minería de datos, se procesarán los datos registrados para identificar patrones anómalos que puedan indicar la presencia de intrusos.
- Alertas de anomalías junto a los datos que se están transfiriendo: Cuando se detecten anomalías intrusivas, el sistema generará alertas y proporcionará información detallada sobre los datos que se están transfiriendo en ese momento.
- Cifrado de archivos para mayor protección dentro y fuera de la red: Con el fin de mejorar la seguridad, el sistema implementará mecanismos de cifrado de archivos para proteger la información confidencial tanto dentro de la red como durante la transferencia de datos.

Al combinar estas características, el Sistema de Detección de Intrusos propuesto ofrecerá una solución efectiva para la protección de redes computacionales de pequeña y mediana escala, aprovechando las capacidades de Python y las técnicas de minería de datos.

Justificación

Los sistemas de detección de intrusos (IDS) desempeñan un papel crucial en la seguridad informática, al monitorear el tráfico de red y detectar actividades maliciosas. Sin embargo, los enfoques tradicionales basados en firmas y anomalías presentan limitaciones, como la incapacidad de detectar amenazas desconocidas o la alta tasa de falsos positivos (Sultana, 2021) (Zuech, 2015).

Las técnicas de minería de datos han demostrado su potencial para mejorar la precisión y eficiencia de los sistemas IDS (Bhuyan, 2014) (Tsai, 2009). Al analizar grandes cantidades de datos de tráfico de red y descubrir patrones complejos, estas técnicas pueden identificar actividades maliciosas de manera más efectiva (Sharma, 2012).

IDS CON MINERIA DE DATOS

Este IDS combinará múltiples técnicas de minería de datos para aprovechar las fortalezas de cada una y compensar sus debilidades individuales. Además, se explorará la implementación de algoritmos híbridos que integren enfoques basados en anomalías con las técnicas de minería de datos, lo que podría mejorar aún más la precisión y la capacidad de detección.

El uso de lenguajes de programación modernos como Python facilitará el desarrollo y la implementación del sistema IDS propuesto, gracias a su sintaxis clara, bibliotecas robustas para minería de datos (Pandas) y una amplia comunidad de desarrolladores. Esto permitirá una codificación eficiente, una fácil integración de diferentes técnicas y una rápida iteración para mejorar continuamente el sistema.

Además, se explorarán estrategias para optimizar el rendimiento del sistema IDS, como el procesamiento en tiempo real, la escalabilidad y la adaptabilidad a nuevas amenazas y entornos de red cambiantes. Esto garantizará que el sistema sea capaz de responder de manera efectiva a los desafíos de seguridad en constante evolución.

La validación del enfoque propuesto se llevará a cabo mediante pruebas exhaustivas utilizando conjuntos de datos de tráfico de red ampliamente reconocidos. Se evaluará el rendimiento del sistema IDS en términos de métricas clave, como la tasa de detección, la tasa de falsos positivos y el tiempo de respuesta, y se comparará con otros enfoques existentes.

Capítulo 1: Redes Computacionales

En el mundo interconectado actual, las redes de computadoras desempeñan un papel fundamental en la comunicación y el intercambio de información. Estas infraestructuras permiten la transferencia de datos entre dispositivos a través de diversos medios y protocolos, facilitando la colaboración y el acceso a recursos compartidos.

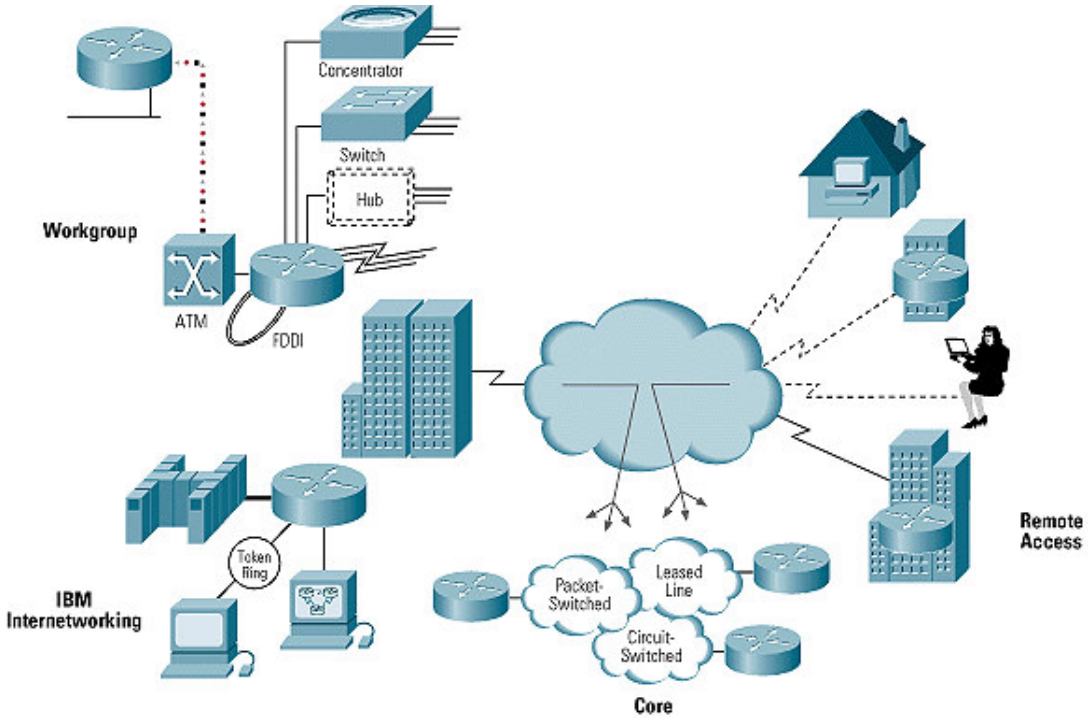

Ilustración 1 Cisco Internetwork Operating System (IOS)[3]

Modelo OSI (Forouzan, 2002)

El modelo OSI define una arquitectura de siete capas que describe las funciones necesarias para la comunicación efectiva en una red. Cada capa tiene responsabilidades específicas y se comunica con las capas adyacentes mediante

[3] "https://www.cisco.com/c/es_mx/support/docs/ios-nx-os-software/ios-software-releases-110/13178-15.html"

interfaces bien definidas. A continuación, se describen las siete capas del modelo OSI:

- Capa física: Es la capa más baja del modelo y se encarga de la transmisión de bits a través de un medio físico. Define las especificaciones eléctricas, mecánicas y funcionales para activar, mantener y desactivar la conexión física.
- Capa de enlace de datos: Proporciona transferencia de datos confiable a través del enlace físico. Se encarga del direccionamiento físico, detección y corrección de errores, control de flujo y acceso al medio.
- Capa de red: Determina la ruta a través de la red y el direccionamiento lógico. Es responsable del enrutamiento y el intercambio de datos entre diferentes redes.
- Capa de transporte: Segmenta los datos en unidades más pequeñas para su transmisión, reconoce y recupera datos perdidos, y controla el flujo de datos para evitar la saturación del receptor.
- Capa de sesión: Establece, mantiene y sincroniza las sesiones de comunicación entre aplicaciones. Gestiona el diálogo entre dispositivos y la recuperación de fallas.
- Capa de presentación: Define la sintaxis de transferencia de datos y formatos de representación. Se encarga de la traducción, cifrado y compresión de datos.
- Capa de aplicación: Proporciona servicios de red para las aplicaciones y define los protocolos de comunicación de alto nivel. Ejemplos de protocolos de esta capa son HTTP, FTP, SMTP, entre otros.

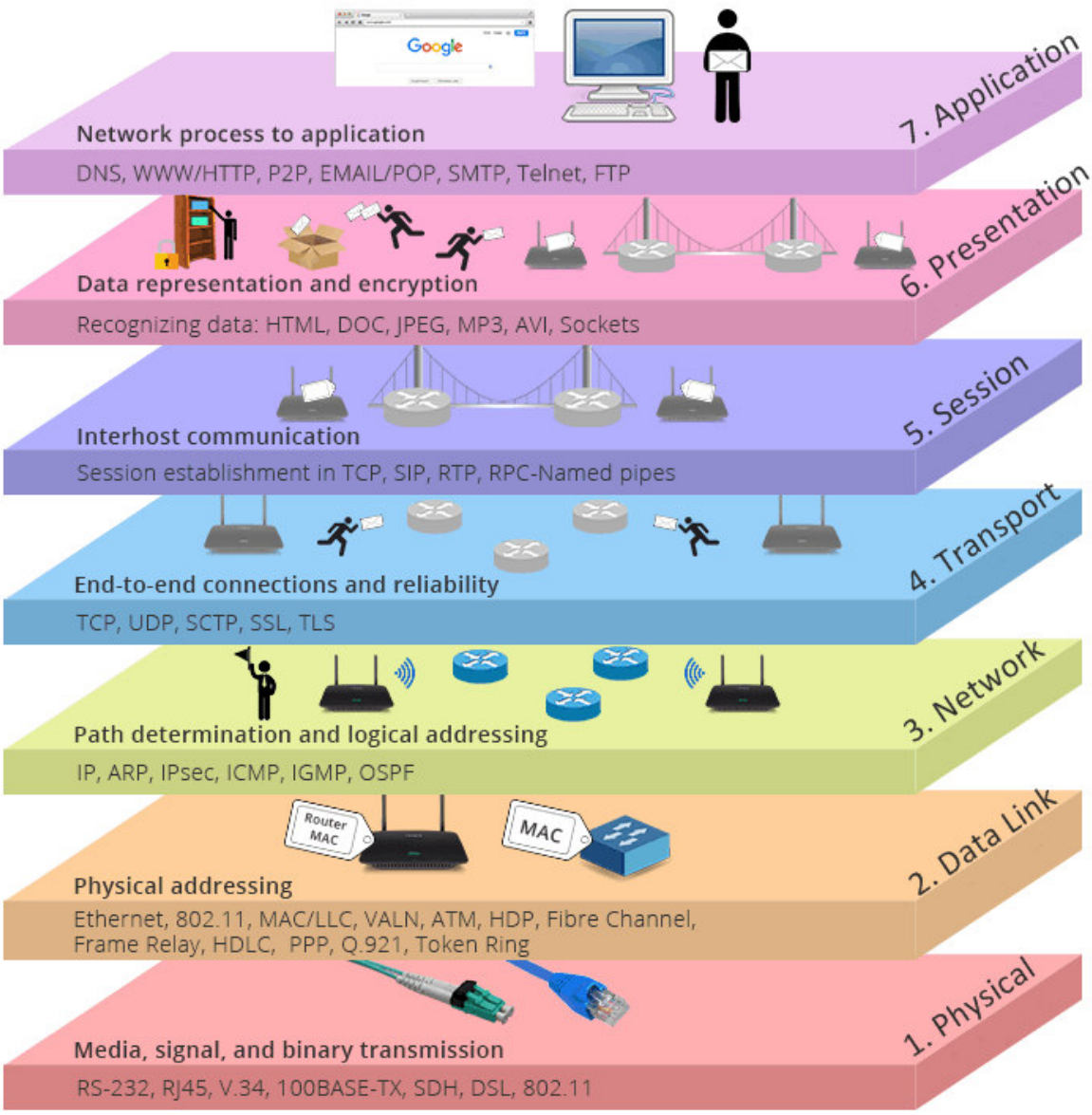

Ilustración 2 Modelo OSI[4]

Red

Los "nodos" son los puntos de conexión de la red, y para conectarse entre sí se necesitan medios de comunicación, que permitan la transmisión de datos entre los

[4] "https://community.cisco.com/t5/blogs-general/protocolos-de-red-b%C3%A1sicos-en-la-comprensi%C3%B3n-del-modelo-osi/ba-p/4810310"

nodos. La comunicación en una red puede realizarse a través de diversos medios, como cables, fibras ópticas o conexiones inalámbricas.

Las redes informáticas pueden variar en tamaño y complejidad desde las redes de área local (LAN) que conectan dispositivos cercanos a las redes globales. El objetivo principal de la red es facilitar la comunicación eficiente y el intercambio de recursos entre nodos. (Wetherall, 2011)

Funcionamiento de una red

El funcionamiento de una red generalmente sigue ciertos principios básicos. Estas redes están diseñadas para conectar dispositivos finales, como computadoras, teléfonos, cámaras, etc., a la red central.

Aspectos generales del funcionamiento de una red de acceso:

Conexión de Dispositivos Finales:

- Los dispositivos finales, también llamados hosts, se conectan a la red de acceso. Esto puede hacerse mediante conexiones por cable (como Ethernet) o de forma inalámbrica (Wi-Fi).

Infraestructura de Red:

- La red de acceso incluye la infraestructura necesaria para permitir la comunicación entre los dispositivos finales y, eventualmente, hacia la red central. Esto puede involucrar dispositivos como switches, routers y puntos de acceso inalámbrico.

Direcciones IP y Configuración:

- Cada dispositivo en la red de acceso suele tener una dirección IP única que le permite ser identificado y comunicarse con otros dispositivos. La

configuración de direcciones IP y otros parámetros de red es esencial para el funcionamiento correcto de la red.

Switching y Enrutamiento:

- En el caso de redes con cable, los switches gestionan la conectividad entre los dispositivos finales dentro de una red local (LAN). Los routers pueden ser necesarios para dirigir el tráfico entre diferentes redes, como la red de acceso y la red central.

Redes Inalámbricas:

- En el caso de redes inalámbricas, los puntos de acceso (Access Points) desempeñan un papel crucial. Estos dispositivos permiten la conectividad inalámbrica y gestionan la comunicación entre los dispositivos inalámbricos y la infraestructura de red.

Gestión de Tráfico:

- Los dispositivos en la red de acceso deben gestionar eficientemente el tráfico para garantizar un rendimiento óptimo. Esto implica tomar decisiones sobre cómo dirigir los datos entre los dispositivos y asegurarse de que los recursos de red se utilicen de manera eficiente.

Seguridad:

- La seguridad es un aspecto crítico en la red de acceso. Se implementan medidas como firewalls, autenticación y cifrado para proteger la integridad de los datos y evitar accesos no autorizados.

Acceso a la Red Central:

- La red de acceso se conecta a la red central, que puede ser una WAN o una red más extensa. Esta conexión se realiza mediante dispositivos como

routers, y puede implicar la transmisión de datos a través de diferentes tecnologías, como líneas de fibra óptica, cables coaxiales o enlaces inalámbricos.

(Róbert Szabó, 2010)

Capítulo 2: Sistema de detección de intrusos (IDS) y Sistema de prevención de intrusos (IPS)

En el entorno actual de crecientes amenazas cibernéticas, el uso de IDS e IPS se ha vuelto fundamental para proteger sistemas y redes. Estas herramientas desempeñan un papel crucial en la detección temprana de actividades maliciosas y la mitigación eficaz de ataques. Los IDS monitorizan el tráfico y generan alertas ante comportamientos sospechosos, mientras que los IPS no solo detectan amenazas, sino que también las bloquean en tiempo real. Su implementación es especialmente importante en sectores que manejan información crítica o sensible, como finanzas, salud y gobierno. Con el aumento de ataques sofisticados, contar con IDS e IPS robustos se ha convertido en un requisito indispensable para mantener una postura de seguridad sólida y proteger los activos digitales de cualquier organización.

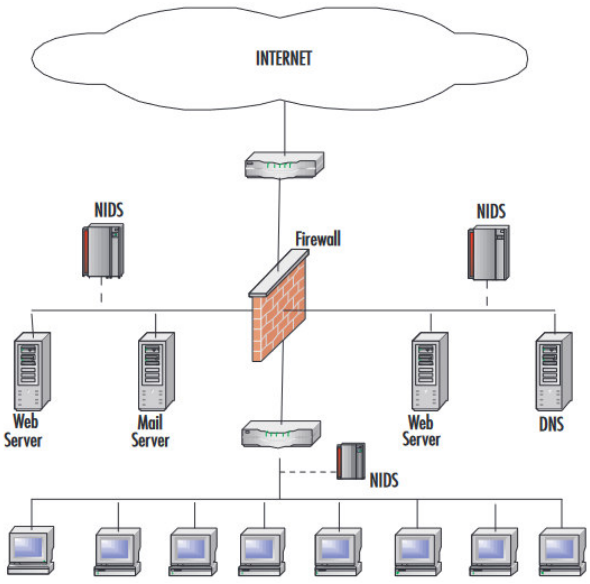

Ilustración 3 "NIDS Network" [5]

[5] *pag. 12 Snort 2.1 Intrusion Detection, Second Edition. by Brian Caswell, Jay Beale. Released June 2004. Publisher(s): Syngress. ISBN: 9780080480992.*

IDS

Un IDS se define como un sistema de seguridad informática diseñado para detectar actividades no autorizadas o malintencionadas en una red y/o sistema informático.

Un IDS monitorea y analiza el tráfico de red, eventos de sistema, registros y otros aspectos relevantes del entorno de tecnología de la información en busca de signos de intrusiones o actividades sospechosas. Cuando se detecta una anomalía o un comportamiento potencialmente malicioso, el IDS emite alertas para notificar a los administradores de seguridad, permitiéndoles tomar medidas correctivas. (Bace, 2000)

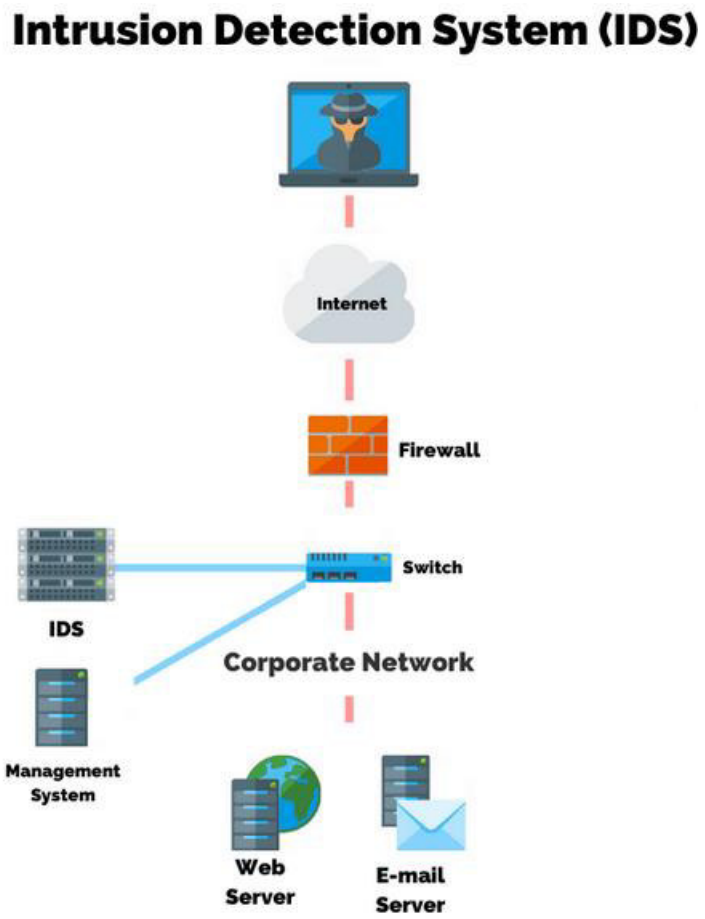

Ilustración 4 Estructura de una red una vez colocado un IDS [6]

[6] "https://purplesec.us/intrusion-detection-vs-intrusion-prevention-systems/"

Tipos de IDS

Sistemas de Detección de Intrusos de Red (NIDS):

- Estos IDS se centran en monitorear el tráfico de red. Examinan los paquetes de datos que fluyen a través de la red y buscan patrones que puedan indicar actividades maliciosas o no autorizadas. Los NIDS son efectivos para detectar ataques en tiempo real y para analizar el tráfico en busca de comportamientos anómalos.

Sistemas de Detección de Intrusos de Host (HIDS):

- Los HIDS se implementan a nivel de un sistema individual (host) y supervisan las actividades en ese host específico. Están diseñados para detectar cambios en archivos del sistema, intentos de acceso no autorizado, comportamientos anómalos del usuario y otras actividades sospechosas a nivel de host. Los HIDS son particularmente útiles para identificar intrusiones que pueden no ser evidentes a nivel de red.

