ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA TELECOMUNICACIONES Y REDES

"ANÁLISIS DE TRÁFICO DE DATOS MALICIOSO MEDIANTE TECNICAS MACHINE LEARNING, UTILIZANDO OPNids EN LA RED DEL EDIFICO DE LA FIE"

TRABAJO DE TITULACIÓN

TIPO: PROPUESTA TECNOLÓGICA

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA TELECOMUNICACIONES Y REDES

AUTOR: JONATHAN PATRICIO HERRERA SILVA

DIRECTOR: Ing. ALBERTO LEOPOLDO ARELLANO AUCANCELA

Riobamba-Ecuador

2021

Yo, Jonathan Patricio Herrera Silva, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor (a) asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación. El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 06 de mayo de 2021.

Jonathan Patricio Herrera Silva

060408960-7

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMATICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES

El Tribunal del trabajo de titulación certifica que: El trabajo de investigación: Propuesta tecnológica "ANALISIS DE TRAFICO DE DATOS MALICIOSO MEDIANTE TECNICAS MACHINE LEARNING, UTILIZANDO OPNids EN LA RED DEL EDIFICIO DE LA FIE", de responsabilidad del señor JONATHAN PATRICIO HERRERA SILVA, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, queda autorizado su presentación.

	FIRMA	**FECHA**
Ing. Pedro Severo Infante Moreira **PRESIDENTE DEL TRIBUNAL**	Firmado electrónicamente por: PEDRO SEVERO INFANTE MOREIRA	06 de mayo 2021
Ing. Alberto Leopoldo Arellano Aucancela **DIRECTOR DEL TRABAJO DE TITULACIÓN**	ALBERTO LEOPOLDO ARELLANO AUCANCELA Firmado digitalmente por ALBERTO LEOPOLDO ARELLANO AUCANCELA Fecha: 2021.05.12 12:05:44 -05'00'	06 de mayo 2021
Ing. Diego Fernando Veloz Cherrez **MIEMBRO DEL TRIBUNAL**	Firmado electrónicamente por: DIEGO FERNANDO VELOZ CHERREZ	06 de mayo 2021

DEDICATORIA

Dedico esta investigación a Dios, por brindarme su fuerza para alcanzar mi objetivo, a mi familia; mis padres y hermano por ser mi pilar fundamental y razón por la cual este largo y duro camino fue superado paso a paso, su trabajo y sacrificio espero recompensarlos de esta manera. Es para mí un orgullo y privilegio el ser su hijo, finalmente a todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito.

Jonathan

AGRADECIMIENTO

Me van a faltar páginas para agradecer a las personas que se han involucrado en la realización de este trabajo, sin embargo, merecen reconocimiento especial mis padres que con su esfuerzo y dedicación me ayudaron a culminar mi carrera universitaria y me dieron el apoyo suficiente para no decaer cuando todo parecía complicado e imposible.

A mis amigos, *Kevin Ortega*, José Rodas, Carlos Estévez, Bryan Trujillo, Jair Vera, Valeria Hernández y Genesis Aldas que me han apoyado a lo largo de mi vida universitaria y fuera de ella, gracias infinitas por toda su ayuda y buena voluntad sin su apoyo no lo habría logrado.

Jonathan

TABLA DE CONTENIDO

CAPÍTULO II

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE GRÁFICOS

ÍNDICE DE ANEXOS

RESUMEN

El presente trabajo de titulación tiene como objetivo analizar el tráfico de datos malicioso mediante la plataforma OPNids que consta con un motor de machine learning para la detección de ataques informáticos en la red del edificio de la Facultad de Informática y Electrónica (FIE) de la Escuela Superior Politécnica de Chimborazo (ESPOCH). Para cumplir con los objetivos propuestos se realizó un análisis del tráfico de datos de la FIE con el IDS Suricata integrado en OPNids y se detectó las amenazas y potenciales ataques informáticos a los que se encuentra expuesta la intranet de la FIE. Se realizó el estudio de sobre la teoría del machine learning aplicado a la detección de intrusos con sus diferentes algoritmos de clasificación entre los cuales destaca Random Forest un algoritmo con características de gran precisión para clasificar diferentes tipos de ataques, a continuación, se llevó a cabo el proceso de entramiento del algoritmo para ajustarlo a las necesidades de la intranet de la FIE tomando en cuenta las amenazas detectadas en el análisis del IDS Suricata. Posteriormente se realizó la simulación en GNS3 de las amenazas detectadas en un escenario similar a un laboratorio de la FIE y para finalizar se evaluó el desempeño y efectividad de la plataforma OPNids en la detección de intrusos a través de ataques informáticos del tipo DoS, R2L, U2R y de escaneo similares a los detectados en el análisis de tráfico de datos en conjunto con los parámetros obtenidos del algoritmo de machine learning. Obteniendo como resultado una precisión y puntaje F1 del 98% del modelo de machine learning. Para finalizar se recomienda realizar pruebas con otros algoritmos de clasificación y comprobar diferencia de detección de intrusos entre ellos, además de verificar el soporte de la plataforma OPNids.

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <PLATAFORMA OPNIDS>, <MACHINE LEARNING>, <SISTEMA DE DETECCION DE INTRUSOS>, <SURICATA IDS>, <GNS3 (SOFTWARE)>, <RANDOM FOREST>, <PRECISION>.

ABSTRACT

This research aims to analyze malicious data traffic through the OPNids platform made of a machine learning engine in charge of detecting computer attacks on the Informatics and Electronics Faculty (FIE) network at Escuela Superior Politécnica de Chimborazo (ESPOCH). An analysis of the Faculty's data traffic was carried out to meet the proposed objectives by using the IDS Suricata integrated with OPNids, and the threats and potential computer attacks to which the Faculty's intranet is exposed were detected. A study based on the machine learning theory applied to detecting intrusions with its different classification algorithms was carried out, being Random Forest, an algorithm with high precision characteristics to classify different types of attacks. Next, the algorithm training process was conducted to adjust it to the FIE intranet's needs, taking into account the IDS Suricata analysis threats. Subsequently, the GNS3 simulation based on threats detected was performed in a scenario similar to a faculty laboratory. Finally, the performance and effectiveness of the OPNids platform in detecting intrusion through computer attacks such as DoS, R2L, U2R, and scanning similar to those detected in the data traffic analysis were evaluated in conjunction with the parameters obtained from the machine learning algorithm. As a result, a precision and F1 score of 98% of the machine learning model was obtained. It is recommended to carry out tests with other classification algorithms and check the difference in intrusion detection between them and verify the OPNids platform's support.

Keywords: < ENGINEERING SCIENCE AND TECHNOLOGY>, <OPNIDS PLATFORM>, <MACHINE LEARNING>, <INTRUSION DETECTION SYSTEM>, <SURICATA IDS>, <GNS3 (SOFTWARE)>, <RANDOM FOREST>, <PRECISION>.

INTRODUCCIÓN

Con el paso del tiempo, tanto los medios de almacenamiento de información como los medios para proteger la misma han ido evolucionando. En el pasado la forma de proteger los datos era colocándolos en una habitación con seguridad física, es decir, candados, alarmas, sensores y demás mecanismos de seguridad, con una persona encargada de vigilar que nadie sin permiso accediera a dichos datos. Pero con la proliferación de las Tecnologías de la Información y Comunicación (TIC), la información pasó de archivadores físicos, a estar en bases de datos digitales, más eficientes, robustas y manipulables desde cualquier lugar del mundo. Además, en la actualidad la información debe ser accesible para todos los usuarios con permisos simultáneamente. Ante esta perspectiva, la seguridad de la información tuvo que desarrollarse a la par que la evolución tecnológica, motivada porque la información enviada a través de internet se tornaba vulnerable ante cualquier ataque informático existiendo el riesgo de exposición de los datos confidenciales. (Carazo, 2017, pp. 5–6)

La evolución de las telecomunicaciones dio como resultado un mundo interconectado y por ende un crecimiento acelerado de todo tipo de servicio en línea; las empresas, organizaciones, bancos, universidades, entre otros, han facilitado al usuario un sin número de aplicaciones en internet que generan gran cantidad de tráfico de datos con información de todo tipo. Uno de los activos más importantes que poseen las universidades es la información. Sin embargo, no necesariamente se cuenta con políticas y controles adecuados para protegerla, generando vulnerabilidades que pueden ser aprovechadas por las amenazas existentes en el entorno y por ende afectar a la integridad, confidencialidad y disponibilidad de la información de la comunidad universitaria. (Reyes et al, 2019, pp.13)

Para las universidades el tener conocimiento sobre los incidentes que se pueden causar aprovechando las vulnerabilidades de sus sistemas es muy importante para poder determinar e implementar medidas preventivas y correctivas para la seguridad de sus sistemas, estas medidas se agrupan y se forman políticas de seguridad de la información las cuales se deben aplicar en la organización. Sin embargo, según un estudio realizado por CEDIA en 2018 (Padilla R., et al, 2019, pp. 80-83) revela que, de un conjunto de universidades encuestadas, únicamente 11 cuentan con políticas de seguridad aprobadas representando el 26%, 24 de manera parcial representando el 57%, mientras que 7 no tiene una política de seguridad representando el 17%. cabe indicar que en este estudio se encuestaron 46 universidades ecuatorianas entre las cuales se encontraba la Escuela Superior Politécnica de Chimborazo (ESPOCH). Este estudio también revela que no se realizan auditorias de seguridad, planes de contingencia y que solo 8 de cada 10 poseen herramientas para análisis de vulnerabilidades. Entre los principales se presenta la implementación de Sistemas de Detección de Intrusos (IDS).

Como no existe un sistema 100% seguro, en la actualidad se ha venido desarrollando y adoptando nuevas técnicas informáticas y estadísticas para la seguridad informática como el machine learning aplicado a la ingeniería con el objetivo de mejorar los tiempos de detección de intrusos. Machine Learning con sus algoritmos de aprendizaje supervisado y no supervisado pretenden mejorar la detección de intrusos en la red y disminuir el porcentaje de falsos positivos generados por las alertas comunes en los IDS. Además de incrementar la rapidez de detección de un posible ataque a través del entrenamiento de sus algoritmos de clasificación por medio de flujos generados por el tráfico de la red. Machine Learning nace de la inteligencia artificial y es aplicado en varias ciencias con gran éxito y en la ciberseguridad no puede ser la excepción, los algoritmos con un correcto entrenamiento pueden llegar a complementar la seguridad informática y automatizar procesos para que el administrador de red no tenga la necesidad de revisar una a una las alertas de posibles ataques. Si una intranet no posee al menos una de estas opciones puede quedar expuesta por falta de políticas de seguridad como: Antivirus, credenciales temporales, socialización de medidas con el personal, control de acceso, entre otros.

ANTECEDENTES

Arthur Samuel, visionario en el área de los juegos de computadora y la inteligencia artificial, asoció el término "Aprendizaje automático" o ML por sus siglas en inglés, en 1959. (Samuel, 1959, pp. 211-212) El ML nació en conjunto con la inteligencia artificial (IA). Los investigadores querían que las computadoras "aprendieran" de los datos. Pero dicho proceso conllevó a largos análisis y estudios de alta complejidad con la utilización de métodos simbólicos, llegando a definir las "redes neuronales"; estos fueron en su mayoría perceptrones y otros modelos que fueron adaptados de los modelos lineales de estadística. (Sarle, 1994, pp. 1) Sin embargo, un estudio en el enfoque lógico y basado en el conocimiento causó la división entre la IA y el ML. Los sistemas probabilísticos estaban repletos de errores teóricos y prácticos de adquisición y representación de datos. (Russell y Norvig, 2009, pp. 5–8) Desde 1995 hasta 2005, existió mucho realce en el uso del lenguaje natural, la búsqueda y la recuperación de información. Las herramientas de aprendizaje automático eran más simples que las utilizadas en la actualidad; están compuestos por procesos como regresión logística, SVM (máquinas de vectores de soporte), núcleos con SVM y PageRank. Google fue el pionero en hacer uso de dicha tecnología, implementando ML en procesos como Google News y Gmail. (Bosagh Zadeh, 2008, pp. 2)

Varias empresas desde 1980 han desarrollado varios avances como Microsoft que desarrollo *Kinect* un dispositivo compuesto por una cámara y sensores capaz de reconocer 20 características humanas a una velocidad de 30 v/s, permitiendo una interacción con la computadora a través de movimientos y gestos, Google con el laboratorio X desarrolla un algoritmo de ML para navegar

e identificar videos de YouTube con datos, todo de forma autónoma, Facebook desarrolla su poderoso algoritmo *DeepFace* utilizado para reconocimiento facial en las fotos de la red social a un nivel de exactitud como si lo hiciera un humano por su parte Amazon lanzó su plataforma de aprendizaje automático. (Marr, 2016)

Con el conocimiento de los párrafos anteriores, el avance tecnológico abordó todo nivel social buscando mejorar la calidad de vida de los usuarios, pero toda acción tiene una reacción y provocó el interés de personas con altos conocimientos a realizar ataques informáticos y nuevas modalidades delictivas en contra de la información de los usuarios, por esta razón todo dispositivo conectado a internet es potencialmente una víctima. (Mieres, 2009, pp. 3) Para lograr mitigar el impacto causado por los ataques informáticos, se debe estudiar y conocer la forma en la que se realiza un ataque y determinar las debilidades de un sistema explotado en los que se deben enfocar los esfuerzos de seguridad tendientes a la prevención de estos. (Mieres, 2009, pp. 3) Durante los últimos años, internet ha sido testigo de un aumento considerable en el tráfico malicioso, como el generado por cualquier ataque en mayor o menor medida; entre ellos podemos mencionar la denegación de servicio (DoS), la propagación del tráfico de gusanos y malware. Se define el tráfico normal como el tráfico de red generado por los servicios y aplicaciones conocidos, por ejemplo, web, ftp, nntp y smtp. Los ataques DDoS causan que las latencias de DNS aumenten en un 230% y que las latencias de la web aumenten en un 30%. (Lan et al., [sin fecha], pp. 1)

La Escuela Superior Politécnica de Chimborazo registra trabajos en el área de seguridad de redes, pero con poca relación al tema propuesto, por esta razón la presente investigación destaca una propuesta tecnológica para el análisis de tráfico de datos malicioso mediante técnicas machine learning, utilizando OPNids en la red del edificio de la FIE para detectar posibles ataques informáticos en la intranet.

FORMULACIÓN DEL PROBLEMA

¿Es factible utilizar OPNids con técnicas Machine Learning para mejorar la detección de ataques en la red del edificio de la FIE, mediante el análisis de tráfico de datos malicioso?

SISTEMATIZACIÓN DEL PROBLEMA

¿Es necesario analizar las técnicas actuales con el propósito de detectar tráfico malicioso?

¿Cuáles son las plataformas utilizadas para la prevención de ataques informáticos?

¿Es factible analizar la seguridad de la red del edifico de la Facultad de Informática y Electrónica (FIE)?

¿Cuáles son los parámetros que determinan que la red se encuentra bajo ataques informáticos?

JUSTIFICACIÓN DEL TRABAJO DE TITULACIÓN

JUSTIFICACIÓN TEÓRICA

En el Workshop de Investigadores en Ciencias de la Computación de 2014 se presentó la investigación *"Machine Learning aplicado en Sistemas de Detección de Intrusos"* la misma se basa en la revisión bibliográfica del estado del arte sobre el Machine Learning y/o minado de datos y el área de seguridad en sistemas. Como resultado se obtiene la implementación de los algoritmos propuestos en la investigación para validar las soluciones obtenidas mediante su aplicación a problemas de seguridad concretos en diversos entornos mediante la utilización de herramientas como software libre buscando con esto perfeccionar los algoritmos inteligentes. (Peluffo, Capobianco y Echaiz, 2014, pp. 3-4)

En Ecuador se registra la investigación *"Implementación del sistema de gestión y administración de seguridad para la Dirección de Tecnologías de la Universidad Central del Ecuador (DTIC)."*, esta investigación utilizó el sistema OSSIM una distribución Linux que provee todas las capacidades de seguridad en un SIEM de código abierto, el cual permite la recopilación de eventos, la normalización, la correlación y respuesta a incidentes. Como resultado obtuvieron que OSSIM prioriza un grupo determinado de eventos lo cual facilita enormemente la tarea del administrador de seguridad de la red informática, al permitir obtener información precisa sobre los incidentes de seguridad. (Chanaluisa, Meza y Tasipanta, 2012, pp.127-128)

La ESPOCH registra la investigación *"Modelo basado en las Técnicas de Minería de Datos aplicada a la detección de ataques en las redes de datos de la Facultad de Informática y Electrónica"* utilizando la herramienta WEKA, obtuvo como resultado de aplicar técnicas de minería de datos a un conjunto de datos previamente preparados para extraer conocimiento y generar reglas de detección de intrusos. De la implantación del modelo se obtuvo como resultado que el 86.10 % de ataques fueron detectados en menos de 8 milisegundos. Se concluye que el modelo desarrollado permite la detección oportuna de los ataques a una red de datos. (Carranza y Naranjo, 2014, pp. 177)

Sin embargo, en la presente investigación nos enfocamos al análisis de tráfico malicioso para la detección de ataques en la FIE – ESPOCH a través de la plataforma OPNids; que posee diferentes herramientas integradas entre las más importantes destacan su Sistema de Detección de Intrusos (IDS) Suricata y el motor de Machine Learning Dragonfly MLE, una tendencia creciente en la actualidad y que puede ser aplicada en la seguridad informática.

El Machine Learning o Aprendizaje automático no es una tecnología nueva, si bien en la actualidad su utilidad ha crecido de forma exponencial en casi todos los ámbitos tecnológicos y sociales debido a su gran potencial con el manejo de los datos. El ser humano siempre ha querido

dotar de conocimiento y lograr que las computadoras "aprendan" por sí solas por esta y varias razones más se relaciona al Machine Learning con la Inteligencia Artificial (IA); siendo esta una idea equivocada porque son dos ciencias complementarias.

La clasificación del tráfico en la red es un tema muy importante actualmente en el campo de la informática; es el punto de partida para analizar e identificar los diferentes tipos de aplicaciones que fluyen en una red. A través de esta técnica, los proveedores de servicios de Internet (ISP) pueden administrar el rendimiento general de una red. Hay muchos métodos tradicionales para clasificar el tráfico de Internet como la técnica basada en puertos, basada en la carga de pago y basada en el aprendizaje automático siendo esta la tendencia actual. (Shafiq et al., 2016, pp. 2451)

Un Sistema de Detección de Intrusos o IDS (Intrusion Detection System) es una herramienta de seguridad encargada de monitorizar los eventos que ocurren en un sistema informático en busca de intentos de intrusión. Definimos intento de intrusión como cualquier intento de comprometer la confidencialidad, integridad, disponibilidad o evitar los mecanismos de seguridad de una computadora o red. Las intrusiones se pueden producir de varias formas: atacantes que acceden a los sistemas desde Internet, usuarios autorizados del sistema que intentan ganar privilegios adicionales para los cuales no están autorizados y usuarios autorizados que hacen un mal uso de los privilegios que se les han asignado. (Mira, 2001, pp. 13-14)

OPNids es una plataforma o suite Open Source nacida bajo el proyecto del mismo nombre formado por las empresas Deciso y Counterflow AI con el fin de aportar hacia el desarrollo de la ciber seguridad de una manera libre ya que existen varias soluciones de seguridad, pero comerciales.

JUSTIFICACIÓN APLICATIVA

La infraestructura de red de la ESPOCH depende de la seguridad implementada por la Dirección de Tecnologías de la Información y Comunicación – DTIC y la FIE no es la excepción, la seguridad proporcionada por el DTIC está compuesta por Firewalls y DMZ configurados hacia la WAN, es decir, la intranet de la FIE está expuesta a un ataque interno (laboratorios, oficinas, etc.) por esta razón es importante implementar una solución de seguridad informática. La investigación sigue la implementación de la plataforma OPNids con sus herramientas de seguridad como IDS y motor con técnicas de Machine Learning frente a IDS´s tradicionales como Snort un IDS basado en reglas y que no posee Machine Learning.

La infraestructura física de la FIE consta de Planta Baja, 1er Piso y 2do piso, en el cual se ubica el área de servidores que posee una conexión directa desde la Dirección de Tecnologías de la Información y Comunicación (DTIC) a través de un hilo de fibra óptica a un switch Cisco WS-C3850-NM-2-10G; de este switch se distribuye a los laboratorios y oficinas administrativas que están equipadas con Switchs Cisco 2900 del edificio de la FIE. El switch Cisco WS-C3850-NM-2-10G posee varios puertos en modo Mirror, el mismo que será gestionado desde el servidor HP Proliant DL360 Gen9 para el análisis de tráfico malicioso a través de la plataforma Snort como punto de partida y OPNids.

La FIE posee 7 laboratorios distribuidos en su infraestructura física mencionada anteriormente, los laboratorios se encuentran equipados con 30 computadores cada uno, que son utilizados por los estudiantes en un horario de 2 horas por asignatura en el periodo de 7 am a 9 pm. Cuando los estudiantes utilizan los computadores generan un gran volumen de tráfico de datos que debe ser analizado en tiempo real tomando en cuenta las posibles amenazas internas que pueden existir y alertar las mismas.

Para lograr cumplir con los objetivos planteados en el presente trabajo de titulación se seguirán 3 etapas que contemplan desde el análisis de la infraestructura de red e instalación de la plataforma en el edificio de la FIE, análisis y monitoreo del tráfico de datos y entrenamiento del algoritmo de machine learning para obtención de resultados.

OBJETIVOS

OBJETIVO GENERAL

Analizar el tráfico de datos malicioso mediante técnicas Machine Learning, utilizando OPNids en la red del edificio de la FIE.

OBJETIVOS ESPECIFICOS

- Investigar las técnicas actuales de detección de tráfico malicioso en redes corporativas.
- Analizar las características de Machine Learning en la plataforma OPNids para determinar ventajas o desventajas respecto a plataformas tradicionales de detección y prevención de ataques informáticos.
- Diseño e implementación de la plataforma OPNids en la red del edificio la FIE.
- Evaluar el desempeño y efectividad de la plataforma OPNids en la detección de tráfico malicioso en la red del edifico de la FIE.

CAPÍTULO I

1. MARCO TEÓRICO

1.1. Seguridad de la información

Las organizaciones han necesitado sobrellevar la evolución tecnológica, y entre las más importantes la seguridad de la información, la cual ha sufrido dos cambios importantes en las últimas décadas. Antes de los equipos de cómputo para el procesamiento de datos, las organizaciones protegían su información valiosa a nivel físico, es decir, la seguridad de la información consistía en guardar los documentos confidenciales en armarios o bóvedas con altas seguridades y, por otro lado, por medios administrativos, como los procedimientos de protección de datos del personal que se usan durante el proceso de contratación. (Stallings, 2004, pp. 2)

Con la introducción del computador y la revolución que provoco Internet, se hizo evidente la necesidad de disponer de herramientas automatizadas para la protección de archivos y otros tipos de información almacenada en el computador. Esto ocurre especialmente en sistemas compartidos, aplicaciones y servicios; y la necesidad se acentúa en sistemas a los que se puede acceder por medio de una red telefónica pública, una red de datos o Internet. El nombre común adoptado para este conjunto de herramientas de protección de datos y evitar la intrusión de los hackers es el de seguridad informática. (Stalling, 2004, pp. 2)

En la concepción de Internet los aspectos de seguridad de la información no fueron una prioridad y con justificada razón, debido a que en un principio los servicios o aplicaciones ofrecidas eran muy limitados, simples comunicaciones cliente-servidor a nivel de redes LAN, transferencia de archivos y correo electrónico, entre otros. Con el paso de los años, la evolución tecnológica y los servicios implementados en Internet cambiaron todo el sentido de la información, desde el básico correo electrónico hasta redes sociales, ubicación (GPS), telemedicina, transacciones financieras, compras online, etc., es decir, la información ya tenía un valor más allá que ceros y unos dentro de un computador o en un archivador, se convirtió en un bien, el más valioso para una empresa u organización y por consecuencia se debe proteger.

La seguridad de la información debe estar principalmente dirigida a proteger la propiedad intelectual y la información de las organizaciones y de los usuarios. Los riesgos de la información están presentes cuando se juntan dos elementos: amenazas y vulnerabilidades. Las amenazas y vulnerabilidades están íntimamente ligadas, y no puede haber ninguna consecuencia sin la

presencia conjunta de éstas. Las amenazas deben tomar ventaja de las vulnerabilidades y pueden venir de cualquier parte, interna o externa, relacionada con el entorno de las organizaciones. (Tarazona, 2007, pp. 137)

Las vulnerabilidades son una debilidad en la tecnología o en los procesos relacionados con la información, y como tal, se consideran características propias de los sistemas de información o de la infraestructura que la contiene. En términos generales, una amenaza, es un evento que puede afectar la posibilidad de que las organizaciones o las personas puedan desarrollar sus actividades afectando directamente la información o los sistemas que la procesan. (Tarazona, 2007, pp. 137-138)

1.2. Amenazas informáticas

Las amenazas a las que un sistema de seguridad de redes de una organización está expuesto se pueden categorizar desde: desastres naturales (terremotos, tsunami), factores humanos (involuntarios, errores), fallas en los sistemas informáticos y actos maliciosos o malintencionados; que por lo general son los más abundantes en la red. (Tarazona, 2007, pp. 138)

1.2.1. Internas

Las redes internas se representan como una red LAN, así como se observa en la Figura 1-1, suponiendo que este tipo de redes son las menos susceptibles a ataques son las menos protegidas, no obstante, se debe tener en cuenta e implementar sistemas de seguridad de red.

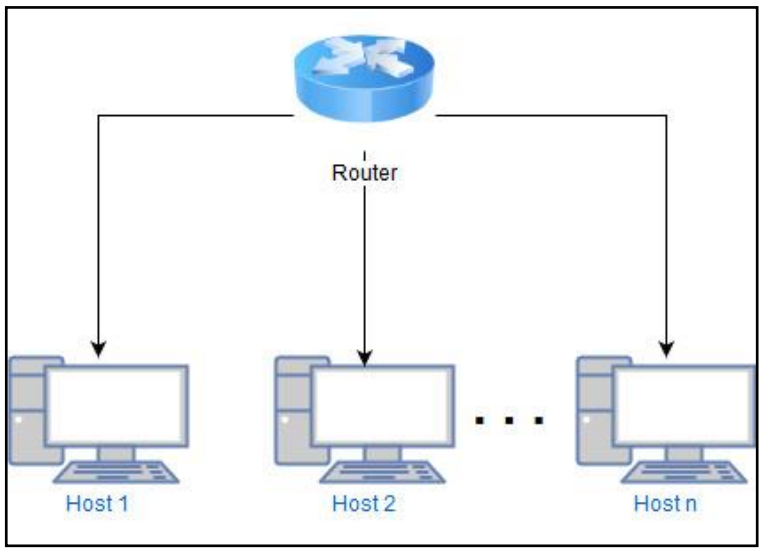


Figura 1-1: Red interna

Realizado por: Herrera, Jonathan 2021.

Las amenazas internas muchas veces pueden considerarse más perjudiciales que cualquier otra, debido a la exposición directa de toda la red a través de los usuarios cotidianos. La mayoría de las veces y de forma involuntaria un usuario que posee acceso a la red de la organización puede por desconocimiento de la política de seguridad o despiste otorgar información confidencial, generar accesos no permitidos que dejan vulnerable la seguridad de dicha red. Por otro lado, algunos ataques se producen por parte del personal que tienen la intención de sustraer la información confidencial de la empresa u organización. (Equipo de expertos, 2018)

1.2.2. Externas

Las amenazas externas se encuentran principalmente en redes que el administrador de red de una organización no tiene control, por ejemplo, una red LAN conectada a Internet, como se observa en la Figura 2-1.

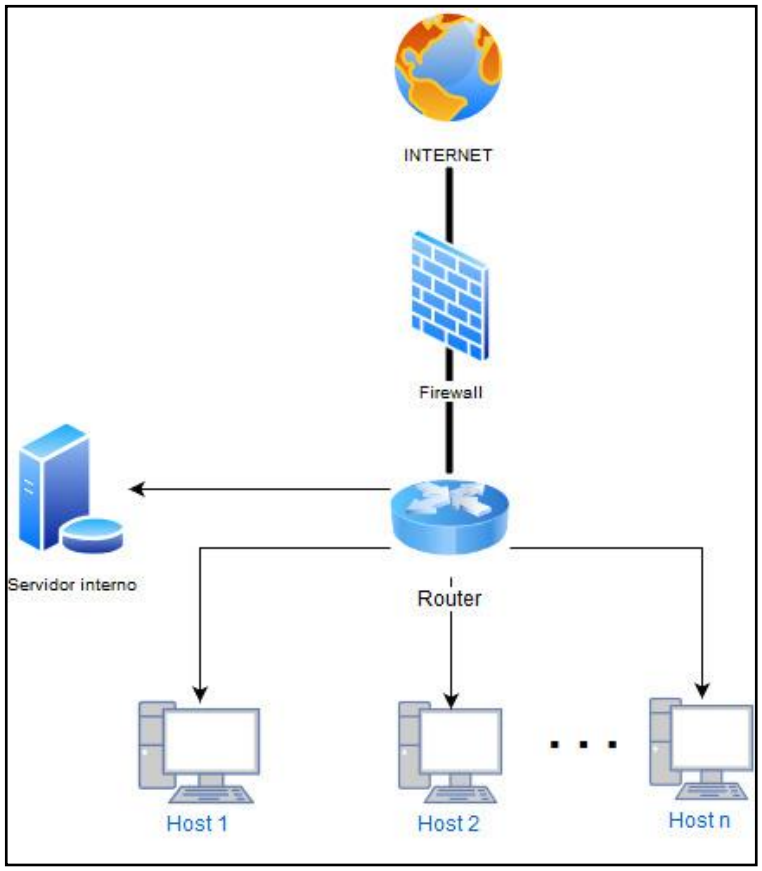


Figura 2-1: Red externa

Realizado por: Herrera, Jonathan 2021.

Este tipo de amenazas son a las que se enfrenta un sistema de seguridad de redes de una organización. Lo más común es la infección de los sistemas operativos de la organización a través de diferentes tipos de códigos maliciosos conocidos como virus de diferentes tipos como son: troyanos, gusanos, malware, entre otros. Por otro lado, los ataques de un hacker son más organizados con el fin de explotar una vulnerabilidad del sistema. El fin principal es recopilar los datos privados e importantes que se almacenan en la empresa para extorsionar, vender esa información o realizar fraudes. (Equipo de expertos, 2018)

1.2.3. Tipos de amenazas

En la tabla 1-1 se observa las amenazas más comunes que se pueden presentar en el sistema de una organización.

Tabla 1-1: Tipos de amenazas

Amenazas	Descripción
Troyanos, virus y gusanos	Son programas de código malicioso, que se alojan en los computadores infectados comúnmente por otro programa o archivo (un ejecutable, imagen, video, descargas, etc.) con el objetivo de permitir el acceso no autorizado o el control de forma remota del sistema. Además, puede ser destructivo, dañando o capturando la información del computador, y generando el consumo de recursos de manera incontrolada para bloquear o negar servicios.
Phishing	El objetivo es obtener de manera fraudulenta información confidencial de un usuario, principalmente del tipo financiero, redes sociales, credenciales corporativas, entre otros. El atacante aprovecha el desconocimiento de los usuarios, los cuales pecan en confiar en servicios tecnológicos con bajos niveles de seguridad. En la actualidad el phishing es muy sofisticado, con herramientas de confirmación como: mensajes de correo electrónico y falsos sitios Web.
Spam	Básicamente consiste en el envío masivo de mensajes basura, el servicio más utilizado para este tipo de amenazas es el correo electrónico, con el propósito de difundir mensajes comerciales, financieros, servicios streaming.
Spywre	Se encarga de recopilar la información de navegación de un usuario, para invadir el navegador web con ventanas emergentes de propaganda de mercadeo, o para robar información personal de tipo financiera,

Fuente: (Tarazona, 2007)

Realizado por: Herrera, Jonathan 2021.

1.3. Servicios de la seguridad

El RFC 2828 define a los servicios de la seguridad como: un servicio de procesamiento o de comunicación proporcionado por un sistema para dar un tipo especial de protección a los recursos del sistema; los servicios de seguridad implementan políticas de seguridad y son implementados, a su vez, por mecanismos de seguridad. (Stallings, 2004, pp. 9) En la tabla 2-1 que se observa a continuación, se presentan todos los servicios de la seguridad que deben cumplirse:

Tabla 2-1: Servicios de la seguridad

Servicios de la seguridad.	Descripción
Autentificación	La seguridad de que la entidad que se comunica es quien dice ser. Es decir, que existe una verificación de las dos entidades involucradas asegurando que no exista intervención de una tercera entidad no autorizada.
Control de acceso	Permite que una entidad autenticada obtenga acceso a varios recursos de acuerdo con su nivel de privilegio otorgado.
Integridad de los datos	La certeza que los datos recibidos son exactamente los mismos que fueron enviados por una entidad autorizada, es decir, no contienen modificación, inserción, omisión ni repetición.
Confidencialidad de los datos	La protección de los datos contra la revelación no autorizada.
Disponibilidad – accesibilidad	La propiedad que posee un sistema, servicio o recurso de estar accesible en todo momento a la solicitud de una entidad autorizada.
No repudio	Evita que el emisor o receptor nieguen la transmisión de un mensaje, por consecuencia de una interrupción de las dos entidades que participan en la comunicación.