(Bace, 2000)

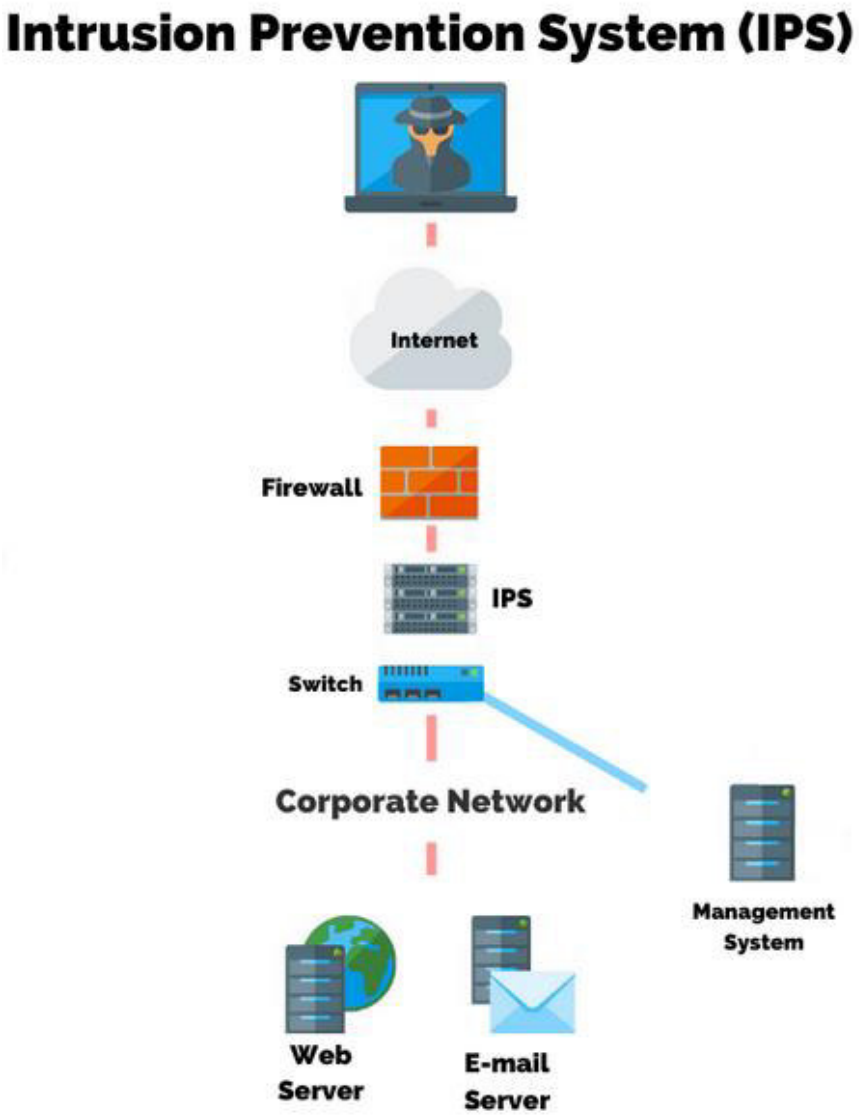

Ilustración 5 Estructura de una red una vez colocado un IPS [7]

Un IPS (Sistema de Prevención de Intrusiones) es un tipo de sistema de seguridad que tiene como objetivo detectar, prevenir y responder a actividades maliciosas o no autorizadas en una red o sistema informático. (Rash, 2005)

[7] "https://purplesec.us/intrusion-detection-vs-intrusion-prevention-systems/"

Diferencias entre IDS e IPS

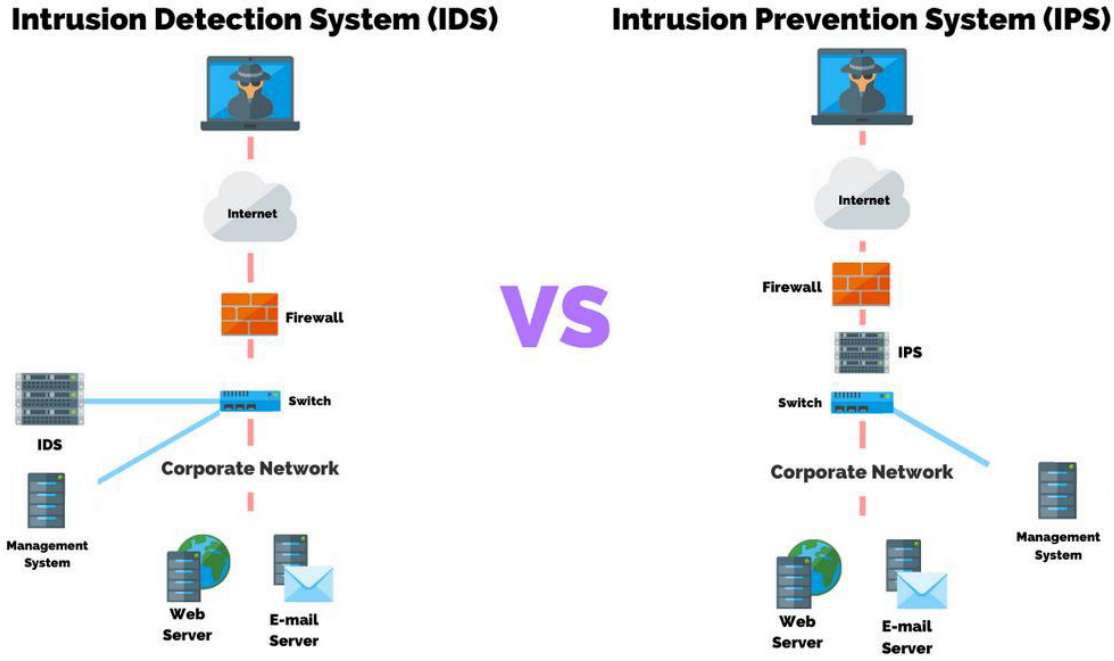

Ilustración 6 Comparación de estructuras de un IDS y un IPS [8]

Sistema de Detección de Intrusos (IDS):

- El objetivo principal es supervisar y analizar el tráfico en busca de comportamientos sospechosos o actividades maliciosas. Identifica y advierte sobre posibles amenazas, pero no actúa directamente para bloquear o prevenir dichas amenazas. Suministra información valiosa a los administradores de seguridad, quienes pueden tomar medidas manuales basadas en las alertas generadas.

Sistema de Prevención de Intrusos (IPS):

- Ofrece una funcionalidad más avanzada que la simple detección al permitir la implementación de medidas activas para prevenir o bloquear intrusiones. Puede aplicar reglas de filtrado, bloquear direcciones IP, cerrar conexiones u otras acciones automatizadas para detener amenazas en tiempo real.

[8] "https://purplesec.us/intrusion-detection-vs-intrusion-prevention-systems/"

Proporciona una capa extra de seguridad al ofrecer una respuesta activa y automática a eventos de seguridad.

(Anderson, 1 Abril 2008)

Capítulo 3: Virus informático de host

Los virus de host representan un riesgo tangible para la seguridad de los sistemas informáticos individuales y, por lo tanto, deben abordarse con medidas de protección adecuadas, como el uso de software antivirus actualizado, la aplicación de parches de seguridad y la concientización de los usuarios sobre las mejores prácticas de seguridad.

Virus informático

Un virus informático es un programa malicioso que infecta archivos ejecutables y tiene la habilidad de replicarse y propagarse entre diferentes programas y sistemas. Los virus modifican y replican su código cuando el archivo o programa huésped es ejecutado

Las principales características de los virus informáticos:

- Auto-replicación: el virus se duplica a sí mismo e infecta otros programas cuando se ejecuta.
- Infectividad: inserta su código malicioso dentro de archivos ejecutables sanos.
- Activación al ejecutar el programa infectado: permanecen inactivos hasta que se ejecuta el programa o archivo infectado por el virus.
- Efectos dañinos: destruyen, corrompen o roban datos y recursos del sistema.
- Ocultación: algunos utilizan técnicas para evadir la detección por software antivirus.

(Cohen, 1984)

Malware de host

El malware de host es un tipo de software malicioso diseñado para infectar y comprometer la seguridad de un sistema informático específico. Reside y se ejecuta localmente en el dispositivo host que infecta.

Las características principales del malware de host son:

- Infecta directamente el sistema operativo del dispositivo a nivel local.
- No requiere propagación a través de una red, aunque algunas variantes pueden comunicarse con servidores remotos de comando y control (C2).
- Puede obtener control total del sistema, evadir software antivirus y permanecer persistente en el dispositivo.
- Ejemplos comunes: virus, troyanos, gusanos, spyware, adware, ransomware, rootkits, keyloggers, bots, etc.

(Bahl, 2021)

Riesgos de tener un malware

Algunos de los principales riesgos que conlleva tener un malware de host en un dispositivo:

- Robo de información personal y datos confidenciales almacenados en el dispositivo infectado, incluyendo contraseñas, historial de navegación, información bancaria, etc.
- Monitorización de la actividad del usuario sin su consentimiento, como pulsaciones de teclas, sitios web visitados, archivos accedidos, etc.
- Uso de los recursos del dispositivo para actividades maliciosas como ataques de denegación de servicio, envío de spam, minería de criptomonedas, etc.
- Daños y alteraciones del sistema operativo, archivos y configuraciones del dispositivo debido a modificaciones realizadas por el malware.

- Instalación de puertas traseras para permitir acceso remoto no autorizado al dispositivo infectado.
- Imposibilidad de usar el dispositivo infectado de manera normal debido al comportamiento malicioso del malware.

(Dash Bahl, 2021)

Como detectar un Malware

Algunas formas de detectar malware de host en un:

- Emplear técnicas de análisis estático y dinámico de malware para estudiar y clasificar muestras sospechosas. El análisis estático estudia el código sin ejecutarlo, el dinámico observa su comportamiento en tiempo real.
- Utilizar honeypots[9] para atraer y detectar malware analizando cualquier tráfico entrante no autorizado hacia estos señuelos.
- Monitorear llamadas a API y funcionamiento del sistema mediante hooks[10] para identificar comportamientos anómalos indicativos de malware.
- Emplear machine learning entrenando modelos con conjuntos de datos etiquetados para detección automática de amenazas.
- Analizar la reputación de dominios, IPS y hashes de archivos consultando bases de datos de amenazas conocidas.

(Chen, 2021)

[9] Honeypot: Es un sistema informático que se "sacrifica" para atraer ciberataques, como un señuelo. Simula ser un objetivo para los hackers y utiliza sus intentos de intrusión para obtener información sobre los cibercriminales y la forma en que operan o para distraerlos de otros objetivos.
[10] Hooks: El código que maneja tales llamadas de función, eventos o mensajes interceptados.

Capítulo 4: Python

Ilustración 7 Logotipo página oficial[11]

Python se ha convertido en un lenguaje de programación fundamental en el campo de la ciberseguridad, gracias a sus numerosas librerías y su capacidad para automatizar tareas, analizar datos, manipular paquetes de red y crear herramientas de seguridad personalizadas. Su versatilidad, facilidad de uso y una comunidad activa lo convierten en una elección popular para profesionales y entusiastas de la ciberseguridad.

Python como lenguaje de programación

Python es un lenguaje de programación poderoso y fácil de aprender que se utiliza para desarrollar aplicaciones de todo tipo, desde videojuegos hasta interfaces de usuario y software científico. Tiene una sintaxis simple y elegante que hace énfasis en la legibilidad y reduce la complejidad del código. (Sweigart, 2019)

Librerías en Python

De acuerdo con la documentación oficial de Python y libros sobre este lenguaje, una librería en Python es un conjunto de código reutilizable, compuesto por módulos y funciones, que permite extender las capacidades de Python sin tener que reescribir el código desde cero.

Las librerías son paquetes o módulos de Python que contienen funciones y métodos ya implementados para realizar tareas específicas. Al importar una librería en un programa Python se obtiene acceso a ese código reutilizable.

Algunas características de las librerías en Python:

[11] "https://www.python.org/"

- Permiten abstraer y encapsular código complejo en funciones simples de usar.
- Son compartidas y utilizadas por la comunidad de Python.
- Ahorran tiempo al evitar tener que reescribir código para problemas comunes.
- Extienden las capacidades de Python para necesidades específicas.
- Se instalan fácilmente usando pip o el gestor de paquetes de Python.

(Sweigart, 2019)

Scapy

Ilustración 8 Logotipo de la librería Scapy [12]

Scapy es una librería avanzada de Python que permite manipular paquetes en la red a nivel de bits para generar y diseccionar tráfico personalizado. Puede forjar o decodear una amplia variedad de protocolos, descubrir hosts, escanear puertos, fingerprinting de red, construir herramientas de sniffing y llevar a cabo inyección de paquetes y ataques MITM.

Algunas características clave de Scapy:

- Permite construir paquetes a mano especificando valores de headers TCP/IP.
- Soporta gran cantidad de protocolos y decodificación de paquetes.

[12] "https://thepythoncode.com/article/getting-started-with-scapy"

- Capacidades avanzadas de sniffing y manipulación de tráfico en tiempo real.
- Interfaz para realizar escaneo de puertos, traceroute, ping sweep y otros tests.
- Utilizado ampliamente en seguridad informática y ethical hacking con Python.

(Seitz, 2021)

Pyhsark

Pyshark es un wrapper de Python para la biblioteca tshark de Wireshark que permite decodificar paquetes de red capturados en objetos Python para su posterior análisis e investigación. Proporciona capacidades para sniffing en vivo, lectura de capturas PCAP, extracción y filtrado de paquetes según criterios específicos.

Ilustración 9 Logotipo de la librería pyshark[13]

Algunas características clave de Pyshark:

- Acceso a los cientos de decodificadores y filtros de Wireshark.
- Lectura de capturas PCAP y extracción de información de paquetes.
- Sniffing en vivo de datos a medida que fluyen por la red.
- Filtrado de paquetes por dirección, protocolo, puerto u otros criterios.
- Extracción de campos, valores y metadatos de paquetes capturados.
- Integración con Python para automatizar tareas de análisis de tráfico.

(Khrais, 2019)

[13] "https://pypi.org/project/pyshark/"

Scapy vs Pyshark

Las principales diferencias son:

- Scapy permite construir paquetes desde cero, mientras que Pyshark analiza paquetes previamente capturados.
- Scapy funciona a bajo nivel, creando y diseccionando paquetes a nivel de bytes. Pyshark utiliza Wireshark para análisis a alto nivel.
- Scapy es mejor para manipulación de tráfico y ataques. Pyshark es más adecuado para análisis pasivo de tráfico.
- Scapy tiene mayor flexibilidad y control total sobre cada campo de los paquetes. Pyshark depende de las capacidades de Wireshark.
- Scapy requiere mayor conocimiento técnico para construir y decodificar paquetes manualmente. Pyshark es más sencillo de usar.
- Scapy posee módulos para algunos protocolos específicos. Pyshark soporta cientos de protocolos gracias a Wireshark.

(Kathiravelu, 2021)

En pocas palabras, Scapy es más eficiente y de bajo nivel, mientras que Pyshark ofrece una API sencilla para aprovechar las capacidades de análisis de Wireshark desde Python.

OS library

El módulo OS de Python proporciona docenas de funciones para listar directorios, conocer y cambiar el directorio actual, obtener información del sistema operativo, administrar procesos, renombrar, crear y eliminar archivos y directorios, etc. Incluye funciones portátiles entre SOs (del plural sistemas operativos) como os.listdir(), os.getcwd() y otras específicas para Windows, Linux, macOS. (Ramalho, 2015)

Por lo tanto, podemos concluir que os permite acceder a funcionalidades del sistema operativo para administrar archivos, procesos, rutas y ejecutar comandos de una manera portable.

Subproces library

La librería subprocess de Python permite ejecutar procesos externos desde un programa Python y conectarse a sus flujos de entrada/salida. Ofrece mayor control y flexibilidad que simplemente usar os.system().

Subprocess permite:

- Lanzar nuevos procesos y conectarse a sus pipes stdin, stdout y stderr.
- Obtener la salida de un programa externo y trabajar con ella en Python.
- Sustituirá shells de forma más segura, controlada y portable que os.system().
- Verificar el código de retorno de un proceso para determinar si terminó correctamente.

Algunos usos comunes de subprocess:

- Ejecutar comandos de Linux/Unix y trabajar con su salida.
- Lanzar scripts Python externos como subprocess y manejar su input/output.
- Automatizar y administrar procesos de forma portable entre sistemas operativos.
- Realizar tareas de administración del sistema y mantenimiento.

(Jones, 2013)

Pyqt5

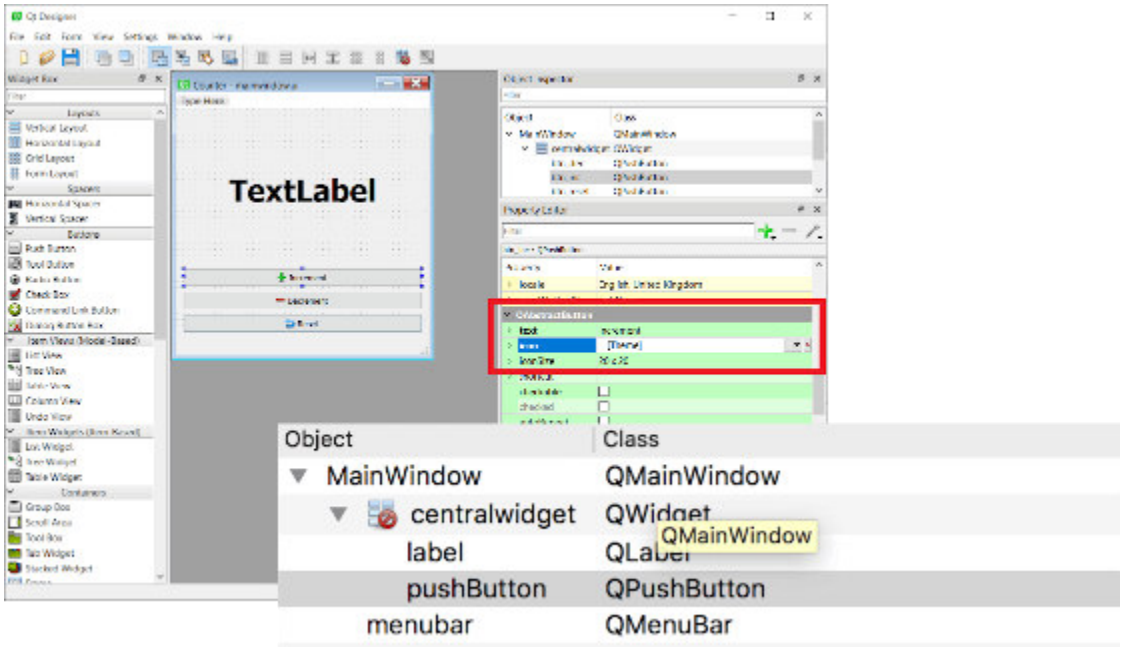

Ilustración 10 Interfaz de Pyqt5 para desarrollo de GUIs[14]

PyQt5 es un conjunto de módulos que permiten crear aplicaciones gráficas de usuario (GUI) multiplataforma utilizando Python. Se basa en el toolkit Qt y provee una API para acceder a elementos gráficos como ventanas, botones, menús, etc.

PyQt5 permite:

- Crear interfaces gráficas complejas de forma rápida y sencilla.
- Acceder a widgets como texto, imágenes, tablas, sliders, botones, etc.
- Manejar eventos producidos por acciones del usuario.
- Incluir estilos personalizados CSS.
- Integrar matplotlib y otras librerías gráficas para plots.
- Desarrollar GUIs que funcionen en Windows, Mac y Linux.

(Fitzpatrick, 2022)

[14] "https://www.pythonguis.com/pyqt5-tutorial/"

Pandas

Pandas es una popular librería de Python open source para análisis y manipulación de datos. Provee estructuras de datos de alto rendimiento y herramientas de análisis de datos fáciles de usar.

Pandas permite:

- Estructuras de datos flexibles para almacenar y manipular tablas de datos con filas y columnas heterogéneas.
- Funciones integradas para análisis estadístico y limpieza de datos faltantes.
- Herramientas para cargar datos de CSV, Excel, bases de datos y formatos custom.
- Métodos para fusionar, combinar y remodelar conjuntos de datos.
- Una API similar a NumPy que funciona bien con otros paquetes científicos de Python.

(McKinney, 2018)

Python y la ciberseguridad

Ilustración 11 Representación de la combinación de Python y la ciberseguridad[15]

[15] "https://www.cec.es/python-en-ciberseguridad-cinco-motivos-para-aprender-este-lenguaje/"

Algunas razones para usar Python como herramienta de ciberseguridad, basadas en libros recientes:

- Python tiene una amplia cantidad de librerías dedicadas a tareas de ciberseguridad, como scapy, pwntools, nmap, entre otras. (Welsh, 2021)
- Permite automatizar tareas repetitivas en el análisis de malware, monitoreo de red, ethical hacking, etc., ahorrando tiempo. (MichaelTu, 2020)
- Es un lenguaje multiparadigma que permite programación procedimental, orientada a objetos y funcional para distintas necesidades. (Welsh, 2021)
- Cuenta con frameworks web como Django y Flask que pueden usarse para crear aplicaciones de seguridad y honeypots. (MichaelTu, 2020)
- Es multiplataforma y se puede usar en sistemas Linux, ideales para testing de penetración y análisis de malware. (Welsh, 2021)
- Existe gran cantidad de documentación y comunidad activa para aprender y resolver consultas sobre Python en seguridad informática. (MichaelTu, 2020)

Para finalizar se puede concluir el capítulo sabiendo que tenemos los conocimientos necesarios para entender la factibilidad del porque el uso de este lenguaje de programación para implementarlo en el desarrollo de la tesis.

Capítulo 5: Minería de datos

La minería de datos se ha convertido en una herramienta fundamental en la era digital actual. Con el creciente volumen de datos generados por diversas fuentes, la capacidad de extraer información valiosa y conocimientos relevantes a partir de estos datos masivos se ha vuelto esencial para las organizaciones.

Minería de datos

La minería de datos es el proceso de extraer patrones interesantes de grandes conjuntos de datos utilizando métodos de la estadística y el aprendizaje automático. (Ullman, 2020)

Técnicas de minería de datos

Aquí algunas de las técnicas más comunes de minería de datos:

- Clasificación: asigna elementos en categorías o clases previamente definidas. Algoritmos: regresión logística, árboles de decisión, Naive Bayes, redes neuronales.
- Clustering: agrupa elementos similares en clusters sin categorías predefinidas. Algoritmos: K-means, DBSCAN, agrupamiento jerárquico.
- Reglas de asociación: descubre relaciones interesantes entre variables. Se usa en market basket analysis.
- Reducción de dimensionalidad: selecciona las características más relevantes de los datos. Técnicas: análisis de componentes principales, selección de características.
- Detección de anomalías: identifica datos atípicos que se desvían significativamente del resto. Técnicas: análisis estadístico, aislamiento forestal.
- Aprendizaje por refuerzo: optimiza estrategias para maximizar una recompensa. Algoritmos: Q-learning, SARSA.

- Redes neuronales profundas: modelos predictivos basados en redes neuronales multicapa.

(Ullman, 2020)

Aplicación de la minería de datos

Algunos ejemplos de aplicaciones de la minería de datos:

- Detección de fraude: identificar transacciones fraudulentas analizando patrones históricos y datos del cliente.
- Recomendación de productos: sugerir artículos que podrían interesar al usuario según su comportamiento previo. Algoritmos como filtrado colaborativo.
- Análisis de sentimientos: determinar la actitud u opinión general sobre un tema analizando texto no estructurado.
- Puntuación de crédito: asignar una puntuación de solvencia crediticia a un cliente utilizando su historial financiero.
- Predecir abandono de clientes: identificar clientes con mayor probabilidad de cancelar un servicio mediante la predicción de churn.
- Personalización web: ofrecer contenidos adaptados al usuario individual analizando su comportamiento en un sitio web.
- Detección de intrusos: identificar actividades anormales en redes y sistemas que podrían indicar un ataque.
- Diagnóstico médico asistido: ayudar en el diagnóstico de enfermedades en base a síntomas y exámenes de pacientes.

(Ullman, 2020)

Pandas y la minería de datos

Pandas puede ser muy útil para la minería de datos ya que facilita el preprocesamiento y análisis exploratorio de los datos previo a aplicar los algoritmos de minería.

Pandas permite:

- Cargar rápidamente conjuntos de datos de diferentes fuentes como CSV, SQL, JSON y Excel.
- Limpiar, transformar y normalizar los datos para prepararlos para análisis.
- Realizar análisis descriptivo y generar estadísticas resumen de manera sencilla.
- Manipular y filtrar datos de forma simple mediante indexación y slicing.
- Combinar y agregar múltiples conjuntos de datos para enriquecerlos.
- Visualizar y obtener insights de los datos de forma rápida.

(VanderPlas, 2016)

Capítulo 6: Ciberseguridad - Hacking (técnicas y herramientas)

Mientras la ciberseguridad busca defender y proteger sistemas y datos, el hacking implica explotar vulnerabilidades[16], ya sea con fines maliciosos o para evaluar y fortalecer la seguridad. Herramientas como Nmap, Snort, Suricata y Fail2ban son ampliamente utilizadas por profesionales de ciberseguridad y hackers éticos para tareas de evaluación de seguridad, detección de amenazas y prevención de ataques. (Stavroulakis, 2020).

Ciberseguridad

La ciberseguridad se refiere a la protección de sistemas informáticos, redes y datos de ataques o accesos no autorizados. Su objetivo es garantizar la confidencialidad, integridad y disponibilidad de la información en el ciberespacio.

La ciberseguridad implica:

- Tecnologías, procesos y prácticas diseñadas para proteger redes, dispositivos, programas y datos de ataques o intrusiones.
- Medidas para detectar, documentar y contrarrestar dichas amenazas a través de soluciones técnicas y políticas de seguridad.
- Un campo interdisciplinario que abarca software, hardware, inteligencia, educación, conciencia y aspectos éticos y legales.
- Una responsabilidad compartida entre usuarios, profesionales de TI y diversas organizaciones.
- Acciones continuas para anticipar y adaptarse a un panorama de amenazas en constante evolución.
- (Stavroulakis, 2020)

[16] Vulnerabilidad: es una debilidad existente en un sistema que puede ser utilizada por una persona malintencionada para comprometer su seguridad.

El Cubo de MacCumber

El Cubo de MacCumber es un modelo que representa las tres principales características de la seguridad de la información: confidencialidad, integridad y disponibilidad.

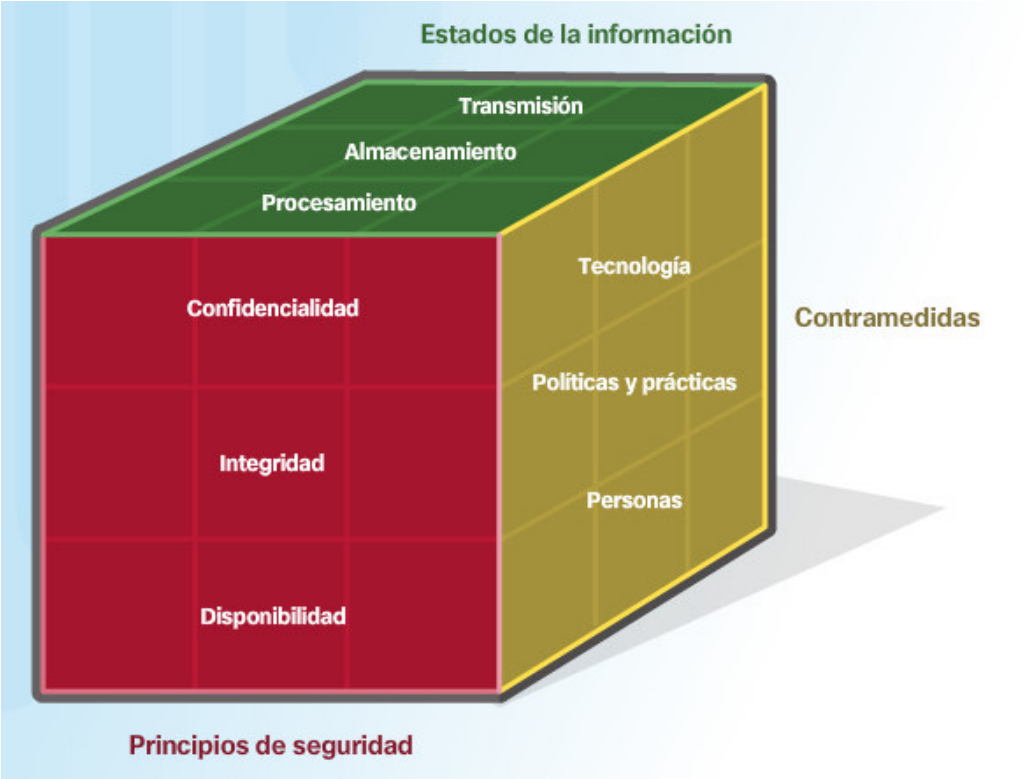

Ilustración 12 Cubo de McCumber[17]

Cada una de las tres características principales del Cubo de MacCumber (confidencialidad, integridad y disponibilidad) se puede dividir en subsecciones más específicas:

- Confidencialidad: a. Privacidad de datos: Protección contra la divulgación no autorizada de información. b. Privacidad de las comunicaciones: Protección contra la interceptación de comunicaciones. c. Anonimato: Ocultación de la identidad de los usuarios o entidades.

[17] "https://itsector300.blogspot.com/2018/05/ciberseguridad-john-mccumber.html"

- Integridad: a. Integridad de datos: Garantizar que los datos no sean modificados de manera no autorizada. b. Integridad de origen: Verificación de que el origen de los datos es genuino y no ha sido suplantado. c. Integridad del sistema: Asegurar que el sistema funciona correctamente y no ha sido comprometido.
- Disponibilidad: a. Disponibilidad de servicios: Asegurar que los servicios y recursos estén disponibles cuando se necesiten. b. Capacidad de recuperación: Capacidad de restaurar los servicios y datos en caso de fallas o interrupciones. c. Rendimiento: Garantizar un rendimiento adecuado y tiempos de respuesta aceptables.

Hacking

El hacking se puede definir como:

- El acto de modificar un sistema o dispositivo de una forma no prevista o no autorizada.
- Aprovechar una tecnología de maneras innovadoras para hacerla funcionar de forma diferente al diseño original.
- El proceso de encontrar debilidades o vulnerabilidades en sistemas informáticos o redes y explotarlas para obtener acceso no autorizado.
- Una actividad multifacética que abarca un amplio espectro de metodologías, motivaciones y objetivos.

Algunas características clave del hacking:

- Puede realizarse con fines maliciosos (black hat hacking) o benéficos (white hat hacking).
- Requiere un profundo conocimiento técnico de sistemas y redes informáticas.
- Implica el uso de herramientas y técnicas avanzadas para descubrir y explotar vulnerabilidades.
- Permite identificar problemas de seguridad antes de que sean explotados por atacantes reales.

- Es una práctica controversial y potencialmente ilegal dependiendo de la motivación y el método.

(Erickson, 2008)

Ciberseguirdad vs Hacking

La ciberseguridad busca defender y proteger activos de información de ataques y amenazas. El hacking implica explotar vulnerabilidades en sistemas informáticos, a menudo con fines malintencionados.

Algunas diferencias son:

- La ciberseguridad adopta una postura defensiva mientras que el hacking es ofensivo.
- La ciberseguridad es legal y ética, el hacking puede ser ilegal.
- El objetivo de la ciberseguridad es preservar la confidencialidad, integridad y disponibilidad de los datos. El hacking busca comprometer estas propiedades.
- La ciberseguridad emplea herramientas y conocimientos técnicos para proteger sistemas. El hacking utiliza habilidades similares para atacarlos.
- Los profesionales de ciberseguridad analizan vulnerabilidades y las corrigen. Los hackers explotan esas mismas debilidades.

(Easttom, 2021)

Ataque informático

Un ataque informático se puede definir como:

- Cualquier intento de comprometer la confidencialidad, integridad o disponibilidad de un sistema informático o red de computadoras.
- Un acto intencionado para tomar el control, causar daño o acceder sin autorización a recursos digitales ajenos.

- La explotación de vulnerabilidades en software, hardware, protocolos o prácticas de seguridad deficientes.
- El empleo de técnicas como malware, phishing, denegación de servicio, entre otras para lograr un objetivo malintencionado.
- Una violación de las políticas de seguridad establecidas mediante el uso de herramientas, scripts o metodologías intrusivas.
- Una actividad cuyo fin puede ser robo de datos, espionaje, fraude, vandalismo, interrupción de servicios u otros delitos informáticos.
- Un problema grave que causa pérdidas financieras y de reputación a individuos, empresas y gobiernos.

(Dileep Kumar G, 2021)

Tipos de ataques informáticos

Algunos de los tipos más comunes de ataques informáticos:

- Malware: Software dañino como virus, troyanos, spyware, ransomware.
- Phishing: Suplantación de identidad para engañar a las víctimas y robar datos confidenciales.
- Denegación de servicio (DoS): Sobrecarga de recursos para interrumpir el acceso a un servicio.
- Ataques de día cero: Explotan vulnerabilidades desconocidas anteriormente en software o hardware.
- Fuerza bruta: Método de prueba y error para descifrar contraseñas.
- Inyección SQL: Inserción de código malicioso en consultas SQL para acceder a bases de datos.
- Desbordamiento de buffer: Provoca fallos en la memoria de una aplicación para ejecutar código dañino.
- Suplantación de IP & MAC (Spoofing): Falsifica la dirección IP y/o MAC de origen para ocultar la identidad.
- Man in the Middle: Intercepta y manipula la comunicación entre dos sistemas.
- (Gupta, 2022)

Minería de datos en la ciberseguridad

Mediante técnicas como clasificación, clustering, detección de anomalías y aprendizaje por refuerzo, se aplica en ciberseguridad para tareas como detección de fraude, análisis de sentimientos, detección de intrusos y predicción de abandono de clientes, mientras que Pandas facilita el manejo de datos para estos fines. (Ullman, 2020)

Nmap como herramienta de ciberseguridad y hacking

Nmap es una poderosa herramienta de código abierto para exploración de redes y seguridad informática, muy utilizada por profesionales de la ciberseguridad y hackers éticos.

Nmap permite:

- Descubrir hosts activos en una red para mapearla.
- Detección de puertos abiertos y servicios disponibles en dispositivos.
- Detección de sistemas operativos y versiones de servicios remotos.
- Verificación de configuraciones de firewall y detección de reglas.
- Búsqueda de computadoras con puertos abiertos vulnerables.
- Escaneo de redes para detectar puntos de acceso no autorizados.
- Búsqueda de vulnerabilidades conocidas en servicios remotos.
- Generación de firmas de redes para monitoreo de cambios.

Nmap es una herramienta clave en evaluaciones de seguridad de redes, testing de penetración, hardening, investigación de incidentes, entre otros usos relacionados con hacking ético y ciberseguridad. (Fyodor, 2022)

Snort

Snort es una popular herramienta de código abierto para detección de intrusiones y monitoreo de tráfico de red, ampliamente utilizada para ciberseguridad y hacking ético.

Algunas capacidades clave de Snort son:

- Análisis en tiempo real de tráfico en red para detectar ataques y anomalías.
- Detección de exploits[18], escaneo de puertos, worms, virus y otros tipos de malware.
- Monitoreo flexible y personalizable basado en firmas y reglas de detección.
- Registro de paquetes y sesiones de red para su posterior análisis forense.
- Funciona como IDS (Sistema de Detección de Intrusos) en red o NIDS.
- Alto rendimiento y bajo uso de recursos para monitoreo de redes de cualquier tamaño.
- Capacidad de prevenir ataques en tiempo real configurándolo en modo IPS.
- Interfaz web para administrar y ver alertas en tiempo real.
- Amplia base de datos de firmas de ataques actualizada constantemente.

(Angela Orebaugh, 2005)

Suricata

Suricata es una herramienta de código abierto para detección y prevención de intrusiones en red (IDS/IPS), alternativa a Snort. Suricata ofrece:

- Monitoreo en tiempo real de tráfico en red para detección de amenazas.
- Motor de reglas y firmas para identificar ataques conocidos y malware.
- Detección de anomalías mediante análisis estadístico de flujos de red.
- Alto rendimiento utilizando capacidades multihilo y aceleración por hardware.
- Integración con herramientas de seguridad como Bro, Zeek e Intrusion Detection Framework de Snort.
- Capacidad de prevenir ataques en línea configurándolo en modo IPS.
- Soporte de protocolos de red modernos como HTTP/2, TLS 1.3, QUIC, MQTT, entre otros.

[18] Exploit: es un software diseñado para aprovechar un fallo en un sistema informático, normalmente con fines maliciosos, como la instalación de malware.

- Interfaz web para administración, generación de reportes y alertas en tiempo real.
- Amplia documentación y comunidad activa de soporte.

(Diehl, 2021)

Fail2ban

Ilustración 13 Logotipo de fail2ban[19]

Fail2ban es una herramienta de seguridad open source para proteger sistemas Unix de ataques de fuerza bruta y otros abusos. Fail2ban funciona de la siguiente manera:

- Monitorea logs del sistema buscando patrones sospechosos como múltiples intentos fallidos de login.
- Cuando detecta actividad maliciosa, actualiza firewalls como iptables para bloquear temporalmente la IP origen.
- Evita accesos no autorizados al sistema al banear IPs que muestren signos de un ataque.
- Soporta servicios como SSH, HTTP, SMTP, FTP, Wordpress, entre muchos otros.
- Permite personalizar los filtros de baneo mediante expresiones regulares en los archivos de configuración.
- Es altamente personalizable en cuanto a duración de los baneos, umbrales y acciones a tomar.

[19] "https://github.com/fail2ban/fail2ban"

- Puede integrarse con herramientas de monitoreo y alertas como Nagios para reportar incidentes.

(Bauer, 2015)

Esta información proporciona la forma de comportamiento de los IDS comunes para poder diferenciarlo del IDS propuesto, de esta manera se ve que la optimización e implementación de multiplataforma es clara dado el lenguaje de programación implementado.

Capítulo 7: Computadoras y Sistemas Operativos

Una computadora es una máquina que puede ser instruida para llevar a cabo secuencias de operaciones aritméticas o lógicas automáticamente mediante programación. Las computadoras modernas son capaces de ejecutar billones de cálculos por segundo y almacenar vastas cantidades de datos.

Sus características principales son:

- Responde a una secuencia específica de instrucciones en un programa de computadora.
- Puede almacenar datos e instrucciones.
- Puede realizar operaciones aritméticas y lógicas simples sobre los datos.

Una computadora moderna está compuesta por al menos una unidad de procesamiento, unidades de almacenamiento de datos e instrucciones, dispositivos de entrada/salida y módulos de control." (David A. Patterson (Autor), 2013)

Hardware

Hardware se refiere a los componentes físicos, tangibles y electrónicos de una computadora. Es la parte material que proporciona la capacidad de procesamiento, almacenamiento y entrada/salida de datos.

Los componentes básicos de hardware necesarios para una computadora son:

- Unidad de procesamiento (CPU - Central Processing Unit): Es el componente encargado de ejecutar las instrucciones de un programa de computadora. Interpreta y lleva a cabo operaciones lógicas y aritméticas.