Fuente: (Stallings 2004)

Realizado por: Herrera, Jonathan 2021.

1.4. Ataques Informáticos

El avance tecnológico y en particular en el área de las comunicaciones han permitido un crecimiento mundial en aplicaciones y servicios siendo Internet el principal artífice para albergar y compartir toda la información, conocimiento, herramientas, entre otros. Junto con este avance se han desarrollado nuevos ataques y modalidades delictivas, en donde, personas con las

herramientas y conocimientos en el área informática han buscado distintas maneras de apoderarse de información confidencial con fines negativos, con el objetivo de obtener un beneficio económico, a raíz de estos acontecimientos el término "Ataque Informático" es la manera más común de describir estos actos. Un ataque informático se define como el aprovechamiento de una vulnerabilidad en el software, hardware, e incluso, en las personas que forman parte de un ambiente informático. (Mieres, 2009, pp. 3-4) Debido a estos motivos, las organizaciones deben implementar diferentes técnicas de defensa para tratar en lo posible mitigar todo acceso no deseado.

1.4.1. Antecedentes

En la actualidad todo dispositivo conectado a Internet posee un alto riesgo de sufrir un ataque informático, aun mas si este no se encuentra protegido bajo alguna medida de seguridad como un software especializado que puede ser un Antivirus, Malware, spyware o una Red Privada Virtual (VPN) por sus siglas en ingles. La razón se debe a la proliferación de todo tipo de virus, gusanos, troyanos, ransomware, por falta de educación en la seguridad informática.

Definir el comienzo de los ataques informáticos se remonta a mayo de 1950, cuando John Von Neuman desarrolla el concepto de programas auto replicantes, pero todo bajo el marco de una investigación científica, la misma que se desvirtúa por la mala aplicación de sus conceptos a cargo de personas mal intencionadas para causar daños en los sistemas informáticos a niveles universitarios propios de la época. (Manzo, 2010) No es hasta 1971, que el ingeniero Robert "Bob" Thomas creo "Creeper" considerado el primer virus informático, cabe recalcar que no era un código malicioso, al contrario, su objetivo era demostrar que podía migrar de un computador a otro y reproducirse sin autorización del usuario dentro de la red de ARPANET mostrando el mensaje: "Soy una enredadera... ¡atrápame si tú puedes!". (Yubal, 2018) A pesar de que Creeper no representaba ningún riesgo de seguridad, su idea de esparcirse de un computador a otro y de manera relativamente fácil ocasionó que en 1974 se creara "Rabbit" el primer virus informático de tipo malicioso, con el objetivo de reproducirse en el computador para dejarlo sin recursos y bajar su rendimiento. (Yubal, 2018) A lo largo del tiempo, los ataques informáticos se han ido desarrollando muchas veces de forma casual hasta ataques organizados y específicos, entre los más destacados podemos mencionar los siguientes:

Yahoo la popular compañía de correo electrónico, admitió ser víctima de un hackeo total de su plataforma en el 2013, en donde se comprometieron 3000 millones de cuentas cuyos nombres, direcciones de e-mail, números de teléfono, contraseñas y, en algunos casos, preguntas de seguridad fueron robados. (BBC News Mundo, 2016)

Sony Pictures Entertainment fue víctima de dos grandes ataques a su infraestructura, el primero a su división de PlayStation Network en abril de 2011, en el cual el usuario quedo sin acceso total a la plataforma, la misma que estuvo fuera de servicio por casi un mes presentado perdidas aproximadas de 120 millones de dólares. El segundo ataque vino por parte del grupo "Guardianes de la paz" en el 2014, donde tuvieron acceso a información sobre empleados, e-mails confidenciales, direcciones e información financiera. (De la Torre, 2018, pp. 1)

Uno de los ataques que han causado más relevancia en los últimos años fue el que se presentó en mayo de 2017 con el ransomware "WannaCry", por el cual se vieron afectados aproximadamente 150 países y alrededor de 200,000 computadores de distintas empresas y organizaciones, entre las más importantes Telefónica de España, la red de hospitales del Reino Unido, Latam Airlines y varias universidades de Europa y América. Este programa malicioso consistía en "secuestrar" el computador, es decir, bloqueaba y cifraba la información de un computador infectado para pedir un rescate a través de Bitcoins. (Deloitte, 2017, pp. 33-36)

1.4.2. Tipos de ataques

Los ataques informáticos son de diferente naturaleza aplicando técnicas y métodos para vulnerar los sistemas informáticos, actúan con forme al objetivo, pero con un fin común, obtener la información de este. Por tal motivo resulta difícil nombrar todos los ataques que existe, sin embargo, para su estudio se realiza una clasificación general como ataques pasivos y activos.

1.4.2.1. Ataques pasivos

Un ataque pasivo, consiste en obtener la información que se esté transmitiendo entre dos o más computadores, una especie de escucha u observación en el canal de comunicación, pero de forma oculta y sin alterar los recursos del sistema. Dos tipos de ataques pasivos son la obtención de contenidos de mensajes y el análisis del tráfico, siendo este el más importante. (Stallings 2004, pp. 5–6) En la Figura 3 -1 se observa el análisis de tráfico, el atacante pretende capturar la información que se envía en un canal de comunicación, suponiendo que dicha información se encuentra protegida mediante la técnica de cifrado, a pesar de que este logre capturarla no podrá obtener los datos. Se debe tomar en cuenta que a pesar de entregar información cifrada el atacante puede analizar el patrón de envío y de esa menara determinar la localización y la identidad de los servidores que se comunican y descubrir la frecuencia y la longitud de los mensajes que se están intercambiando. Esta información puede ser útil para averiguar la naturaleza de la comunicación que está teniendo lugar. (Stallings, 2004, pp. 7-8)

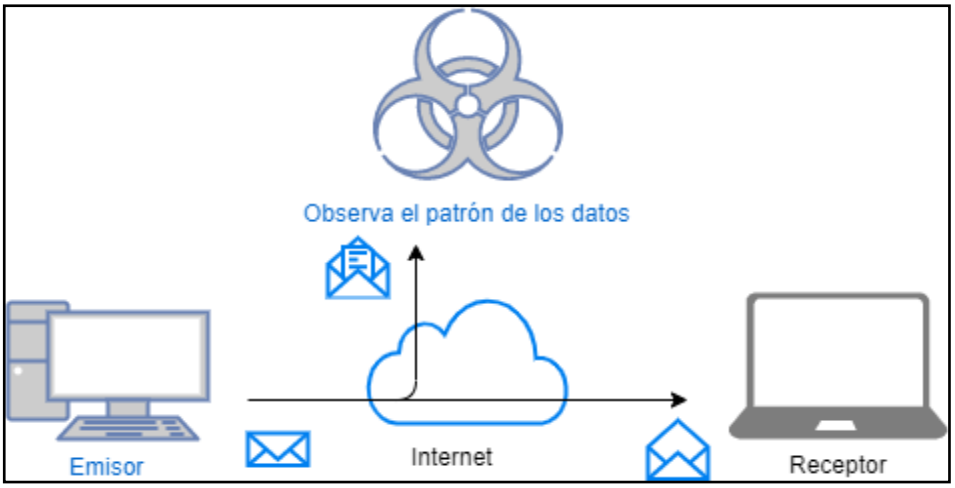


Figura 3-1: Análisis de tráfico

Fuente: (Stallings, 2004)

Realizado por: Herrera, Jonathan 2021.

1.4.2.2. Ataques activos

Un ataque activo busca la modificación o creación de falsa información e intenta alterar los recursos del sistema y afectar a su funcionamiento. Se pueden dividir en cuatro categorías: suplantación o fabricación, repetición o interceptación, modificación de mensajes e interrupción de servicio. La suplantación o fabricación como se observa en la Figura 4-1, se produce cuando una entidad dice ser otra, es un ataque directo contra la autenticidad, porque el atacante puede insertar datos al sistema como, por ejemplo, añadir registros a una base de datos, direcciones IP, e-mail, etc. (Pardo, 2016)

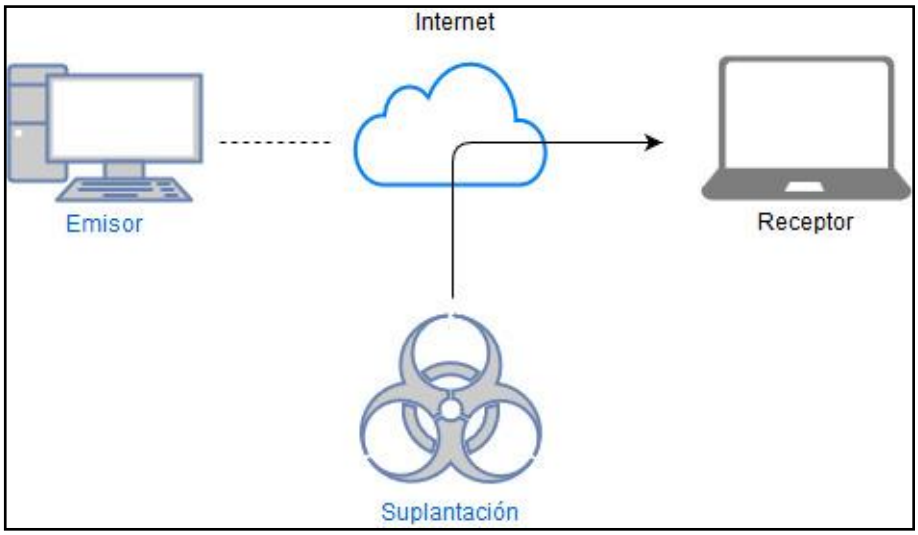


Figura 4-1: Esquema de suplantación

Realizado por: Herrera, Jonathan 2021.

La interrupción del servicio se observa en la Figura 5-1, detiene de forma temporal o definitiva el flujo normal de las comunicaciones. Es uno de los ataques activos más populares como, por ejemplo, la denegación de un servicio, una página web o un servidor de archivos, a través de la inundación de peticiones hacia el servidor provocando una sobrecarga y el fallo de este. Afecta a la *disponibilidad*.

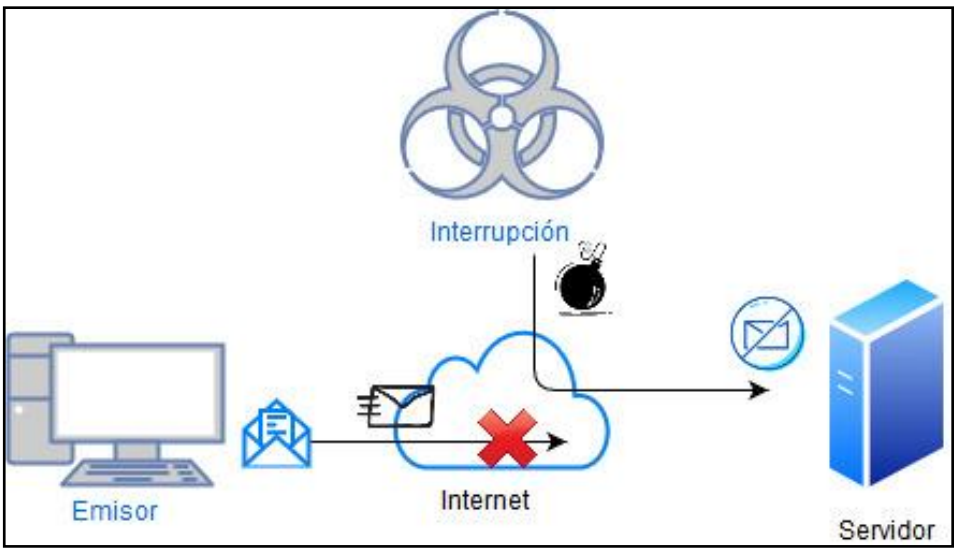


Figura 5-1: Esquema de interrupción

Realizado por: Herrera, Jonathan 2021.

1.5. Sistema de detección de intrusos (IDS)

Un gran número de intentos de intrusión a todo tipo de organizaciones crece día a día, debido al manejo de grandes volúmenes de información a nivel de usuario, los atacantes buscan diferentes métodos de acceso ilegitimo que van desde un simple barrido de ping hasta técnicas sofisticadas buscando vulnerabilidades en las redes. (Jackson et al, 2005, pp. 9)

Por esta razón es necesario el desarrollo de arquitecturas, técnicas y sistemas de detección y prevención. Como herramienta para la detección se crearon los Sistemas de Detección de Intrusos (IDS). (Instituto Nacional de Ciberseguridad, 2017, pp.7)

Un IDS es una herramienta de monitoreo de eventos de la red, en busca de posibles intrusiones, utilizado para la seguridad informática. El IDS genera alertas hacia la consola de administración de red o sistema, dichas alertas pueden ser actividades maliciosas o violación de políticas. (Gopalkrishna et al., 2014, pp. 69) Los IDS trabajan identificando los posibles ataques, monitorea la información; si ha sido comprometida por una intrusión, trata de terminar dicha intrusión en caso

de ser un IDS/IPS, ya que no todos los IDS están en la capacidad de detener una intrusión y finaliza alertando, todo este proceso se realiza en tiempo real y de forma transparente. (Abdullah et al., 2009, pp. 1–2)

Un intento de intrusión se entiende como la búsqueda de comprometer los servicios de la seguridad, es decir, la autentificación, integridad y confidencialidad de los datos, disponibilidad, entre otros. (Mira, 2001, pp.13) Según la investigación de IBM-X Force Threat Intelligence Index 2018 por lo general, el 95% de las intrusiones son causadas por descuido o errores humanos; en un periodo de análisis desde el 2015 hasta 2018 como se observa en el grafico 1-1, los círculos indican el impacto económico aproximado por la filtración de información financiera perdida. (IBM, 2018, pp. 27)

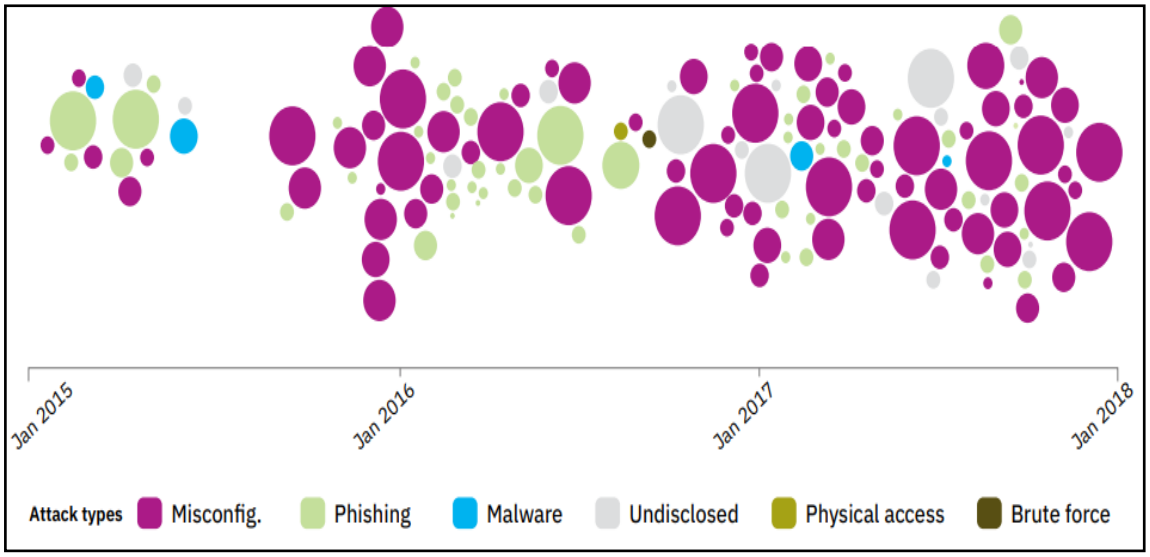


Gráfico 1-1: Impacto de incidentes de seguridad como consecuencia de errores involuntarios.
Fuente: (IBM, 2018)

En la Figura 6-1 se observa un esquema básico de un IDS representado por dos grupos según su enfoque: Detección de uso indebido; son los sistemas basados en normas a través de la comparación de firmas y la detección de anomalías; son sistemas adaptables que utilizan técnicas estadísticas. (Jiménez, 2009, pp. 17)

En la actualidad, con el auge de los ataques informáticos herramientas como los IDS se han vuelto muy populares, cabe recalcar que un IDS no solucionaría todos los problemas de seguridad de una empresa debido a su naturaleza de acción, es un sistema de defensa pasivo, por consecuencia es un complemento a la seguridad de la información de un sistema, en el cual debe existir diferentes formas de protección activas como, por ejemplo, Firewalls, Sistema de Prevención de

Intrusos (IPS), Machine Learning, Yara, encriptación de datos, antivirus, antimalware, entre otros. (Ashoor y Gore, 2011, pp. 6)

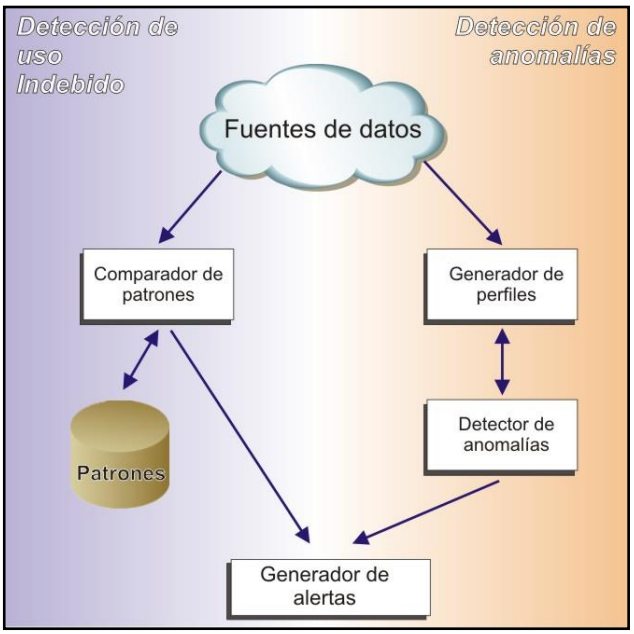


Figura 6-1: Esquema de IDS

Fuente: (Jiménez, 2009)

1.5.1. Antecedentes

El campo de la seguridad informática se ha desarrollado de menos a más, en los últimos 20 años la detección de intrusos de manera específica a obtenido un crecimiento sustancial. En 1980, James Anderson fue de los primeros en publicar su investigación "Monitoreo y vigilancia de amenazas de seguridad informática". (McHugh, Christie y Allen, 2000, pp. 42)

Entre 1983 y 1986, Dorothy Denning y Peter Neumann empezaron al desarrollo de IDS a nivel gubernamental; la investigación realizada permitió la creación del primer IDS bautizado como "Sistema Experto en detección de Intrusiones" o IDES por sus siglas en ingles. (Bruneau, 2019, pp. 3) A partir de los años 2006 a 2010, como se observa en el Gráfico 2-1 los problemas cibernéticos aumentaron de 5000 a 35000 aproximadamente, creando una necesidad de un control mejorado a través de un IDS dentro del modelo de seguridad de red. (Ashoor y Gore, 2011, pp. 3-4)

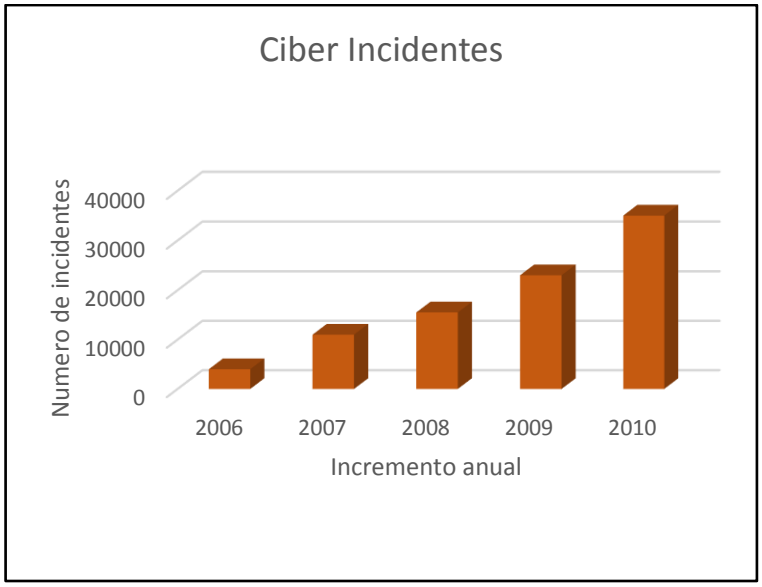


Gráfico 2-1: Reportes de ciber incidentes de la US-Cert

Fuente: (Ashoor y Gore, 2011)

Realizado por: Herrera, Jonathan 2021.

1.5.2. Tipos de IDS

Los IDS se clasifican en diferentes formas, en la Figura 7-1 se observa cuatro tipos principales que son: según su enfoque, origen de datos, estructura y comportamiento, a continuación, se detallaran cada uno de ellos.

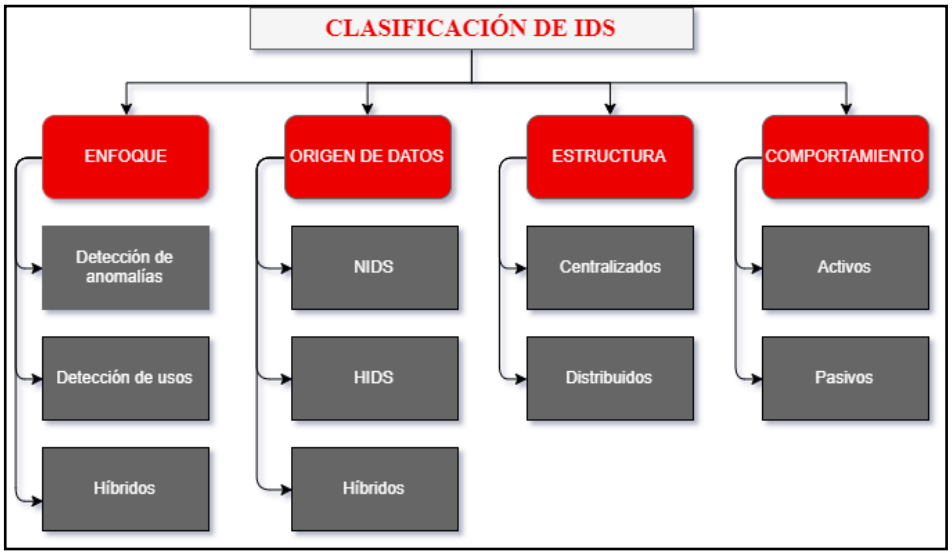


Figura 7-1: Métodos y clasificación de IDS

Fuente: (Jiménez, 2009)

Realizado por: Herrera, Jonathan 2021.

1.5.2.1. IDS según su enfoque

Los IDS poseen dos enfoques fundamentales para su funcionamiento y un tercer enfoque de forma hibrida, que son:

- **Detección de mal uso o firmas**

También conocida como detección de reglas, dicho enfoque se centra en la comparación de los comportamientos rutinarios de los usuarios versus los comportamientos sospechosos de los intrusos que pretenden acceder a la red o sistema. (Stallings, 2004, pp. 311–312) La información obtenida es comparada con grandes bases de datos de firmas las cuales contienen información de ataques documentados, cabe recalcar que no es capaz de detectar ataques o intrusiones nuevas. (Beal, 2005)

- **Detección de anomalías**

El enfoque basado en detección de anomalías consiste en el estudio de los comportamientos de los usuarios reales en un periodo de tiempo, para definir perfiles de actividad individual que ayuden a detectar cambios en caso de existir un ataque. Se realizan pruebas estadísticas para desarrollar un mayor grado de veracidad y definir los límites para determinar la frecuencia de las actividades, pero cabe mencionar que dicho enfoque posee una alta tasa de errores. (Stallings, 2004, pp. 311)

- **Hibrido**

Un enfoque hibrido como su palabra lo indica, busca el funcionamiento en conjunto del enfoque de mal uso o firmas y anomalías. El enfoque basado en anomalías quiere definir la aproximación a un comportamiento normal, pero con rendimiento inferior por otro lado la detección basada en firmas pretende definir un rendimiento mejorado frente a ataques conocidos y vulnerable a desconocidos. El uso de los 2 enfoques permitirá enfrentar los múltiples ataques que existen en la actualidad. (Stallings, 2004, pp. 312)

1.5.2.2. IDS según el origen de datos

En esta clasificación de igual manera que la anterior existen tres tipos de IDS, los cuales pueden obtener eventos de varias fuentes de información, los IDS funcionaran a través del monitoreo y captura de paquetes y otros por el análisis del sistema operativo nativo.

- **IDS basado en host (HIDS)**

Funciona a través del uso de la información del sistema operativo de un único host, con el objetivo de detectar procesos no autorizados. (Peluffo, Capobianco y Echaiz, 2014, pp. 2) Como se observa en la Figura 8-1, los HIDS utilizan las bitácoras o logs generados de forma automática por el sistema operativo (SO) o las aplicaciones que ejecute el usuario. Además del análisis de logs principales como son los de un servidor web, de correo, demonios de red inetd en caso de Linux y servicios en Windows. El análisis implica verificar la integridad de los ficheros importantes, como son el de usuarios y contraseñas, permisos del sistema y tareas programadas. (Chablé, 2013, pp. 21)

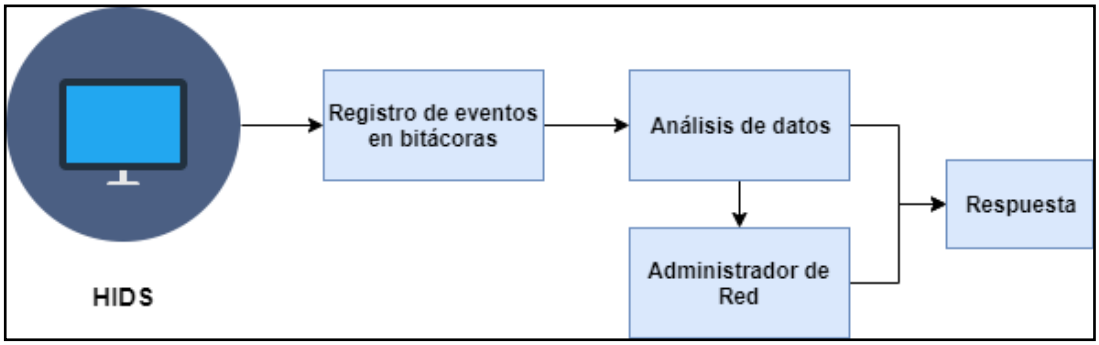


Figura 8-1: IDS basado en host

Fuente: (Chablé, 2013)

Realizado por: Herrera, Jonathan 2021.

- **IDS basados en red (NIDS)**

Se enfoca en el monitoreo, captura y análisis del tráfico de red al cual se encuentra conectado, con una detección en tiempo real de las conexiones entrantes y salientes. (Gopalkrishna, et al., 2014, pp. 70) Los NIDS son trasparentes para la red, es decir, no interfieren en el funcionamiento, para realizar el análisis utiliza comúnmente un switch con un puerto o puertos en modo promiscuo, el cual será el encargado de enviar todo el tráfico de red hacia el NIDS, el mismo que almacenará la información de los paquetes en búsqueda de patrones de ataque y generar una alerta, dicho proceso se evidencia en la Figura 9-1. (Chablé, 2013, pp. 23)

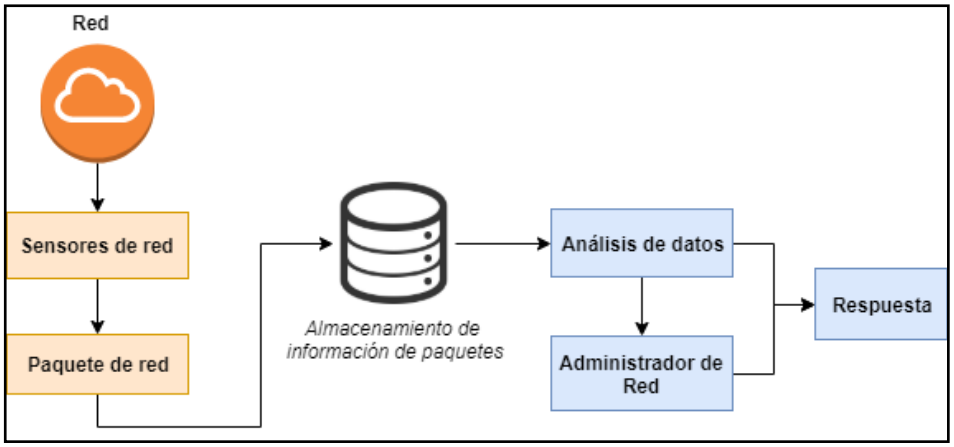


Figura 9-1: IDS basado en red
Fuente: (Chablé, 2013)
Realizado por: Herrera, Jonathan 2021.

- **IDS híbridos**

Como su nombre lo indica es una combinación entre un IDS de host y red. Se junta lo mejor de ambas arquitecturas para salvaguardar una infraestructura que será monitoreada en conjunto y de forma local.

1.5.2.3. IDS según su estructura

- **IDS centralizado**

Este IDS se caracteriza por la administración a través de un sistema central, toda la información es enviada por sensores implementados en la red. En la Figura 10-1 se observa un sensor por cada segmento de red y un único administrador.

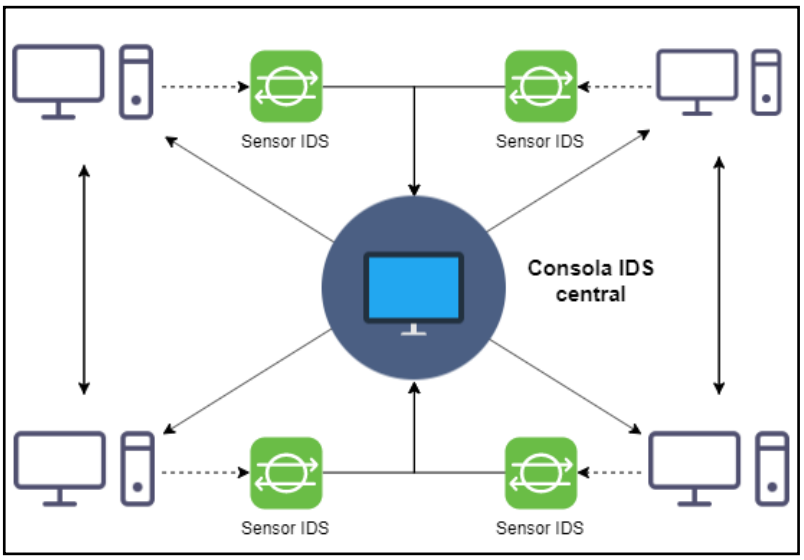


Figura 10-1: IDS centralizado
Fuente: (Instituto Nacional de Ciberseguridad, 2017)
Realizado por: Herrera, Jonathan 2021.

- **IDS distribuido**

Es la evolución de un sistema centralizado; estos producían un notable degrado en el rendimiento por su análisis a toda la red. Los IDS distribuidos monitorean la red, pero poseen colectores de eventos los cuales son canalizados a la consola de administración. Este tipo de IDS como se aprecia en la Figura 11-1 pretende proporcionar una visión globalizada de un sistema. (Jiménez, 2009, pp. 9)

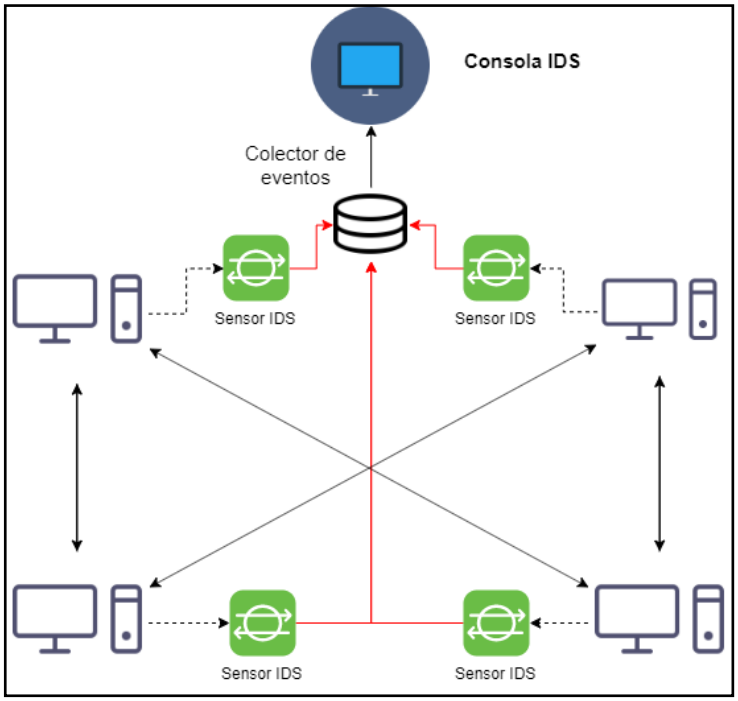


Figura 11-1: IDS distribuido
Fuente: (Instituto Nacional de Ciberseguridad, 2017)
Realizado por: Herrera, Jonathan 2021.