- Memoria: Proporciona espacio de almacenamiento para programas e instrucciones a ser ejecutados por la CPU, así como datos de entrada y salida. Existen dos tipos principales:
- Memoria principal (RAM - Random Access Memory): Memoria volátil de acceso rápido.
- Memoria secundaria (discos duros, SSD, etc.): Memoria no volátil de gran capacidad, pero acceso más lento.
- Dispositivos de entrada/salida (E/S): Permiten la comunicación entre la computadora y el mundo externo. Ejemplos: teclado, mouse, pantalla, impresora, etc.
- Buses: Circuitos que permiten la transferencia de datos entre los componentes principales de la computadora.
- Otros componentes como fuente de poder, carcasa, tarjetas de expansión, etc.
- (David A. Patterson (Autor), 2013)

Software

El software se refiere a las instrucciones o programas que controlan el funcionamiento de un sistema de computadora. Es la parte lógica e intangible que le dice al hardware qué hacer y cómo hacerlo.

En cuanto a las características del software, los autores mencionan las siguientes:

- Portabilidad: La capacidad de ejecutar el mismo software en diferentes tipos de hardware.
- Diseño jerárquico: El software se organiza en capas o niveles, donde cada nivel depende de los servicios proporcionados por el nivel inferior.
- Modularidad: El software se divide en módulos o componentes separados que pueden ser desarrollados y probados de forma independiente.
- Abstracciones: El software proporciona abstracciones que ocultan los detalles complejos del hardware subyacente.

- Interpretabilidad: El software es un conjunto de instrucciones que pueden ser interpretadas y ejecutadas por el hardware.
- Facilidad de desarrollo: El software puede ser desarrollado, probado y mantenido más fácilmente que el hardware.
- Mal funcionamiento: El software puede contener errores o fallas que pueden causar un comportamiento inesperado o incorrecto en el sistema.
- Ciclo de vida: El software tiene un ciclo de vida que incluye etapas como requisitos, diseño, implementación, pruebas, mantenimiento y actualización.
- (David A. Patterson (Autor), 2013)

Sistemas Operativos

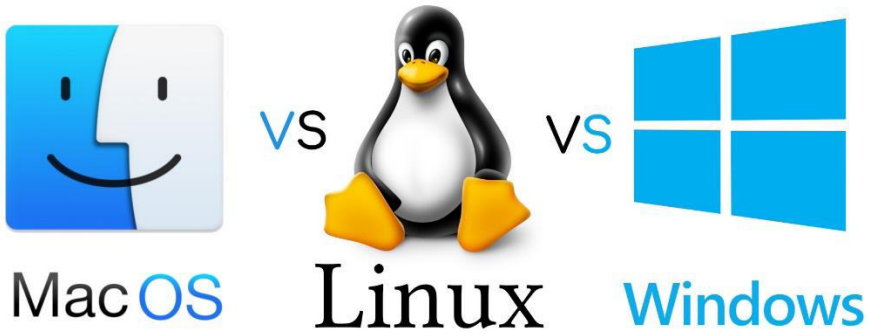

Ilustración 14 Logo de los sistemas operativos más Comunes[20]

Windows: Es un sistema operativo de Microsoft diseñado principalmente para computadoras personales y servidores. Utiliza una interfaz gráfica de usuario basada en ventanas y está orientado a ofrecer una experiencia de usuario intuitiva y amigable. Algunos de sus componentes clave son el Administrador de Ventanas, el Administrador de Objetos, el Administrador de Memoria y el Administrador de Procesos.

[20] "https://www.stemprinting.com/windows-mac-llnux/"

macOS: Es el sistema operativo desarrollado por Apple para sus líneas de computadoras Mac. Está basado en el núcleo Unix y combina una interfaz gráfica elegante con características de seguridad y estabilidad heredadas de Unix. Cuenta con un administrador de ventanas llamado Quartz Compositor, un sistema de archivos jerárquico y servicios como el gestor de arranque, el instalador de software y el administrador de energía.

Linux: Es un sistema operativo de código abierto basado en el núcleo Linux. Existen numerosas distribuciones de Linux, como Ubuntu, Fedora, Debian, etc. Está diseñado para ser portable, multitarea y multiusuario, con soporte para una amplia gama de arquitecturas de hardware. Utiliza el sistema de archivos jerárquico y ofrece una gran selección de aplicaciones gratuitas y de código abierto. (Andrew S. Tanenbaum, 2009)

Desarrollo óptimo de software en Sistemas Operativos

Un programa óptimo en un sistema operativo se refiere a un programa de software que ha sido diseñado e implementado de manera que maximiza su eficiencia, rendimiento y uso adecuado de los recursos del sistema, al tiempo que minimiza su impacto negativo en el funcionamiento general del sistema operativo.

Las características clave de un programa óptimo en un sistema operativo incluyen:

- Uso eficiente de recursos como CPU, memoria, E/S y ancho de banda de red.
- Alto rendimiento con tiempos de respuesta rápidos y alta capacidad de procesamiento.
- Baja latencia en operaciones críticas.
- Escalabilidad para manejar aumentos en la carga de trabajo sin degradación significativa del rendimiento.
- Eficiencia energética, especialmente en sistemas embebidos o dispositivos móviles.
- Seguridad adecuada para proteger datos y el sistema.

- Modularidad y mantenibilidad del código.
- Compatibilidad con el sistema operativo, hardware y otros componentes de software.

(Greg Gagne, 2018)

Raspberry Pi

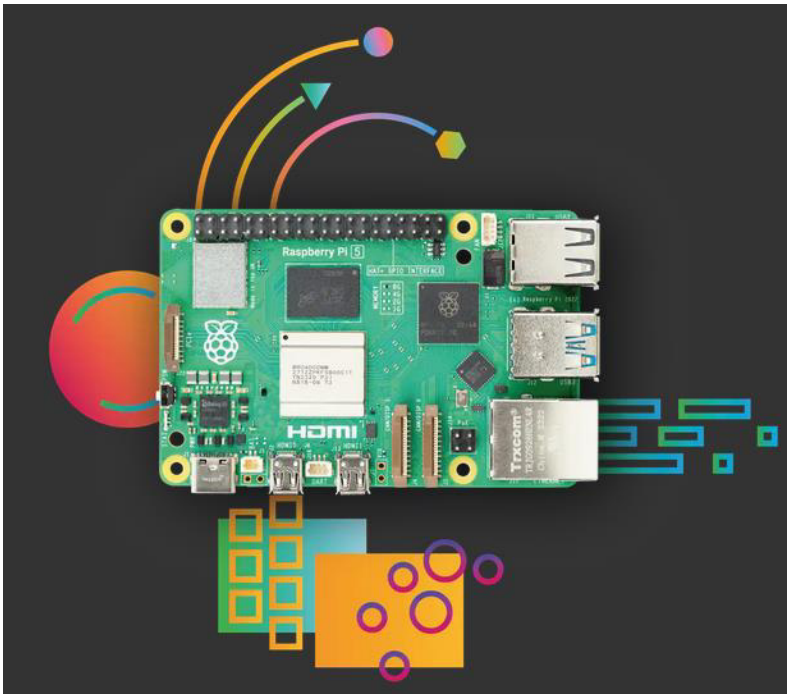

Ilustración 15. Estructura de Rasberry Pi[21]

Es una pequeña computadora del tamaño de una tarjeta de crédito que se puede utilizar para diversos proyectos y aplicaciones. Algunas de sus principales características son:

- Procesador ARM de bajo consumo energético
- Puertos GPIO (entrada/salida de propósito general) que permiten conectar sensores, actuadores, etc.
- Salidas de video y audio para conectar pantallas y parlantes

[21] "https://www.raspberrypi.com/"

- Puertos USB para conectar dispositivos como teclados, ratones, cámaras web, etc.
- Conectividad Ethernet y WiFi
- Capacidad para ejecutar sistemas operativos como Raspbian (basado en Debian Linux)

Debido a su bajo costo, tamaño compacto y consumo eficiente de energía, las Raspberry Pi son muy populares en proyectos de electrónica, robótica, domótica, servidores web/multimedia caseros, y para aprender programación y desarrollar habilidades de informática. (Foundation, 2024)

Capítulo 8: Desarrollo e Implementación

El desarrollo de este IDS se dará en base al backend[22] del programa y estará basado en diversas técnicas de minería de datos como lo son:

- Técnica de Agrupación (Clustering): Esta técnica permite obtener datos del tráfico de red en su forma cruda, como direcciones IP, direcciones MAC y el tipo de paquete, tanto de emisor como de receptor. Estos datos se organizan en tablas almacenadas en archivos CSV, lo que facilita la aplicación de la técnica de asociación.
- Técnica de Asociación (Association): Mediante esta técnica, se asocian los datos obtenidos en la etapa de agrupación, generando listas reducidas que, además de contener la información básica (direcciones IP, MAC y tipos de paquetes), incluyen contadores de paquetes enviados en total y contadores de los diferentes tipos de paquetes.
- Técnica Análisis de Series Temporales: Esta técnica permite navegar entre los archivos de datos y aplicar las técnicas de asociación y reducción de dimensionalidad en procesos específicos, como el "salto de día" y las estadísticas resultantes de dicho proceso.
- Técnica de Reducción de Dimensionalidad: Mediante esta técnica, se eliminan las redundancias presentes en la información de los archivos, preparándolos para la generación de estadísticas complementarias al programa, como el análisis continuo de la red en tiempo real, el resumen del día que se tomará en consideración para la detección de anomalías y los registros de las alertas generadas por estas circunstancias.
- Técnica de Minería Web: Esta técnica permite interpretar e interactuar con la información del escaneo de servicios ejecutados en los puertos de cada

[22] Un programa se divide en dos fases: El funcionamiento del programa de forma de procesos de un sistema (backend) y la interfaz gráfica que liga el backend con la interacción de un usuario (Frontend).

dispositivo perteneciente a la red. Esto se logra con el apoyo de la herramienta de ciberseguridad Nmap, que obtiene información de dichos puertos, y la página VirusTotal[23], que permite identificar la existencia de algún virus.

Se empleará la herramienta Nmap con el fin de identificar los dispositivos conectados a la red y realizar una comparación con la base de datos de VirusTotal. Esta acción permitirá verificar que, previo al análisis del escenario inicial (día cero), no existan dispositivos comprometidos por algún código malicioso que pudiese alterar los resultados obtenidos. Esta precaución es crucial para garantizar la integridad de los datos que servirán como entrada al algoritmo de detección de intrusos, evitando así que posibles infecciones previas distorsionen las estadísticas y afecten la detección de anomalías de la red.

Así como se ha mencionado desde un principio, esta tesis este está basado en dos fases (Algoritmo de día cero y Algoritmo base del IDS) que serán explicadas en las siguientes páginas.

[23] VirusTotal: página web que analiza archivos, dominios, IP y URL sospechosos para detectar malware y otras infracciones y compártalos automáticamente con la comunidad de seguridad.

ALGORITMO DE DIA CERO DEL IDS

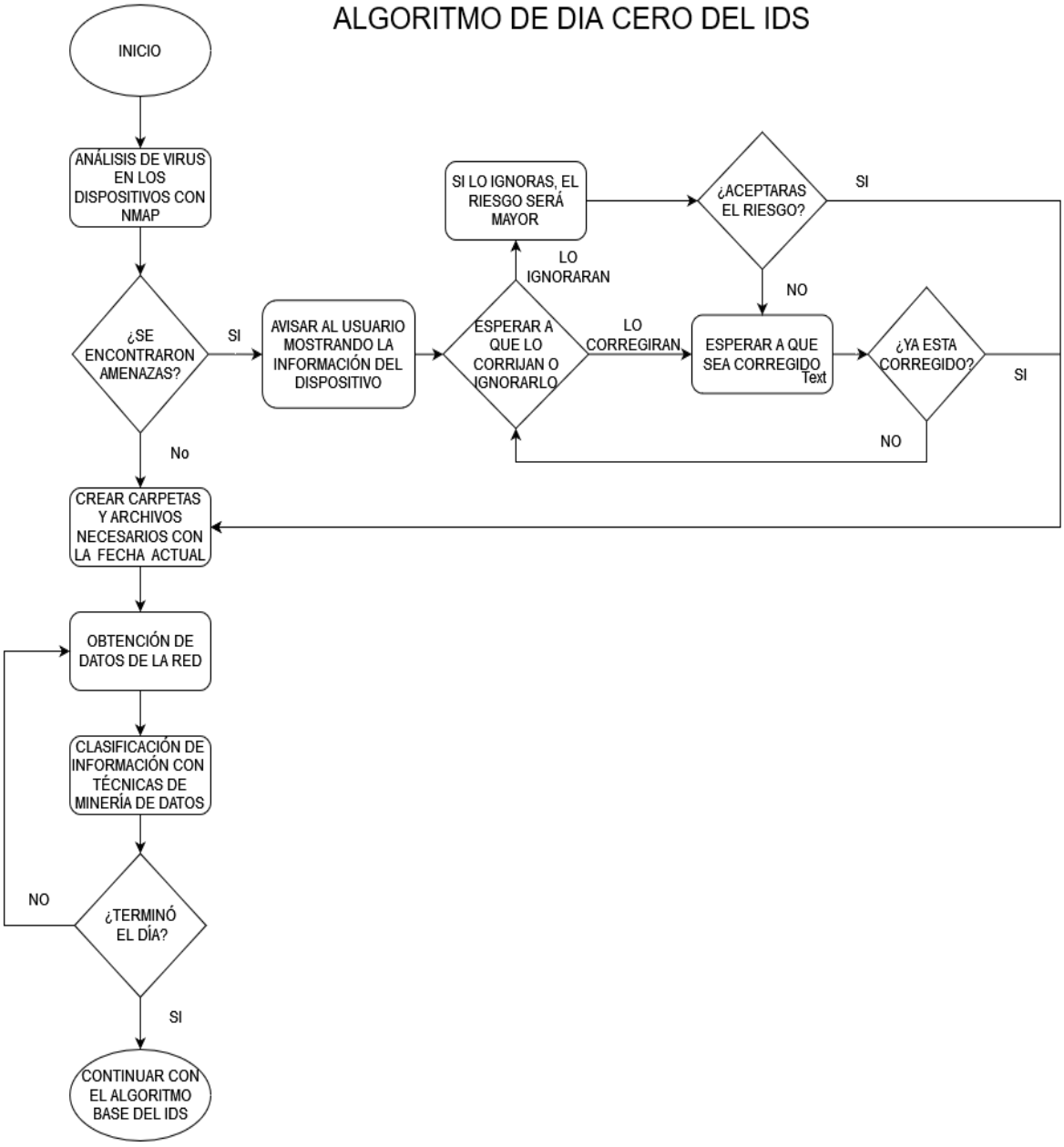

Ilustración 16 Algoritmo del día cero del IDS

En este algoritmo de pueden percibir varios procesos:
1.- El algoritmo propuesto contempla diversos procesos clave. En primer lugar, se lleva a cabo un "Análisis de dispositivos con Nmap" con el objetivo de examinar los servicios y puertos abiertos en cada host de la red, como los puertos 8080, 8181, entre otros. Este paso es fundamental para asegurar que el análisis de tráfico de red en el escenario inicial (día cero) no se vea contaminado por el registro de paquetes generados por códigos maliciosos o técnicas de hacking, tales como el movimiento lateral o pivoteo en red.

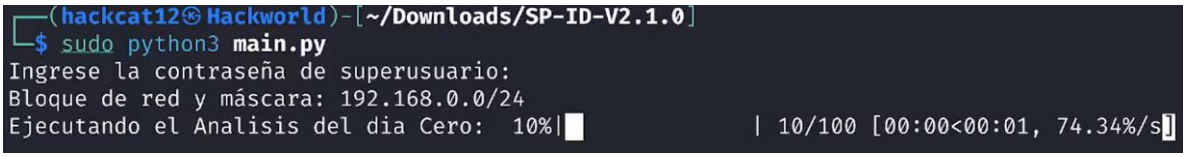

Ilustración 17 Vista principal del análisis de Nmap en la red dentro del Backend del programa

2.- En caso de que el análisis con Nmap detecte la presencia de algún virus o amenaza en los servicios de los dispositivos, se generará una alerta al administrador. Este podrá entonces investigar el incidente con mayor profundidad o, si se trata de un riesgo aceptable, optar por continuar con el proceso.

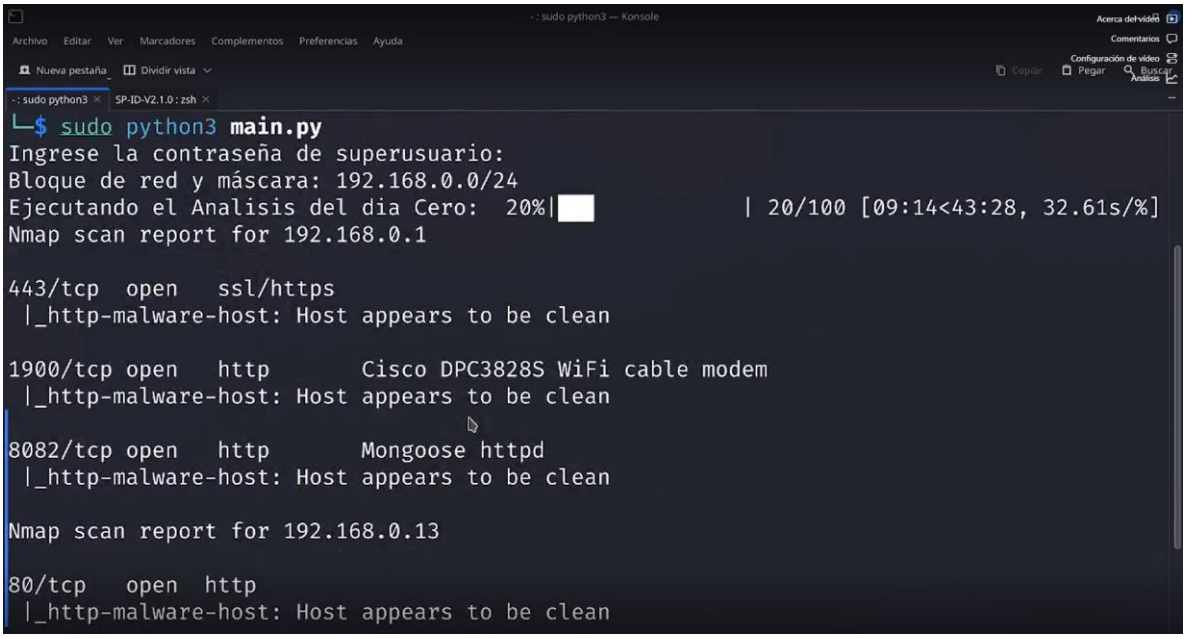

Ilustración 18 Vista de resultados del Análisis de Nmap (parte 1 de 2)

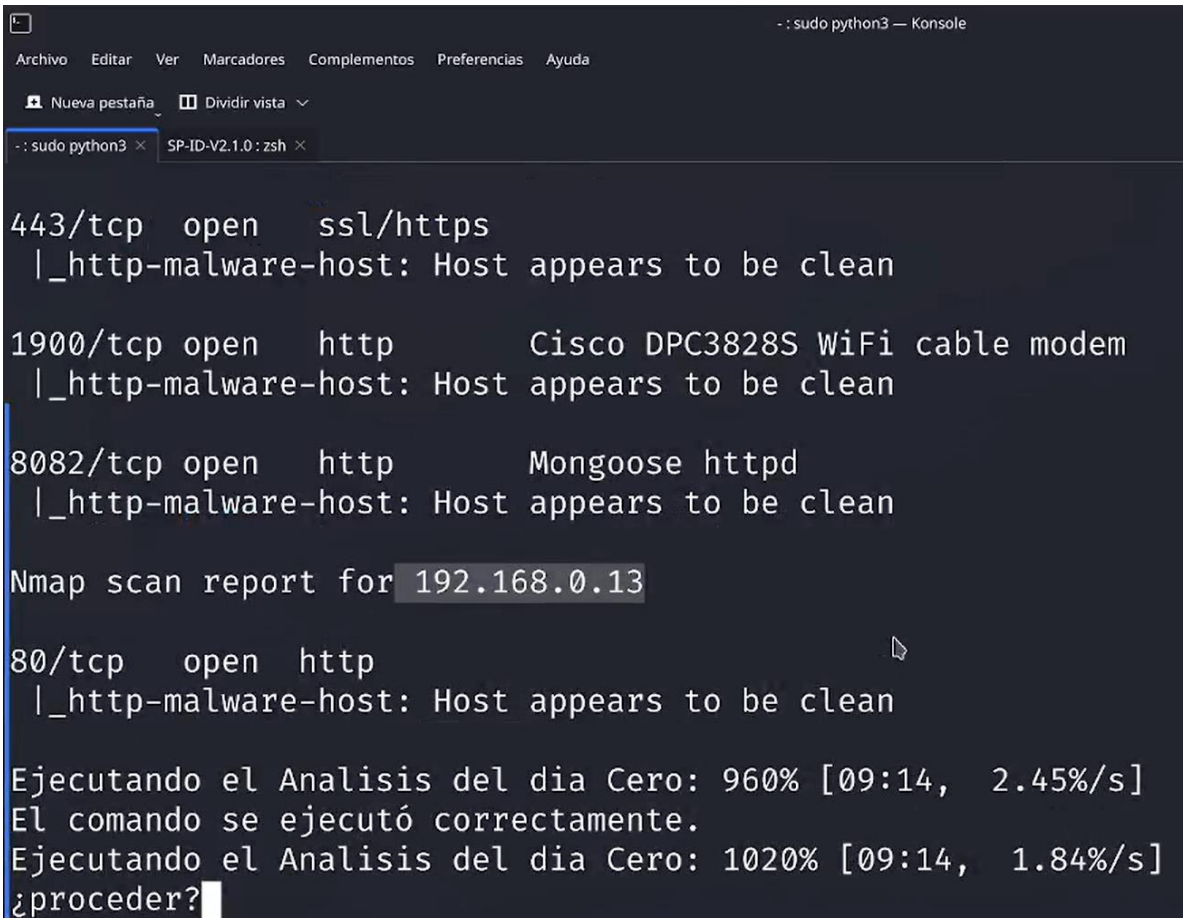

Ilustración 19 Vista de resultados del Análisis de Nmap (parte 2 de 2)

3.- Independientemente de la decisión tomada, el sistema procederá a crear los archivos necesarios para iniciar el registro de los paquetes de red. Uno de estos archivos es "Análisis de día cero.txt", en el cual se almacena información detallada sobre el estado de los servicios y la presencia de virus en los dispositivos analizados. Adicionalmente, se generarán archivos de registro encriptados para garantizar una mayor protección de los datos recopilados. Esta medida de seguridad es crucial, ya que el dispositivo encargado de ejecutar este programa podría convertirse en un objetivo atractivo para los ciberdelincuentes, quienes podrían intentar acceder a él y corromper los registros para ocultar sus actividades maliciosas y evitar ser detectados por el administrador.

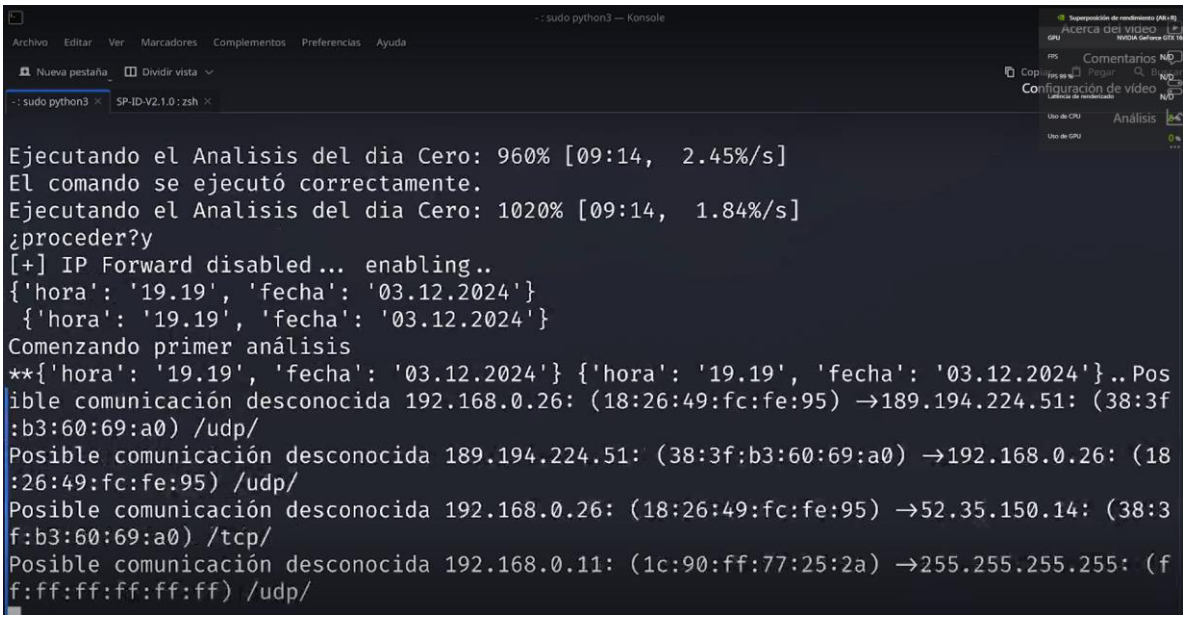

Ilustración 20 Vista del Backend del programa al ejecutar el primer análisis (análisis de día cero)

Una vez completado el período de 24 horas de registro, el sistema avanzará al Proceso General (Algoritmo Base del IDS) para el análisis y detección de intrusos.

Colocar el QR del funcionamiento del backend

Para más comprensión puede ver el video del funcionamiento del backend del programa que se encuentra en el siguiente QR:

ALGORITMO BASE DEL IDS

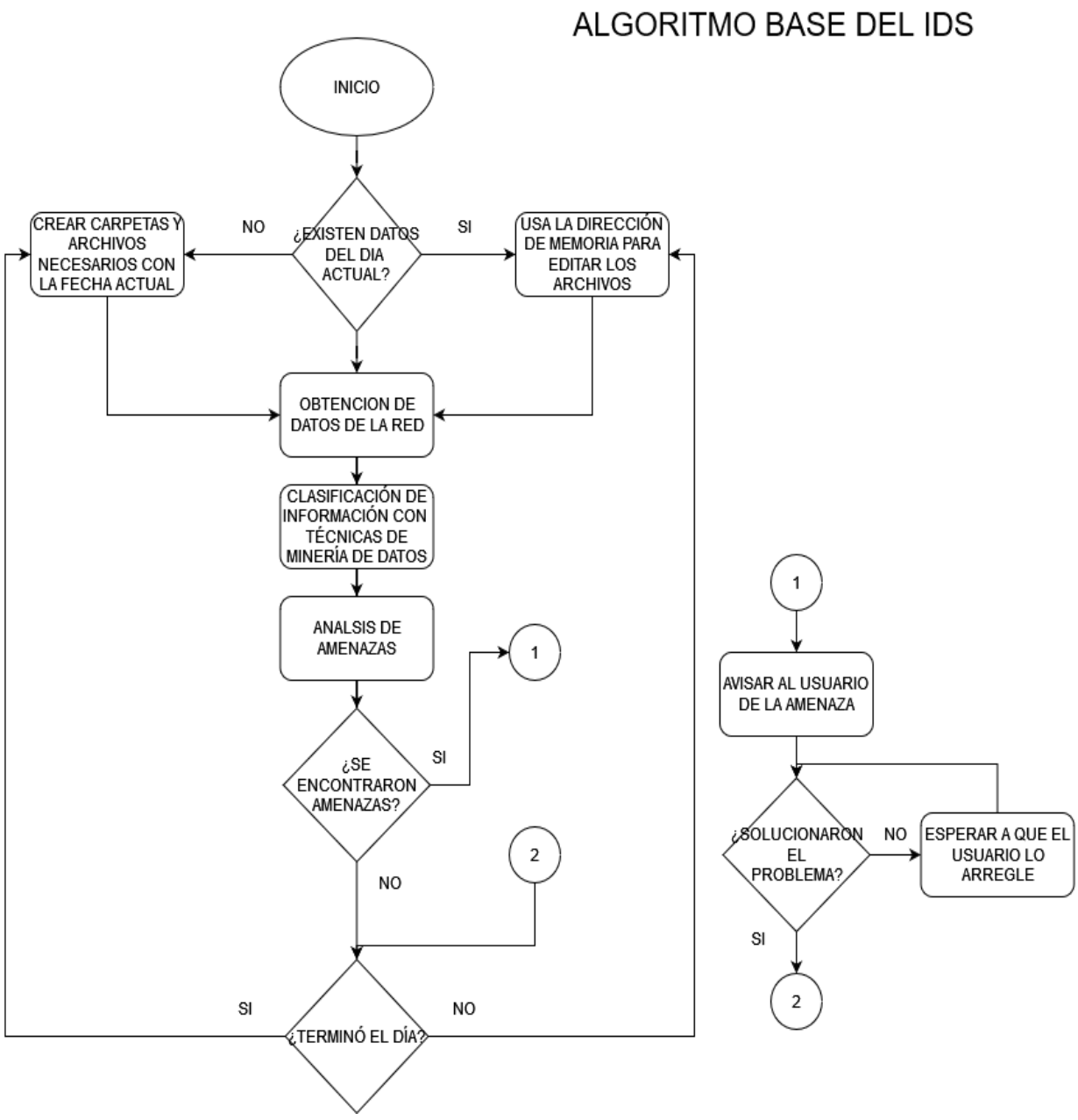

Ilustración 21 Algoritmo Base del IDS

Procesos implicados:

- Recopilación de tráfico de red: Se utiliza la biblioteca Scapy de Python para capturar y analizar el tráfico de red en tiempo real. Los datos recopilados incluyen la fecha, las direcciones MAC de origen y destino, las direcciones IP de origen y destino, los puertos de origen y destino, así como los paquetes ICMP, SSH, FTP, SFTP y Telnet, entre otros. (Para más información consultar Tipos de protocolos de paquetes en una red en el Apéndice A)
- Almacenamiento de datos en bruto: Los datos de tráfico de red capturados se almacenan en su formato original en un archivo CSV denominado "sniffing.csv" utilizando la librería Pandas de Python de esta manera aplicamos la técnica de agrupación para obtener listas de datos en crudo.
- Aplicación de técnicas de minería de datos: Se aplican técnicas de asociación (Association) de minería de datos al archivo "sniffing.csv" para identificar patrones de dichos datos.
- Generación de datos procesados: Los resultados del proceso anterior se almacenan en un nuevo archivo CSV llamado "count.csv".
- Procesamiento de alertas: Se analizan los datos procesados para detectar posibles amenazas o actividades sospechosas y generar alertas correspondientes.
- Verificación del fin del día: Se determina si se ha alcanzado el final del período de recopilación de datos (24 horas).
- Iteración o finalización del proceso:
- Si no se ha alcanzado el final del día, el proceso regresa al paso 1 para continuar recopilando y procesando datos.
- Si se ha alcanzado el final del día, se compara el archivo "count.csv" del día actual con el del día anterior. Se calcula el promedio por columna y se genera un archivo de referencia para el siguiente día. Este archivo de

referencia se utilizará como conjunto de entrenamiento para el algoritmo de detección de intrusos.

Este proceso se aplica tanto para el escenario inicial (día cero) como para el funcionamiento continuo del sistema de detección de intrusos (IDS).

Para la generación de alertas, se ha implementado una técnica de hacking conocida como "man-in-the-middle" como técnica de apoyo para el usuario o administrador. Esta técnica permite capturar y analizar los paquetes de red utilizando la herramienta Wireshark, lo cual facilita la identificación de falsos positivos o falsos negativos en las alertas generadas por el sistema.

Las alertas se generan mediante la comparación de los registros de tráfico de red con los patrones esperados.

Tipos de alertas y sus condiciones de activación (Para comprender mejor dichas alertas léase el Apéndice A):

- Alerta de "Intruso detectado": Se activa cuando se detecta una dirección MAC no registrada en la red. El formato de la alerta es "$MAC:ip_intrusa:Protocolo$, registro de error (activa man-in-the-middle)".
- Alerta de "Precaución, los puertos no coinciden": Se activa cuando los puertos de origen y destino de una comunicación no coinciden con los esperados. El formato de la alerta es "$MAC:ip:puerto$(origen) - $MAC:ip:puerto$(destino), registro de error, (activa man-in-the-middle)".
- Alerta de "Posible movimiento lateral": Se activa cuando no existe un registro previo de comunicación entre dos direcciones MAC. El formato de la alerta es "$MAC:ip:puerto$(origen) - $MAC:ip:puerto$(destino), registro de error (activa man-in-the-middle)".
- Alerta de "Paquetes excedidos, posible extracción de datos": Se activa cuando el número total de paquetes excede en un 15% el umbral esperado. El formato de la alerta es "$MAC;ip:puerto$(origen) - $MAC:ip:puerto$(destino), registro de error, (activa man-in-the-middle)".

- Alerta de "Ataque DDoS": Se activa cuando el número de paquetes ICMP excede en un 15% el umbral esperado. El formato de la alerta es "$MAC:ip:puerto$(origen) - $MAC:ip:puerto$(destino), registro de error".
- Alerta de "Posible Ataque de fuerza bruta": Se activa cuando el número de paquetes SSH, FTP, SFTP o Telnet excede en un 15% el umbral esperado. El formato de la alerta es "$MAC:ip:puerto$(origen) - $MAC:ip:puerto$(destino), (activa man-in-the-middle)".
- No se generan alertas por MAC spoofing, ya que, para acceder a la red, el dispositivo debe utilizar su dirección MAC real, la cual sería detectada como un intruso.

Registro de alertas del día cero:

- Se crea una carpeta con el nombre "alerta_dia_cero".
- Se crea un archivo llamado "registro_de_malware_dia_cero_fecha".En este archivo, se registran los siguientes datos: fecha, hora, IP, puerto, estado del puerto, presencia de malware en el servicio, dirección MAC, indicación de falso positivo, alerta ignorada, alerta pendiente, alerta de excepción, y un campo de observaciones.

Algoritmo de registro de errores

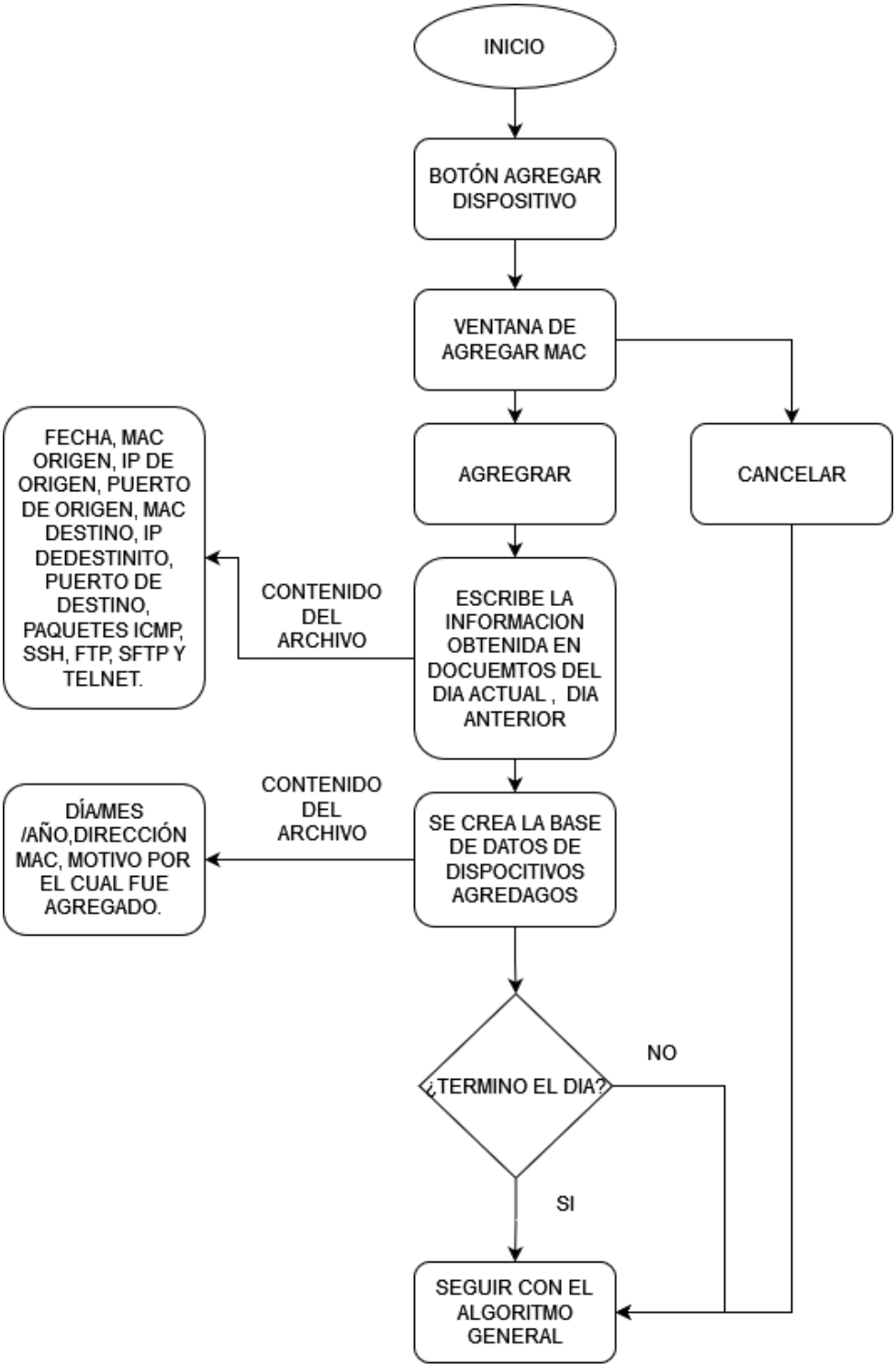

Ilustración 22 Algoritmo de registro de errores

- Crear una carpeta con el nombre de la alerta
- Crear un archivo llamado registro_fecha_Nombre de la alerta

- Guardar datos de fecha, hora, protocolo, MAC, ip y puerto(origen), MAC, ip y puerto(destino), Revisada, Ignorada, pendiente, excepción, observación (campo de observación escrito), en el archivo anterior.

Si da click en el botón revisado:

- En el de registro se colocará una x en el campo, rompe el man in the middle y en observación se escribe el error de inconsistencia.

Nos lleva a excepción:

- Si es excepción se marca con una x en el campo nos lleva a observación, rompe el man in the middle y elimina todo lo que estaba ocurriendo con en Man in the middle y se llena la observación del falso positivo y lo registra en archivo ayer y archivo hoy

Botón observación:

- Colocar por qué dio el error o si se soluciono

Quedará pendiente:

- Si queda pendiente no se borra la alerta, deja pendiente las observaciones, continua con el man in the middle y conserva todo lo que estaba ocurriendo con en Man in the middle y se llena la observación y RECUERDA dentro de 5, 10, 30, 1 hr.

Dar click en el botón ignorar:

- Se marca el campo de ignorar con una x y los demás campos se marcan en nada, rompe el man in the middle y elimina todo lo que estaba ocurriendo con en Man in the middle y se llena la observación y lo registra en archivo ayer y archivo hoy.

Respecto a los botones de agregar excepción se crea un algoritmo de la siguiente manera:

Se presiona el botón agregar dispositivo o ignorar (en caso de detectar alguna dirección MAC no reconocida “intruso”), en este se despliega un recuadro de dirección MAC y los botones agregar y cancelar

En caso de cancelar, se cierra el recuadro

En caso de agregar, se agrega toda la información recabada en tres documentos (día anterior, día actual y base de datos de dispositivos agregados) de esta manera al final se hace un promediado final del cual ya habíamos hablado respecto al del día siguiente para mantener un orden y que no se alteren las métricas de ambos. En la base de datos se agrega día/mes/año, dirección MAC, motivo por el cual fue agregado.