1.5.2.4. IDS según de su comportamiento

- **IDS pasivo**

A través de su análisis a la red y en caso de detectar una intrusión únicamente genera alertas hacia la consola de administración mas no actúa sobre la amenaza. Se necesita del personal capacitado para observar y responder en el momento exacto. (Instituto Nacional de Ciberseguridad, 2017, pp.31)

- **IDS activo**

Conocido como IPS, se diferencia de un IDS en su forma de respuesta, además de generar alertas de una posible intrusión permite responder a las mismas con manejo simultaneo de Firewalls y routers, podrá crear reglas y aplicar métodos tanto ofensivos como defensivos al instante y de forma automática. (Instituto Nacional de Ciberseguridad, 2017, pp. 31)

1.5.3. Técnicas actuales de detección de tráfico malicioso en redes corporativas

Tanto los IDS basados en Host, Red e Híbridos están destinados a proveer beneficios para mejorar a la seguridad de la red o el sistema de una organización, empresa o corporación, pero dichos IDS trabajan bajo la conformación de reglas estáticas definidas por el desarrollador, comunidad libre o usuarios. Las reglas configuradas en los IDS son producto de pruebas y análisis con un uso específico, por esta razón crear nuevas reglas para ataques sofisticados implica un gran desgaste de recursos, tiempo y talento humano. (Peluffo, Capobianco y Echaiz, 2014, pp. 2-3)

En la actualidad existen varias técnicas estudiadas para poder ser utilizadas con un IDS creadas según la necesidad del usuario y el ambiente de red cooperativa dentro del cual se encuentra operando, en la Tabla 3-1 se resumen las técnicas existentes en la actualidad.

Tabla 3-1: Técnicas actuales de los IDS

Técnicas	**Funcionamiento**
Redes Neuronales Artificiales (ANNs)	Las ANNs permiten crear patrones flexibles, dichos patrones arbitrarios serán datos de entrada que servirán para entrenar al sistema que buscara coincidencias con la salida para determinar si ha existido o no una intrusión.
Machine Learning	La implementación de algoritmos supervisados y no supervisados de clasificación capaces de mejorar la detección de intrusos en tiempo real a través de un entramiento por un conjunto de datos.
Tablas de Estado de Transmisión	Permite observar las acciones realizadas por un intruso dentro del sistema y representadas en forma de diagrama de estado, al coincidir con un estado corrompido se detecta la intrusión.
Algoritmos Genéticos (GAs)	Pretende imitar el sistema de selección natural, es decir, el más fuerte sobrevive, en otras palabras, busca crear una firma capaz de reconocer patrones de intrusión.
Red Bayesian	Se introducen modelos gráficos con reglas de transición para la creación de una tabla probabilística con variables en cada nodo, puede contener datos incompletos.

Fuente: (Ahmad et al. 2014, pp. 22), (Al-Subaie y Zulkernine, 2006, pp. 3)

Realizado por: Jonathan Herrera, 2021.

Como se detalla en la Tabla 3-1 existe varias técnicas para la detección de intrusos, sin embargo, en la investigación realizada se detalla que no todas se pueden implementar, en primer lugar,

dependerá de la capacidad de integración que exista entre la técnica y la herramienta IDS a utilizar, es decir, el trabajo de programación para implementar alguna de las técnicas en un IDS Open Source como pueden ser Suricata, Snort o Zeek; los más utilizados en el mercado. En segundo lugar, es escoger la técnica de IDS, se debe analizar los diferentes recursos informáticos (software y hardware) que va a consumir el despliegue de la técnica en conjunto con el IDS, de las mencionadas en la Tabla 7-1 Redes Neuronales Artificiales, Algoritmos Genéticos y Red Bayesian son aquellas que necesitan de grandes recursos además de un gran estudio para su correcto funcionamiento, en el mercado no existe ningún IDS que utilice este tipo de técnicas. Machine learning y Tablas de Estado de Transmisión de igual manera necesitan de recursos informáticos, pero en menor medida que las otras tres técnicas; Tablas de estado es una técnica poco desarrollada y no se ha implementado estudios para integrarlo con otras herramientas, lo contrario sucede con machine learning que en la actualidad posee muchos estudios y sus utilidades se desarrollan en todos los campos de la informática, la seguridad informática es uno de ellos y por tal razón la plataforma OPNids integro esta técnica. En busca de la autonomía de los IDS se propone la inclusión de técnicas Machine Learning, el cual cuenta con un amplio campo de desarrollo aplicable a todo nivel, la seguridad de redes no es la excepción obteniendo resultados, como, por ejemplo, detección de tarjetas de crédito falsas, defensa contra el cyber-terrorismo, detección de botnets, entre otros. (Thuraisingham et al. 2008, p. 585)

1.6. Machine Learning

Los programas informáticos fueron creados para satisfacer las necesidades de los usuarios y cumplir tareas específicas; dichas tareas son procesos transparentes para el usuario pero que el computador ejecuta y cumple con el objetivo programado. El campo del ML se enfoca a un nivel superior de cómo construir programas informáticos que aprendan automáticamente con experiencia. Se asume que un programa informático aprende de la experiencia “E” con respecto a alguna clase de tarea “T” y alguna medida de rendimiento “P”, si su rendimiento en “T”, medido por “P”, mejora con la experiencia “E”. (Mitchell, 1997, pp. 11-12) Por lo tanto, los algoritmos de ML se alimentan de forma automática de cierta información proporcionada por el sistema analizado. Desafortunadamente, su veracidad depende de la cantidad y calidad de los datos con los cuales son entrenados. Por esta razón se debe escoger la fuente correcta para alimentar los algoritmos de ML y obtener resultados precisos. (Hall y Smith, 1996, pp. 1)

Este aprendizaje personalizado se puede utilizar para hacer predicciones sobre cualquier tendencia social en el futuro y para proporcionar información sobre el comportamiento natural de los conceptos objetivo. Los conceptos que manejan los algoritmos de ML, en primera fase siendo

entrenados a través de datos de ejemplo, para establecer características y relaciones que serán puestas a prueba en la fase dos, que consiste en la clasificación en grupos según las características detectadas. El éxito de un algoritmo de ML para cumplir con su determinada tarea dependerá, de lo mencionado en párrafos anteriores; la cantidad y calidad de los datos manejados esto se logra en la fase de análisis de datos, filtrando la información para ajustar un algoritmo soñado. (Hall y Smith, 1996, pp. 1)

En el Grafico 3-1 se observa que Machine Learning no es un concepto aislado, al contrario, es un conjunto de aportes de varias ramas como: la estadística; con sus modelos estadísticos versus la minería de datos; con algoritmos de aprendizaje automático, estableciendo enlaces hacia las redes neuronales modernas; las cuales están fuertemente ligadas a la neurociencia computacional y cognitiva. Sin dejar de lado a la Inteligencia artificial como origen del Machine Learning.

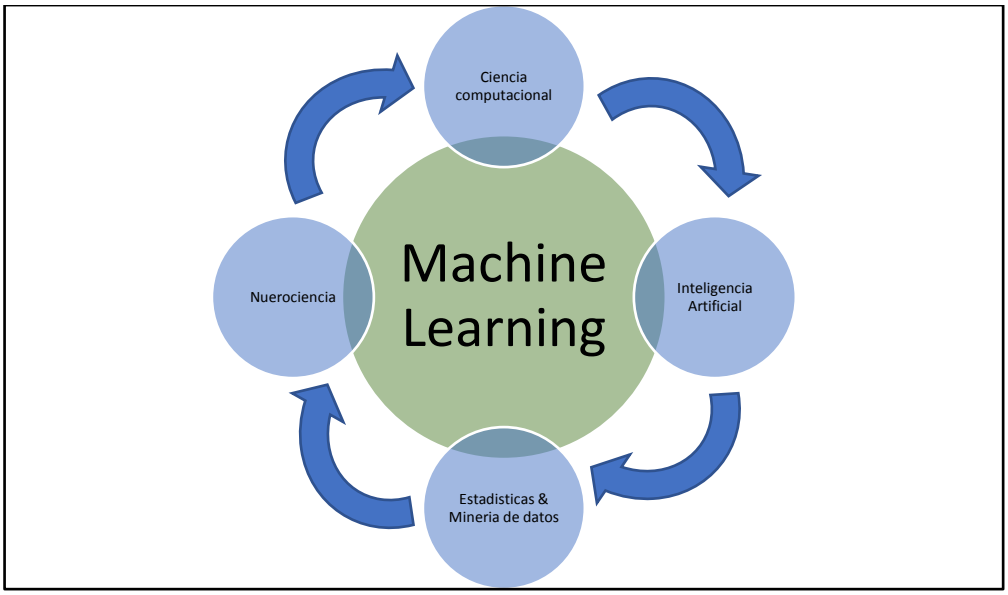


Gráfico 3-1: Machine Learning y relación con otras ramas

Realizado por: Herrera**,** Jonathan 2021.

1.6.1. Machine Learning en la seguridad informática

Machine Learning y sus algoritmos estadísticos no fueron desarrollados con el objetivo de ser aplicados en la seguridad informática, sin embargo, su gran potencial de procesamiento de datos ha generado la adaptación de estos para cumplir funciones de seguridad.

Cabe recalcar, que el ML no será la varita mágica en la seguridad informática en comparación con el reconocimiento fácil o procesamiento del lenguaje; campos con fuerte presencia y resultados positivos. Siempre existirá individuos en busca de vulnerabilidades en los sistemas o los algoritmos de ML para evadir los mecanismos de defensa. (Polyakov, 2018)

Además, en la actualidad los atacantes utilizan algoritmos de ML en ofensiva, es decir, son utilizados para evadir la detección y defensas del sistema, aprendiendo el comportamiento de estas y con la capacidad de hacer que sus algoritmos sean entrenados de forma errónea. Por tal motivo métodos adaptativos y dinámicos siempre conllevan un riesgo en la seguridad. (Chio y Freeman 2018, p. 11-12)

El aprendizaje automático en la seguridad informática se puede clasificar en dos categorías que son:

1.6.1.1. Reconocimiento de patrones

El reconocimiento de patrones busca detectar características explícitas u ocultas en la información. Dichas características son separadas por conjuntos de características semejantes, los cuales sirven de ejemplo, para entrenar a los algoritmos a detectar en otro tipo de datos que presentan similitud al conjunto de entrenamiento. (Chio y Freeman, 2018, pp. 12)

1.6.1.2. Detección de anomalías

La detección de anomalías no quiere depender de aprender patrones específicos que determina un cierto conjunto de datos, el objetivo es crear una proyección de normalidad que describa la mayoría de un conjunto de datos dado. Establecido dicho umbral de normalidad, cualquier desviación será considerada una anomalía. (Chio y Freeman, 2018, pp. 12-13)

1.6.2. Enfoques de Machine Learning

Con el conocimiento sobre el Machine Learning en la seguridad informática es necesario conocer el enfoque de sus algoritmos y el manejo de los datos. En una red poder diferenciar todo el tráfico normal versus el potencialmente malicioso es casi imposible, como saber si un archivo contiene malware, si una solicitud a los servidores es un ataque de Denegación de Servicio (DoS) o si una conexión saliente es un bot que pretende conectarse a un servidor de Command – Control. (Chio y Freeman 2018, p. 26) Por tal motivo el aprendizaje automático se divide en dos principales categorías para la solución de tareas, estos son: algoritmos *supervisados* y *no supervisados*.

1.6.2.1. Algoritmos de machine learning supervisados

Los algoritmos de aprendizaje supervisado necesitan hacer uso de la información histórica del sistema, la cual posee características definidas que servirán para entrenar el algoritmo.

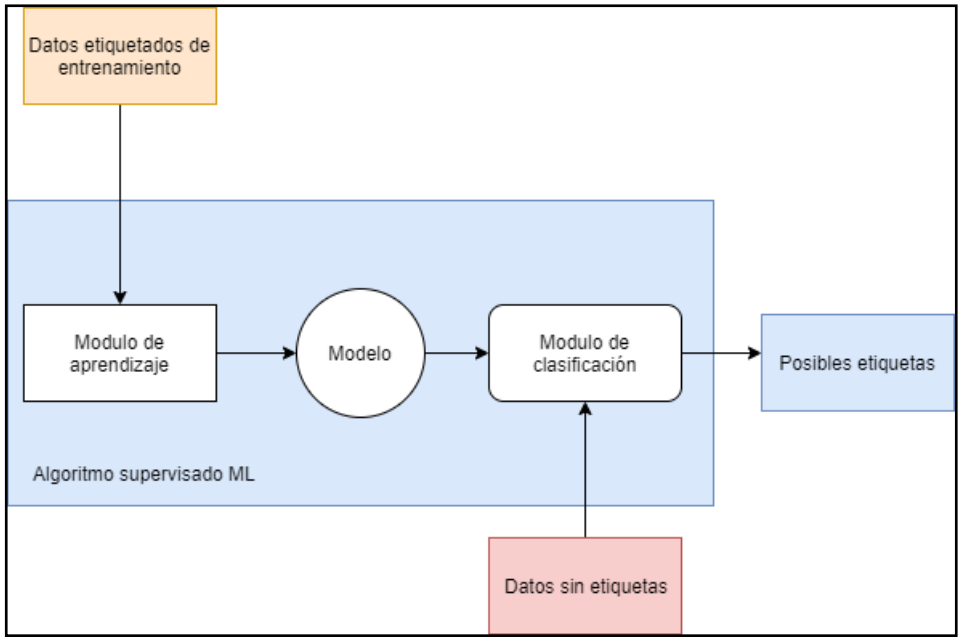


Figura 12-1: Aprendizaje supervisado

Fuente: (Hendrickx, 2005)

Realizado por: Herrera, Jonathan 2021.

Como se observa en la Figura 12-1, un algoritmo de ML supervisado está formado por tres partes: un módulo de aprendizaje, un modelo y un módulo de clasificación. El módulo de aprendizaje crea un modelo basado en un conjunto de características para entrenamiento con etiquetas. Este modelo consta de una función construida por el módulo de aprendizaje y contiene un conjunto de asignaciones asociativas; la información de tal conjunto es obtenida de los "logs" almacenados del sistema. (Hendrickx, 2005, pp. 2) Estas asignaciones, cuando se ejecutan a un conjunto de prueba no etiquetado, predicen las etiquetas, la exactitud dependerá del nivel de entrenamiento del modelo, es decir, la información con la que fue alimentado el módulo de aprendizaje deberá ser lo suficientemente precisa y en gran cantidad para obtener resultados reales. La predicción de las etiquetas del conjunto de prueba lo realiza el módulo de clasificación. (Hendrickx, 2005, pp. 2)

1.6.2.2. Algoritmos de machine learning no supervisados

Los algoritmos de ML no supervisados son un enfoque basado en datos, a diferencia del aprendizaje supervisado, este algoritmo se entrena a través de datos no etiquetados, con la misión de que el algoritmo sea capaz de predecir patrones para clasificar un conjunto de datos. (López, 2015) El objetivo del aprendizaje no supervisado puede ser agrupar ejemplos en función de su similitud. (Hendrickx, 2005, pp. 2)

1.6.3. Tareas de Machine Learning

Existen varios métodos para realizar las diferentes tareas de Machine Learning. Las tareas de *clasificación* y *regresión* son aprendizaje supervisado, como la *agrupación* o *clusters y reducción dimensional* son una forma de aprendizaje no supervisado. (Chio y Freeman, 2018, pp. 26) En el presente trabajo de titulación se utilizará tareas o algoritmos de clasificación por su naturaleza y como su nombre lo dice se basan en la clasificación de manera muy precisa a los diferentes tipos de variables que puede presentar un conjunto de datos y en este caso específico los tipos de ataques o amenazas informáticas. A continuación, se explica la tarea de clasificación y dos de sus algoritmos más utilizados los mismos que se encuentran integrados en la plataforma OPNids.

1.6.3.1. Clasificación

En la tarea de *clasificación* de eventos en el tráfico de red, el algoritmo de ML supervisado ya entrenado con información de posibles ataques pasados busca asociarlos a eventos futuros. La clasificación puede ser binaria, en la que solo se identifica dos clases, o multi categoría; en el caso de que un objeto posea variantes como, saber qué clase de malware es: ransomware, un keylogger o un troyano de acceso remoto. (Chio y Freeman, 2018, pp. 26) La Figura 13-1 muestra una clasificacion binaria, en la que el algoritmo supervisado de clasificación logra indentificar las variables categoricas y asi precedir las variables discretas, como resutado se obtiene la clasificación. (Epicalsoft Instance, 2018)

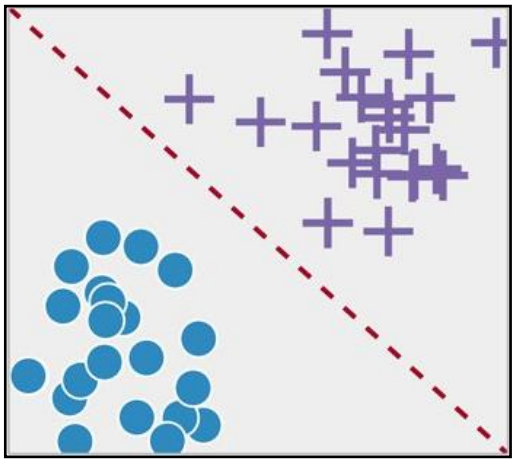


Figura 13-1: Algoritmo de clasificación

Fuente: (Epicalsoft Instance, 2018)

A continuación, se describen los dos algoritmos de ML supervisados de clasificación que se encuentran en la plataforma OPNids y estudiaremos cual es mejor para ser utilizado en la red de la FIE.

Regresión logística

La regresión logística es un algoritmo de clasificación utilizado para predecir la probabilidad de una variable dependiente categórica. La variable dependiente es del tipo binaria que puede contener valores codificados como 1 – 0, sí – no, verdadero – falso. Este algoritmo calcula la probabilidad de una respuesta binaria basada en una o más variables independientes. (Gonzáles, 2018a)

Al tratar con problemas de clasificación binaria únicamente se obtendrá dos posibles valores, por lo tanto, existe un valor de umbral el cual determinará la clasificación, si supera el umbral la variable obtendrá el valor de 1, caso contrario será 0, dicho umbral por lo general es de 0.5 el cual puede ser ajustado conforme los requerimientos para reducir los falsos positivos y negativos. La variable dependiente se relaciona con la variable independiente a través de la función sigmoide, como se observa en la Figura 14-1 es una curva en forma de S que puede tomar cualquier valor entre 0 y 1. (Rodríguez, 2018a)

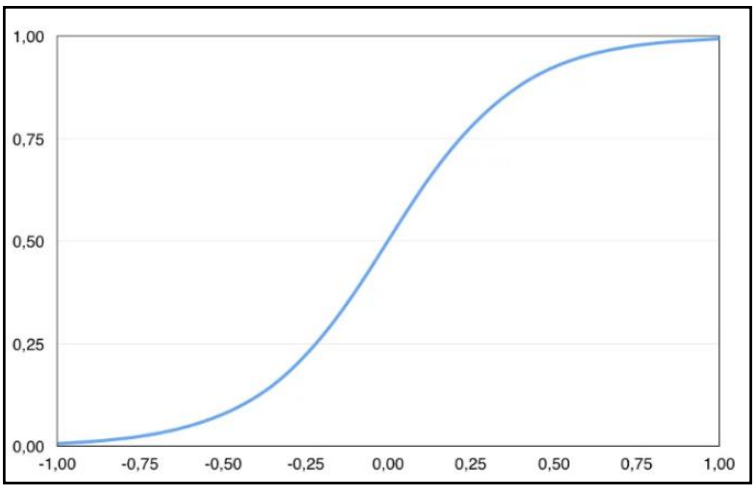


Figura 14-1: Función sigmoide

Fuente: (Rodríguez, 2018)

La regresión logística es un algoritmo popular, a pesar de los algoritmos avanzados como redes neuronales profundas, es escogido por su bajo consumo de recursos en hardware y software, su eficiencia depende del entrenamiento realizado a través de una gran cantidad de datos, es capaz de manejar miles de características. (Gonzáles, 2018a) Además, se debe conocer ciertas características importantes para cumplir con un desempeño correcto del algoritmo supervisado de regresión logística, que son:

- La variable de salida debe ser del tipo binario.
- El algoritmo no es capaz de detectar los errores en la variable de salida, es decir, debe existir un proceso de limpieza de datos, eliminar valores atípicos y posibles relaciones entre categorías del conjunto de entrenamiento. (Gonzáles, 2018a)
- La regresión logística es un algoritmo lineal, con una transformación no lineal en la salida. Las transformaciones de datos de sus variables de entrada que exponen mejor esta relación lineal pueden dar como resultado un modelo más preciso. (Gonzáles, 2018a)
- Eliminar las características que presentan multicolinealidad, las variables deben ser independientes entre sí. (Chio y Freeman, 2018, pp. 41)

Bosques aleatorios (Random Forest)

El algoritmo de bosques aleatorios o Random Forest está formado por varios árboles de decisión individuales y que en conjunto forman un bosque como se observa en la Figura 15-1, cada árbol utiliza los mismos nodos, pero con diferentes datos que conducen a diferentes hojas. Combina las decisiones de múltiples árboles de decisión para encontrar una respuesta, que representa el promedio de todos los árboles involucrados. (Schott, 2019)

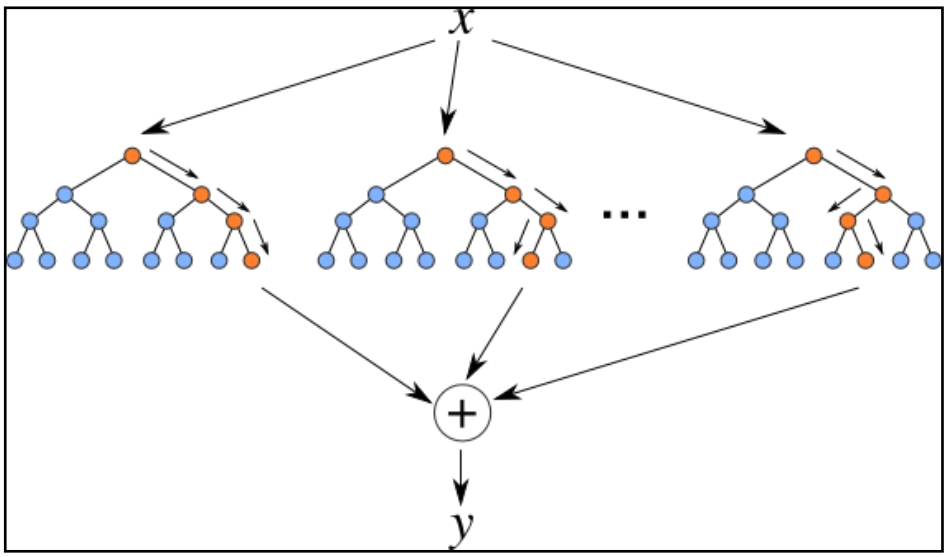


Figura 15-1: Algoritmo Random forest

Fuente: (Schott, 2019)

Los árboles de decisión son versátiles al momento de tratar con variables continuas o categóricas ya que no necesitan de un continuo entrenamiento para predecir y modelar los datos. Sin embargo, su desventaja se presenta al momento en que dichos arboles crecen de manera profunda llegando

a fallar en sus predicciones hasta el punto de provocar un sobreajuste. (Aguirre, 2019b) Al igual que todos los modelos de aprendizaje automático los árboles de decisión también sufren de problemas de sesgo o bias y varianza; donde el sesgo es la diferencia entre los valores pronosticados y los valores reales y la varianza es la diferencia de la predicción de un modelo al ser entrenado con un conjunto diferente de datos. (Orellana, 2018)

Los valores de sesgo y varianza variaran dependiendo el modelo, es decir, si se aumenta la complejidad del modelo, el sesgo se reduce y aumenta la precisión, pero una vez que este modelo se vuelva más robusto producirá un sobre ajuste y la varianza será alta. (Orellana, 2018) Por esta razón un árbol de decisión tiende a ser un estimador con bajo sesgo, pero alta varianza. (Aguirre, 2019b) En la Figura 16-1 se observa el concepto de un modelo óptimo en donde se busca un punto medio entre el sesgo y la varianza, a esta técnica se la conoce como *trade-off* o equilibrio, para lograr esto se utiliza métodos ensambladores (ensembles). (Orellana, 2018)

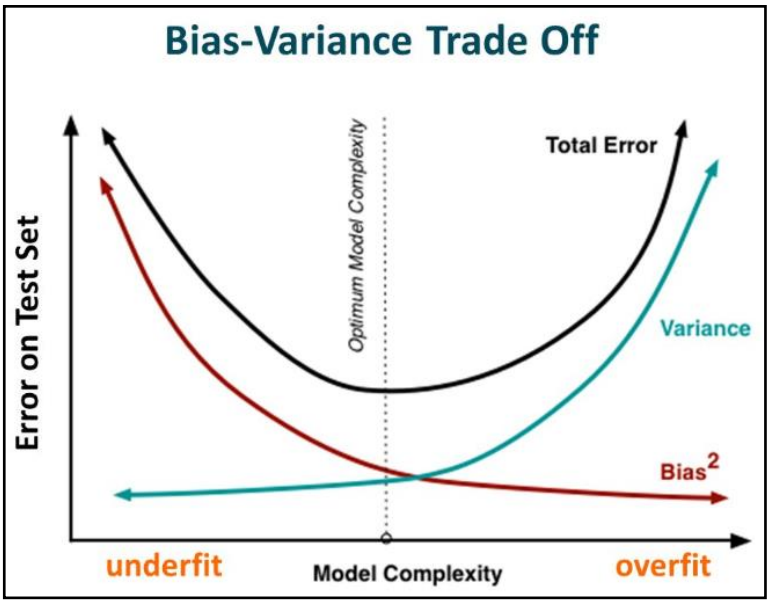


Figura 16-1: Modelo óptimo

Fuente: (Orellana, 2018)

Un ensamble es la combinación de múltiples modelos que analizan muestras diferentes, por cada modelo agregado se reduce de forma considerable el error general al tener menor varianza. Existen 3 métodos de ensamble Bagging, Boosting and Stacking, pero el más utilizado para Random forest es bagging. (Aguirre, 2019b) Bagging es una técnica usada para reducir la varianza de las predicciones a través de la combinación de los resultados de varios clasificadores, cada uno de ellos modelados con diferentes subconjuntos tomados de la misma muestra. La Figura 17-1 representa esta técnica. (Orellana, 2018)

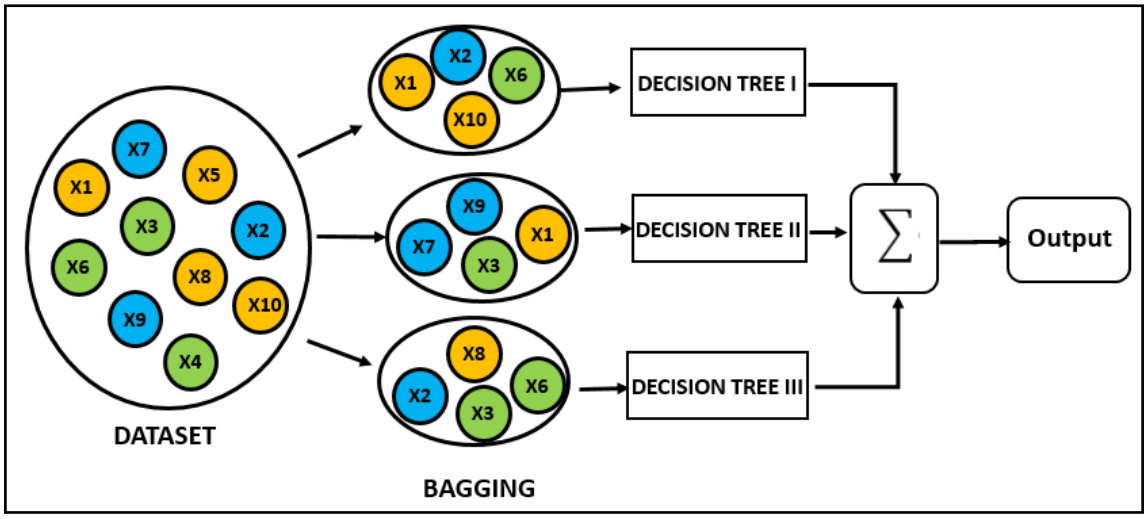


Figura 17-1: Técnica bagging

Fuente: (Orellana, 2018)

1.6.4. Comparación de algoritmos de machine learning en la plataforma OPNids

Una vez estudiada la teoría de los dos algoritmos de clasificación existentes en la plataforma OPNids se realiza una comparativa entre los mismos para analizar sus ventajas y desventajas y obtener un mejor criterio de decisión, a continuación, en la Tabla 4-1 se detalla de mejor manera.

Tabla 4-1: Comparativa de algoritmos ML de clasificación

Características	**Algoritmos**	
	Regresión Logística	Random Forest
Clasificación de variables binaria	Si	Si
Clasificación de variables múltiples	No	Si
Reducción de sobre ajuste	No, pero se puede logar con un correcto dataset y entrenamiento	Si
Costo de hardware y software alto	No	Si
Dataset de gran volumen	No	Si
Tiempo de entramiento	Bajo	Alto

Realizado por: Herrera, Jonathan 2021.

En la Tabla 4-1 se puede apreciar ciertas características importantes que posee cada algoritmo si bien en la mayoría el algoritmo de Random Forest es el ganador y se consideraría el mejor hay que tener en cuenta que una de sus principales desventajas es el costo de hardware y software y a la vez el tiempo de entramiento son problemas muy importantes que deben ser tomados en cuenta al momento de implementar dicho algoritmo y que con su correcta supervisión se logrará resultados muy precisos. Por el contrario, el algoritmo de regresión logística es muy bueno en

cuanto a su nivel de consumo de recursos y tiempo, pero con una desventaja importante es que no necesariamente necesita un gran volumen de datos para su entrenamiento exponiéndolo a un gran margen de error además de que para realizar la clasificación únicamente lo hace de manera binaria. Los dos algoritmos son muy buenos en diferentes casos de estudio, si bien se podría escoger el algoritmo de regresión logística por su ventaja en el consumo de recursos no sería una garantía ante el nivel de clasificación que se quiere obtener, por el contrario, Random forest al ser un algoritmo más robusto, con capacidad de analizar y clasificar múltiples variables en conjunto con sus múltiples arboles de decisión nos acerca hacia un menor grado de error y es el que se utiliza en el presente trabajo de titulación.

1.7. OPNids

OPNids es un sistema de detección de intrusos (IDS) de código abierto, fácil de usar, que representa la primera integración de Suricata Signature Inspection con un motor de secuencias de comandos de aprendizaje automático (Dragonfly MLE). (OPNids, 2018)
OPNids incluye la mayoría de las funciones disponibles en los costosos sistemas comerciales de detección de intrusiones y más en muchos casos. La inclusión de Machine Learning Scripting Engine representa la innovación para la comunidad de código abierto al permitir un camino para que Data Science Security Analyst aproveche el motor de detección de red basado en Suricata maduro y robusto mientras experimenta con los modelos de amenazas de aprendizaje automático en la primera plataforma con un Motor de secuencias de comandos de aprendizaje automático de código abierto. (OPNids, 2018)

1.7.1. Herramientas de la plataforma OPNids

La plataforma OPNids en su interfaz gráfica de control y administración cuenta con varias herramientas configurables, como se observa en la Figura 18-1 a continuación, se explicarán y analizarán las utilizadas en esta investigación.

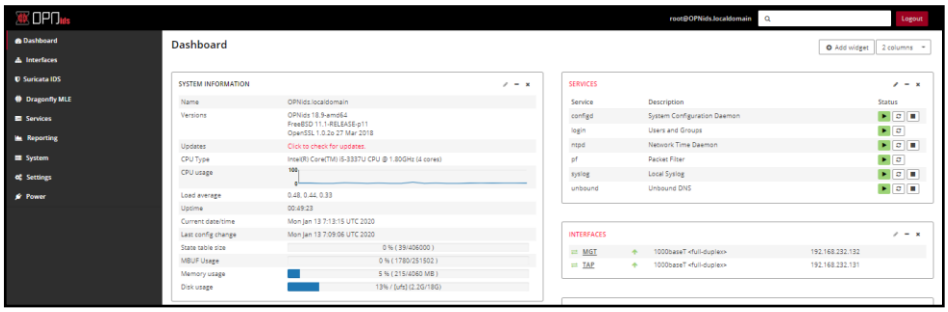


Figura 18-1: Interfaz de administración OPNids
Realizado por: Herrera, Jonathan 2021.