Funcionamiento del programa

Este programa funciona en base a los algoritmos descritos en la tesis y sigue el siguiente flujo:

- Solicitud de bloque de red: El programa solicita el bloque de red a analizar, por ejemplo, 192.168.0.0/24 (donde /24 indica que se analizará el último octeto 192.168.0.0). Ilustración 24 Pantalla de análisis del día cero de la GUI del IDS propuesto
- Solicitud de clave de administrador o usuario root: Debido a que el programa requiere ejecutar procesos con privilegios elevados, se solicita la clave de administrador o usuario root. Ilustración 24 Pantalla de análisis del día cero de la GUI del IDS propuesto
- Opciones de análisis:
    - a. Análisis de malware de día cero: Esta opción permite la búsqueda de virus en todos los dispositivos conectados a la red, proporcionando un resumen de si existe o no algún virus en ellos.
    - b. Análisis de host de día cero: Esta opción recopila el comportamiento de la red, analizando el número y tipo de paquetes que se envían entre diferentes dispositivos. Este proceso dura 24 horas.

- Captura de tráfico de red y detección de anomalías: Al terminar el análisis seleccionado, el programa captura el tráfico de red y compara las cifras de paquetes recaudados para detectar anomalías. Se tiene en cuenta una tolerancia del 15% de la cifra total de cada tipo de paquete para activar la alerta correspondiente.
- Actualización diaria y reducción de alertas falsas: Al finalizar el día, el programa calcula el promedio de las cifras comparando los datos del día actual con los del día anterior. De esta manera, se mantiene un seguimiento de las cifras actuales, reduciendo la probabilidad de generar alertas falsas. Ilustración 25 Pantalla de Alertas del día cero de la GUI del IDS propuesto

Mientras el programa esté en funcionamiento y recaudando información de la red, se podrán realizar diversas tareas de seguimiento y análisis. Algunas de las capacidades que ofrece el programa son:

Seguimiento de análisis Ilustración 27 Pantalla de Seguimiento de Alertas de la GUI del IDS propuesto:

- Monitorear el progreso de los análisis de malware de día cero y de host de día cero.
- Visualizar el estado actual de los análisis y los dispositivos o hosts que ya han sido analizados.
- Revisar los resultados preliminares de los análisis en curso.

Seguimiento de alertas Ilustración 27 Pantalla de Seguimiento de Alertas de la GUI del IDS propuesto:

- Recibir y revisar las alertas generadas por el programa cuando se detectan anomalías en el tráfico de red.
- Acceder a detalles específicos sobre las alertas, como el tipo de anomalía detectada, los dispositivos involucrados, el tráfico sospechoso, entre otros.

- Realizar un seguimiento histórico de las alertas generadas en diferentes períodos de tiempo.

Generación de estadísticas Ilustración 28 Pantalla de Estadísticas de la GUI del IDS propuesto:

- Obtener estadísticas detalladas sobre el tráfico de red analizado, incluyendo el número y tipo de paquetes transmitidos, direcciones IP involucradas, protocolos utilizados, entre otros.
- Generar informes estadísticos para diferentes intervalos de tiempo (diario, semanal, mensual, etc.).
- Visualizar gráficas y representaciones gráficas de las estadísticas para facilitar el análisis y la identificación de patrones.

Estas capacidades de seguimiento y análisis permiten a los administradores de red mantener un control exhaustivo sobre la seguridad de la red, detectar amenazas de manera oportuna y tomar medidas preventivas o correctivas según sea necesario. Además, la generación de estadísticas puede ser útil para identificar tendencias, realizar ajustes en las configuraciones de seguridad y optimizar el rendimiento de la red.

Implementación

A partir de la información presentada en el Capítulo 1: Redes Computacionales, se observa que existen tres dimensiones de redes: pequeñas (LAN), medianas (WAN) y grandes (Para más información consultar Tipos de redes en el Apéndice A). Tomando en cuenta el objetivo de la presente tesis, el programa propuesto está diseñado para su implementación en redes pequeñas y medianas. Si bien estas redes comparten ciertas similitudes, existen diferencias clave que deben considerarse, como la dimensionalidad de dispositivos y la factibilidad de implementar herramientas de protección adicionales.

Con el fin de validar el desempeño del sistema desarrollado, se llevaron a cabo su implementación en una empresa de tamaño mediano ubicada en León, Guanajuato. Es importante destacar que, como condición para su colaboración, la empresa

solicitó mantener su identidad anónima durante y después de la implementación. Esta decisión se fundamenta en la posibilidad de que los resultados obtenidos puedan ser utilizados a favor o en contra de los intereses de la organización.

La elección de una empresa de tamaño mediano se alinea con el enfoque del programa propuesto, ya que este tipo de redes presentan una complejidad y requisitos de seguridad más exigentes en comparación con las redes pequeñas, pero sin alcanzar la escala y complejidad de las redes grandes. Esto permitió evaluar el rendimiento del sistema en un entorno real, con una infraestructura de red y requisitos de seguridad más representativos de las condiciones que enfrentarían las organizaciones objetivo.

También tomando el objetivo de mejorar la experiencia de usuario y facilitar la operación del sistema de detección de intrusos (IDS) propuesto, se desarrolló una interfaz gráfica de usuario (GUI, por sus siglas en inglés). La implementación de esta interfaz gráfica brinda múltiples beneficios:

- Facilidad de uso: La interfaz gráfica proporciona una manera intuitiva y amigable para que los usuarios interactúen con el sistema, lo que reduce la curva de aprendizaje y aumenta la accesibilidad del software.
- Visualización de información: La GUI permite la representación visual de los datos y alertas generados por el IDS, lo que facilita la comprensión y el análisis de la información por parte de los administradores de seguridad.
- Detección y corrección de errores: Al centralizar la interacción con el sistema en una interfaz gráfica, se simplifica el proceso de identificación y solución de posibles errores o problemas que puedan surgir durante la ejecución del IDS.
- Guía de operación: La interfaz gráfica actúa como una guía visual que ilustra el flujo de operación del sistema, lo que permite a los usuarios comprender de manera más clara cómo se llevan a cabo los diferentes procesos y funcionalidades del IDS.
- Personalización y configuración: La GUI proporciona opciones de personalización y configuración, permitiendo a los usuarios ajustar los

parámetros del sistema según sus necesidades específicas, como la selección de interfaces de red a monitorear, la definición de umbrales de alerta, entre otros.

- Escalabilidad y mantenibilidad: Al separar la lógica del sistema de la interfaz de usuario, se facilita el mantenimiento y la escalabilidad del software, ya que los cambios o actualizaciones en la interfaz gráfica no afectan directamente el núcleo del sistema de detección de intrusos.

Considerando el objetivo principal de la tesis, es imperativo que el programa desarrollado sea óptimo en términos de eficiencia y escalabilidad. Al tratarse de la implementación de lenguajes de programación modernos, en este caso Python como lenguaje base tanto para el backend como para el frontend, es crucial abordar el desafío del consumo de recursos de hardware y software que puede presentarse al procesar grandes cantidades de datos.

En las siguientes imágenes se muestra la interfaz gráfica que se desarrolló para la comprensión de problemas y así mejorar el manejo del IDS propuesto referenciadas en la sección Capítulo 8: Desarrollo e Implementación:

Ilustración 23 Pantalla de Bienvenida de la GUI del IDS propuesto

Ilustración 24 Pantalla de análisis del día cero de la GUI del IDS propuesto

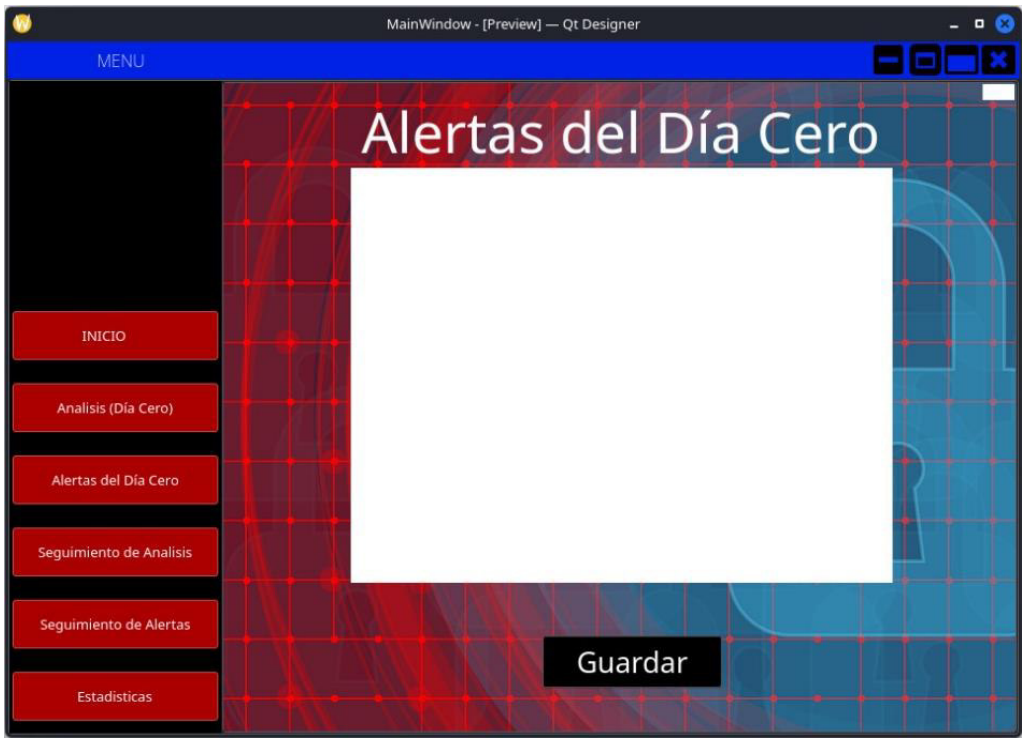

Ilustración 25 Pantalla de Alertas del día cero de la GUI del IDS propuesto

Ilustración 26 Pantalla de seguimiento de análisis de la GUI del IDS propuesto

Ilustración 27 Pantalla de Seguimiento de Alertas de la GUI del IDS propuesto

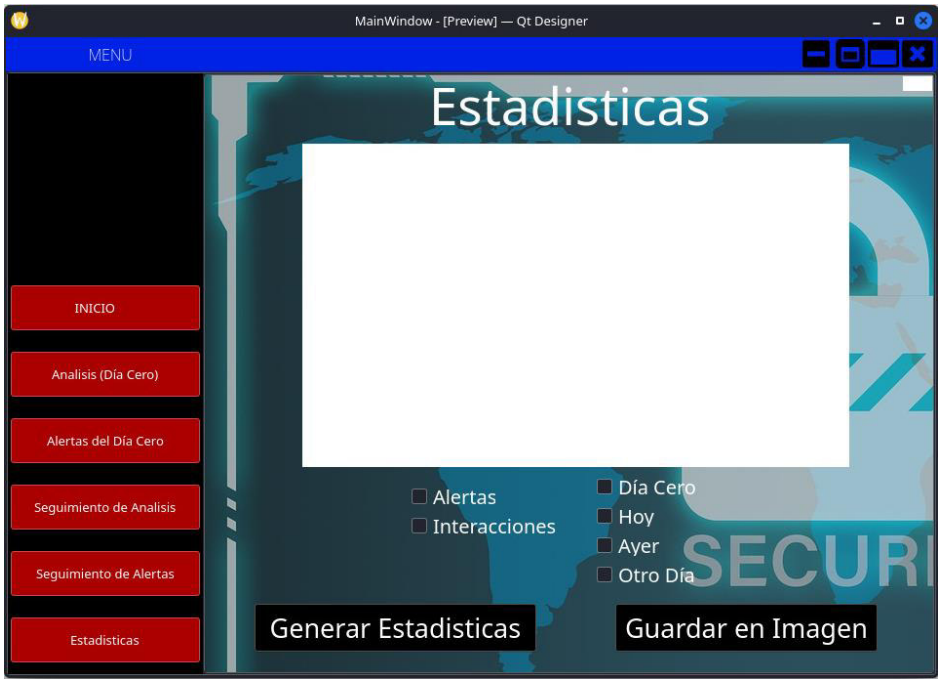

Ilustración 28 Pantalla de Estadísticas de la GUI del IDS propuesto

Con el fin de demostrar la eficiencia y escalabilidad del sistema propuesto, tanto a nivel de software como de hardware, se optó por utilizar una Raspberry Pi 4 como

plataforma de implementación. Esta elección se fundamenta en las siguientes razones:

- Optimización de recursos: Las Raspberry Pi son dispositivos de bajo consumo de energía y recursos limitados, lo que representa un entorno ideal para evaluar la eficiencia del programa desarrollado en Python. Al lograr un rendimiento adecuado en esta plataforma de recursos reducidos, se demuestra la capacidad del sistema para operar de manera óptima en entornos con restricciones de hardware.
- Escalabilidad de hardware: A pesar de sus recursos limitados, las Raspberry Pi ofrecen la posibilidad de escalabilidad mediante el uso de múltiples unidades en clúster o la integración con dispositivos de mayor capacidad de procesamiento. Esto permite evaluar la capacidad del sistema para adaptarse a diferentes configuraciones de hardware y demandas de procesamiento.
- Portabilidad y flexibilidad: Las Raspberry Pi son dispositivos compactos y portátiles, lo que facilita su implementación en diferentes entornos y escenarios de prueba. Además, su arquitectura abierta y compatibilidad con diferentes sistemas operativos proporcionan flexibilidad para realizar pruebas y evaluaciones en diversas configuraciones.
- Costos reducidos: En comparación con equipos de escritorio o servidores convencionales, las Raspberry Pi ofrecen una solución de bajo costo, lo que las convierte en una opción atractiva para el desarrollo y prueba de prototipos o sistemas de detección de intrusos en entornos académicos o de investigación.

Al demostrar la capacidad del sistema desarrollado para operar de manera eficiente y escalable en la plataforma de la Raspberry Pi 4, se respalda la viabilidad de su implementación en entornos de producción más exigentes, ya sea mediante el uso de clústeres de Raspberry Pi o mediante su integración con hardware de mayor capacidad de procesamiento.

Capítulo 9: Resultados

Es importante destacar que, con el objetivo de incrementar la seguridad y controlar el acceso a los datos obtenidos, se implementaron medidas de cifrado en los resultados. Esta encriptación asegura que solo el programa desarrollado en esta tesis pueda consultar y procesar los documentos que contienen los datos recopilados. De esta manera, se garantiza la confidencialidad y la integridad de la información, evitando accesos no autorizados o modificaciones indebidas (Ilustración 29 Ejemplo de archivo encriptado).

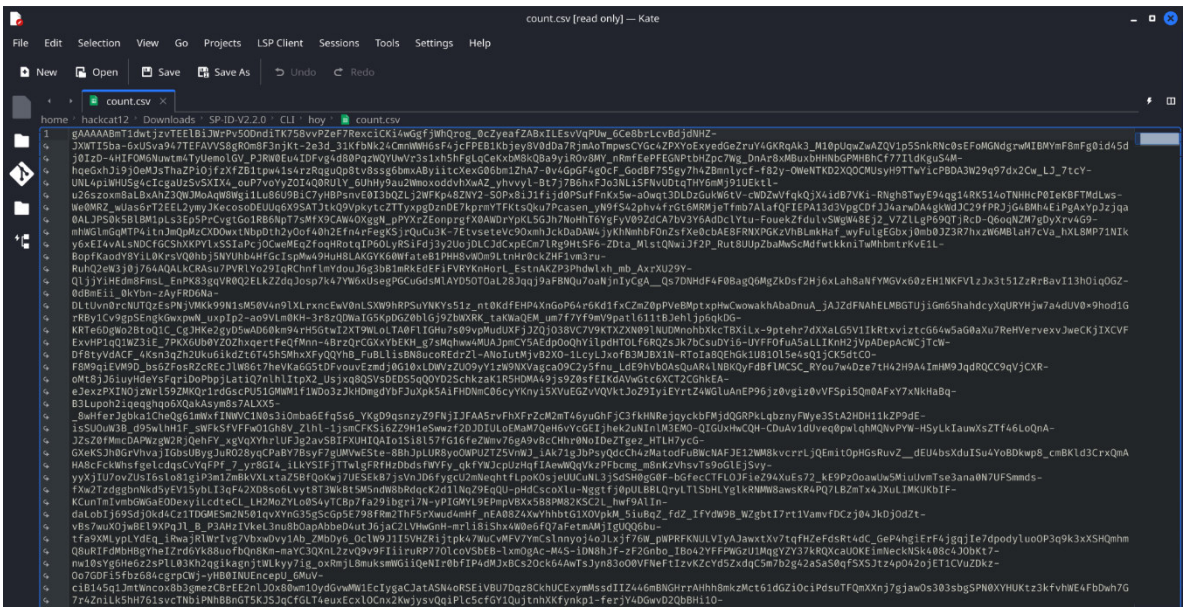

Ilustración 29 Ejemplo de archivo encriptado

Como se mencionó en capítulos anteriores, el administrador del sistema tiene la capacidad de decidir si realizar un escaneo inicial de la red en busca de virus y amenazas potenciales. Este escaneo preliminar, conocido como "escaneo del primer día", tiene como objetivo identificar la presencia de amenazas y determinar si estas son aceptables o no dentro del entorno de red específico. En caso de detectar amenazas no deseadas, el administrador puede tomar las medidas necesarias para mitigarlas o eliminarlas antes de proceder con la implementación completa del Sistema de Detección de Intrusos (IDS) propuesto. Esta funcionalidad

permite asegurar que el IDS se ejecute en un entorno de red lo más limpio y seguro posible, mejorando así su eficacia y minimizando la posibilidad de falsos positivos.

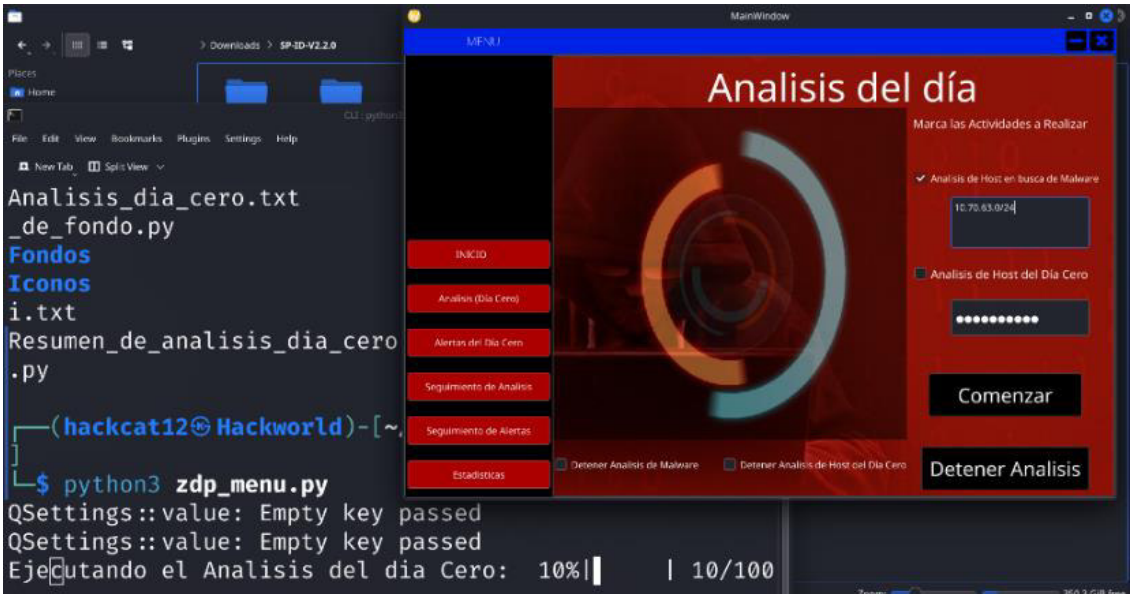

Ilustración 30 ejecución del Análisis de Malware del día cero

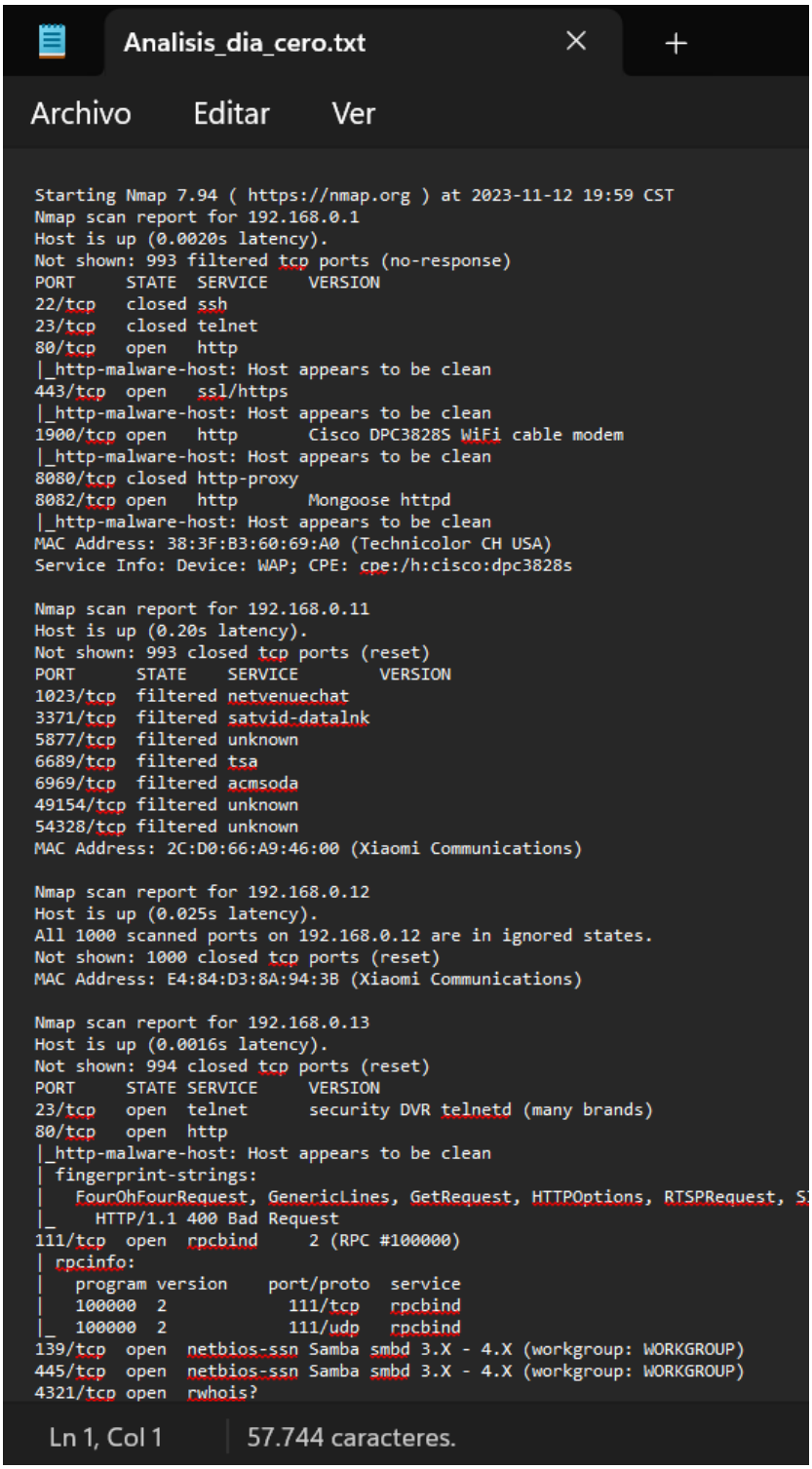

Ilustración 31 Análisis de día cero sin haber sido resumido

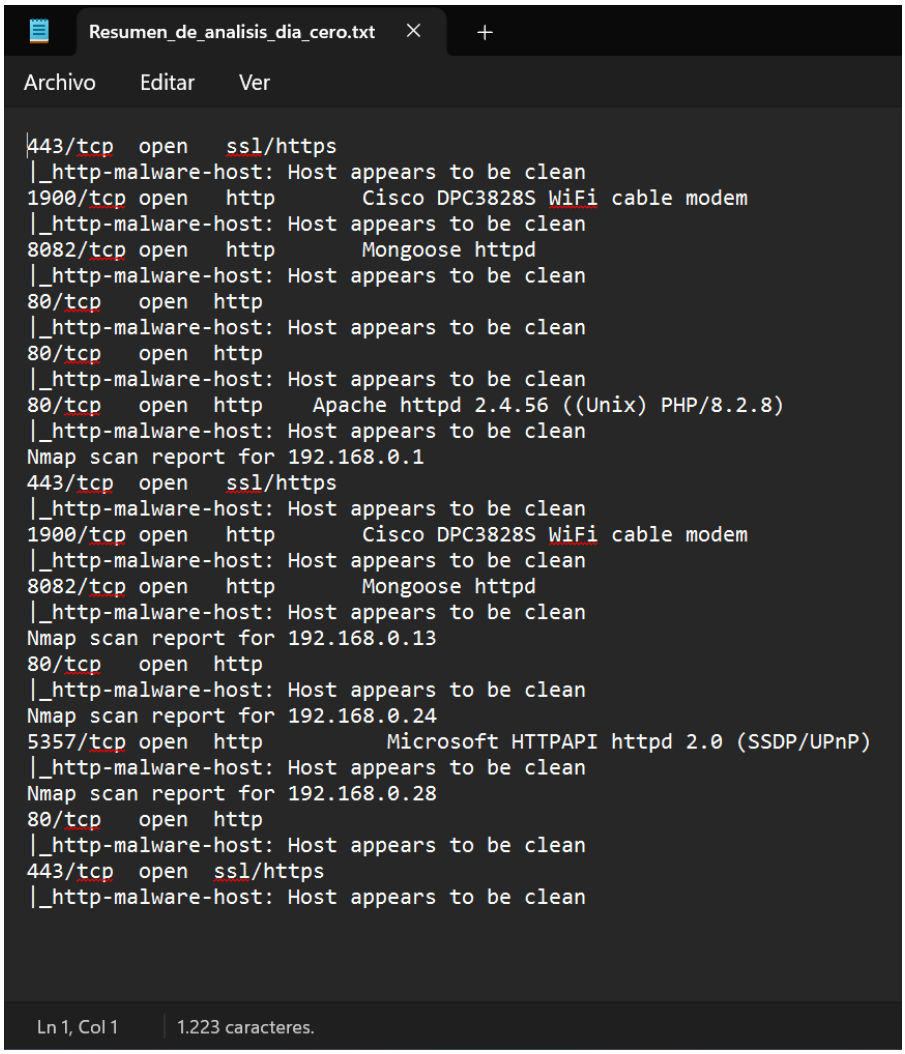

Ilustración 32 Resumen de Análisis de día cero

En las imágenes presentadas anteriormente, se puede observar que el análisis inicial de un segmento de la red ha sido exitoso (Ilustración 31 Análisis de día cero sin haber sido resumido) de manera que se puede recabar un resumen de este presentado en la (Ilustración 32 Resumen de Análisis de día cero). Cabe destacar que este análisis se realizó de manera parcial, abarcando únicamente un segmento de la red y no la totalidad de esta. Esto se logró mediante la utilización de la herramienta Nmap, la cual redirigió el tráfico de red hacia VirusTotal, permitiendo realizar un escaneo exhaustivo de cada servicio proveniente de los dispositivos presentes en dicho segmento[24]. En esta etapa preliminar, no se detectaron amenazas significativas. Sin

[24] Es necesario recordar que VirusTotal evalúa todo tipo de archivo basándose en los motores de antivirus de marcas empresariales reconocidas, como Panda, Norton, Avast, ESET, entre otras.

embargo, es importante mencionar que, a lo largo de los días posteriores, el administrador del sistema decidió ampliar el análisis a la red completa. Como resultado de este escaneo más extenso, se identificó la presencia de malware dentro de la red, lo cual requirió la implementación de medidas adicionales de mitigación y protección.

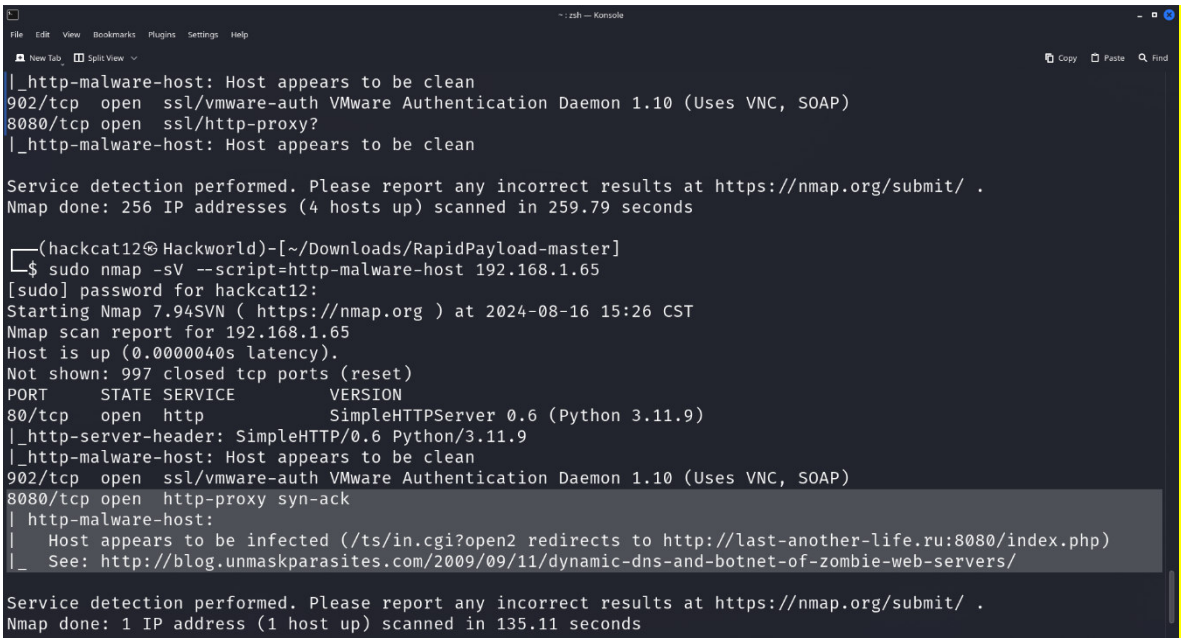

Ilustración 33 Detección de malware al escanear la red

Tras detectar la presencia de malware en la red, se procedió a evaluar la amenaza de manera exhaustiva. Como resultado de esta evaluación, se determinó que el malware se originó debido a la instalación de un software no autorizado en uno de los dispositivos de la red. Dicho software había sido obtenido a través de un "crack" o versión pirata, lo que facilitó la infección del sistema. En consecuencia, la empresa tomó medidas disciplinarias y correctivas en relación con el dispositivo comprometido y su usuario, incluyendo una capacitación enfocada en concientizar sobre los riesgos asociados con el uso de software no autorizado y las prácticas de seguridad adecuadas.

Una vez que se confirmó la eliminación del malware y se aseguró que no existían amenazas persistentes en la red, el administrador procedió a activar el análisis de host de "día cero". Este proceso tenía como objetivo adquirir información detallada

sobre el tráfico de red en un estado inicial libre de amenazas conocidas. Los datos recopilados durante esta etapa se resguardaron en los archivos previamente mencionados (sniffing.csv y count.csv) (Ilustración 34 Archivos del IDS), los cuales servirían como línea base para futuras comparaciones y análisis dentro del Sistema de Detección de Intrusos (IDS) propuesto.

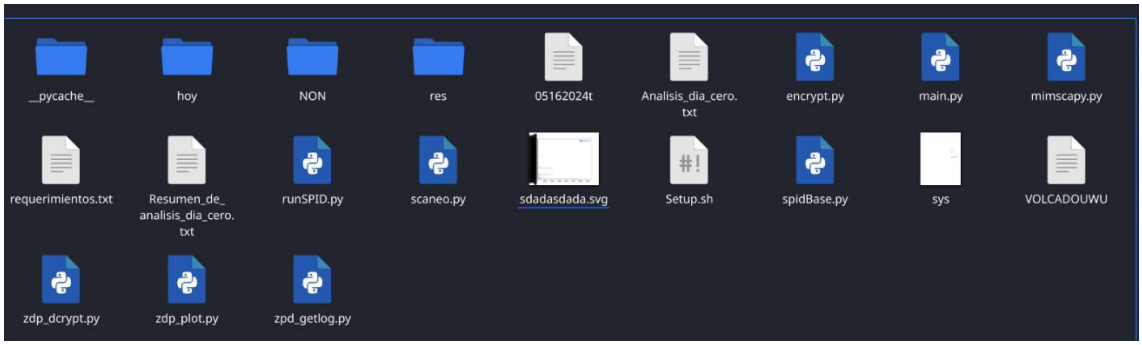

Ilustración 34 Archivos del IDS

Al examinar los archivos generados durante el proceso de análisis de "día cero", se pudo constatar que el programa desarrollado efectivamente almacena los datos en bruto recopilados del tráfico de red en el archivo denominado "sniffing.csv". Este archivo contiene la información sin procesar, incluyendo detalles sobre los paquetes de red capturados, sus cabeceras y cargas útiles.

La preservación de los datos en su forma original es fundamental para garantizar la integridad y la trazabilidad del proceso de análisis. Contar con un registro detallado del tráfico de red en su estado inicial permite establecer una línea base confiable para futuras comparaciones y detecciones de anomalías o actividades sospechosas. Además, el almacenamiento de estos datos brutos facilita la realización de auditorías y la revisión de los resultados obtenidos, lo cual es esencial para validar el correcto funcionamiento del Sistema de Detección de Intrusos (IDS) y asegurar la confiabilidad de sus hallazgos

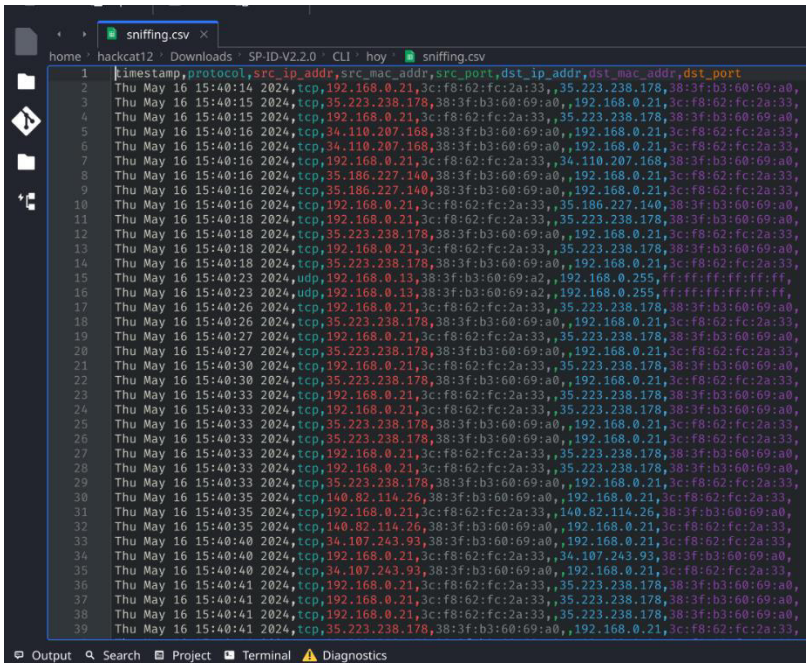

Ilustración 35 Registro de Paquetes en bruto en archivo "sniffinf.csv"

Además de almacenar los datos en bruto (Ilustración 35 Registro de Paquetes en bruto en archivo "sniffinf.csv"), el programa aplicó técnicas de minería de datos, como se mencionó anteriormente, para agrupar y procesar la información recopilada. Este proceso generó un archivo denominado "count.csv" (Ilustración 36 Registro de Alertas del algoritmo base), el cual contenía listas reducidas y resumidas de los datos originales. Estas listas condensadas facilitaron su posterior uso en la detección de cambios y anomalías en el tráfico de red, tanto para el análisis de "día cero" como para los análisis en días posteriores. [Insertar imagen del archivo "count.csv" aquí]

La imagen mostró un extracto del archivo "count.csv", donde se pudo observar cómo los datos se agruparon y resumieron en forma de conteos y estadísticas. Cada fila representaba una entrada específica, ya fuera una dirección IP, un puerto, un protocolo o cualquier otro elemento relevante para el análisis del tráfico de red.

Después de transcurrir 24 horas desde la activación del análisis de "día cero", el programa comenzó a comparar los resultados del día anterior con los del día actual. Este proceso tuvo como objetivo identificar posibles anomalías o desviaciones en el comportamiento normal del tráfico de red. Durante esta fase de prueba, el administrador realizó el análisis de "día cero" en un segmento de red y, al día

siguiente, se conectó a otro segmento diferente. Esto generó algunas alertas erróneas, las cuales, si bien no representaban amenazas reales, sirvieron como ejemplo ilustrativo de cómo se visualizarían las alertas en caso de detectarse actividades sospechosas genuinas.

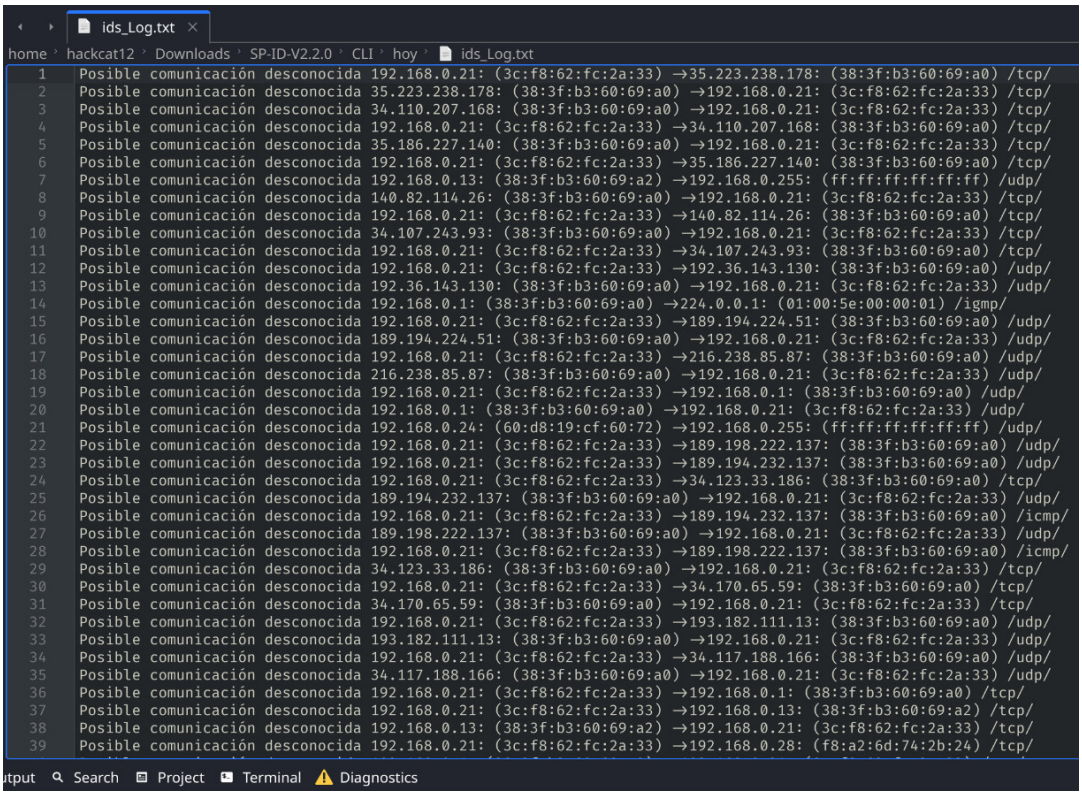

Ilustración 36 Registro de Alertas del algoritmo base

Capítulo 10: Conclusiones y Trabajo Futuro

Conclusiones

Las técnicas de minería de datos empleadas pueden ser implementadas de diversas formas al ser programadas, lo que permite ajustar el gasto de recursos y el grado de efectividad del sistema. Esta flexibilidad se logra principalmente a través del diseño y la optimización de los algoritmos subyacentes, los cuales definen en gran medida el rendimiento y la eficiencia del IDS.