1.7.2. Suricata IDS

Suricata es un motor de detección de amenazas de red Open Source que proporciona capacidades que incluyen detección de intrusos (IDS), prevención de intrusos (IPS) y monitoreo de seguridad de red. Funciona muy bien con la inspección profunda de paquetes y la coincidencia de patrones, lo que lo hace increíblemente útil para la detección de amenazas y ataques. Suricata pertenece a la Open Information Security Foundation (OISF). (Bricata, 2019) Las características de Suricata se detallan en la Tabla 5-1.

Tabla 5-1: Características de Suricata IDS/IPS

Característica	Descripción
Multi-Threading	La característica principal y por la cual suricata es considerado uno de los mejores IDS/IPS es Multi-Threading consiste en procesamiento de paquetes en uno o más hilos aprovechando el procesamiento múlti-nucleo de un procesador actual, potenciando su capacidad de detección. Esto no es posible realizar en IDS/IPS Uni- Threading
Estadísticas de rendimiento	Modulo que se encarga de registrar y almacenar nuevos eventos para ser presentados como estadísticas. Este módulo recolecta datos como: bytes/sec, packets/sec, alerts/sec, dropped packets, entre otros.
Detección de protocolos automático	Suricata maneja palabras clave para protocolos como: IP, TCP, UDP, HTTP, ICMP, FTP, TLS y SMB con el objetivo de detectar en un flujo de datos. Este módulo es efectivo para la detección de malware.
Métodos de entrada estándar	Soporte para NFQueu, IPFRing y LibPcap standard para la captura de tráfico.
Unified2 Output	Soporte para métodos y herramientas de salida estándar unified2.
Fast Ip matching	Utiliza un preprocesador para validar más rápido las reglas que marquen coincidencia de IP, por ejemplo, RBN o las listas IP de Emerging Threats.
Aceleración por tarjetas de video	Utilizando CUDA y OpenCL se puede utilizar el poder de procesamiento de las tarjetas gráficas para liberar recursos del procesador.
Windows Binaries	Compatible con el sistema operativo de Windows.

Tabla 5-1 (continuación)

Módulos de detección extendidos	Además de analizar paquetes Suricata es capaz de examinar certificados TLS/SSH, peticiones DNS y solicitudes HTTP

Fuente: (Astudillo, Jimenez y Ortiz, 2011, pp. 15-19)

Realizado por: Herrera, Jonathan 2021.

Suricata como todo IDS/IPS basado en firmas necesita un conjunto de reglas para la detección de paquetes por esta razón es importante conocer y entender la forma como está formada una regla y su funcionamiento, por lo general se utilizan reglas existentes y a su vez es posible crear reglas específicas según el requerimiento de la red analizada. Una regla o firma está formada por tres partes:

- **Action (Acción):** Determina la acción a tomar cuando existen una coincidencia o match.

- **Header (Cabecera):** Se define el protocolo, direcciones IP,puertos y dirección de la regla.
  En la Figura 31-1 el texto resaltado de color verde representa la cabecera, en él se encuentran diferentes parámetros que suricata buscará coincidencias al momento del análisis, en primer lugar, se tienen los diferentes protocolos como TCP, UDP, ICMP y también los protocolos de capa de aplicación como HTTP, FTP, TLS, DNS, entre otros. (OISF, 2016a)
  A continuación, el parámetro de dirección IP de origen ($HOME_NET) y destino ($EXTERNAL_NET) bajo los nombres por defecto. Y para finalizar la cabecera se tiene los puertos de origen y destino, estos cambiaran según el protocolo y pueden ser el puerto 80 de HTTP o el 443 de HTTPS, entre otros. (OISF, 2016a)

- **Rules options (Opciones de reglas):** Detalles de la regla. En la Figura 16-1 el texto resaltado de azul son las diferentes opciones que varían dependiendo hacia donde está orientada. (OISF, 2016a)

En la Figura 19-1 el texto resaltado de color rojo es la acción, esta propiedad define el comportamiento de la alerta, existen cuatro opciones, que son detalladas en la Tabla 6-1: (OISF, 2016a)

Tabla 6-1: Acciones de las reglas de un IDS/IPS

Acción	Descripción
Pass (Pasar)	Si la regla/firma coincide Suricata deja de analizar el paquete y salta de hacia el siguiente.

Tabla 6-1 (continuación)

Drop (Bloquear)	Esta propiedad se utiliza más en el modo IPS y al momento de coincidir Suricata bloquea el paquete y no se enviará más, muchas veces esta acción puede ser equivocada y bloquea tráfico normal dejando en espera una sesión; en el caso de ser TCP. Se genera una alerta.
Reject (Rechazar)	Se rechaza el paquete de manera activa, es decir, el emisor y receptor recibirán un paquete de rechazo. Si el paquete era TCP recibirá un paquete de rechazo Reset y los demás protocolos error ICPM. También se generan alertas.
Alert (Alerta)	Si coincide con la propiedad de Alerta, el paquete pasará sin problemas solo el administrador de red conocerá la información.

Fuente: (OISF, 2016b)

Realizado por: Herrera, Jonathan 2021.

Un ejemplo de una regla/firma se observa en la siguiente Figura 19-1:

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN Likely Bot Nick in IRC (USA +..)";
flow:established,to_server; flowbits:isset,is_proto_irc; content:"NICK "; pcre:"/NICK .*USA.*[0-9]
{3,}/i"; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; sid:2008124;
rev:2;)
```

Figura 19-1: Formato de reglas

Fuente: (OISF, 2016)

La plataforma OPNids en sus recomendaciones de hardware menciona que debe contar con un procesador de mínimo 4 núcleos, la razón de este requerimiento es aprovechar el Multi-hilo de suricata y potenciar su funcionamiento. Suricata posee varios modos de ejecución (Runmodes) y al ser multiproceso, mantiene varios subprocesos ejecutándose a la vez. Un paquete puede ser procesado por más de un hilo. El paquete se pasará al siguiente hilo a través de una cola. Los paquetes serán procesados por un hilo a la vez, pero el motor puede procesar varios paquetes a la vez. Un hilo puede tener uno o más módulos de hilo. Si tienen más módulos, solo pueden estar activos a la vez. La forma en que los hilos, módulos y colas se organizan juntos se llama Runmode. (OISF, 2016c) Suricata posee 3 modos de ejecución, pero para poder listar en línea de comandos se puede usar el comando -list-runmodes. A continuación, en la Tabla 7-1 se detallan los mismos.

Tabla 7-1: Modos de ejecución de Suricata

Modo de ejecución	**Descripción**
Single	Este modo es similar al workers, la diferencia radica en la utilización de un solo subprocesador de paquete.

Tabla 7-1 (continuación)

Workers	Es el modo por defecto y que posee mejor rendimiento. El controlador de la NIC asegura un correcto balanceo de los paquetes hacia los hilos de procesamiento de Suricata.
Autofp	Este modo se utiliza al momento de procesar archivos PCAP o en algunas configuraciones de IPS. Varios hilos capturan los paquetes y los decodifican, después los envían hacia otros hilos llamados flow worker.

Fuente: (OISF, 2016c)

Realizado por: Herrera, Jonathan 2021.

1.7.3. Características del motor de machine learning Dragonfly

Dragonfly MLE es un motor de aplicaciones de transmisión escalable, programable para la detección de amenazas de red creado en Redis y LuaJIT. MLE proporciona poderosos algoritmos de detección de anomalías, búsquedas de inteligencia de amenazas y predicciones de aprendizaje automático con modelos entrenados. MLE es ligero, rápido y flexible.

Está diseñado para funcionar en conjunto con Suricata. Al ejecutar analizadores definidos por el usuario implementados en Lua, puede procesar cientos de miles de eventos por segundo. (Counterflow, 2018)

Dragonfly MLE cuenta con las siguientes características:

- Diseñado para integrarse con Suricata.
- Implementado en C con rutas de ejecución escalables de subprocesos múltiples.
- Secuencias de comandos LuaJIT definidas por el usuario con soporte nativo para json y redis.
- Soporte nativo para operaciones de Redis ML.
- Capaz de ejecutarse como una aplicación Dockerized. (Counterflow, 2018)

Dragonfly MLE está formado por una arquitectura pipeline con un sistema de colas configurable con tres procesadores de eventos que son:

- Entrada: obtiene los mensajes de una fuente, normaliza los datos en formato JSON y los dirige a la cola del analizador asignada para su procesamiento. Las fuentes de mensajes son archivos, sockets de Unix o intermediarios de kafka. Las operaciones de normalización y ETL se realizan mediante un script Lua definido por el usuario.
- Análisis: extrae los mensajes de la cola, analiza el evento y enruta los resultados a la cola de salida adecuada para su procesamiento. Los analizadores se implementan como scripts Lua definidos por el usuario.

- Salida: extrae los mensajes de la cola y los entrega al receptor apropiado. Los receptores de mensajes son archivos, sockets de Unix o intermediarios de Kafka. (Counterflow, 2018)

El motor de Machine Learning Dragonfly viene integrado con dos algoritmos de aprendizaje automático supervisados de clasificación que son Regresión logística (Logistic Regression) y Bosques aleatorios (Random forest), que fueron estudiados y comparados en la sección 1.6. El algoritmo de clasificación Random Forest es el recomendado por el motor de machine learning DragonFly.

1.7.3.1. Entrenamiento del modelo de machine learning para el motor DragonFly

El proceso de entrenamiento de un modelo de machine learning consiste en proporcionar datos de entrenamiento de los cuales va a aprender un algoritmo de machine learning, es decir, el algoritmo de aprendizaje que por lo general es 20% de los datos del conjunto total y el restante, 80% se utiliza para evaluar el algoritmo de entramiento. (Amazon Web Services, 2021) En el motor DragonFly estos datos de entramiento son obtenidos a través del procesador de entrada que es alimentado por un archivo JSON con la información detectada por parte del IDS Suricata.

Los datos de entrenamiento deben contener la respuesta correcta, que se conoce como destino o atributo de destino. El algoritmo de aprendizaje encuentra patrones en los datos de entrenamiento que asignan los atributos de los datos de entrada al destino (la respuesta que desea predecir) y genera un modelo de machine learning que captura dichos patrones. (Amazon Web Services, 2021) El modelo de machine learning busca mejorar la detección de intrusos de la plataforma OPNids.

1.8. Ventajas y desventajas de OPNids con respecto a plataformas tradicionales IDPS comerciales y open source

En la actualidad es indispensable implementar sistemas de seguridad informática a nivel de hardware y software en cualquier infraestructura con manejo de información sensible y expuesta a internet. Pero no siempre toda institución, organización o empresa poseen los recursos necesarios para implementar soluciones de IDPS comerciales, las cuales varían de precio según su robustez, servicios, herramientas y el fabricante; además que si se trata de Machine Learning los recursos humanos e informáticos aumentan. Las soluciones OpenSource son las llamadas a cubrir estas necesidades, pero dependen del apoyo de sus comunidades de desarrollo para crecer y mejorar continuamente.

1.8.1. Análisis comparativo de los IDPS comerciales versus OPNids

Para un análisis detallado de soluciones comerciales nos basamos en la metodología de la investigación desarrollada por la empresa consultora Gartner y su famoso "Cuadrante mágico de Gartner" que proporciona una gráfica con cuatro campos en los cuales serán ubicados los diferentes fabricantes. Para asignar a cada fabricante a un cuadrante, estos son evaluados bajo dos criterios: *Habilidad de ejecución* e *Integridad de visión*, después de un análisis de estos dos criterios los diferentes fabricantes caen en uno de los 4 cuadrantes que llevan los siguientes nombres: (Astudillo, et al., 2011, pp. 23)

- Líderes: Alta habilidad de ejecución e integridad y visión.
- Competidores: Alta habilidad de ejecución y baja habilidad de visión.
- Visionarios: Baja habilidad de ejecución y alta integridad de visión.
- Jugadores de nicho: bajo en ambos criterios. (Astudillo, Jimenez y Ortiz, 2011, pp. 23)

En la figura 20-1 se observa el cuadrante mágico de Gartner actualizado hasta enero de 2018 con sus diferentes fabricantes, ubicando a: Cisco con la compra de la empresa SourceFire fortaleciendo su división de seguridad, Mc Afree con la compra de la empresa StoneSoft complementando su IDPS Network Security Platform y Trend Micro con la compra de HP TipppingPoint, como los lideres IDPS comerciales del mercado. Gartner en su estudio dice:

- Cisco tiene una amplia cartera de productos de seguridad y ha tenido ofertas IDPS durante muchos años. La adquisición de Sourcefire ha continuado siendo una influencia positiva y fuerte en la cartera de seguridad de red de Cisco, dando a la compañía tracción en el mercado de cortafuegos que no habría obtenido de otra manera. (Gartner, 2018)

- McAfee tiene una importante cartera de productos en red, servidor, nube, web, información de seguridad y gestión de eventos (SIEM), análisis de red, prevención de pérdida de datos (DLP) y seguridad de endpoints. Su IDPS, llamada Network Security Platform (NSP), es un elemento principal de su oferta de productos de seguridad de red. (Gartner, 2018)

- Trend Micro es un gran proveedor de seguridad de TI global. El IDPS también se beneficia de las sinergias entre tippingpoint y los equipos de investigación de Trend Micro sobre malware, que está mejorando la capacidad del IDPS para abordar específicamente los elementos basados en la red de amenazas de malware. Además, la tecnología de amenazas avanzadas trend micro (sandbox) para su IDPS, llamada Deep Discovery, ahora tiene

integraciones a su IDPS para poder recibir telemetría en tiempo real que se puede utilizar para casos de prevención y detección de casos de uso. (Gartner, 2018)

Figura 20-1: Cuadrante mágico para Sistemas de detección y prevención de intrusiones

Fuente: (Gartner, 2018)

En la Tabla 8-1 se comparan algunas de las principales características de los IDPS comerciales lideres versus Suricata-OPNids el propuesto en el presente trabajo de titulación.

Tabla 8-1: Comparación de IDPS comerciales vs OPNids

Características	IDPS comerciales (Cisco, Mc Afree y Trend Micro)	OPNids
Soporte	Si (24/7)	Comunidad
Machine Learning	Mc Afree, Trend Micro	Si
Protección e-mail	Cisco	No
Protección Cloud	Cisco, Mc Afree, Trend Point	No
Multi-Threading	No	Si
Soporte para IPv6	Si	Si
IP Reputation	Si	Si

Tabla 8-1 (continuación)

Detección automática de protocolos	Si	Si
Aceleración con GPU	No	Si
URL Reputation	Si	Si
GeoIP	Mc Afree	Si
Análisis avanzado de HTTP	Cisco, Mc Afree	Si
Detección de anomalías	Si	Si
Alta disponibilidad	Si	No
Escalabilidad	Si	Si
Interfaz gráfica de administración	Si	Si

Fuente: (Astudillo, et al., 2011, pp. 26)

Realizado por: Herrera, Jonathan 2021.

El estudio realizado por la empresa Gartner sitúa a Cisco, Mc Afee y Trend Micro como los líderes IDPS comerciales, en base a este estudio se analizó y realizo una comparativa de las características similares entre estos IDPS con la plataforma OPNids, en la tabla 8-1 se presentan las características y se encuentra muchas similitudes entre ellas. Cada IDPS comercial cuenta con características únicas y comunes que los hacen fuertes en diferentes campos de la seguridad, pero todo ellos buscan abarcar gran parte de las necesidades que una empresa u organización necesitan. Se debe considerar que un IDPS comercial posee una ventaja principal que es el soporte; cuando una empresa te vende el servicio en realidad te ofrece la herramienta con soporte 24/7 para resolver cualquier inconveniente en la brevedad posible, a diferencia de cualquier IDS/IPS Open Source que dependen de la comunidad para la resolución de problemas como lo es Suricata, que obtiene su soporte de la Open Information Security Foundation (OISF), pero que a pesar de ser una comunidad sin fines de lucro es una de las más activas.

El estudio revela que los IDPS lideres han evolucionado en su habilidad de ejecución e integridad de visión, hacia la utilización de técnicas innovadoras como protección de cloud, escalabilidad, inteligencia ante amenazas y destacando entre todas ellas es la integración de machine learning en las plataformas Network Security Platform (Mc Afee) y Tipping Point (Trend Micro). Las características del machine learning son orientados hacia la detección de amenazas de día cero que son cambiantes por naturaleza y avance en técnicas de los atacantes.

La plataforma OPNids después del análisis comparativo de las características realizado en los párrafos anteriores en base al estudio realizado por Gartner se concluye que presenta las ventajas de su licencia Open Source, Suricata IDS integrado con el motor de machine learning DragonFly y comunidad activa versus las desventajas de la falta de soporte inmediato, protección en Cloud y email, mejores técnicas integradas de machine learning e integración de nuevas herramientas

de control y análisis de seguridad por parte de los IDPS comerciales. Los IDPS comerciales y la plataforma OPNids presentan varias herramientas funcionales para el análisis, control y monitoreo del tráfico de datos de una empresa, organización o universidad, pero se debe analizar la infraestructura de red y los servicios que van a ser controlados, por esta razón los IDPS comerciales son perfectos para infraestructuras grandes y concisas que además cuenten con el financiamiento para su compra, caso contrario siempre los IDS/IPS Open Source serán los escogidos.

1.8.2. Análisis comparativo de los IDS/IPS Open Source versus OPNids

En el presente trabajo de titulación nos enfocamos en el uso de IDS/IPS Open Source, por esta razón es necesario analizar las características entre la plataforma Suricata-OPNids y sus principales rivales como son Snort y Zeek, este último conocido anteriormente como Bro. A continuación, en la Tabla 9-1 se observa la comparación de las características entre las herramientas mencionadas.

Tabla 9-1: Comparación de características Zeek vs Snort vs Suricata-OPNids

Características	Zeek	Snort	Suricara-OPNids
Soporte - Comunidad	Si	Si	No
Multi-Threading	No	No	Si
Machine learning	No (Script)	No	Si
Detección automática de protocolos	Si	No	Si
Aceleración con GPU	No	No	Si
Multiplataforma	No	Si	Si
Detección de anomalías	Si	Si	Si
Alta disponibilidad	No	Si	Si
Escalabilidad	Si	Si	Si
Interfaz gráfica de administración	No	No	Si

Fuente: (Astudillo, et al., 2011, pp. 28)

Realizado por: Herrera, Jonathan 2021.

En la tabla 9-1 se observa las características más importantes de cada IDS/IPS y se obtiene como resultado que la plataforma Suricata-OPNids es la mejor punteada seguida por Snort y Zeek respectivamente. Suricata-OPNids y Snort comparten gran similitud en sus capacidades y son considerados los mejores IDS/IPS Open Source del mercado.

Las principales ventajas de Suricata-OPNids frente a los IDS/IPS mencionados son 3 características muy importantes:

- Multi-Threading o multi hilo: se puede configurar el número de procesadores a utilizar en el análisis de tráfico de datos, es decir, aumenta la capacidad para el procesamiento del tráfico de datos de la red.
- Aceleración por hardware de video: la utilización de la tarjeta de video para el rastreo de redes.
- Motor de machine learning Dragon Fly: el motor de machine learning a través de sus algoritmos supervisados de clasificación pretende mejorar la detección de ataques informáticos a través del IDS Suricata en la intranet del edifico de la FIE.

Después de analizar las ventajas de Suricata-OPNids frente a las plataformas tradicionales IDS/IPS comerciales y Open Source. La principal desventaja del IDS Suricata frente a Snort y Zeek es la utilización de recursos informáticos, ya que Suricata al utilizar procesamiento multi hilo puede llegar a sobrecargar el procesador, además que Zeek es capaz de implementar procesos de machine learning, aun así, esto presenta costos en recursos humanos e informáticos, ya que no existe estudios sobre una integración de la técnica machine learning a la plataforma en producción, solo existen pequeños ejemplos de laboratorio. Otra de las plataformas famosas en el mercado Open Source con características simulares a OPNids, es Security Onion una suite dirigida hacia la seguridad informática que consta con varias herramientas, en la tabla 10-1 se detalla una comparativa de sus principales características.

Tabla 10-1: Comparación de características Security Onion vs OPNids

Características	Security Onion	OPNids
Soporte	Si	No
NIDS/IPS	Suricata/Zeek	Suricata
Multi-Threading	Si	Si
Machine Learning	No (script)	Si
Detección automática de protocolos	Si	Si
Aceleración con GPU	Si	Si
Multiplataforma	Linux	Linux
Detección de anomalías	Si	Si
Alta disponibilidad	Si	Si
Escalabilidad	Si	Si
Interfaz gráfica de administración	Si (Kibanna)	Si (OPNsense)

Fuente: (Security Onion Solutions, 2021) (OPNids, 2019)
Realizado por: Herrera, Jonathan 2021.

Las dos plataformas poseen características simulares, la ventaja de Security Onion es el soporte que posee la herramienta, al contrario, OPNids dejo de tener soporte a inicios de 2020. Sin embargo, la ventaja de OPNids otra vez es su motor de machine learning DragonFly teniendo en cuenta que como se explicó en párrafos anteriores Zeek posee la capacidad de integrar la técnica machine learning, pero de manera oficial Security Onion lo considera como un proceso externo, es decir, que el usuario debe realizar la integración, lo que supone gastos en recursos humanos e informáticos hacia la organización, empresa o universidad que desee implementar dicha técnica.

Sin embargo, para obtener un resultado cuantitativo del análisis de las ventajas y desventajas de las características realizado con respecto a los IDS/IPS Open Source versus OPNids, se procedió a realizar una encuesta online con la metodología de investigación de una escala de Likert de tipo "Importancia" y se obtuvo los siguientes resultados. Un ejemplo de la encuesta se encuentra en el anexo F.

Se procedió a realizar una encuesta online basada en las características comparadas en las tablas 9-1 y 10-1, se dimensionó de manera general las características para plantear las 10 preguntas correspondientes y obtener resultados. La encuesta online se realizó a 30 personas con criterio formado con respecto a plataformas tradicionales de IDS/IPS y conocimiento afín.

En el Grafico 4-1 se observa la frecuencia global de la encuesta en la que se aprecia los porcentajes obtenidos en la escala de Likert, en los que las opciones de "Importante" y "Muy importante" destacan con 38% y 34% respectivamente, la opción neutra obtiene un 18% y las opciones negativas "Poco importante" y "No es importante" obtienen un 8 % y 2% respectivamente.

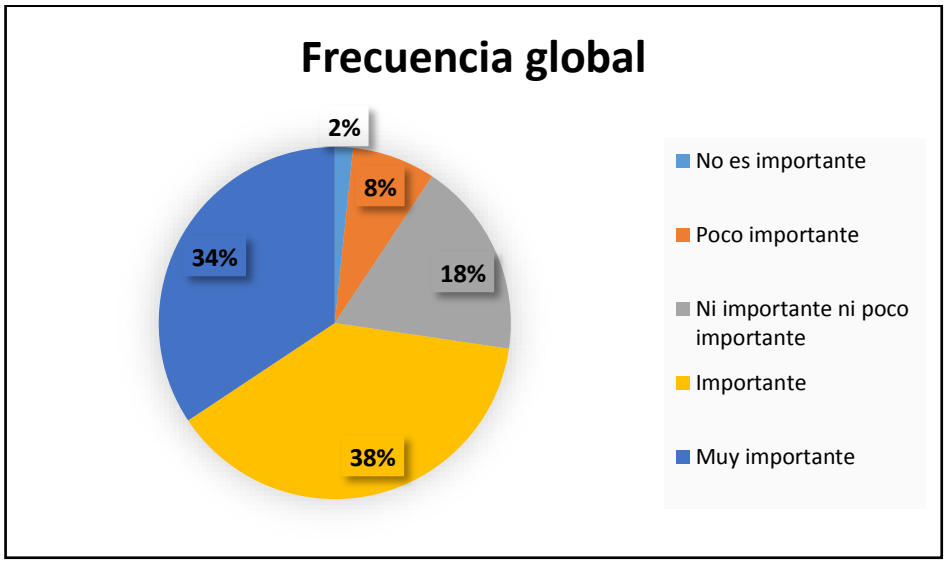


Gráfico 4-1: Frecuencia global de la escala de Likert

Realizado por: Herrera, Jonathan 2021.

Sin embargo, estos porcentajes no nos permiten analizar puntualmente cada una de las características encuestadas para poder compararlas con el estudio realizado previamente. A continuación, se procedió a calcular la frecuencia absoluta de cada pregunta según los niveles de importancia, es decir, la escala de Likert realizada en la encuesta online como se observa en la tabla 11-1. En cada pregunta se calculó la frecuencia con la que el encuestado respondió con respecto a los niveles de importancia, esto nos ayudará a calcular el valor de la encuesta por pregunta y relacionarlo con las características de los IDS/IPS Open Source.

Tabla 11-1: Frecuencia absoluta de cada pregunta según los niveles de importancia

Niveles de Importancia	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
No es importante	0	1	0	0	0	1	0	2	0	1
Poco importante	0	3	1	2	8	3	0	3	1	2
Ni importante ni poco importante	0	0	3	11	12	10	2	3	4	9
Importante	13	13	9	6	8	9	19	13	15	10
Muy importante	17	13	17	11	2	7	9	9	10	8

Realizado por: Herrera, Jonathan 2021.

Como ejemplo de la frecuencia del encuestado, se observa el grafico 5-1 que corresponde a la pregunta 5 de carácter neutra y representa la conducta de los 30 encuestados.

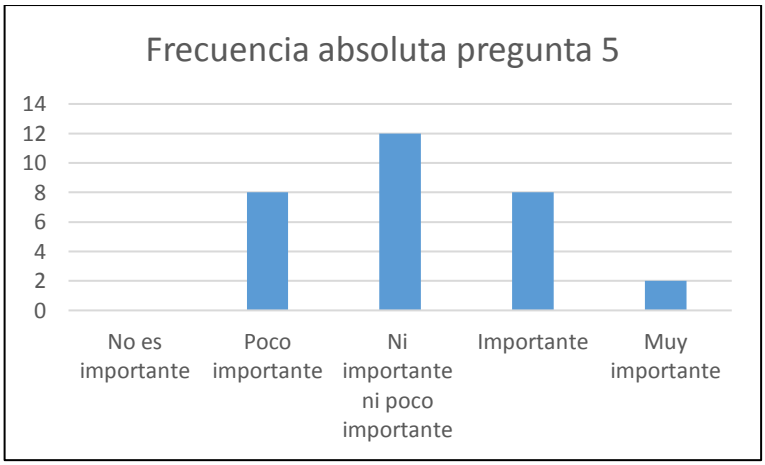


Gráfico 5-1: Frecuencia absoluta de la pregunta 5

Realizado por: Herrera**,** Jonathan 2021.

En la tabla 12-1 se observa la comparación de los resultados obtenidos a través de la encuesta y del cálculo de la frecuencia absoluta por cada pregunta.

Tabla 12-1: Comparación de resultados entre valor estudiado y encuestado

	Plataformas IDS/IPS							
Características	**Snort**		**Zeek**		**Security Onion**		**OPNids**	
	V. Es	V. En	V. Es	V. En	V. Es	V. En	V. Es	V. En
Soporte	1	1	1	1	1	1	0	0
Recursos informáticos	0	0	0	0	1	0.8667	1	0.8667
Machine Learning	0	0	1	0.8667	0	0	1	0.9667
Detección automática de protocolos	0	0	1	0.9333	1	0.9333	1	0.9333
Aceleración GPU	0	0	0	0	1	0.7333	1	0.7333
Multiplataforma	1	0.8667	0	0	0	0	0	0
Detección de anomalías	1	0.9333	1	0.9333	1	0.9333	1	0.9333
Disponibilidad	1	0.7333	0	0	1	0.7333	1	0.8333
Escalabilidad	1	0.8333	1	0.8333	1	0.8333	1	0.8333
GUI de administración	0	0	0	0	1	0.6	1	0.6
Total	**5**	**4.37**	**5**	**4.57**	**7**	**6.63**	**7**	**6.70**

Realizado por: Herrera, Jonathan 2021.

Después del cálculo de la frecuencia absoluta de cada pregunta realizada como se observa en la tabla 11-1, se procedió a relacionar cada pregunta con su respectiva característica de los IDS/IPS como se observa en la tabla 12-1. El valor estudiado fue comparado con el valor encuestado, en las 4 plataformas IDS/IPS evaluadas se obtuvieron los siguientes resultados: Snort obtuvo un valor estudiado de 5 versus 4.37 encuestado, Zeek obtuvo un valor estudiado de 5 versus 4.57 encuestado, Security Onion obtuvo un valor estudiado de 7 versus 6.63 encuestado y OPNids obtuvo un valor estudiado de 7 versus 6.70 encuestado. Los resultados nos confirman que existe una aproximación entre los valores estudiados y encuestados confirmando un análisis correcto.

Las plataformas Security Onion y OPNids obtuvieron los valores más altos y con muy poca diferencia entre ellos, lo que concluye que cualquiera de las dos plataformas IDS/IPS pueden ser implementadas en cualquier infraestructura de red que se desea analizar su tráfico de datos. Sin embargo, en el presente trabajo de titulación nos enfocamos hacia la utilización de técnicas machine learning para el análisis de tráfico de datos malicioso, por esta razón la plataforma OPNids con su motor de machine learning DragonFly integrado es la primera opción para implementar en la red del edifico de la FIE.

.

CAPÍTULO II

2. MARCO METODOLÓGICO

2.1. Introducción

En este apartado se especifica la metodología utilizada en el presente trabajo de titulación que consta de tres etapas como se observa en el Grafico 1-2. Las tres etapas nos permitirán cumplir con cada uno de los objetivos propuestos utilizando la información investigada previamente en el capítulo 1, lo cual nos permite desarrollar los objetivos correctamente.

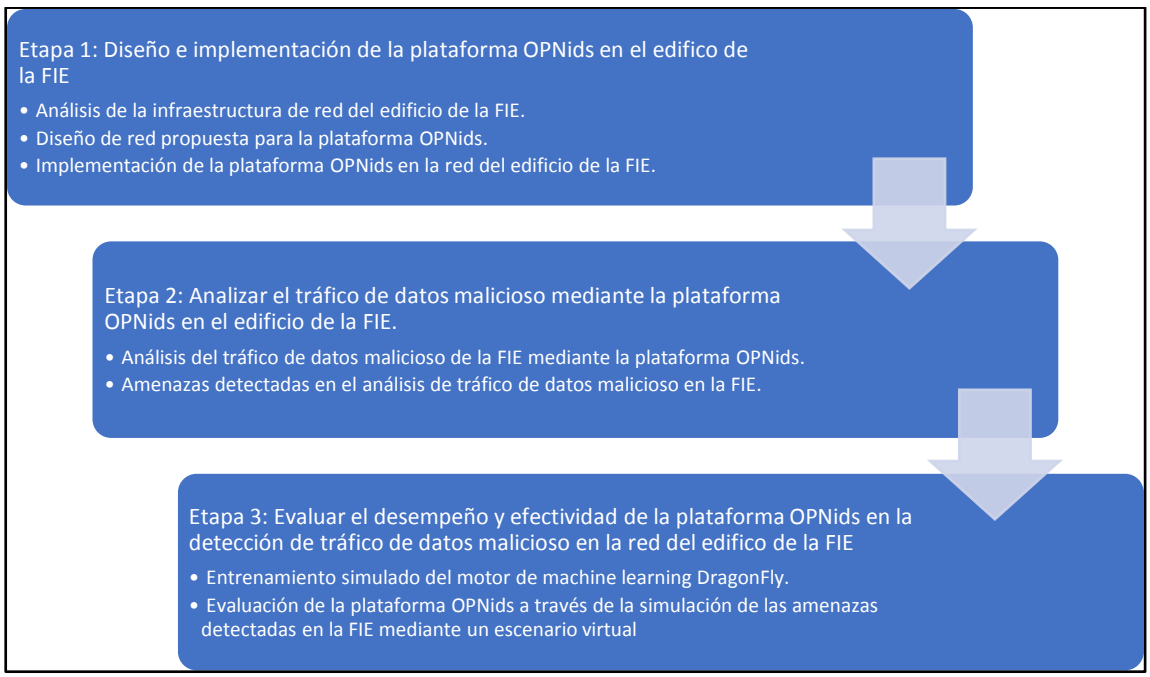


Gráfico 1-2: Etapas para el desarrollo del trabajo de titulación

Realizado por: Herrera, Jonathan 2021.

2.2. Etapa 1: Diseño e implementación de la plataforma OPNids en el edificio de la FIE.

En esta etapa se realiza un análisis de la infraestructura de red implementada en el edificio de la FIE, con la finalidad de recabar toda la información necesaria para realizar el diseño de red e

implementar la plataforma OPNids para el análisis del tráfico de datos malicioso en el edificio de la FIE.

2.2.1. Análisis de la infraestructura de red del edificio de la FIE

Para diseñar e implementar la herramienta OPNids se realizó una inspección de la infraestructura de red junto al personal técnico encargado del edificio de la FIE – ESPOCH, la Ing. Ruth Barba cumpliendo las funciones de "técnico-docente" nos proporcionó la información tanto de hardware como de software existente.

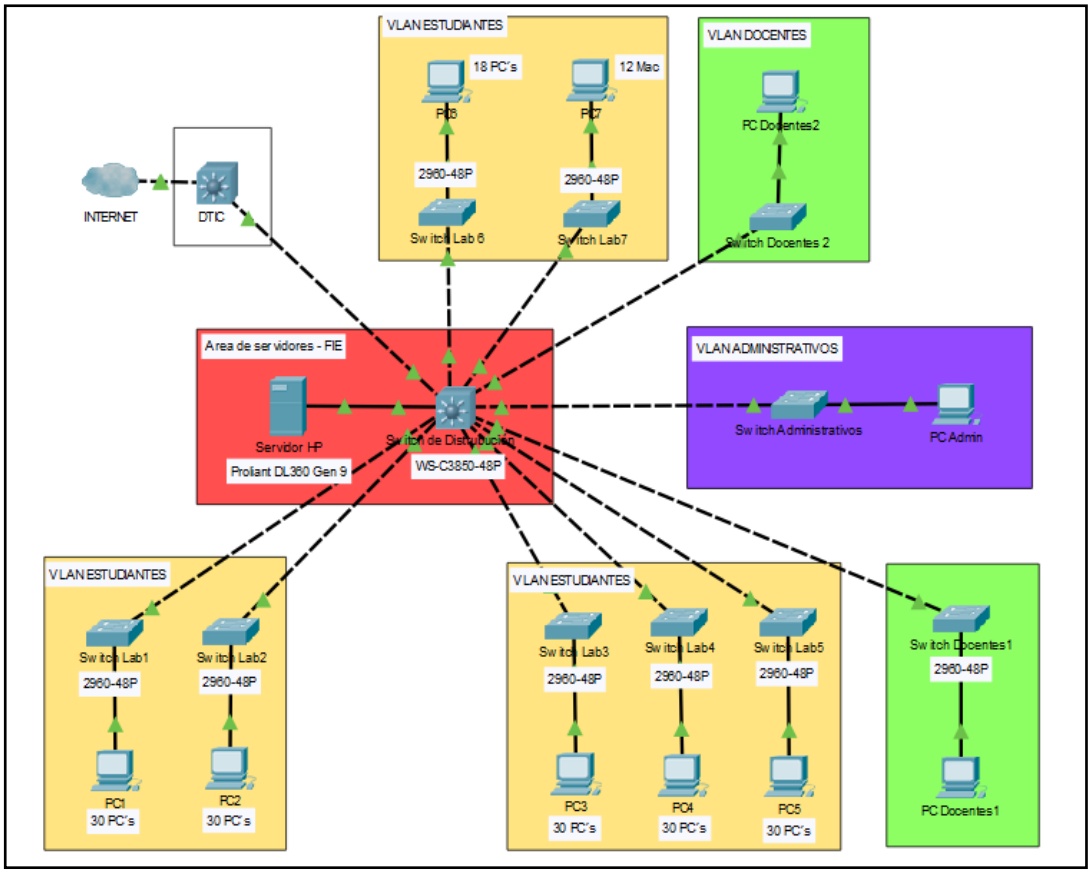


Figura 1-2: Topología de red estrella del edificio de la FIE

Realizado por: Herrera, Jonathan 2021.

El edificio de la FIE tiene una topología de red estrella como se observa en la Figura 1-2 y se estructura de la siguiente manera:

Todo el equipamiento en su mayoría es de marca CISCO distribuido en su infraestructura física, la FIE consta de 9 aulas, 7 laboratorios de cómputo, 2 salas de profesores y oficinas administrativas. El área de servidores es el corazón de la infraestructura; el cual se encuentra

equipado con varios switches y servidores que son gestionados por la Dirección de Tecnologías de la Información y Comunicación (DTIC).

Un switch WS-3850 de 48 puertos denominado *"switch de distribución"* es el encargado de interconectar hacia los diferentes switchs 2960-48P ubicados en los espacios mencionados, estos switchs son conocidos como *"switch de acceso"*. La red posee una configuración con 3 VLAN y cada una con un rango de direccionamiento IP explicado en la en la Tabla 1-2:

Tabla 1-2: Rango de IP de las VLAN de la FIE

VLAN	Rango de direcciones IP
Estudiantes (VLAN 200)	172.25.200.1-172.25.203.254 172.25.202.156-172.25.202.159
Docentes (VLAN 204)	172.25.204.231-172.25.204.254
Administrativos (VLAN 205)	172.25.205.114-172.25.205.254

Realizado por: Herrera, Jonathan 2021.

Las Vlan´s mencionadas son el objetivo de análisis de la plataforma OPNids ya que por ellas se conduce todo el tráfico de datos y por ende posibles intentos de ataques informáticos hacía el edificio de la FIE.

2.2.1.1. Sistemas de seguridad en tiempo real del edificio de la FIE

La FIE en su infraestructura de red consta de gran equipamiento como lo demuestra la sección anterior, cabe recalcar que sus servidores no contienen ningún sistema de seguridad en tiempo real encargados de monitorear el comportamiento de la red ante cualquier amenaza o ataque informático que pueda exponer una vulnerabilidad de cualquier tipo, lo que por consiguiente crea una potencial brecha de seguridad de la intranet de la FIE, dicha afirmación es verificada por la "técnico-docente" encargada.

Una red toma valor o se vuelve objetivo para un atacante cuando existen activos de información, es decir, bases de datos de usuarios, información confidencial, sistemas académicos, entre otros. Las principales amenazas son ocasionas por errores humanos, factores naturales o fallas técnicas. Por esta razón toda red por más pequeña que está sea y con mayor razón la FIE debe implementar un sistema de seguridad en tiempo real. La FIE no cuenta con ningún sistema de seguridad en tiempo real, dejando expuesta y vulnerable la intranet para que todo tipo de tráfico de datos circule siendo esto potencialmente peligroso, por esta razón la implementación de la herramienta OPNids para el análisis de tráfico de datos malicioso es de gran importancia para mantener un control y análisis de la red.

2.2.1.2. ¿Porque OPNids?

OPNids es una solución Open Source frente a otros IDS del mercado por sus herramientas integradas en la plataforma. OPNids busca ser una plataforma para estimular el desarrollo y la implementación de modelos sofisticados de aprendizaje automático en problemas de ciberseguridad. Además de sus prestaciones mencionadas posee uno de los mejores IDS Open Source del mercado que es Suricata con características como rapidez, robustez, gratuito, compatibilidad con varios sistemas operativos, capacidad de detección de intrusos en tiempo real, procesamiento de pcaps offline. Las alertas generadas por Suricata se pueden visualizar a nivel de consola o con herramientas más interactivas como Elasticsearch, Kibana, Splunk, entre otros. Suricata emite sus alertas a través del análisis de tráfico de datos por medio de firmas y reglas aportadas por la comunidad OISF (Open Information Security Foundation) un factor fundamental es que se puede crear reglas acordes a las amenazadas detectadas en la red y así corregir las vulnerabilidades este proceso es apoyado para un mejor rendimiento gracias al motor de machine learning; el factor preponderante para la elección de la plataforma OPNids es DragonFly MLE su integración conjunta con Suricata es la única en el mercado y por consiguiente el desarrollo correcto de la herramienta permitirá un mejor rendimiento en eficacia y eficiencia frente a la detección de ataques informáticos en la red del edificio de la FIE.

2.2.2. Diseño de red propuesta para la plataforma OPNids

Se realizo el diseño de red para la infraestructura del edificio de la FIE teniendo en cuenta la topología de red estrella mostrada en la Figura 1-2 y la información analizada en la sección 2.2.1. La solución propuesta se puede observar en la Figura 2-2, que consiste en la implementación de la plataforma OPNids en el servidor HP Proliant DL360 ubicado en el área de servidores, además de la configuración del *switch de distribución* en el cual se creó una sesión de monitoreo de todo el tráfico de datos de entrada de las 3 vlans que fue reflejado hacia los puertos P1, P5 y P16 respectivamente para su posterior análisis de tráfico de datos de la intranet de la FIE. La configuración se la realizó a través de las siguientes líneas de comando con el permiso otorgado por parte del DTIC-ESPOCH:

monitor session 2 destination interface Gi1/0/1, Gi1/0/5, Gi1/0/16 encapsulation replicate dot1q vlan 200
monitor session 2 destination interface Gi1/0/1, Gi1/0/5, Gi1/0/16 encapsulation replicate dot1q vlan 204

monitor session 2 destination interface Gi1/0/1, Gi1/0/5, Gi1/0/16 encapsulation replicate dot1q vlan 205

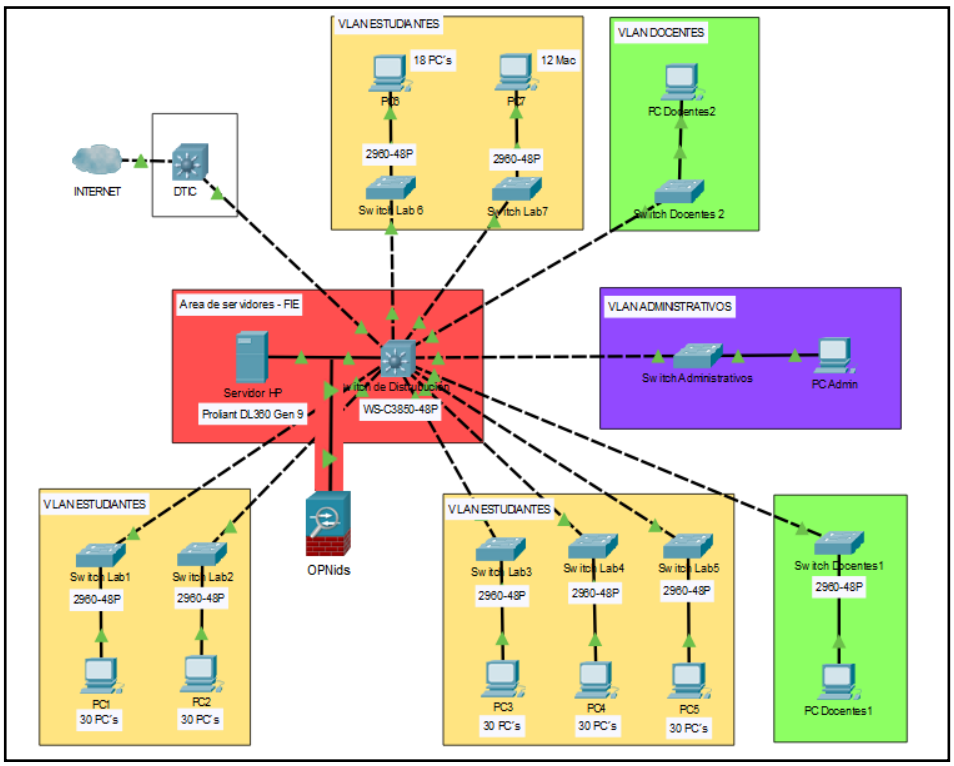


Figura 2-2: Diseño de red propuesto para la implementación de OPNids

Realizado por: Herrera, Jonathan 2021.

2.2.2.1. OPNids como plataforma de seguridad del edificio de la FIE

Para realizar el análisis del tráfico de datos y detectar posibles amenazas e intentos de intrusión se utilizó el sistema de detección de intrusos de red Suricata IDS en conjunto con su motor de machine learning DragonFly, integrados dentro de la plataforma OPNids que se implementó en las máquinas virtuales creadas en el servidor HP Proliant DL360 para el análisis de tráfico de datos de las Vlans. Cada máquina virtual posee dos adaptadores de red virtuales, configurados como se observa en la Tabla 2-2.

Tabla 2-2: Adaptadores de red virtuales

Adaptadores de red virtuales	Configuración
Adaptador de red ens160	Se encuentra configurado la dirección IP correspondiente a cada Vlan.
Adaptador de red ens192	Se encuentra conectado a 3 puertos SPAN (mirror), uno por cada Vlan que reciben todo el tráfico de datos.

Realizado por: Herrera, Jonathan 2021.

2.2.3. Implementación de la plataforma OPNids en la red del edificio de la FIE

2.2.3.1. Requerimientos técnicos para la implementación del servidor con OPNids

El servidor HP Proliant DL360 proporcionado por la FIE y el grupo SEGINTE se observa en la Figura 3-2 y posee las siguientes características detalladas en la siguiente Tabla 3-2.

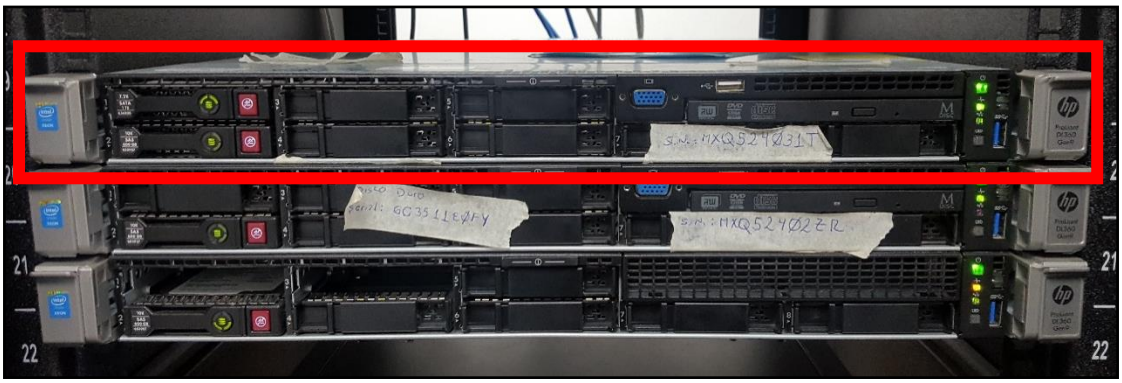


Figura 3-2: Servidor HP Proliant DL360 Gen9

Realizado por: Herrera, Jonathan 2021.

Tabla 3-2: Características del servidor HP

Servidor HP Proliant DL360 Gen 9	Características
Núcleos CPU	8 CPUs x 2.397 GHz
Tipo de procesador	Intel ® Xeon ® CPU E5-2630 v3 2.40GHz
RAM	16 Gb DRR4 SmartMemory
Tarjetas de red	1 NIC 331i – 4 puertos de 1Gb
Almacenamiento	1 HDD SAS 600GB, 1 HDD SATA 1TB
Hipervisor	ESXi v5.5.0

Realizado por: Herrera, Jonathan 2021.

El servidor HP tiene conexión al switch de distribución hacia los puertos P1, P5 y P16 pertenecientes a la Vlan: Estudiantes, Docentes y Administrativos respectivamente. Los puertos tienen la configuración de SPAN o puerto promiscuo para transmitir todo el tráfico de datos de las vlans hacia la herramienta OPNids, en la Figura 4-2 se observa la conexión marcada en los recuadros de color rojo.

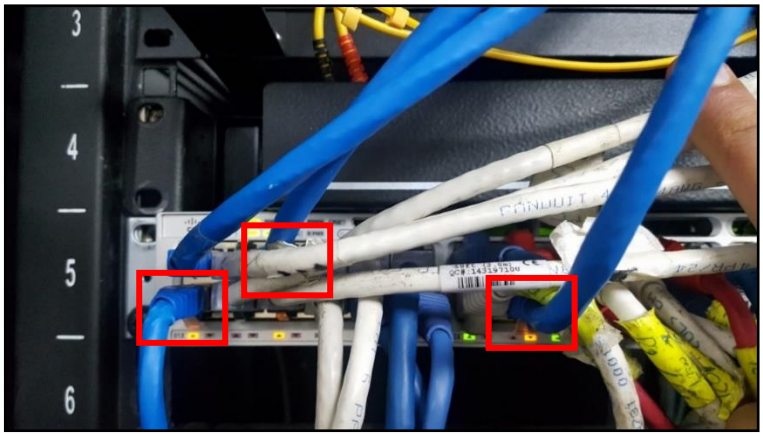


Figura 4-2: Switch Cisco WS-C380

Realizado por: Herrera, Jonathan 2021.

Los requisitos para la herramienta OPNids se listan en la tabla Tabla 4-2.

Tabla 4-2: Requisitos de OPNids

OPNids	Requisitos
Núcleos CPU	4 CPUs x 2.0 GHZ o superior
Tipo de procesador	Intel ® o AMD
RAM	4 – 16 Gb
Tarjetas de red	2
Almacenamiento	120 Gb mínimo, 300 Gb recomendado

Realizado por: Herrera, Jonathan 2021.

Como se observa en las tablas anteriores los requisitos necesarios para el funcionamiento de la herramienta OPNids podrían ocupar un gran número de recursos de hardware del servidor, esto dependerá en gran medida de que cantidad de eventos tenga que procesar y que servicios van a estar activos. En el servidor HP Proliant DL360 Gen 9 se encuentra instalado el hypervisor VMware ESXi v5.5.0 como se observa en la Figura 5-2.

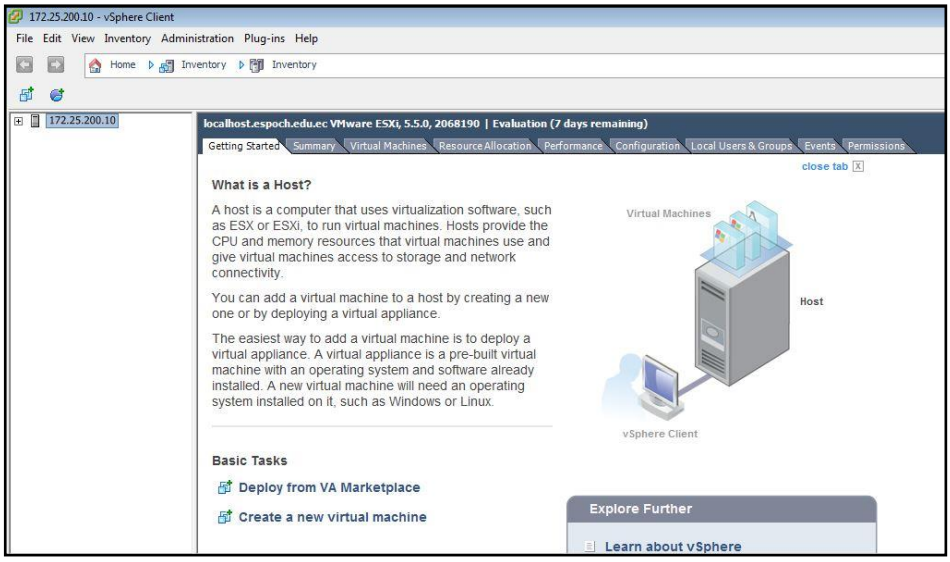


Figura 5-2: Hypervisor VMware EXSi en el servidor HP

Realizado por: Herrera, Jonathan 2021.

2.2.3.2. Instalación de la plataforma OPNids

La herramienta OPNids se instalará y ejecutará de forma virtual desplegada sobre el hypervisor en el servidor HP, sin embargo, esta opción puede presentar algunos problemas en su rendimiento, es decir, que al tratarse de un servidor virtualizado los recursos de hardware (CPU, RAM, RED,etc) serán compartidos para todas las máquinas virtuales presentes en el servidor. Es recomendable que este tipo de herramientas sean instaladas de forma nativa o como en este caso una correcta configuración y limitación de los servicios activos dentro de la herramienta. Se utilizará la versión de OPNids 18.9-amd64 FreeBSD 11.1-RELEASE-p11, disponible en la siguiente dirección: https://opnids.io/donwloads

Para la implementación se requiere de dos interfaces de red, una para administración (MGT) y otra el análisis de tráfico (TAP), configuradas como se muestra en la Tabla 5-2.

Tabla 5-2: Configuración de IP de OPNids

Interfaz	Descripción	IP
Vmx0 (MGT)	La interfaz MGT sirve para gestionar la herramienta a través del navegador, también se puede conectar por ssh. La dirección IP puede ser por DHCP o estática.	172.25.200.166
Vmx1 (TAP)	La interfaz de monitoreo en modo promiscuo (no posee dirección IP) para capturar todo el tráfico de la red.	Sin dirección

Realizado por: Herrera, Jonathan 2021.

Una vez estudiado el hardware, se debe crear la máquina virtual con los requisitos expresados en la Tabla 4-2, se procede a la instalación de la herramienta OPNids a través de una imagen ISO cargada en la unidad de CD/DVD de la máquina virtual.

Al encender la máquina virtual arranca el sistema operativo, en la Figura 6-2 se presenta un menú con 6 opciones, se debe presionar la barra espaciadora para pausar el temporizador y seleccionar la opción que se desea, las opciones se explican en la Tabla 6-2.

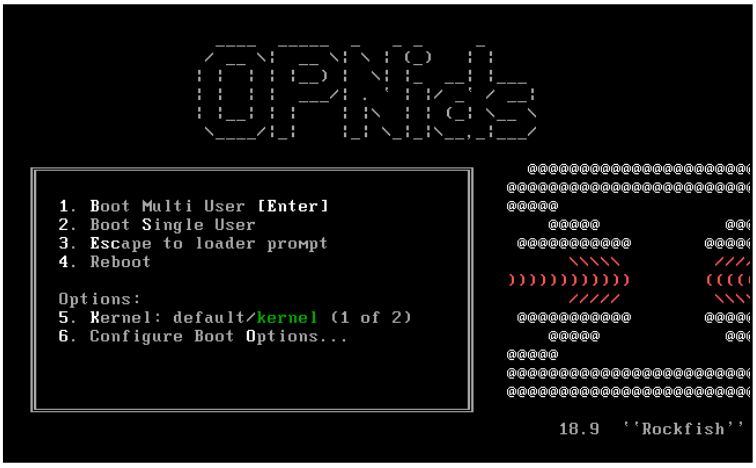


Figura 6-2: Menú de arranque OPNids

Realizado por: Herrera, Jonathan 2021.

Tabla 6-2: Opciones de arranque de OPNids

#	Opción	Descripción
1	**Boot Multi User**	Modo de arranque por defecto.
2	**Boot Single User**	Modo de arranque para reparar una instalación de OPNids-FreeBSD existente.
3	**Escape to loader prompt**	Inicia el sistema con un mensaje para reparación de errores con comandos de bajo nivel.
4	**Reboot**	Reiniciar el sistema.
5	**Kernel: default/kernel (1 of 2)** **Kernel: kernel.old (2 of 2)**	Selecciona el kernel actual de FreeBSD. Selecciona un kernel antiguo de FreeBSD.
6	**Configure Boot Options**	En este menú existen modos de arranque ON/OFF.

Fuente: (FreeBSD, 2019)

Realizado por: Herrera, Jonathan 2021.

A continuación, se procede a escoger la opción 1 para arrancar la herramienta OPNids, se presentan varios mensajes mientras lleva acabo una exploración de hardware y carga el SO,

realiza una preconfiguración por defecto para inicializar los servicios e interfaces de red. La Figura 7-2 presenta el prompt de inicio de sesión de la plataforma con las opciones de "root" que consiste un inicio en modo LIVE idealmente para pruebas e "installer" para instalar en el disco virtual de forma permanente. Se inicia sesión con las credenciales login: "installer" password: "opnids" para continuar con la instalación.

Figura 7-2: Modo live o instalador

Realizado por: Herrera, Jonathan 2021.

La siguiente ventana nos presenta un mensaje de bienvenida con anuncios de que se realizaran preguntas necesarias al momento de instalar OPNids, presionamos la tecla "enter". La configuración de consola que se muestra en la Figura 8-2, con las opciones de cambiar el mapa de teclas y escoger una configuración de video, se asume los valores por defecto y seleccionamos la primera opción "Accept these settings".

Figura 8-2: Configuración de consola

Realizado por: Herrera, Jonathan 2021.

En la Figura 9-2 se debe seleccionar el tipo de instalación que se desea seguir, escogemos la instalación guiada y presionamos la tecla “enter”.

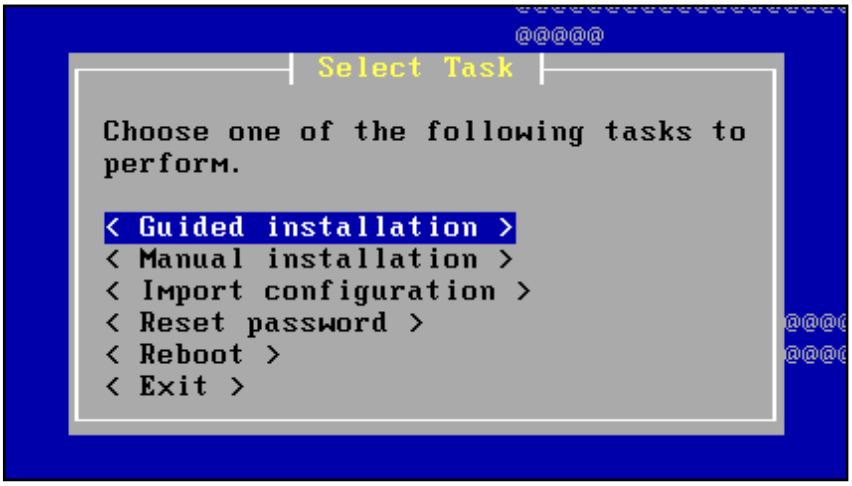


Figura 9-2: Tipos de instalación

Realizado por: Herrera, Jonathan 2021.

Se debe seleccionar el disco virtual en el cual va a instalar, en caso de existir varias unidades de almacenamiento, en este caso como se observa en la Figura 10-2 presenta un único disco virtual.

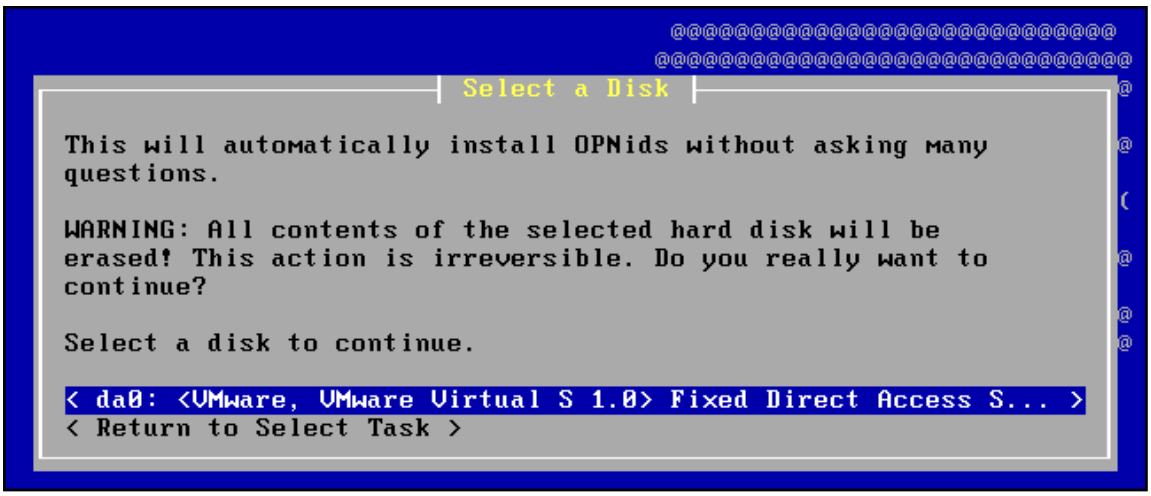


Figura 10-2: Selección del almacenamiento

Realizado por: Herrera, Jonathan 2021.

El siguiente paso es escoger un modo de instalación como se observa en la Figura 11-2, la opción GPT/UEFI es la recomendada a diferencia de la opción MBR que es para hardware antiguo, seleccionamos la opción recomendada y al pulsar la tecla “enter” comienza el proceso de instalación con un tiempo de instalación de 10 minutos, el mismo puede variar dependiendo los recursos disponibles.

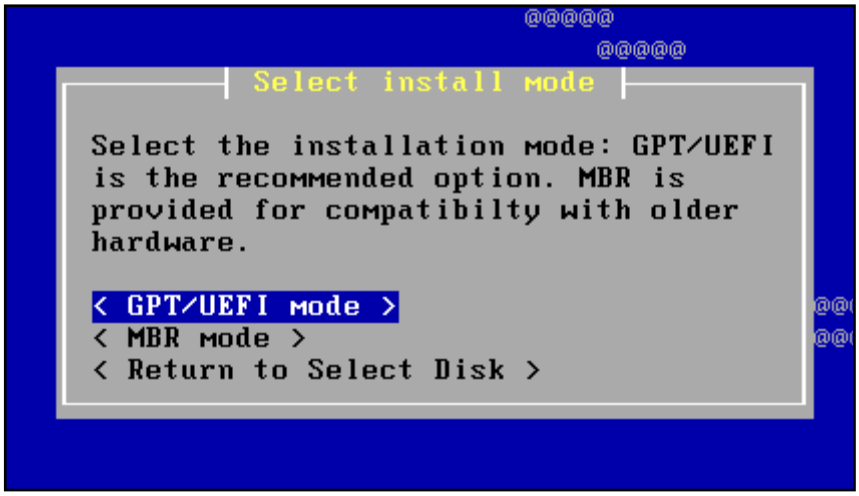


Figura 11-2: Modos de instalación

Realizado por: Herrera, Jonathan 2021.

Una vez finalizada la instalación nos presenta la posibilidad de crear una nueva contraseña de "root", en caso de dejarlo vacío la contraseña seguirá siendo la misma que se utilizó al principio de la instalación. Para finalizar la instalación solicita un reinicio del sistema y retirar la imagen ISO del adaptador CD/DVD y arrancar desde el HDD, después de cargar todo el sistema operativo de OPNids e iniciar sesión con las credenciales mencionadas en párrafos anteriores se muestra un menú de configuración en línea de comandos. La Figura 12-2 indica la IP de la interfaz de administración (MGT) desde la cual se maneja una interfaz gráfica para configurar todos los servicios de OPNids.

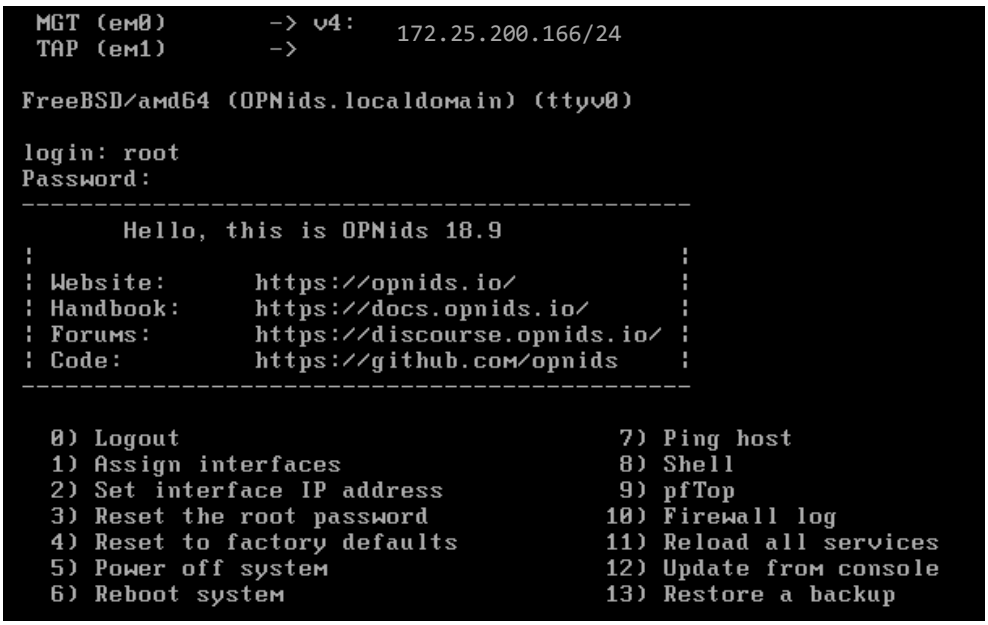


Figura 12-2: Menú principal de OPNids

Realizado por: Herrera, Jonathan 2021.

2.2.3.3. Configuración de las interfaces y herramientas de análisis de la plataforma OPNids

En esta sección como su nombre lo dice nos enfocamos en la configuración de las interfaces y lo más importante para esta investigación las herramientas de análisis de OPNids como son Suricata IDS y Dragonfly MLE.

La IP de administración (MGT) se la configura en línea de comandos, para realizar este proceso se presiona la opción 2 a continuación, como se observa en la Figura 13-2 se presentan las 2 interfaces, escogemos la opción 1, se muestra una pregunta si se desea configurar una dirección IPv4 a través de DHCP en esa interfaz, con las opciones yes "y" o no "n"; se escribe la letra "y" y pulsamos la tecla "enter", de igual forma se puede configurar para IPv6 pero en la en la FIE no existe tal direccionamiento, por esta razón se escribe la opción "n".

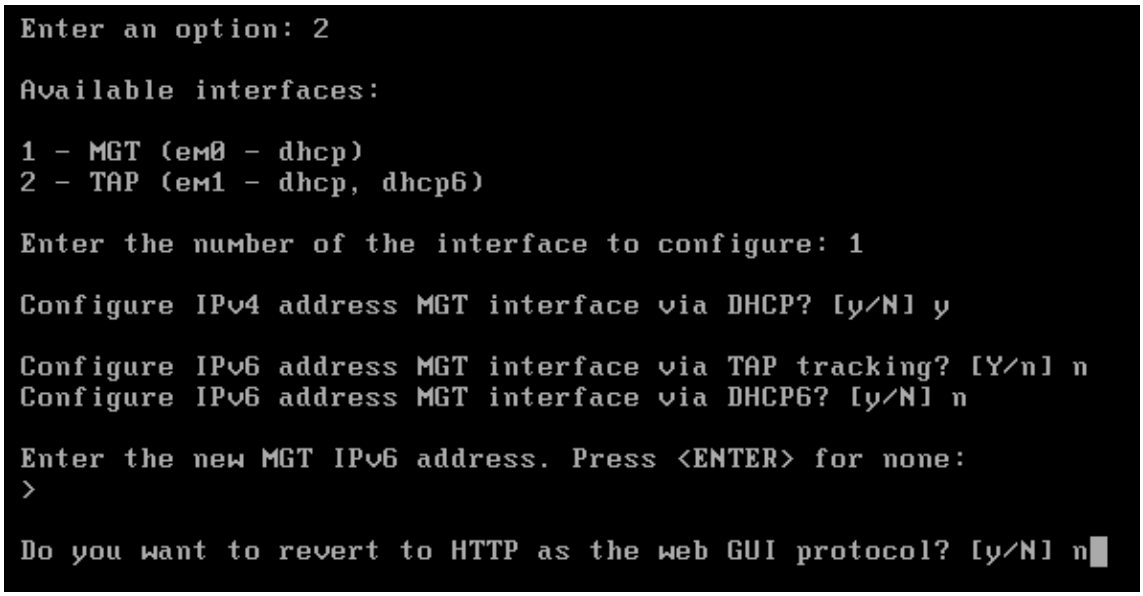


Figura 13-2: Configuración de la interfaz de administración

Realizado por: Herrera, Jonathan 2021.

Para la interfaz de análisis (TAP) no es necesario configurar una dirección IP debido a que como su nombre lo dice se encargara del análisis de tráfico de la red de forma transparente, es decir, que no exista tráfico propio de la interfaz.

Con la dirección IP de administración asignada por el DHCP podemos ingresar a la interfaz gráfica de control a través de cualquier navegador, ubicando en la parte del URL la dirección "172.25.200.166" ingresamos ala interfaz principal, como se aprecia en la Figura 14-2.

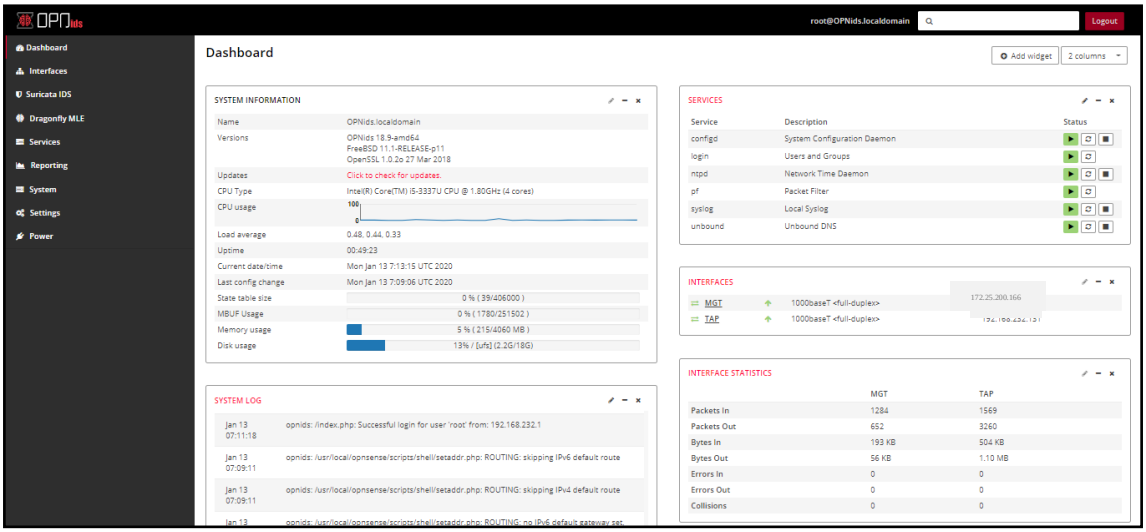


Figura 14-2: Interfaz de administración y configuración de OPNids

Realizado por: Herrera, Jonathan 2021.

La interfaz gráfica se encuentra dividido en 2 secciones, en la parte derecha se muestran widgets de: información del sistema, Log del sistema, servicios, interfaces y estadísticas de interfaces un enfoque resumido del funcionamiento de OPNids.

En la parte izquierda se ubica un menú de configuración de toda la herramienta OPNids, explicados en siguiente Tabla 7-2.

Tabla 7-2: Menú de la interfaz gráfica de OPNids

Opción	Descripción
Dashboard	Tablero principal de monitoreo.
Interfaces	Configuración de las interfaces MGT y TAP, direccionamiento IPv4, IPv6, DHCP, etc.
Suricata IDS	Configuración del IDS Suricata, herramienta de gran importancia de OPNids.
Dragonfly MLE	Motor de Machine Learning.
Servicios	Beats, monitoreo, tiempo de red.
Reportes	Genera gráficos del tráfico de datos in/out en las interfaces.
Sistema	Manejo de usuarios y grupos, configuración del sistema como backups, actualizaciones de firmware, logs de sistema y diagnóstico de servicios.
Ajustes	Configuraciones generales de la herramienta.
Energía	Opciones de Apagado y reinicio.

Realizado por: Herrera, Jonathan 2021.

OPNids potencia su funcionamiento a diferencia de otras herramientas tradicionales como son Snort, The Zeek que son IDS Open Source con características similares entre ellos. La herramienta OPNids integra uno de los mejores IDS Open Source del mercado, usado a nivel mundial y por grandes empresas, este es Suricata un IDS multi-hilo de gran rendimiento además integra el motor de machine learning DragonFly. A continuación, se detalla la configuración y funcionamiento de las 2 herramientas en la red de la FIE.

2.2.3.4. Configuración del sistema de detección de intrusos Suricata

El servicio de Suricata se encuentra deshabilitado por defecto para proceder a activarlo y configurarlo nos dirigimos hacia su opción en el menú grafico del tablero principal. Como se observa en la Figura 15-2 los ajustes del IDS presentan varias opciones, se procede a configurar de la siguiente manera:

- Se marca la casilla para habilitar el servicio.
- Se marca la casilla para habilitar el log de registro de alertas.
- Se elije un emparejador de patrones, los algoritmos integrados son Aho-Corasick por defecto e Hyperscan un algoritmo de Intel integrado a Suricata desde la versión 2.x, por motivos de la versión el algoritmo Hyperscan presenta errores al ser seleccionado, se escoge el algoritmo por defecto.
- Elegimos la interfaz de análisis TAP.
- La frecuencia en la cual se creará un nuevo log se escoge de forma semanal.
- Definimos el número de log que van a guardar, en este caso 4.

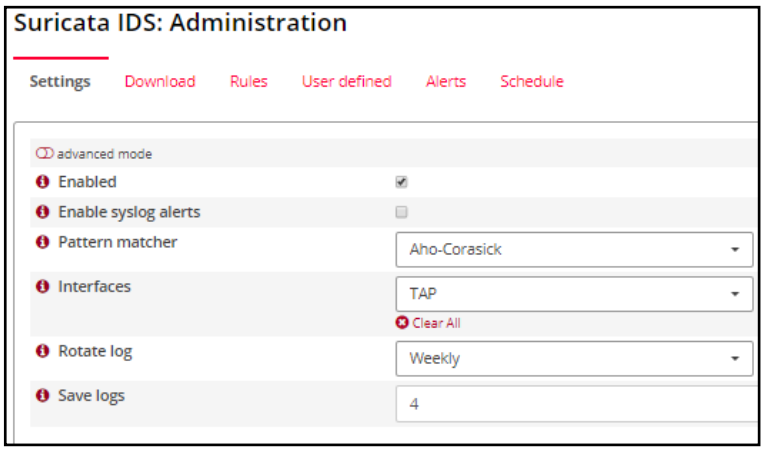


Figura 15-2: Ajustes Suricata IDS

Realizado por: Herrera, Jonathan 2021.

Se aplican los cambios y el IDS comienza a analizar el tráfico de la red, pero para que este funcione necesita de un conjunto de reglas. Nos ubicamos en la pestaña "Descargas" en la cual se muestra el conjunto de reglas disponibles para descargar, habilitar y deshabilitar. Una vez que se seleccionan todas las reglas se las habilita, después se procede a presionar el botón "Descargar y actualizar reglas" cuando finaliza la descarga se instalan con la hora y fecha, todo el proceso se encuentra numerado en la Figura 16-2.

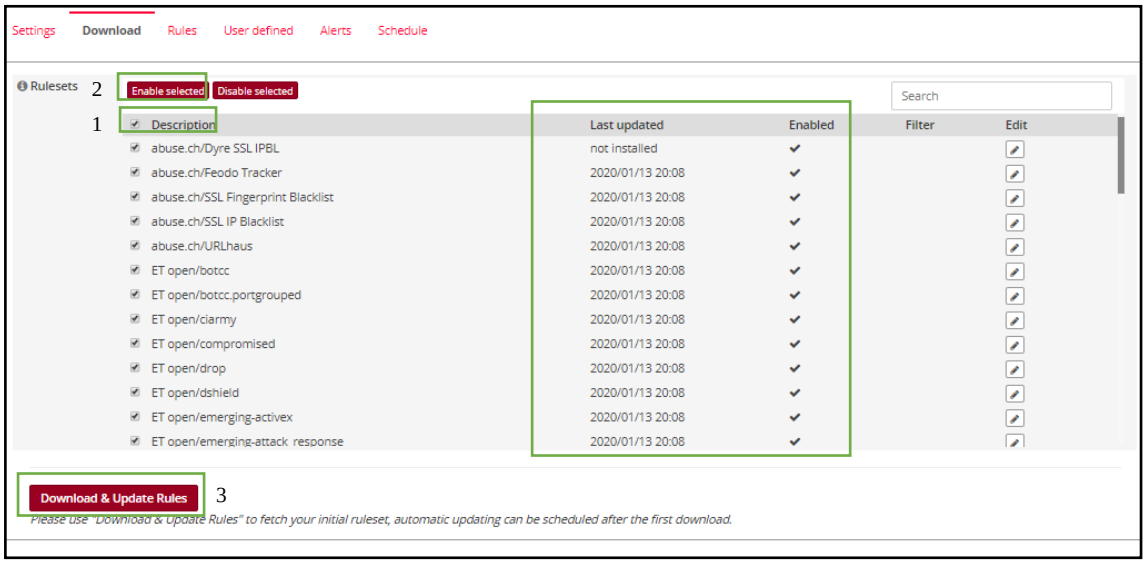


Figura 16-2: Conjunto de reglas de Suricata IDS

Realizado por: Herrera, Jonathan 2021.

En la pestaña "Reglas" de la Figura 17-2 se muestra una lista con todas reglas instaladas las cuales pueden ser habilitadas y deshabilitadas por el administrador.

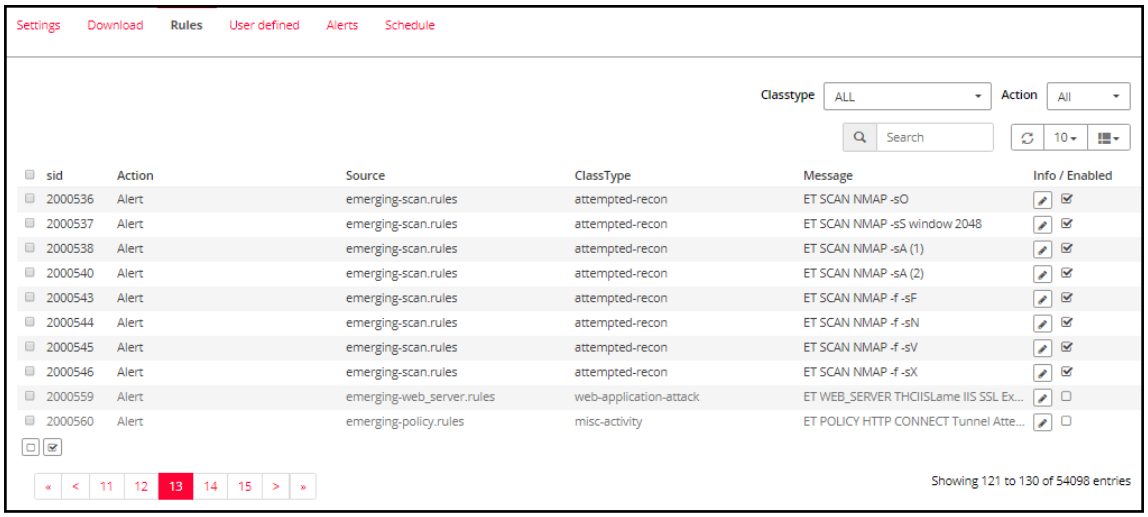


Figura 17-2: Reglas de Suricata IDS

Realizado por: Herrera, Jonathan 2021.

Suricata IDS maneja 3 conjuntos de reglas de diferentes tipos que son:

- Abuse.ch: Organización sin fines de lucro creada para combatir el malware en internet a través de listas negras.
- Emerging Theats (ET): Es un servidor de reglas patrocinado por la OSIF, comparte su conjunto de reglas con cualquier IDS, entre los más usados Suricata y Snort.
- OPNsense: Es un conjunto de reglas para IDS/IPS que tiene soporte para trabajar con las reglas de ET y Abuse.ch. (Deciso BV, 2019)

Por motivos de análisis de tráfico, utilización de recursos y para disminuir en mayor cantidad los falsos positivos no se recomienda que un IDS trabaje con todas las reglas habilitadas por esta razón, el IDS Suricata fue sometido a un tiempo de afinamiento y observación con respecto al comportamiento de la red de la FIE. En la sección 2.3.2.4 se explica del análisis y funcionamiento del IDS. En la pestaña con el nombre "Alerts" o Alertas se observa todo el registro en tiempo real de las alertas generadas por las reglas habilitadas. En la Figura 18-2 se aprecian los siguientes campos:

- Timestamp: registra la fecha y hora de la alerta.
- Acción: Si es permitida o bloqueada.
- Interfaz: nombre de la interfaz de análisis.
- Ip / Puerto de origen: Ip que envía la solicitud de comunicación por un puerto determinado.
- Ip / Puerto de destino: Ip que recibe la solicitud de comunicación por un puerto determinado.
- Alerta: nombre de la alerta detectada.
- Información: se visualiza información detallada de la alerta.

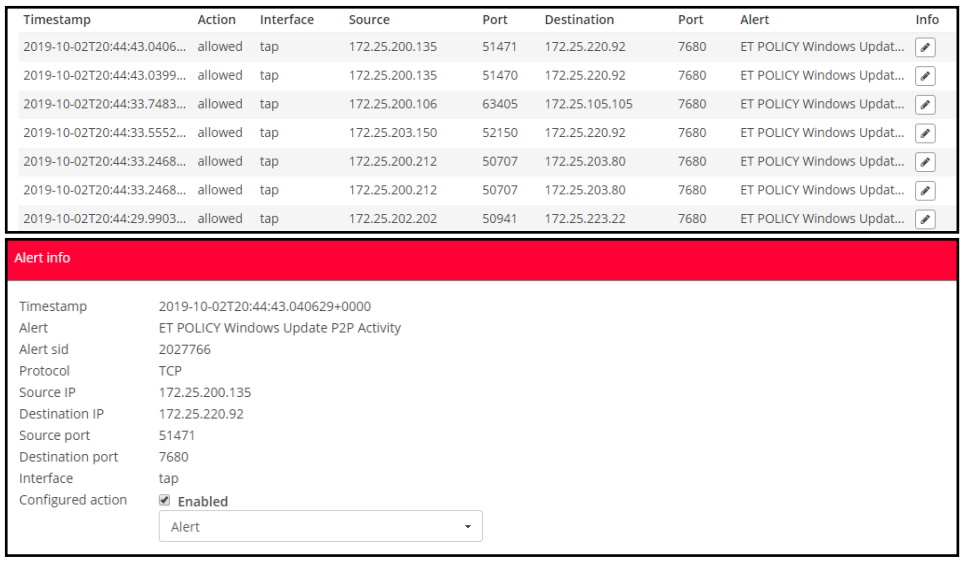


Figura 18-2: Alertas del IDS

Realizado por: Herrera, Jonathan 2021.

2.2.3.5. Configuración del motor de machine Learning DragonFly

Dragonfly MLE es el motor de Machine Learning integrado de la herramienta OPNids, al igual que Suricata IDS este servicio viene deshabilitado, nos ubicamos sobre el menú grafico del tablero principal para ir a sus opciones.

En el modo grafico de la herramienta OPNids no muestra un panel de configuración al igual que el IDS, únicamente se puede habilitar o deshabilitar el servicio de Dragonfly MLE, como se observa en la Figura 19-2.

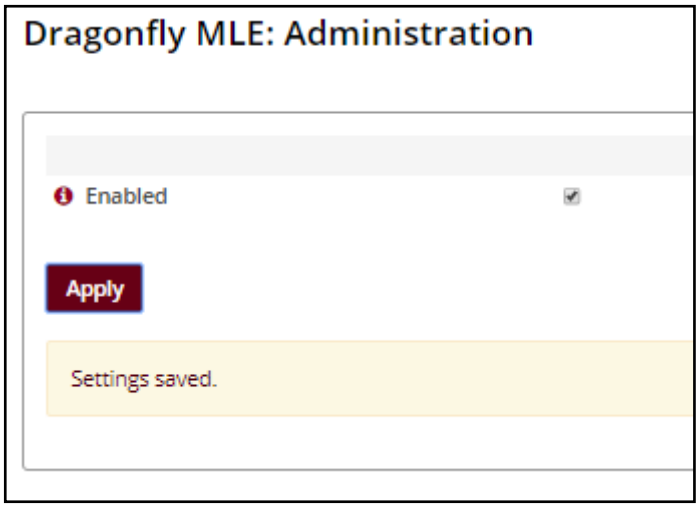


Figura 19-2: Motor Dragonfly MLE

Realizado por: Herrera, Jonathan 2021.

Por esta razón debemos realizar la configuración a través de la consola como se observa en la Figura 20-2, nos dirigirnos hacia el directorio donde se encuentra instalado, ingresando al Shell con la opción 8 y escribimos el siguiente comando:

cd /usr/local/dragonfly-mle – cambiamos del directorio raíz hacia el de Dragonfly MLE.

ls – se muestra el contenido del directorio

Figura 20-2: Directorio raíz de Dragonfly MLE

Realizado por: Herrera, Jonathan 2021.

La Figura 20-2 nos muestra el directorio de Dragonfly MLE en el cual se encuentra el directorio "config" el mismo que posee un archivo de configuración "config.lua", como se explicó en la sección 1.7.3. este archivo está formado por 3 partes, entrada, analizador y salida y procedemos a editar el archivo con los siguientes comandos:

vi config.lua – Ingresamos al archivo con el editor de texto vi

Nos ubicamos sobre la sección de entrada o "input" y modificamos la línea marcada con el cuadro rojo como se observa en la Figura 21-2.