Cabe mencionar que, si bien la implementación de seguridad en archivos (encriptamiento de archivos utilizados) no formaba parte de los objetivos iniciales de esta tesis, su incorporación resultó beneficiosa. Dado que el IDS consume y analiza tráfico de red, el dispositivo en el que se ejecuta podría ser un objetivo potencial para ciberdelincuentes. Por lo tanto, se implementaron medidas de seguridad adicionales, como el encriptamiento de archivos, para proteger la integridad del sistema, especialmente al considerar que este IDS estaría formando parte de un sistema complejo como son las redes de mediano rango.

Es importante señalar que el programa desarrollado aún requiere la atención y supervisión de un administrador de TI, ya que no es capaz de tomar decisiones de forma autónoma. Esta característica representa un área de mejora para futuras iteraciones del sistema, con el objetivo de lograr un mayor grado de autonomía y automatización en la detección y respuesta a posibles intrusiones.

Trabajo futuro

En el transcurso del desarrollo de esta tesis, se identificaron diversas oportunidades de mejora que podrían potenciar el sistema propuesto y ampliar su alcance. A continuación, se describen algunas de las principales áreas de mejora:

- Mejora de la experiencia de usuario: Durante la realización de este trabajo, la interfaz gráfica de usuario (GUI) se desarrolló con el objetivo principal de facilitar la validación práctica del sistema y cumplir con los objetivos específicos de la tesis. Sin embargo, no se siguieron estrictamente las normas y buenas prácticas de usabilidad y experiencia de usuario para un público general. Por lo tanto, una mejora sustancial sería el rediseño y la optimización de la GUI, incorporando principios de diseño centrado en el usuario y ergonomía, con el fin de brindar una experiencia más intuitiva y amigable para los administradores de sistemas.
- Implementación de Inteligencia Artificial: En el sistema actual, el administrador desempeña un papel crucial en la toma de decisiones, desde tareas básicas hasta situaciones complejas. La incorporación de técnicas de Inteligencia Artificial (IA) podría representar un avance significativo, al proporcionar un apoyo automatizado al administrador y reducir la carga de trabajo en la gestión de alertas y la toma de decisiones. La IA podría encargarse de tareas como el análisis de patrones, la correlación de eventos y la clasificación de amenazas, permitiendo al administrador centrarse en los casos más críticos que requieran su intervención.
- Incremento de la seguridad: Si bien la implementación de Inteligencia Artificial tiene el potencial de mejorar la eficiencia y la precisión del sistema, también introduce nuevos desafíos de seguridad. Es fundamental abordar estos riesgos para garantizar que el sistema siga cumpliendo con sus objetivos principales de protección. Algunas áreas clave a considerar son la seguridad del modelo de IA, la protección contra ataques adversarios y la preservación de la privacidad de los datos utilizados para el entrenamiento y la operación del sistema.

Estas oportunidades de mejora representan desafíos y líneas de investigación futuras que podrían ampliar el alcance y el impacto del sistema de detección de intrusos propuesto en esta tesis. Abordando estas áreas, se puede lograr un sistema más robusto, escalable y centrado en el usuario, capaz de adaptarse a las crecientes amenazas y demandas de seguridad en entornos de red complejos.

Apéndice A

Tipos de redes

Redes de área personal (PAN):

- Son las redes más pequeñas y se limitan a un área personal, generalmente dentro de unos pocos metros. Ejemplos incluyen redes Bluetooth y redes de área corporal (BAN).

Redes de Área Local (LAN - Local Area Networks):

- Son redes que cubren un área geográfica limitada, como un edificio o un campus. Los dispositivos en una LAN pueden compartir recursos y comunicarse directamente entre sí.

Redes de Área Extensa (WAN - Wide Area Networks):

- Abarcan áreas geográficas más extensas, como ciudades, países o incluso a nivel global. Las WAN utilizan tecnologías como líneas telefónicas, satélites o conexiones de fibra óptica para conectar dispositivos distantes.

1. Redes de Área Metropolitana (MAN - Metropolitan Area Networks):

- Tienen un alcance mayor que las LAN pero menor que las WAN, cubriendo una ciudad o una región metropolitana. Suelen utilizar tecnologías de alta velocidad como la fibra óptica.

2. Redes de Almacenamiento (SAN - Storage Area Networks):

- Diseñadas específicamente para la transferencia eficiente de datos entre sistemas de almacenamiento y servidores. Son comunes en entornos empresariales para manejar grandes volúmenes de datos.

3. Redes de Acceso (Access Networks):

- Se refieren a las redes que conectan dispositivos finales (como computadoras, teléfonos, etc.) a la red central. Ejemplos incluyen redes de acceso a Internet DSL, cable o inalámbricas.

4. Redes de Interconexión (Internetworks):

- Se componen de múltiples redes individuales interconectadas mediante dispositivos como enrutadores y conmutadores para formar una red más grande, como Internet.

5. Redes Inalámbricas (Wireless Networks):

- Utilizan ondas de radio o señales de infrarrojos para la comunicación en lugar de cables físicos. Incluyen WLAN (Wireless LAN) y tecnologías móviles como 4G y 5G.

6. Redes Peer-to-Peer (P2P):

- En estas redes, los nodos tienen roles similares y pueden actuar tanto como clientes como servidores. Se utilizan comúnmente en entornos donde no hay una infraestructura centralizada.

(Wetherall, 2011)

Dispositivos de una red

En el libro "Computer Networking: Principles, Protocols and Practice" de Olivier Bonaventure, se aborda la variedad de dispositivos que conforman una red de computadoras. Estos dispositivos desempeñan roles específicos para permitir la comunicación y el intercambio de datos en la red. Como lo son los siguientes:

Computadoras (Hosts):

- Las computadoras individuales, ya sea una PC, una estación de trabajo o un servidor, son dispositivos finales que generan o consumen datos en la red.

Router:

- Un router es un dispositivo que conecta diferentes redes y dirige el tráfico entre ellas. Juega un papel crucial en la interconexión de redes y en la toma de decisiones sobre cómo enviar datos de un lugar a otro.

Switch:

- Los switches se utilizan para conectar varios dispositivos en una red local (LAN). A diferencia de los hubs, los switches pueden tomar decisiones inteligentes sobre el envío de datos solo a los dispositivos específicos que necesitan la información.

Hub:

- Aunque menos comunes en redes modernas, los hubs simplemente repiten los datos a todos los dispositivos conectados a ellos. No toman decisiones inteligentes sobre el enrutamiento de datos.

Access Point (Punto de Acceso):

- Utilizado en redes inalámbricas (Wi-Fi) para proporcionar conectividad a dispositivos inalámbricos, como computadoras portátiles, tabletas y teléfonos móviles.

Modem:

- Convierte señales digitales de una computadora en señales analógicas que pueden transmitirse a través de líneas telefónicas o cables coaxiales, y viceversa. Se utiliza comúnmente para la conexión a Internet.

Firewall:

- Un firewall se encarga de gestionar el tráfico de red según ciertas reglas de seguridad. Puede ser hardware o software y ayuda a proteger la red contra amenazas externas.

Servidores:

- Estos son dispositivos dedicados que proporcionan servicios específicos a la red, como servidores de archivos, servidores web, servidores de correo electrónico, entre otros.

Repetidores y Extensores:

- Se utilizan para amplificar o extender la señal en redes inalámbricas, especialmente en grandes áreas donde la señal puede debilitarse.

Tipos de protocolos de paquetes en una red

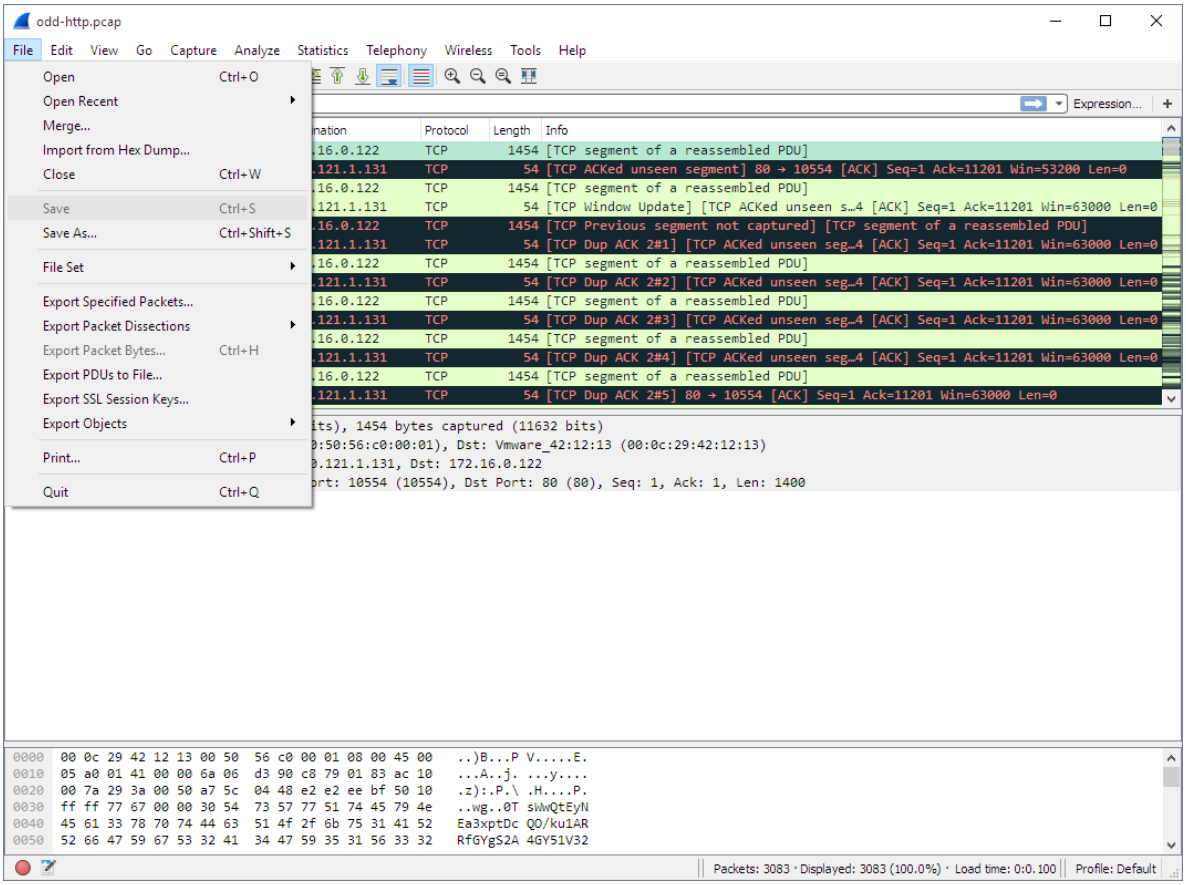

Ilustración 37 Captura de pantalla del funcionamiento de Wireshark[25]

1. TCP (Transmission Control Protocol): TCP es un protocolo orientado a conexión que proporciona un servicio de entrega confiable de datos. Garantiza que los paquetes se envíen y reciban en el orden correcto y sin duplicados. Establece una conexión punto a punto antes de transmitir datos.
2. UDP (User Datagram Protocol): UDP es un protocolo no orientado a conexión y no confiable. No garantiza la entrega de paquetes ni el orden de estos. Se utiliza en aplicaciones en tiempo real donde la entrega rápida es más importante que la fiabilidad.

[25] "https://www.wireshark.org/"

3. IP (Internet Protocol): IP es el protocolo principal de la suite de protocolos TCP/IP. Se encarga de enrutar y entregar los paquetes de datos a través de una red. Proporciona una entrega de "mejor esfuerzo" sin garantías.
4. ICMP (Internet Control Message Protocol): ICMP es un protocolo de mensajes de control y error utilizado para enviar informes de error en redes IP. Por ejemplo, informar sobre una ruta inaccesible o un tiempo de espera agotado.
5. IGMP (Internet Group Management Protocol): IGMP se utiliza para gestionar las comunicaciones de multidifusión en redes IP. Permite que un host se una o abandone un grupo de multidifusión.
6. ARP (Address Resolution Protocol): ARP mapea las direcciones IP a direcciones físicas de la capa de enlace de datos. Es necesario para la comunicación dentro de una red local.
7. RARP (Reverse Address Resolution Protocol): RARP realiza la función opuesta a ARP, mapea direcciones físicas a direcciones IP. Se utiliza cuando un host no conoce su dirección IP.
8. DHCP (Dynamic Host Configuration Protocol): DHCP es un protocolo que permite a los hosts obtener automáticamente una dirección IP y otros parámetros de configuración de red de un servidor DHCP.
9. DNS (Domain Name System): DNS es un sistema de nomenclatura jerárquico que traduce los nombres de dominio legibles por humanos a direcciones IP numéricas.
10. FTP (File Transfer Protocol): FTP es un protocolo estándar para transferir archivos de manera confiable entre sistemas conectados a una red. Permite a los usuarios cargar, descargar y administrar archivos en servidores remotos.
11. SMTP (Simple Mail Transfer Protocol): SMTP es un protocolo utilizado para la transferencia confiable de correo electrónico a través de Internet. Define la manera en que los mensajes de correo son enviados de un cliente a un servidor de correo y enviados de un servidor a otro.

12. HTTP (Hypertext Transfer Protocol): HTTP es el protocolo subyacente de la World Wide Web, utilizado para la transferencia de documentos hipermedia (páginas web) a través de Internet. Define cómo los clientes (navegadores web) solicitan recursos de los servidores web y cómo los servidores responden a esas solicitudes.
13. PPP (Point-to-Point Protocol): PPP es un protocolo de enlace de datos utilizado para establecer una conexión directa entre dos nodos. Se usa comúnmente para conectar hosts a un proveedor de servicios de Internet a través de una línea telefónica o conexión dedicada.
14. SLIP (Serial Line Internet Protocol): SLIP es un protocolo antiguo de enlace de datos que permitía a los hosts acceder a Internet a través de una línea serial. Fue uno de los primeros protocolos utilizados para conectar computadoras a Internet, pero ha sido reemplazado en gran medida por PPP.

(Bonaventure, October 31 (2011))

Bibliografía

Anderson, R. (1 Abril 2008). *Security Engineering: A Guide to Building Dependable Distributed Systems.* Wiley.

Andrew S. Tanenbaum, H. B. (2009). *Sistemas Operativos Modernos.* Pearson Educacion.

Angela Orebaugh, S. B. (2005). *Snort Cookbook.* O'Reilly Media. Obtenido de https://www.oreilly.com/library/view/snort-cookbook/0596007914/

Bace, R. G. (2000). *Intrusion Detection.* Sams Publishing.

Bahl, P. y. (2021). *Evolution of malware and its detection techniques: A survey.* IEEE.

Bauer, M. D. (2015). *Linux Server Security .* Sebastapol, CA O'Reilly .

Bhuyan, M. H. (2014). *Network anomaly detection: methods, systems and tools.* IEEE Communications Surveys & Tutorials. Obtenido de https://ieeexplore.ieee.org/document/6524462

Bonaventure, O. (October 31 (2011)). *Communications and Networks.* The Saylor.

Chen, J. W. (2021). *MATIC: API sequence driven malware detection via natural language processing.* IEEE.

Cohen, F. (1984). *Computer Viruses - Theory and Experiments.* Computers & Security. Obtenido de https://doi.org/10.1016/0167-4048(87)90122-2

Dash Bahl, P. y. (2021). *volution of malware and its detection techniques: A survey.* IEEE.

David A. Patterson (Autor), J. L. (2013). *Computer Organization and Design MIPS Edition: The Hardware/Software Interface.* Mk Morgan Kaufmann.

Department of Computer Science & Information Technology, A. P. (Abril 25 (2015)). *Intrusion detection of masqueraders using data mining and soft computing techniques.* Journal of Computing Sciences in Colleges.

Diehl, S. H. (2021). *Network Security with Suricata.* Technische Universität München University Press.

Dileep Kumar G, M. K. (2021). *Network Security Attacks and Countermeasures.* IGI Global.

Easttom, D. C. (2021). *Computer Security Fundamental.* Pearson IT Cybersecurity Curriculum (ITCC).

Erickson, J. (2008). *Hacking: The Art of Exploitation.* No Starch Press.

Fitzpatrick, D. M. (2022). *Create GUI Applications with Python & Qt5 (5th Edition, PyQt5): The hands-on guide to making apps with Python.* Martin Fitzpatrick.

Foundation, R. P. (2024). *¿Qué es una Raspberry Pi?* Raspberry Pi Foundation. Obtenido de https://www.raspberrypi.com/help/what-%20is-a-raspberry-pi/

Fyodor, G. L. (2022). *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning.* Nmap.org. Obtenido de https://nmap.org/

Greg Gagne, A. S. (2018). *Operating System Concepts.* Wiley.

Gupta, B. B. (2022). *Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives.* CRC Press.

Jones, D. B. (2013). *Python Cookbook: Recipes for Mastering Python 3.* O´REILLY.

Kathiravelu, A. R. (2021). *Python Network Programming.* Packt.

Khrais, H. (2019). *Python for Offensive PenTest: A practical guide to ethical hacking and penetration testing using Python .* Packt.

Kumar, D. (2021). *Network Security Attacks and Countermeasures.* Springer Nature.

McKinney, W. (2018). *Python for Data Analysis.* Jupyter.

MichaelTu, R. A. (2020). *Cyber Security Tool Kit (CyberSecTK): A Python Library for Machine Learning and Cyber Security.* Purdue University Northwest.

Ramalho, L. (2015). *Fluent Python.* SPD.

Rash, A. O. (2005). *Intrusion Prevention and Active Response: Deploying Network and Host IPS.* Syngress Publishing.

Róbert Szabó, H. Z. (2010). *Access Networks .* Springer.

Sahay, A. S. (2021). *Evolution of Malware and Its Detection Techniques.* IEEE.

Seitz, J. (2021). *Black Hat Python: Python Programming for Hackers and Pentesters.* No Starch Press.

Sharma, A. &. (2012). *A survey of modern techniques for intrusion detection systems.* International Journal of Computer Science & Engineering Survey. Obtenido de https://www.airccse.org/journal/ijcses/papers/3412ijcses05.pdf

Stavroulakis, P. &. (2020). *Handbook of Information and Communication Security.* Pearson.

Sultana, N. C. (2021). *A survey on hybrid intrusion detection techniques.* International Journal of Computational Intelligence Systems. Obtenido de https://link.springer.com/chapter/10.1007/978-981-15-7527-3_77

Sweigart. (2019). *Automate the Boring Stuff with Python.* Sweigart .

Tsai, C. F. (2009). *Intrusion detection by machine learning: A review.* Expert Systems with Applications. Obtenido de https://tarjomefa.com/wp-content/uploads/2017/11/8000-English-TarjomeFa.pdf

Ullman, J. L. (2020). *Mining of Massive Datasets.* A Rajaraman.

VanderPlas, J. (2016). *Python Data Science Handbook.* Jupyter.

Welsh, J. (2021). *Hacking With Python: The Complete and Easy Guide to Ethical Hacking, Python Hacking, Basic Security, and Penetration Testing - Learn How to Hack Fast!* Joshua Welsh.

Wetherall, A. S. (2011). *Computer networks.* Prentice Hall.

Zuech, R. K. (2015). *Intrusion detection and big heterogeneous data: a survey.* Journal of Big Data. Obtenido de https://link.springer.com/article/10.1186/s40537-015-0013-4?utm_source=xmol&utm_medium=affiliate&utm_content=meta&utm_campaign=DDCN_1_GL01_metadata