```
-- ------------------------------------------------------------
-- Input queues/processors
-- ------------------------------------------------------------
inputs = {
   { tag="eve", uri="tail:///var/log/suricata/eve.json", script="suricata-filter
.lua", default_analyzer="alert"}, --Split messages based on type
   --{ tag="flow2", uri="ipc://flow-ipc.log", script="passthrough-filter.lua"}
}
```

Figura 21-2: Procesador de entrada de Dragonfly MLE

Realizado por: Herrera, Jonathan 2021.

Al editar la configuración del procesador de entrada, definimos el origen de los datos en este caso el algoritmo va a ser alimentado por los datos del log de alertas de Suricata en formato Json, a continuación, se define el script que de igual manera es el manejado por Suricata y asignamos un analizador por defecto, en esta parte es donde podemos definir entre los diferentes analizadores que posee Dragonfly MLE, escogeremos el algoritmo de clasificación supervisado: Random Forest. A continuación, nos ubicamos en el procesador de analizadores como se observa en la Figura 22-2 y des comentamos la línea siguiente:

{tag="dga", script= "dga/dga-rf-mle.lua"} –Detector DGA de Random forest

```
   -- ------------------------------------------------------------
   -- Advanced Analytics examples using Redis
   -- ------------------------------------------------------------
   -- { tag="flow1", script="example-hll.lua" }, -- Counting Unique Connections
with HyperLogLog
   -- { tag="flow2", script="example-mad.lua" }, -- Flow Outliers using Median A
bsolute Deviation (MAD)
   -- ------------------------------------------------------------
   -- Machine learning examples using Redis-ML
   -- ------------------------------------------------------------
   -- { tag="dga", script="dga/dga-lr-mle.lua" }, --DGA detector w/ Logistic Reg
ression
   -- { tag="dga", script="dga/dga-rf-mle.lua" }, --DGA detector w/ Random Fores
```

Figura 22-2: Procesador de analizadores de Dragonfly MLE

Realizado por: Herrera, Jonathan 2021.

Para finalizar la configuración, en la sección del procesador de salida se puede modificar la forma en la cual se van a presentar los resultados, en este caso utilizaremos el log de ejemplo de Dragonfly MLE al igual que en la Figura 23-2.

Una vez realizada la configuración, se reinicia el servicio de Dragonfly MLE y procedemos a visualizar el paso de las alertas de Suricata en el log asignado.

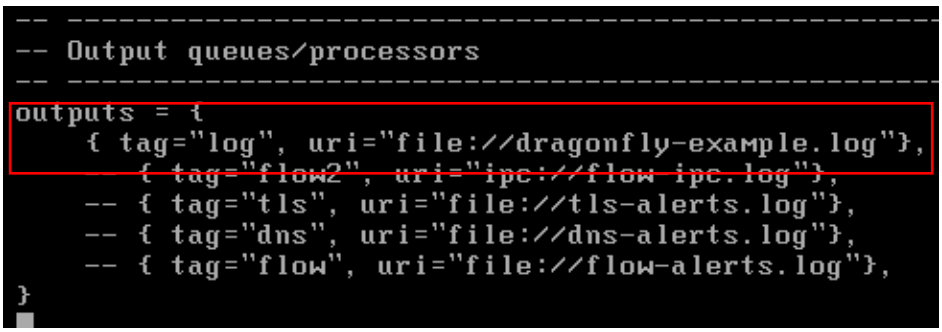


Figura 23-2: Procesador de salida de Dragonfly MLE

Realizado por: Herrera, Jonathan 2021.

2.3. Etapa 2: Analizar el tráfico de datos malicioso mediante la plataforma OPNids en el edificio de la FIE.

2.3.1. Análisis de tráfico de datos malicioso de la FIE mediante la plataforma OPNids

Después de la instalación de la plataforma OPNids y su configuración mencionada en las secciones anteriores se procede a realizar el periodo de pruebas y afinamiento del IDS Suricata y el motor de machine learning. El análisis del tráfico de datos de la FIE empezó de forma oficial el día 02 de diciembre de 2019 con el siguiente itinerario:

- Inicio del servicio de la plataforma OPNids y el IDS Suricata a las 8 am.
- Análisis del tráfico de datos en la Vlan Estudiantes.
- Revisión y registro de las alertas el día 03 de diciembre y de forma consecutiva a las 8:30 am.

Este proceso fue repetitivo hasta el 31 de enero de 2020 de manera constante, es decir, las 24 horas del día, los fines de semana incluido en días feriados por festividades de navidad y fin de año, cabe recalcar que existieron algunos inconvenientes de carácter técnico que sufrió el servidor con respecto al daño de su almacenamiento (HDD) por una semana, sin embargo, del 02 al 09 de diciembre de 2019 se mantuvo un constante análisis del IDS Suricata como prueba de

afinamiento, esta evaluación consistió en verificar el correcto funcionamiento de las reglas y las alertas emitidas con el objetivo de obtener información del comportamiento del tráfico de datos del edificio de la FIE para utilizarlo como punto de partida hacia el entrenamiento del algoritmo de clasificación de machine learning con lo que se pretende reducir el porcentaje de error y aumentar la precisión. Se llevo a cabo un registro de todas las alertas generadas en el tiempo de análisis, cada alerta posee varios campos de interés, como su dirección IP de origen y destino, identificador SID de la alerta, cantidad de alertas, fecha y hora correspondiente y de ser el caso documentar alguna observación realizada durante el día de análisis.

Para el análisis de las vlan de docentes y administrativos, el proceso fue el mismo al mencionado en los párrafos anteriores con la única diferencia en su tiempo de duración, el análisis para las 2 vlan fue de aproximadamente 1 mes, iniciando del 03 de febrero al 02 de marzo de 2020 y del 02 al 16 de marzo de 2020, respectivamente. El tiempo de análisis vario de una vlan a otra por la cantidad de alertas que se registraron, la vlan estudiantes presento mayor volumen de tráfico de datos y variedad en alertas de amenazas, al contrario, en la vlan docentes y administrativos se presentó un tráfico de datos de menor volumen y de igual manera de alertas, por tal motivo solo se realizó el análisis durante el tiempo mencionado. Cabe recalcar que el análisis de tráfico de datos de la vlan administrativos no se concluyó de manera correcta por motivos de salud pública mundial, la cuarentena a raíz de la pandemia de COVID-19 obligo a finalizar el análisis a mitad del mes de marzo. A continuación, se procedió a la implementación de manera offline, es decir, una simulación en un escenario virtual replicando las condiciones del laboratorio de microondas de la FIE para evaluar el desempeño y efectividad de la plataforma OPNids en la detección de tráfico de datos malicioso.

2.3.2. Amenazas detectadas en el análisis de tráfico de datos malicioso en la FIE

El análisis de tráfico de datos malicioso explicado en la sección anterior detectó las amenazas y vulnerabilidades a las cuales se exponen los usuarios de la intranet del edificio de la FIE. Existe una gran variedad de posibles ataques informáticos registrados, por esta razón en el presente trabajo de titulación se procedió a codificarlos o categorizarlos en 4 tipos de ataques explicando los más comunes según su vector de ataque, estos son: Denegación de Servicio (DoS), Usuario a Root (U2R), Remoto a local (R2L) y Escaneo (Scan o Probin).

2.3.2.1. Ataques de denegación de servicio (DoS)

Denegación de servicio o DoS por sus siglas en inglés, es uno de los ataques más utilizados por su difícil rastreo y en muchos casos, no se necesita conocimientos avanzados. Este ataque afecta

la disponibilidad de un servicio. Básicamente un ataque DoS consiste en saturar los recursos de un sistema, ya sea a nivel de hardware como un servidor, software como una página web o ambos, provocando que los usuarios reales de dicho servicio no puedan acceder. Los ataques DoS logran su objetivo inundando a la víctima con tráfico que el servidor no alcanza a procesar lo que generara un bloqueo. (Palo alto networks, 2019) En la Tabla 8-2 se explica brevemente los tipos más comunes de ataques DoS que pueden ejecutarse, por lo general los ataques DoS tienen un patrón en común y es que en su mayoría se los realiza utilizando los protocolos TCP/IP.

Tabla 8-2: Tipos de ataques DoS

Ataque	Descripción
Inundación SYN (SYN Flood)	SYN Flood es un ataque que funciona mediante la explotación del proceso de protocolo TCP. Al intentar establecer una sesión TCP (SYN, SYN/ACK, ACK) el servidor crea sesiones individuales por cada solicitud y al no completarse mantiene la sesión activa con el puerto abierto, así se repite el proceso hasta saturar el servicio.
Inundación ICPM (ICMP Flood)	ICMP Flood es un ataque en el que el atacante intenta cargar a un dispositivo con paquetes de solicitud de eco ICMP, haciendo que la víctima sea inaccesible para brindar un servicio normal.
SMURF	Smurf es un ataque DoS en el cual se intenta inundar un servidor victima con paquetes ICMP. Se envían solicitudes de IP falsificada del host final a varias redes, las cuales responden al servidor víctima, para amplificar el tráfico de ataque inicial y colapsar el servicio. Se considera una vulnerabilidad resuelta.
Inundación UDP (UDP Flood)	UDP Flood es un ataque en el que se envía una gran cantidad de paquetes UDP a un servidor de destino para sobrecargar la capacidad de ese dispositivo para procesar y responder.

Fuente: (Cloudflare, 2020)

Realizado por: Herrera, Jonathan 2021.

Una evolución de los ataques de DoS tradicionales son los ataques de denegación de servicio distribuidos o DDoS, que consisten en la coordinación de múltiples ataques DoS desde distintos orígenes contra uno o varios objetivos. (Cloudflare, 2020)

2.3.2.2. Ataques de acceso remoto a local (R2L)

Un ataque R2L se produce cuando un atacante ingresa a un sistema sin autorización comúnmente desde internet, la víctima de manera remota busca obtener acceso por medio de malware o

backdoors hacia una víctima con privilegios o root, en sistemas Linux. En este tipo de ataques prima la ingeniería social como principal vector para conseguir el objetivo, manipular al personal para recolectar la información necesaria e infiltrarse en el sistema víctima. (De la Hoz et al. 2012, p. 91) En la Tabla 9-2 se explica brevemente los tipos más comunes de ataques R2L que pueden ejecutarse.

Tabla 9-2: Tipos de ataques R2L

Ataque	Descripción
IMAP Attack	Los ataques de fumigación de contraseñas son vectores de riesgo potencialmente altos a sufrir ataques de fuerza bruta sin generar alertas o algún bloqueo, con la finalidad de intentar conseguir las credenciales de los usuarios. Las instituciones con Office365 deben deshabilitar IMAP, en cambio las que poseen correo electrónico interno, si no es posible deshabilitar IMAP, las conexiones al servidor se deben vigilar en caso de existir un gran número de conexiones de una fuente similar, se puede ser víctima de un ataque.
Gusano informático (Worm)	El gusano informático es un malware que se multiplica de un computador a otro. La infección puede ocurrir a través de vulnerabilidades de software o ingeniería social, la cual consiste en engañar a la víctima para que descargue software malicioso e infectar al objetivo. Un gusano infecta el computador de forma transparente al usuario y este puede tener varios propósitos, desde multiplicarse para agotar recursos informáticos, hasta instalar backdoors para que un atacante pueda controlar remotamente el computador de la víctima.
Warezmaster Attack (FTP)	Warezmaster es un tipo de ataque R2L dirigido a sistemas con un servidor de transferencia de archivos o FTP. El ataque aprovecha una mala configuración que otorga a un usuario invitado permisos de escritura, cuando por lo general todo usuario invitado posee solo permisos de lectura. El atacante crea un directorio oculto en el servidor y sube "warez" (software malicioso) el que será descargado por los usuarios sin conocimiento del software infectado.
Warezclient Attack (FTP)	Es la continuación del *warezmaster attack,* en la que le atacante logra de manera exitosa infectar el objetivo con el software descargado, es una operación legal dentro del servidor, ya que un usuario "legal" cargo archivos al mismo y cualquier otro usuario puede acceder a ellos. La característica para detectar dicho ataque consiste en revisar y monitorear constantemente los directorios de descarga.

Fuente: (Jett, 2019), (Norton, 2020), (Sabhnani y Serpen 2003, p. 312), (Sabhnani y Serpen 2003, p. 313)

Realizado por: Herrera, Jonathan 2021.

2.3.2.3. Ataques de usuario aq root (U2R)

Un ataque U2R se produce cuando un atacante que ya se encuentra autenticado en el sistema informático e intenta ganar privilegios de los permitidos. Este ataque se puede dar por vulnerabilidades no detectadas en el sistema, un backdoor, troyano o software instalado. (De la Hoz et al., 2012, pp. 91) En la Tabla 10-2 se explica brevemente los tipos más comunes de ataques U2R que pueden ejecutarse.

Tabla 10-2: Tipos de ataques U2R

Ataque	Descripción
Ataque de desbordamiento de buffer (Buffer overflow attack)	Un Buffer Overflow Attack se utiliza para sobrescribir o desbordar la pila de ejecución de una aplicación. Consiste en enviar código modificado para forzar a ejecutar código malicioso que permite al atacante apoderarse del objetivo. Estos ataques pueden causar bloqueos del sistema, mal funcionamiento u omitir un servicio de seguridad. Algunas de las firmas conocidas de este ataque son: HTTP_Accept_Language_Overflow, HTTP_Apache_DOS, HTTP_Apache_LF_Memory_DoS, entre otras.
Rootkit	Un rootkit es software diseñado para permanecer oculto (backdoor) en el computador mientras facilita acceso y control remotos. Los atacantes utilizan los rootkits para controlar un equipo sin el conocimiento del usuario. Por lo general son introducidos en el computador de la víctima a través de la descarga de malware especifico.
Ataques de Inyección (Injection attacks)	Consiste en inyectar código en un programa, puede ser malware con el objetivo de ejecutar comandos remotos que pueden leer o modificar una base de datos, o cambiar datos en un sitio web. Existen varios tipos de este ataque como: SQL, SSI, Blind SQL and XPath injection, OS Commanding.
Powershell Attack	Son ataques más difíciles de detectar, porque utiliza las herramientas de confianza de Windows, específicamente PowerShell. PowerShell es un lenguaje de scripting eficaz que proporciona total acceso al núcleo interno de un dispositivo y a las API´s de Windows. La capacidad de PowerShell para ejecutarse de forma remota a través de WinRM lo convierte en una herramienta más potente de ejecutar cualquier ataque con casi el 100% de efectividad y de ocultamiento.

Fuente: (IBM Knowledge center, 2018a), (Belcic, 2020), (IBM Knowledge center, 2018b), (O´Connor, 2017)
Realizado por: Herrera, Jonathan 2021.

2.3.2.4. Ataques probin o escáner de redes

Este tipo de ataques se basan en el escaneo de redes de datos con la finalidad de identificar y recolectar toda la información como direcciones IP, puertos abiertos, direcciones MAC, Sistema operativo, fabricante del hardware, etc. A partir de esto el atacante identifica las vulnerabilidades potenciales y prepara las herramientas para explotarlas. (De la Hoz et al., 2012, pp. 91)

En la Tabla 11-2 se explica brevemente los tipos más comunes de ataques de escaneo que pueden ejecutarse.

Tabla 11-2: Tipos de ataques de escaneo

Ataque	Descripción
Nmap	Nmap es una herramienta open source para el escaneo de redes y la auditoría de seguridad. Utiliza paquetes IP para determinar los hosts disponibles en la red, qué servicios se ejecutan en esos hosts, qué sistemas operativos, qué tipo de firewalls, entre otras características.
Barrido de IP (IP Sweep)	Ocurre cuando una dirección IP de origen envía un número definido de paquetes ICMP a diferentes hosts dentro de un intervalo (0.005 s por defecto). El objetivo es enviar los paquetes ICMP a distintos hosts conectados a la red en busca de una respuesta, para descubrir una dirección IP de destino.
Barrido de puertos TCP (Port Sweep TCP)	Ocurre cuando una dirección IP de origen envía paquetes IP que contienen segmentos SYN de TCP a 10 puertos de destino diferentes en un intervalo (5 milisegundos por defecto). El objetivo es verificar puertos abiertos y por consecuencia los servicios disponibles puedan ser identificados.
Barrido de puertos UDP (Port Sweep UDP)	La exploración de puertos UDP proporciona información estadística sobre un umbral de sesión. A medida que llegan los paquetes, se establecen las sesiones. El número de sesiones que se exigen en el umbral se basa en la zona, IP de origen y el período de umbral, y este permite hasta 10 sesiones, para cada zona y dirección IP de origen.

Fuente: (Nmap, 2020), (Juniper, 2020a), (Juniper, 2020b), (Juniper, 2020c)

Realizado por: Herrera, Jonathan 2021.

2.4. Etapa 3: Evaluar el desempeño y efectividad de la plataforma OPNids en la detección de tráfico de datos malicioso en la red del edifico de la FIE

En las etapas anteriores se detalló el procedimiento que se realizó para el diseño e implementación de la plataforma OPNids y el análisis del tráfico de datos malicioso en el edificio de la FIE. Pero como se detalla en la investigación en el capítulo 1, el motor de machine learning DragonFly, debe crear un modelo de entramiento para su algoritmo de clasificación Random Forest con datos de enteramiento correctos para evitar errores de sub-ajuste o sobre-ajuste que provocaría que el modelo falle y no cumpla con su objetivo. A continuación, se procede a detallar el procedimiento que se llevó a cabo para evaluar el desempeño y efectividad de la plataforma OPNids en la detección de tráfico de datos malicioso en la FIE a través del entramiento simulado del motor de machine learning DragonFly en conjunto del script adicional y ataques informáticos simulados. Cabe recalcar que los dos procesos se realizaron de forma simulada debido a la pandemia del COVID-19.

2.4.1. Entrenamiento simulado del motor de machine learning DragonFly

La plataforma OPNids es un proyecto Open Source que depende en la totalidad de sus desarrolladores y la comunidad para avanzar en la integración de sus herramientas, en especial el motor de machine learning DragonFly el cual está compuesto por varias dependencias y módulos para su correcto funcionamiento como filtros de paquetes de Suricata, analizadores de red DNS, HTTPS y el motor de machine learning. El proceso de machine learning es realizado de forma transparente para el usuario, se comprueba la falta de integración entre DragonFly y OPNids en la sección 2.3.3.5 que detalla la configuración del mismo; la cual es muy sencilla, carente de visualización de los parámetros de evaluación de error que son necesarios para evaluar el desempeño y efectividad de la plataforma OPNids cuando se utiliza algoritmos de machine learning aplicados a la detección de intrusos a través del IDS Suricata, por esta razón en el presente trabajo de titulación se realizó la programación de un script complementario para el motor de machine learning DragonFly que nos permita obtener los parámetros de evaluación como: precisión, recall, puntaje F1 todos calculados a través de la matriz de confusión.

Se realizó el entrenamiento del motor de machine learning DragonFly en conjunto con el script adicional de manera simulada por motivos de la pandemia del COVID-19, no se realizaron pruebas en tiempo real y de forma física en el servidor de la FIE debido a la ausencia de estudiantes y el nulo tráfico de datos en la red, sin embargo, para cumplir con los objetivos del

presente trabajo de titulación se realizó una simulación en GNS3 como se observa en la Figura 24-2 del laboratorio de microondas del edifico de la FIE.

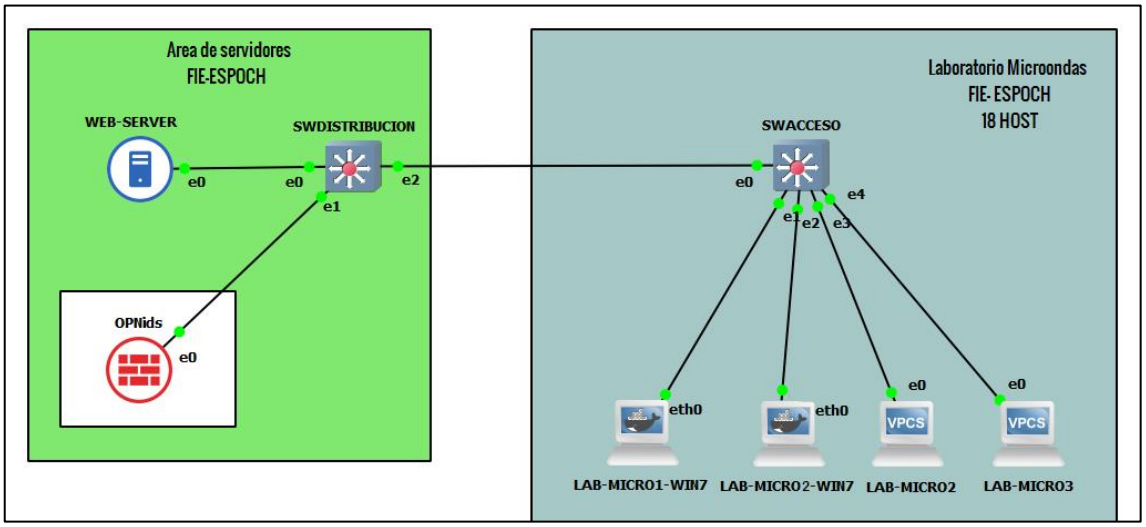


Figura 24-2: Laboratorio de la FIE simulado para entramiento del motor Dragonfly

Realizado por: Herrera, Jonathan 2021.

Para la simulación se utilizó dos dataset o conjuntos de datos para obtener un mejor resultado del entramiento y poder implementarlo en el servidor de la FIE en un futuro. Los datasets fueron: KDD99 y OPNids-eve.json (logs) para testear el algoritmo Random Forest. La diferencia entre estos datasets es que el log eve.json es el archivo del IDS Suricata que almacena todo el registro de eventos de la red, es decir, la alertas que se generaron durante el mes y dos meses de análisis del tráfico de datos de la vlan docentes y vlan estudiantes respectivamente de la red de la FIE, dicho conjunto fue bajado del servidor para su procesamiento offline. Cabe recalcar que en los dos procesos al manejar diferentes tipos de variables de salida es necesario codificar a los ataques en 4 categorías como son DoS, U2R, R2L y Probe los mismos que ya fueron explicados en la sección 2.3.2.

2.4.1.1. Entrenamiento simulado del motor de machine learning DragonFly con dataset KDD99

En primer lugar, nos basamos en el análisis de tráfico de datos de la red de la FIE con sus respectivas alertas detectadas en el periodo asignado a la semana de afinamiento del IDS Suricata por la razón de no existir registro previo sobre ataques informáticos verificados o amenazas como malware, troyanos, gusanos, entre otras. Se realizo una búsqueda exhaustiva en los diferentes sitios webs especializados en el procesamiento de tráfico de datos malicioso para estudio y

entrenamiento de algoritmos de machine learning, si bien lo más difícil fue encontrar similitud entre un dataset y el log de alertas de dicha semana se escogió el más aproximado a las amenazas detectadas en la intranet de la FIE. Entrenar el modelo en simulación controlada es una forma correcta para reducir el porcentaje de error; si un modelo no es correctamente entrenado existirá inconsistencia en sus predicciones y en el caso de la seguridad informática no se puede dar el lujo de existir un gran margen de imprecisión. Además de que para asegurar un algoritmo fiable se necesita un dataset con un gran volumen de datos correctos de entrada y salida, esto es una característica de todo algoritmo de clasificación, por esa razón no se puede utilizar un dataset limitado. El conjunto de datos con tráfico malicioso utilizado para el entrenamiento se encuentra en el siguiente enlace web: http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data_10_percent.gz, como se observa la Figura 25-2.

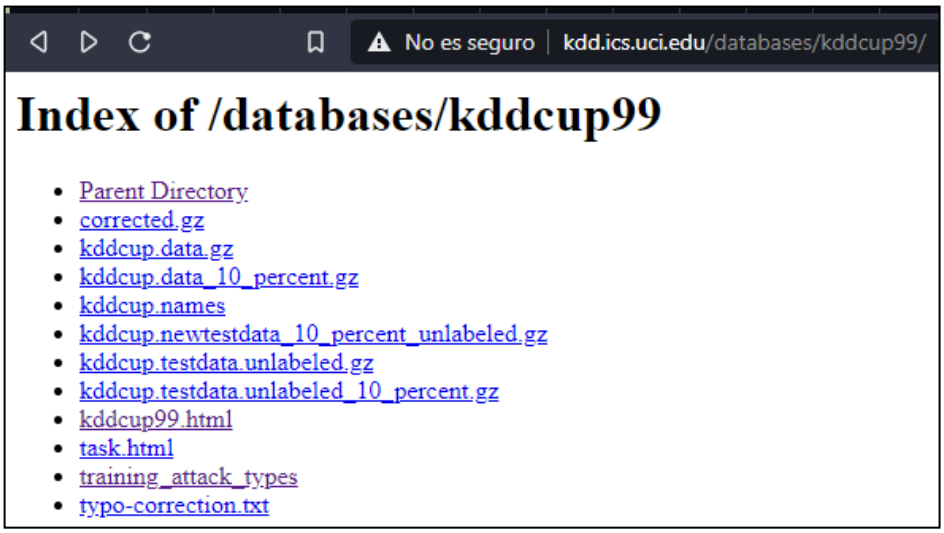


Figura 25-2: Dataset KDD99 con tráfico de datos malicioso

Realizado por: Herrera, Jonathan 2021.

La página web presenta varios conjuntos de información sobre tráfico malicioso, es una de las más utilizadas para la creación de sistemas de detección de intrusos y en este caso particular para el entrenamiento de modelos de machine learning.

La Figura 26-2 muestra parte de las alertas registradas en la semana del 02 al 09 de diciembre de 2019 de la plataforma OPNids-Suricata con el objetivo de correlacionar datos entre este registro y el dataset KDD99, en el registro llevado a cabo de dicha semana se trató de identificar hasta la mínima amenaza para acercarnos a un modelo de machine learning lo más preciso posible.

Timestamp	Alerta	SID	Protocolo	IP origen	IP destino	Puerto origen	Puerto destino	Observaciones
2019-12-02T17:59:29	SURICATA STREAM ESTABLISHED packet out of window	2210020	TCP	172.25.200.4	172.25.200.132	80	61503	
2019-12-02T17:59:29	SURICATA STREAM SHUTDOWN RST invalid ack	2210046	TCP	172.25.200.132	172.25.200.4	61503	80	
2019-12-02T17:59:29	SURICATA STREAM Packet with invalid ack	2210045	TCP	172.25.200.132	172.25.200.4	61503	80	
2019-12-03T17:59:29	SURICATA STREAM FIN invalid ack	2210030	TCP	172.25.200.132	172.25.200.4	61503	80	
2019-12-03T14:11:45	SURICATA zero length padN option	2200094	IPV6-ICMP	fe80:0000:0000:0000:78db:7c13:d59c:fe12	ff02:0000:0000:0000:0000:0000:0016			Se repitio la misma alerta desde las 6pm/12-12-2019 hasta las 2:12pm/13-12-2019
2019-12-03T14:13:40	SURICATA STREAM CLOSEWAIT invalid ACK	2210017	TCP	172.25.203.223	172.25.200.166	55741	443	
2019-12-04T14:13:40	SURICATA STREAM Packet with invalid ack	2210045	TCP	172.25.203.223	172.25.200.166	55741	443	
2019-12-04T14:13:40	SURICATA STREAM CLOSEWAIT ACK out of window	2210015	TCP	172.25.200.166	172.25.203.223	443	55741	
2019-12-05T14:13:40	SURICATA STREAM SHUTDOWN RST invalid ack	2210046	TCP	172.25.203.223	172.25.200.166	55741	443	
2019-12-06T16:31:20	SURICATA STREAM Packet with	2210045	TCP	172.25.201.82	192.188.46.165	42616	443	

Figura 26-2: Registro de alertas en la semana de afinamiento de OPNids

Realizado por: Herrera, Jonathan 2021.

El script adicional está formado por varias partes que se ejecutan de manera progresiva, se lo llevo a cabo en una simulación de un laboratorio de la FIE con la herramienta GNS3. En primer lugar, se debe importar todas las librerías y dependencia que se va a utilizar como son: pandas, numpy, scipy, sklearn, matplotlib y seaborn, en segundo lugar, se procede a descargar y leer el dataset KDD99 del URL, así como se observa en la Figura 27-2.

```
   0    1     2   3    4     5   6  ...    35   36   37   38   39   40       41
0   0  tcp  http  SF  181  5450   0  ...  0.11  0.0  0.0  0.0  0.0  0.0  normal.
1   0  tcp  http  SF  239   486   0  ...  0.05  0.0  0.0  0.0  0.0  0.0  normal.
2   0  tcp  http  SF  235  1337   0  ...  0.03  0.0  0.0  0.0  0.0  0.0  normal.
3   0  tcp  http  SF  219  1337   0  ...  0.03  0.0  0.0  0.0  0.0  0.0  normal.
4   0  tcp  http  SF  217  2032   0  ...  0.02  0.0  0.0  0.0  0.0  0.0  normal.

[5 rows x 42 columns]
```

Figura 27-2: Dataset KDD99 sin etiquetas

Realizado por: Herrera, Jonathan 2021.

El dataset está formado por 42 columnas y 5 filas en las que se observa son datos sin etiquetas, es necesario etiquetar cada columna respectivamente al dato que contiene, como se observa en la Figura 28-2.

```
            duration  ...  dst_host_srv_rerror_rate
count  494021.000000  ...            494021.000000
mean       47.979302  ...                 0.057412
std       707.746472  ...                 0.230140
min         0.000000  ...                 0.000000
25%         0.000000  ...                 0.000000
50%         0.000000  ...                 0.000000
75%         0.000000  ...                 0.000000
max     58329.000000  ...                 1.000000
```

Figura 28-2: Dataset KDD99 etiquetado

Realizado por: Herrera, Jonathan 2021.

Al verificar la variable de salida tiene como resultado 23 respuestas, por esta razón es necesario codificar las mismas para clasificar y manejar los datos de mejor manera, en la Figura 29-2 se observa la categorización de los 4 tipos de ataques y una adicional sobre el tipo de conexión.

```
Counter({'normal.': 97278,
         'u2r': 52,
         'dos': 391458,
         'r2l': 1126,
         'probe': 4107})
```

Figura 29-2: Variables de salida codificadas

Realizado por: Herrera, Jonathan 2021.

Ningún dataset es enviado en su totalidad hacia el algoritmo de machine learning para su entrenamiento, este debe ser limpiado de datos redundantes, vacíos y correlacionados, por eso se imprime la matriz de correlación, como se observa en la Figura 30-2.

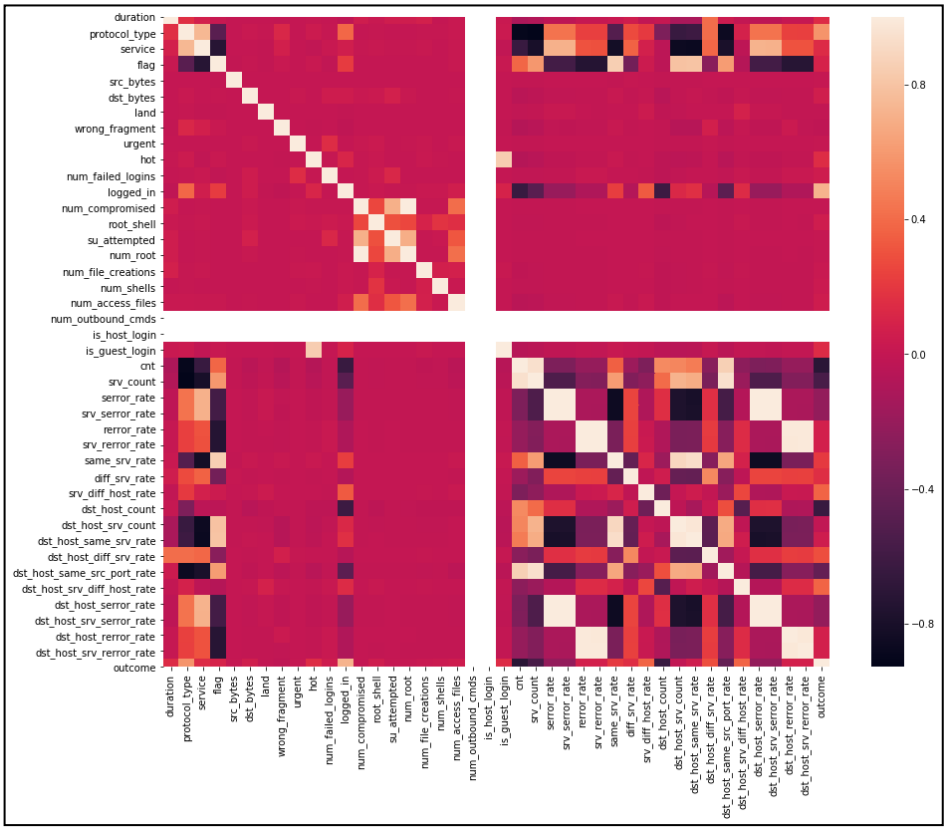


Figura 30-2: Matriz de correlación

Realizado por: Herrera, Jonathan 2021.

La matriz de correlación nos muestra que existe varios campos con una alta correlación como se observa en la figura 31-2, es decir, que existe duplicidad de información, se procede a limpiar los datos y eliminar espacios vacíos.

Feature 1	Feature 2	Correlation
num_compromised	num_root	0.9938277978855619
num_root	num_compromised	0.9938277978855619
serror_rate	srv_serror_rate	0.9983615072774075
srv_serror_rate	serror_rate	0.9983615072774075
rerror_rate	srv_rerror_rate	0.994730953976896
srv_rerror_rate	rerror_rate	0.994730953976896
dst_host_srv_count	dst_host_same_srv_rate	0.9736854572882875
dst_host_same_srv_rate	dst_host_srv_count	0.9736854572882875
dst_host_serror_rate	serror_rate	0.9986729680059853
dst_host_srv_serror_rate	serror_rate	0.9978492485717336
dst_host_rerror_rate	rerror_rate	0.9869947924930166
dst_host_srv_rerror_rate	rerror_rate	0.9851995540727151

Figura 31-2: Correlación de datos

Realizado por: Herrera, Jonathan 2021.

Para finalizar se procedió a la creación de la semilla de entrenamiento la cual representa el 20% de toda la muestra, por consecuencia su restante; el 80% será utilizado para la evaluación del algoritmo Random forest. Entre sus características Random forest nos permite escoger el tamaño del "bosque" que deseamos crear, la técnica de ensamble, máximo y mínimo número de hojas por árbol, entre otras, teniendo en cuenta que mientras más arboles de decisión existan será más preciso el modelo, pero su consumo de recursos es alto y aún más si el mismo debe ejecutarse en tiempo real, se necesitó constante análisis del software corriendo el script y el motor DragonFly. Los resultados serán expuestos en el capítulo 3.

2.4.1.2. Entrenamiento simulado del motor de machine learning con dataset OPNids-eve.json

La información obtenida por el análisis de tráfico de datos de la red de la FIE se utilizó a partir de terminada la semana de pruebas, es decir, desde el 13 de diciembre al 31 de enero de 2020, con el objetivo de tener un dataset completo que permita evaluar al motor de machine learning DragonFly en conjunto con su script adicional, el proceso realizado es el mismo que el detallado para el dataset KDD99, con diferencia de su entrada de datos, la cual fue obtenida del archivo "eve.json" como se observa en la Figura 32-2, en el cual registra todos los eventos en tiempo real el IDS suricata, a continuación seguimos los pasos de codificación de las variables de salida, se limpian los datos redundantes, para finalizar entrenamos y evaluamos el algoritmo, además los datos son enviados hacia el motor DragonFly. De igual manera los resultados serán expuestos en el capítulo 3.

Figura 32-2: Conjunto de datos de la plataforma OPNids

Realizado por: Herrera, Jonathan 2021.

2.4.2. Evaluación de la plataforma OPNids a través de la simulación de las amenazas detectadas en la FIE mediante un escenario virtual

Después del entrenamiento simulado del motor de machine learning DragonFly en conjunto con el script adicional se procedió a evaluar la plataforma OPNids a través de la simulación de 4 ataques informáticos, uno por cada categoría: DoS, U2R, R2L y Probe, los cuales fueron escogidos por medio del registro de las amenazas creado al momento del análisis del tráfico de datos de la FIE realizado en la etapa 2.

La Figura 33-2 muestra la topología de red estrella del laboratorio de microondas de la FIE en el simulador GNS3, realizado en una computadora Asus con 16GB de RAM. El escenario se asemeja al laboratorio de manera física, el cual está formado por un *Switch de acceso* que conecta los 18 host del laboratorio; por motivos de simulación y recursos no fueron simulados en tu totalidad, a su vez el *switch de acceso* se conecta a un *switch de distribución* ubicado en el área de servidores, en dicho switch se configuró un puerto SPAN o mirror para capturar todo el tráfico de datos y dirigirlo a la interfaz de red virtual conectada a la plataforma OPNids. En el área de servidores se encuentra una máquina virtual con distribución Centos 7 ejecutando un servidor web Apache2, el cual será un objetivo para el atacante, que es representado por una máquina virtual con distribución Kali-Linux configurado con dos adaptadores de red virtual. Todos los hosts corren el sistema operativo Windows 7 porque es el que en ese momento se ejecutaba en toda la FIE, el host LAB-MICRO1-WIN7 va a ser la maquita víctima, ya que es una versión explotable de Windows 7 con varias vulnerabilidades entre ellas las detectadas por OPNids, para iniciar con los ataques se comprobó la conectividad de todo el escenario el cual mantiene un direccionamiento de 192.168.193.XXX/24.

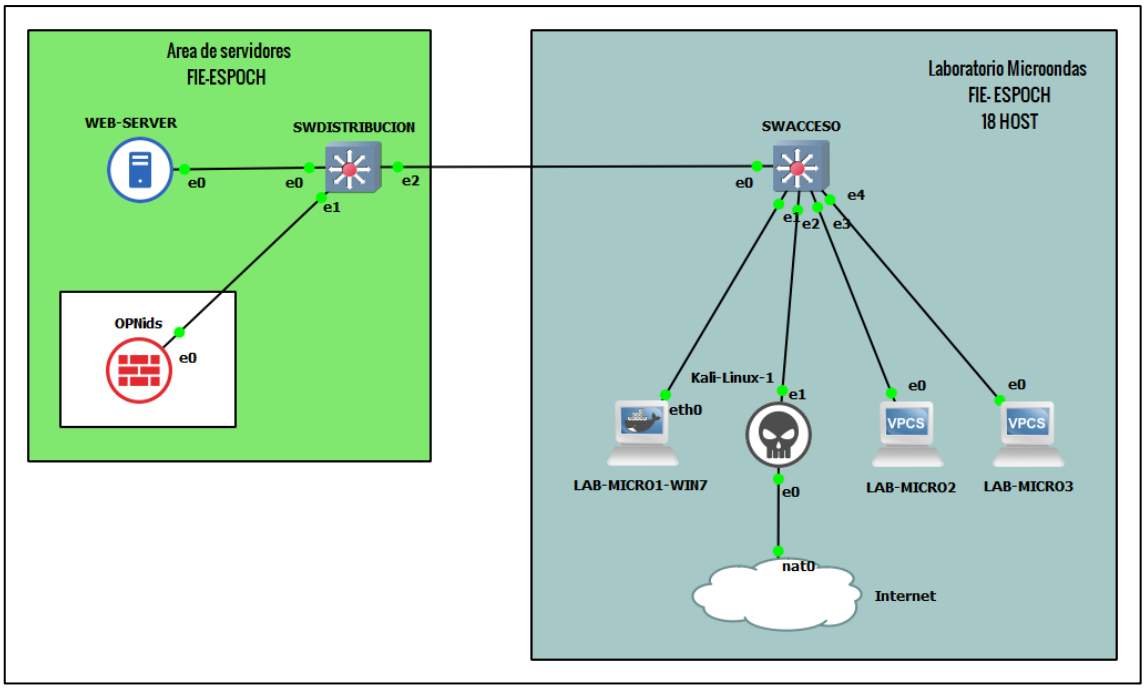


Figura 33-2: Escenario simulado del laboratorio de la FIE para los ataques informáticos

Realizado por: Herrera, Jonathan 2021.

2.4.2.1. Selección de las amenazas detectas en la FIE para la simulación de ataques informáticos

El análisis de tráfico de datos de la FIE presento un sin número de amenazas que fueron detectadas y registradas, muchas de ellas fueron falsos positivos, pero otras si cumplieron con su objetivo de ataque, intentos de DoS, malware y troyanos navegando en toda la red, escaneo de puertos, ip´s y hasta búsquedas de la IP publica fueron detectados, son estas las amenazas que fueron seleccionadas de la vlan estudiantes y docentes para ser simuladas, a continuación en la tabla 12-2 se detallan las amenazas por cada categoría y de las cuales se buscó replicar el mismo ataque o uno de similares características para evaluar el modelo de machine learning con OPNids.

Tabla 12-2: Selección de amenazas para simulación

Alerta	Descripción	Código
ET DOS Possible SSDP Amplification Scan in Progress	Potencial ataque DOS a través del protocolo (SSDP) utilizado para la detección de dispositivos Plug & Play (UPnP). El protocolo es vulnerable a ataques DoS y DDoS.	DoS

Tabla 12-2 (continuación)

ET DELETED Spambot Suspicious 220 Banner on Local Port	Sospecha sobre posible bot de spam utilizando el protocolo SMTP.	
SURICATA Applayer Detect protocol only one direction	Potencial conexión Command-Control detectada en capa de aplicación. Se puede tratar de un malware, mineros o una red Botnet.	R2L
ET TROJAN ZeroAccess, ZeuS udp traffic detected	Es un malware troyano ZeroAccess o Zeus y todas sus versiones. Es utilizado para infectar de ransomware de encriptación a la víctima.	
SURICATA Applayer Detect protocol only one direction	Potencial conexión Command-Control detectada en capa de aplicación. Se puede tratar de un malware, mineros o una red Botnet.	U2R
ET POLICY SMB Remote AT Scheduled Job Create Request – Possible Lateral Movement	Violación de política utilizando posible técnica de movimiento lateral para ataque a través del protocolo SMB.	
ET SCAN NMAP -sA (1)	Se realizo un escaneo de rango de ip´s con la herramienta Nmap.	PROBE
ET POLICY External IP Check myexternalip.com	Escaneo de la IP publica con fines para dirigir un ataque remoto.	

Realizado por: Herrera, Jonathan 2021.

Las amenazas presentadas en la tabla 12-2 son aquellas que representan una prioridad alta y por tal razón se escogieron para la simulación de los ataques informáticos, si bien no se puede replicar todas las amenazas con exactitud debido a que muchas de ellas no son fáciles de encontrar para su estudio por no estar liberadas, en cambio otras no adjuntan ningún número de CVE (Common Vulnerabilities and Exposures) para identificarlas, por esta razón nos basamos en el tipo de ataque y protocolo utilizado para vulnerar al objetivo.

2.4.2.2. Evaluación de la plataforma OPNids mediante ataque DoS Slowloris

Durante el análisis de tráfico de datos de la FIE se presentaron varias alertas de posibles intentos de DoS o denegación de servicio las cuales fueron identificadas en la tabla 12-2, verificando que la red presenta vulnerabilidades ante dicho ataque y pueden ser explotadas. Por esta razón en el

escenario virtual se procedió a instalar un servidor web Apache 2 en una máquina virtual con distribución Centos 7, se configuro su página por defecto y la victima estaría lista, en la Figura 34-2 se observa la página web con su dirección IP.

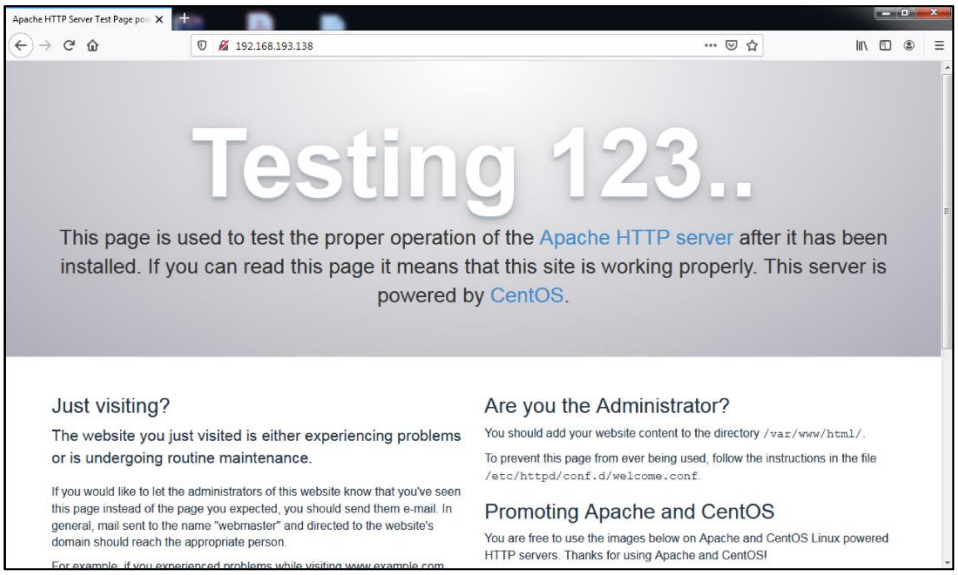


Figura 34-2: Pagina web por defecto de apache2

Realizado por: Herrera, Jonathan 2021.

Después de buscar varias herramientas para ejecutar un DoS se escogió a Slowloris que es una de las más utilizadas para ejecutar este tipo de ataque informático, su proceso se basa en inundar de peticiones HTTP REQUEST al servidor; al recibir miles de peticiones y no poder establecer una conexión Request-Response exitosa la página "se cae", deja de funcionar hasta que las peticiones secén o sean procesadas, slowloris a través de una línea de comandos puede enviar cientos, miles y hasta millones de paquetes en un corto periodo de tiempo, en este caso el ataque se realizó desde la máquina de Kali-Linux desplegada en el escenario virtual. Para comenzar el ataque es necesario descargar la herramienta a nuestro Kali-Linux para ello se abre un terminal y se escribe el siguiente comando:

git clone https://github.com/llaera/slowloris.pl.git.

Una vez que hayamos clonado la herramienta del repositorio nos dirigimos al directorio de la descarga y escribimos la siguiente línea de comando para iniciar el ataque:

perl slowloris.pl –dns 192.168.193.138 –port 80 –time 1 –num 2000 –cache.

El comando indica a slowloris que debe enviar 2000 paquetes cada segundo hacia la dirección ip del servidor por el puerto 80 y que se van a almacenar en el cache de la interfaz de red del servidor. De manera casi inmediata se intenta recargar la página del servidor web, la misma que empieza a presentar problemas hasta que deja de funcionar como se observa en la Figura 35-2, el ataque ha sido exitoso. Los resultados del ataque serán detallados en el capítulo 3.

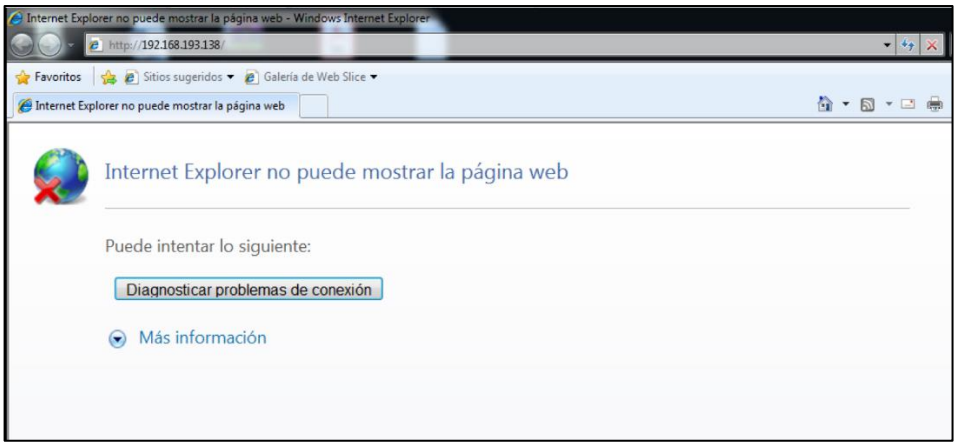


Figura 35-2: Pagina web sin acceso por ataque DoS

Realizado por: Herrera, Jonathan 2021.

2.4.2.3. Evaluación de la plataforma OPNids mediante ataque de Remote to Local R2L

Una de las amenazas detectadas en el tráfico de datos de la FIE fueron los diferentes tipos de malware, este tipo de ataque está en la categoría "R2L", aunque también puede presentarse en la "U2R", todo depende del tipo de malware y el objetivo que tenga el atacante.

La amenaza para la simulación de este ataque fue el malware trojan ZeroAccess, ZeuS, Zbot o cualquiera de sus variantes ya que dicho malware funciona de la misma manera. Se realizo la búsqueda y en el repositorio de investigación de la empresa Gibson Research Corporation y lo descargamos. Si bien este ataque se lo realiza de forma voluntaria sobre el host LAB-MICRO1-WIN7; el cual está sin conexión a internet para evitar que se propague por la red, al contrario, un atacante estudia a la víctima y a través de ingeniería social o phishing logra engañarla para que lo descargue y ejecute. El malware viene enmascarado dentro de un .exe para la instalación del complemento Adobe Flash Player, al momento de instalar se genera un error y el archivo de instalación desaparece, eso se aprecia en los círculos rojos como se observa en la Figura 36-2. Al tratarse de un ataque R2L, el atacante obtiene el control del dispositivo de la víctima, pero de forma silenciosa para ejecutar sus objetivos por tal motivo se concluye el ataque de forma exitosa. Los resultados del ataque serán detallados en el capítulo 3.

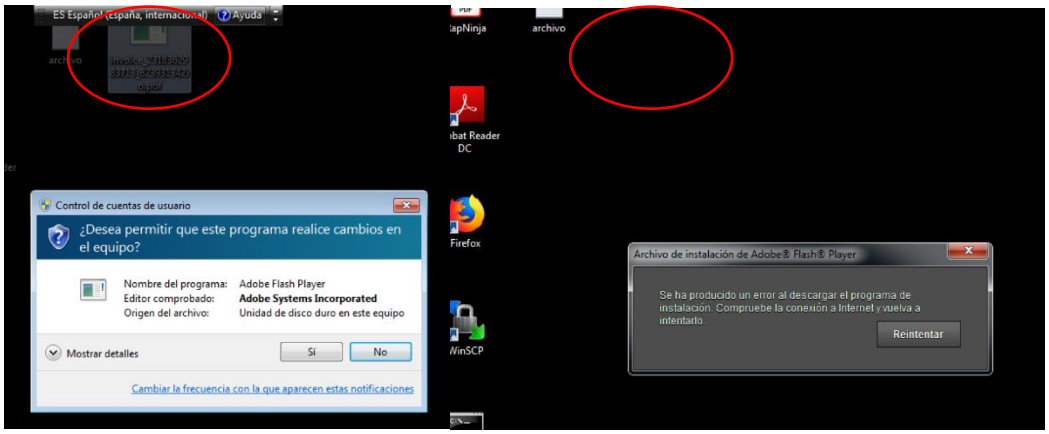


Figura 36-2: Instalación del malware encapsulado en Flash Player

Realizado por: Herrera, Jonathan 2021.

Después de que el malware se encuentra instalado, puede pasar mucho tiempo en que el usuario se dé cuenta que está infectado, ya que ZeroAccess se utiliza de manera remota ya sea para controlar o espiar a la víctima, además que puede instalar backdoors, criptomineros o ser parte de una red Botnet que empezará a utilizar los recursos de hardware de su anfitrión.

2.4.2.4. Evaluación de la plataforma OPNids mediante ataque de User to Root U2R

Una de las amenazas detectadas en el tráfico de datos de la FIE fueron los diferentes tipos de malware, este tipo de ataque está en la categoría "U2R". La amenaza para la simulación de este ataque fue el malware msf Venom, un malware dirigido para tomar control y escalar privilegios en la maquina víctima. El malware fue creado en el framework Metasploit desde la máquina virtual del atacante Kali Linux del escenario simulado como se observa en la Figura 37-2.

```
       =[ metasploit v5.0.41-dev                          ]
+ -- --=[ 1915 exploits - 1074 auxiliary - 330 post       ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 4 evasion                                       ]

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.193.135
LHOST => 192.168.193.135
msf5 exploit(multi/handler) > set exitonsession false
exitonsession => false
msf5 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.193.135:4444
msf5 exploit(multi/handler) >
```

Figura 37-2: Creación y explotación de msf venom

Realizado por: Herrera, Jonathan 2021.

Para crear el malware se usa el exploit multi/handler en conjunto con el payload de Windows para establecer una conexión en reversa tcp, se debe conocer la ip de la víctima, que en este caso es LAB-MICRO1-WIN7, el puerto de escucha será el 4444 por defecto. Una vez creado el malware procedemos a infectar a la víctima y a tener control total del equipo, como se observa en la figura 38-2, podemos revisar el escritorio de la víctima a través del comando run vnc.

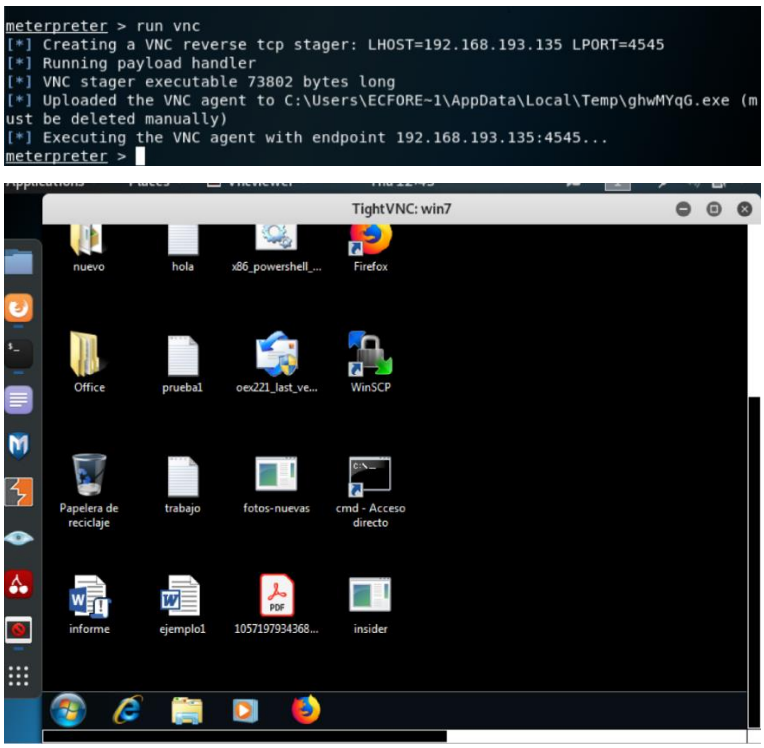


Figura 38-2: Usuario escalando en privilegios, escritorio vnc

Realizado por: Herrera, Jonathan 2021.

2.4.2.5. Evaluación de la plataforma OPNids mediante ataque de escaneo o probe

Como último ataque se encuentra la categoría de escaneo, que no es un ataque propiamente dicho, pero infringe las propiedades de la seguridad, este ataque es de los más importantes al momento que un atacante planea ejecutar uno y el primero que se realiza para encontrar vulnerabilidades que pueden ser aprovechadas por él. Puertos abiertos, servicios ejecutándose, versión de sistema operativo, software sin actualizar es información que puede obtener un escáner de red. La amenaza para la simulación de este ataque fue a través de la famosa herramienta Nmap. El ataque se ejecutó desde la máquina de Kali Linux en su versión gráfica llamada Zenmap, se utilizó esta herramienta debido a la facilidad que nos proporciona para poder escoger entre varios tipos de escaneos como: rápido, medio, intenso, todos los puertos, todos los servicios, entre otros y que es

una herramienta integrada dentro de la distribución de Kali, en la figura 39-2 se observa el escaneo de la red, su equivalente comando para nmap es: Nmap -p1-65535 -T4 -A -v 192.168.193.0./24

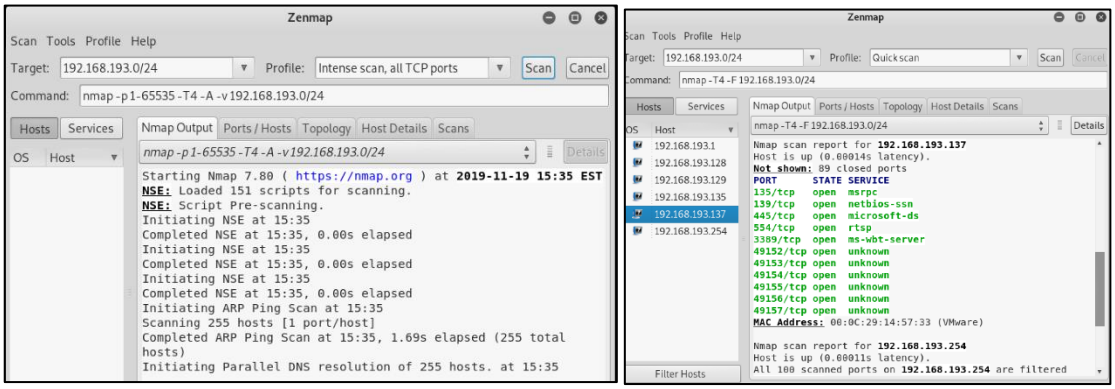


Figura 39-2: Escaneo de la red del escenario simulado

Realizado por: Herrera, Jonathan 2021.

El escaneo se encarga en inundar de tráfico ICMP tipo 7 "echo request" a toda la red. La Figura 39-2 muestra la información obtenida del escaneo realizado a toda la red como es: los hosts conectados a la red, puertos TCP o UDP abiertos, los servicios que utilizan dichos puertos hasta con la versión instalada, la misma que si resulta estar sin actualizar o soporte de seguridad es una potencial vulnerabilidad. El ataque ha sido exitoso, los resultados del ataque serán detallados en el capítulo 3.

CAPÍTULO III

3. GESTIÓN DEL PROYECTO

3.1. Resultados y análisis

En el presente capítulo se muestran los resultados obtenidos del análisis de tráfico de datos en el edificio de la FIE a través de la plataforma OPNids, en las cuales se evidencia el tráfico de datos analizado, las amenazas de ataques informáticos recibidos en la infraestructura de red del edificio de la FIE, resultados del entrenamiento del motor de machine learning DragonFly y los resultados de la evaluación de la plataforma OPNids a través de los ataques informáticos simulados en un escenario virtual. La gestión del proyecto se enmarca a nivel investigativo y desarrollo en laboratorios controlados en búsqueda de nuevas opciones Open Source que ofrece el mercado de la seguridad informática.

3.2. Tráfico de datos y amenazas informáticas detectadas en la red del edificio de la FIE

Mediante el análisis de tráfico de datos del edifico de la FIE explicado en el capítulo 2 se obtuvo como resultados la presencia de tráfico de datos malicioso representado a través de las amenazas informáticas detectadas por la plataforma OPNids. Las amenazas son eventos variables que se pueden presentar en mayor o menor medida, todo depende del comportamiento de la red y la información que se transmitan en ella. El tráfico de datos de la FIE está formado por tres Vlan: Estudiantes, Docentes y administrativos.

La Vlan estudiantes presentó una mayor cantidad de amenazas y es lógico, pues la red está al servicio de todos los estudiantes que la utilizan para desarrollar sus actividades académicas además de que tienen uso libre de conectar cualquier dispositivo a la red, por lo general sus computadores portátiles, al contrario, con la vlan docentes y administrativos que registraron una cantidad menor de amenazas, de igual forma el comportamiento de estas vlan es lógico ya que los docentes utilizan los equipos menos tiempo debido a su carga horaria y el personal administrativo no presento un cambio mayor.

3.2.1. Tráfico de datos y amenazas detectadas en la Vlan Estudiantes

El análisis de tráfico de datos de la Vlan estudiantes comenzó en semana de pruebas desde el 02 de diciembre de 2019 aproximadamente a las 8 AM, se analizó el tráfico durante toda la jornada académica diurna y vespertina con la finalidad de comprobar el funcionamiento de la plataforma, registro de alertas y análisis de paquetes. El análisis se realizó en dos meses; el primero inició el 09 de diciembre de 2019 a las 8 AM y culminó el 02 de enero de 2020, el segundo inició del 02 de enero al 03 de febrero de 2020, siendo estos meses los que se registró mayor tráfico de datos en la red de la FIE con gran cantidad de usuarios activos. El procedimiento se detalló en el capítulo 2 y se obtuvo los siguientes resultados:

Tabla 1-3: Estadísticas del tráfico de datos analizados por OPNids en la Vlan Estudiantes

Mes Diciembre		
Descripción	**# de paquetes**	**Porcentaje del total**
Paquetes in	218776744	100%
Paquetes out	5323	0.0024%
Total Bytes	37.23 GB	
Mes Enero		
Paquetes in	205686423	100%
Paquetes out	4820	0.0023%
Total Bytes	35.50 GB	

Realizado por: Herrera, Jonathan 2021.

En la tabla 1-3 se observa la cantidad de tráfico de datos, los paquetes de entrada y salida que analizo OPNids de los cuales, en el mes de diciembre 2019 analizó un total de 218M de paquetes de entrada siendo el 100% y un 0.0024% de paquetes de salida, esto tiene una explicación ya que el servidor posee configurado un puerto SPAN (promiscuo) y monitorea todo el tráfico de la interfaz de red de bajada mas no de subida. El total de bytes analizados fue de 37.23 Gb. En el segundo mes correspondiente a enero 2020 y los primeros días de febrero, OPNids analizo 205M de paquetes de entrada siendo el 100% del mes de enero 2020 y un 0.0023% de paquetes de salida. El total de bytes analizados fue de 35.50 GB. En el anexo A se puede observar las estadísticas de los paquetes in/out de la vlan estudiantes.

El registro de las alertas detectadas se lo realizo todos los días a las 8:30 am, sin embargo, las alertas de los sábados y domingos se registraban los lunes. Este fue un proceso repetitivo en el que se pudo observar el comportamiento de la red de la FIE y de la vlan estudiantes.

Las amenazas que se registraron son de diferentes tipos en la vlan estudiantes, en general se detectó presencia de malware, gusanos, troyanos, exploits, rootkits, escaneo de red y puertos, intentos de DoS, los cuales fueron detectados por OPNids en el tráfico de datos de la vlan. Las direcciones IP tanto de origen como destino eran diferentes, sus objetivos cambiaban día a día e intentaban buscar vulnerabilidades de un computador a otro conectado en la Vlan Estudiantes. En las tablas siguientes se muestra un resumen de las principales amenazas detectadas por OPNids en los dos meses de análisis respectivamente con su respectivo identificador de seguridad (SID) de Suricata IDS el que nos permite saber que alerta es y a que posible amenaza corresponde.

Tabla 2-3: Resumen de amenazas registradas en la Vlan Estudiantes en diciembre 2019

SID	Alerta	Descripción	Código
2013504	ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management	Violación de política de actualización de paquetes en sistemas GNU/Linux a través del gestor de paquetes APT	Probe
2013505	ET POLICY GNU/Linux YUM User-Agent Outbound likely related to package management	Violación de política de actualización de paquetes en sistemas GNU/Linux a través del gestor de paquetes YUM	
2001569	ET SCAN Behavioral Unusual Port 445 traffic Potential Scan or Infection	Escaneo inusual a través del puerto 445 en el cual puede existir una infección potencial.	
2001579	ET SCAN Behavioral Unusual Port 139 traffic Potential Scan or Infection	Escaneo inusual a través del puerto 139 en el cual puede existir una infección potencial.	
2022973	ET POLICY Possible Kali Linux hostname in DHCP Request Packet	Violación de política, posible host de Kali Linux en petición de paquete DHCP.	
2019980	ET POLICY External IP Check myexternalip.com	Escaneo de la IP publica con fines para dirigir un ataque remoto.	
2260002	SURICATA Applayer Detect protocol only one direction	Potencial conexión Command-Control detectada en capa de aplicación. Se puede tratar de un malware, mineros o una red Botnet.	R2L
2019102	ET DOS Possible SSDP Amplification Scan in Progress	Potencial ataque DOS a través del protocolo (SSDP) utilizado para la detección de dispositivos Plug & Play (UPnP). El protocolo es vulnerable a ataques DoS y DDoS.	DoS

Tabla 2-3 (continuación)

2018131	ET WORM TheMoon.linksys.router 1	Malware The Moon afecta a determinados routers Linksys Wi-Fi y puntos de acceso y enrutadores Wireless-N. The Moon omite la autenticación en el router iniciando sesión. Una vez infectado, el router comienza a inundar la red con el tráfico saliente de los puertos 80 y 8080.	U2R
2018372	ET EXPLOIT Malformed HeartBeat Request	El atacante aprovecha la vulnerabilidad de HeartBeat OpenSSL para evadir la autenticación multifactor de un VPN.	
2015474	ET TROJAN ZeroAccess, ZeuS udp traffic detected	Es un malware troyano ZeroAccess o Zeus y todas sus versiones. Utiliza un rootkit de bajo nivel para ser persistente y realizar fraudes por clic además de ser un backdoor para múltiples propósitos.	R2L

Realizado por: Herrera, Jonathan 2021.

La tabla 2-3 presenta las amenazas registradas en mayor número e importancia, el intento de ataque de comando y control, detectado a través de la capa de aplicación por el OPNids con el SID 2260002; una amenaza de alta prioridad ya que si el atacante logra tener éxito se hace del control de la víctima, llegando a convertirse en una amenaza persistente en busca de escalar privilegios y obtener acceso o puede llegar a consumir recursos informáticos como un criptominero o Botnets , el malware Zeus o troyano ZeroAccess con SID 2015474; identificados para realizar un fraude de pago por clics en conjunto de su backdoor para la instalación de ransomware, envenenamiento web, entre otros, además de un intento de denegación de servicios (DoS) con SID 2019102; que se registró el 20 de diciembre de 2019 a través de una vulnerabilidad del protocolo SSDP utilizado para la detección de dispositivos Plug & Play (UPnP), el protocolo es vulnerable a ataques DoS y DDoS basados en reflexión, los mensajes de respuesta grandes podrían hacer que el servicio consuma recursos del servidor y provocar un ataque exitoso. Escaneos a la red con el SID 2001569; en busca de la vulnerabilidad del protocolo SMB a través de los puertos 445 y 139 para posibles infecciones. Estas amenazas informáticas detectadas por

la plataforma OPNids se relacionan al uso de internet, los estudiantes navegan por todo tipo de páginas sin tener un control e infectan la red y generan potenciales ataques informáticos dirigidos a la infraestructura de red de la FIE y por ende de la ESPOCH.

En el mes de enero las alertas registradas disminuyeron debido a que los estudiantes de la facultad entraron en proceso de evaluaciones académicas y días feriados, por ende, los laboratorios dejaron de ser utilizados de forma habitual, pero aun así existieron amenazas importantes que deben ser analizadas. En la tabla 3-3 se presenta las amenazas detectadas en el mes de enero y primeros días de febrero 2020.

Tabla 3-3: Resumen de amenazas registradas en la Vlan Estudiantes en enero 2020

SID	Alerta	Descripción	Código
2260000	SURICATA Applayer Detect protocol only one direction	Potencial conexión Command-Control detectada en capa de aplicación. Se puede tratar de un malware, mineros o una red Botnet.	R2L
2008060	ET MALWARE DoublePulsar	Intento de intrusión a través del backdoor Doublepulsar, utilizado para ejecutar malware de cualquier tipo en la víctima.	U2R
2000538	ET SCAN NMAP -sA (1)	Se realizo un escaneo de rango de ip´s con la herramienta Nmap.	Probe
2008052	ET MALWARE User-Agent (Internet Explorer)	Malware descargado por el usuario a través de la instalación de barras de herramientas en cubiertas con software malicioso.	U2R
2018372	ET EXPLOIT Malformed HeartBleed Request	Se explota la vulnerabilidad OpenSSL de Heartbleed para eludir la autenticación multifactor en las VPN (CVE-2014-0160).	R2L
2001978	ET POLICY SSH session in progress on Expected Port	Violación de política, conexión a través de SSH por puerto externo.	
2025712	ET POLICY SMB Remote AT Scheduled Job Create Request – Possible Lateral Movement	Violación de política utilizando posible técnica de movimiento lateral para ataque a través del protocolo SMB.	

Tabla 3-3 (continuación)

2027189	ET NETBIOS DCERPC DCOM ExecuteShellCommand Call - Likely Lateral Movement	Posible técnica de movimiento lateral utilizada para tráfico SMB, MSIL/fhRansum, Kaprav, Phishing, Móviles.	U2R
2260002	SURICATA Applayer Detect protocol only one direction	Potencial conexión Command-Control detectada en capa de aplicación. Se puede tratar de un malware, mineros o una red Botnet.	
2001815	ET DELETED Spambot Suspicious 220 Banner on Local Port	Sospecha sobre posible bot de spam utilizando el protocolo SMTP.	DoS

Realizado por: Herrera, Jonathan 2021.

La tabla 3-3 presenta las principales amenazas informáticas detectadas por la plataforma OPNids en el mes de enero y los primeros días de febrero de 2020, de igual manera que en el mes de diciembre 2019 las amenazas presentes son de diferentes tipos, empezando por escaneos identificados con el SID 2000538; realizados con la herramienta Nmap en la red de la FIE en busca de vulnerabilidades a través de los puertos o servicios, intentos de robo de información con el SID 2008052; utilizando una vulnerabilidad de Internet Explore en Windows 7/8/10 esta amenaza es producto de agentes externos, es decir, los estudiantes que conectaban sus laptops en la red mientras se efectuaba el análisis y generaban dicha amenaza, además de intentos de encriptación de información con el SID 2025712; a través del ataque con el Backdoor Doublepulsar explotando la vulnerabilidad del protocolo SMB versión 1 o 2 con el objetivo de ejecutar ransomware, persisten intentos de intrusión de comando y control a través de malware con el SID 2260002; una amenaza de alta prioridad ya que si el atacante logra tener éxito se hace del control de la víctima, llegando a convertirse en una amenaza persistente en busca de escalar privilegios y obtener acceso o puede llegar a consumir recursos informáticos como un criptominero o Botnets y para finalizar con el SID 2001978; una amenaza sobre violación a políticas e intento de conexión SSH. En el anexo B se observa de manera detallada el registro de las amenazas en los dos meses de análisis de la vlan estudiantes.

3.2.2. Tráfico de datos y amenazas detectadas en la Vlan Docentes

El análisis de tráfico de datos de la Vlan docentes tuvo una duración de un mes aproximadamente, empezó del 03 de febrero al 02 de marzo de 2020 a las 08:00 AM; se analizó el tráfico durante toda la jornada laboral. Este mes se evidencio una cantidad de alertas baja acorde al tipo de vlan y los usuarios en ella. El procedimiento se detalló en el capítulo 2 y se obtuvo los siguientes resultados:

Tabla 4-3: Estadísticas del tráfico de datos analizados por OPNids en la Vlan Docentes

Mes Febrero		
Descripción	**# de paquetes**	**Porcentaje del total**
Paquetes in	23200341	100%
Paquetes out	4761	0.021%
Total Bytes	4.30 GB	

Realizado por: Herrera, Jonathan 2021.

En la tabla 4-3 se observa la cantidad de tráfico de datos, los paquetes de entrada y salida que analizo OPNids de los cuales, en el mes de febrero 2020 analizó un total de 23M de paquetes de entrada siendo el 100% y un 0.021% de paquetes de salida, sus valores son considerablemente bajos en comparación con la Vlan estudiantes, pero lógicos con respecto a lo mencionado en párrafos anteriores. El total de bytes analizados fue de 4.30 Gb. En el anexo C se observa las estadísticas de los paquetes in/out de la vlan docentes. El registro de las alertas detectadas se lo realizo todos los días a las 8:30 am, sin embargo, las alertas de los sábados y domingos se registraban los lunes. Este fue un proceso repetitivo en el que se pudo observar el comportamiento de la red de la FIE y de la vlan docentes.

Las amenazas detectadas en la Vlan docentes son muy similares a las que se encontraron en la vlan estudiantes, pero en menor cantidad, se detectó presencia de escaneos de red, malware, troyanos, los cuales fueron detectados por OPNids en el tráfico de datos de la vlan. Las direcciones IP tanto de origen como destino eran diferentes, sus objetivos cambiaban día a día e intentaban buscar vulnerabilidades de un computador a otro conectado en la Vlan docentes. En la tabla siguiente se muestra un resumen de las principales amenazas detectadas por OPNids en el mes de análisis con su respectivo identificador de seguridad (SID) de Suricata IDS el que nos permite saber que alerta es y a que posible amenaza corresponde.

Tabla 5-3: Resumen de amenazas registradas en la Vlan docentes en febrero 2020

SID	Alerta	Descripción	Código
2260000	SURICATA Applayer Detect protocol only one direction	Potencial conexión Command-Control detectada en capa de aplicación. Se puede tratar de un malware, mineros o una red Botnet.	R2L
2000538	ET SCAN NMAP -sA (1)	Se realizo un escaneo de rango de ip´s con la herramienta Nmap.	Probe
2018372	ET EXPLOIT Malformed HeartBleed Request	Se explota la vulnerabilidad OpenSSL de Heartbleed para eludir la autenticación multifactor en las VPN (CVE-2014-0160).	U2R
2001978	ET POLICY SSH session in progress on Expected Port	Violación de política, conexión a través de SSH por puerto externo.	R2L

Realizado por: Herrera, Jonathan 2021.

La tabla 5-3 presenta las principales amenazas informáticas detectadas por la plataforma OPNids en el mes de febrero de 2020, empezando por escaneos identificados con el SID 2000538; realizados con la herramienta Nmap en la red de la FIE en busca de vulnerabilidades a través de los puertos o servicios, intentos de conexión remota con el SID 2001978; utilizando una violación de políticas en una sesión SSH establecida por un puerto externo, además de uso de exploit con el SID 2018372; a través de la vulnerabilidad de OpenSSL para evadir autenticación a través de VPN y para finalizar persisten intentos de intrusión de comando y control a través de malware con el SID 2260000; una amenaza de alta prioridad ya que si el atacante logra tener éxito se hace del control de la víctima, llegando a convertirse en una amenaza persistente en busca de escalar privilegios y obtener acceso o puede llegar a consumir recursos informáticos como un criptominero o Botnets. En el anexo D se observa de manera detallada el registro de las amenazas del análisis de tráfico de datos de la vlan docentes.

3.3. Resultados del entramiento simulado del motor de machine learning DragonFly

En esta sección se muestra los resultados obtenidos del entrenamiento simulado del motor de machine learning DragonFly en conjunto con el script adicional a través de los dos conjuntos de datos (datasets). El procedimiento se detalla en el capítulo 2.

3.3.1. Resultados del entramiento del motor de machine learning con KDD99

El motor de machine learning en conjunto con el script adicional utilizando el dataset KDD99 para el entrenamiento del algoritmo de clasificación supervisado Random Forest obtuvo los resultados que se observa en la Figura 1-3. El procedimiento para obtener estos resultados fue explicado en el capítulo 2 y el algoritmo programado se encuentra en el Anexo E.

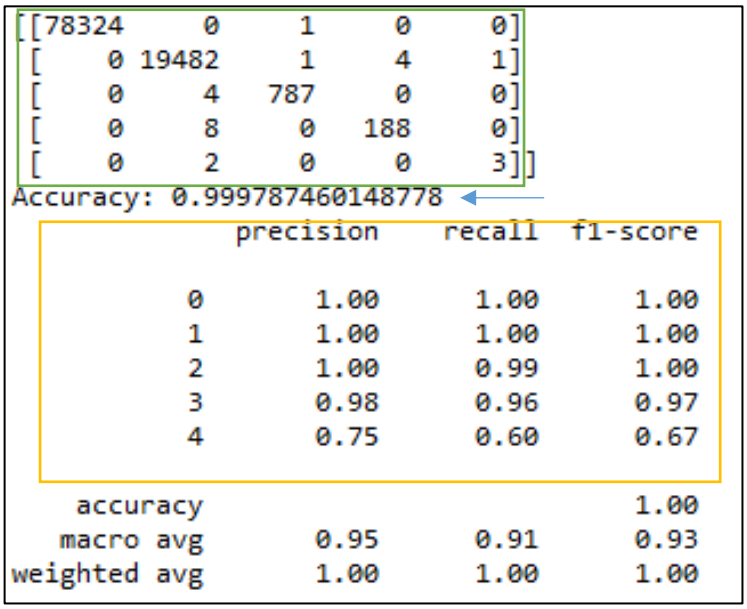


Figura 1-3: Resultados del entrenamiento del algoritmo con KDD99

Realizado por: Herrera, Jonathan 2021.

La Figura 1-3 consta de tres partes, en primer lugar, dentro del recuadro de borde verde está la matriz de confusión del algoritmo, a través de ella se puede calcular todos los parámetros que necesitamos para la evaluación de este, a continuación, se observa la "Accuracy" o precisión y para finalizar en el recuadro de borde amarillo se observan los parámetros calculados por categoría.

La "Accuracy" o precisión es uno de los parámetros más importantes ya que como su nombre lo indica, nos afirma que tan preciso puede llegar a ser el modelo, su valor tiende a aproximarse a 1 o su equivalente al 100%, ese valor se lo puede obtener después de realizar varias pruebas al modelo de machine learning, sin embargo, es algo que únicamente se puede lograr de forma ideal, ya que en la práctica se busca una aproximación lo más cercana al 100%. La precisión se define como la relación entre muestras clasificadas correctamente y el número total de muestras (Farnaaz & Jabbar, 2016, pp. 215). La precisión calculada con el conjunto KDD99 es de 99% y se puede calcular con la ecuación 1-3:

$$A = \frac{(SumDP)}{(SumDP + SumRV\ )}$$

Ecuación 1-3: Precisión del algoritmo RF.

Donde:

$SumDP =$ Suma de los valores de la diagonal principal.

$SumRV =$ Suma del resto de los valores de la matriz.

Para calcular la sensibilidad o recall, el puntaje F1 y cualquier otro parámetro como ya se mencionó en párrafos anteriores se utilizó la matriz de confusión, como se observa en la tabla 6-3.

Tabla 6-3: Matriz de confusión.

		Clasificación	
		Normal	Ataque
Dataset	Normal	TP	FN
	Ataque	FP	TN

Realizado por: Herrera, Jonathan 2021.

Donde:

$True\ Positive\ (TP) =$ Verdaderos positivos, no existe amenazas.

$False\ positive\ (FP) =$ Falsos positivos, genera una alerta de ataque a pesar de no existir una amenaza.

$False\ negative\ (FN) =$ Falsos negativos, no genero una alerta de ataque a pesar de existir una amenaza.

$True\ negative\ (TN) =$ Verdadero negativo, existe amenazas.

La sensibilidad o recall nos permite saber si el algoritmo de machine learning está clasificando de una manera correcta y de cuan sensible puede llegar a ser de existir algún cambio. Se calcula con la ecuación 2-3:

$$S = \frac{TP}{TP + FN}$$

Ecuación 2-3: Sensibilidad del algoritmo RF.

Ejecutar varias veces el script cada vez que se necesita observar cambios de los parámetros con respecto al algoritmo de machine learning, es sencillo; cuando se maneja conjuntos de datos predeterminados como en este caso en particular un dataset predeterminado KDD99, pero cuando él conjunto de datos posea un volumen de información muy alto como lo es el de la FIE representará un gran consumo de recursos informáticos y de tiempo, por tal razón se puede calcular el parámetro puntaje F1 que representa la relación entre la precisión y sensibilidad en un único valor. Se calcula con la ecuación 3-3.

$$F1 = \frac{2 * presicion * sensibilidad}{presicion + sensibilidad}$$

Ecuación 3-3: Puntaje F1 del algoritmo RF.

En la tabla 7-3 se observa los parámetros de precisión, sensibilidad o recall y puntaje F1 cada uno de ellos calculado por categoría. Al tratarse de una simulación con un conjunto de datos predeterminado para entrenamiento de los algoritmos de machine learning es común obtener valores del 100% como en las categorías DoS y Probe, un porcentaje más apegado a la realidad es el que muestra "U2R" con un 98%, al contrario lo que sucede con "R2L" con el 75% es un valor que no es aceptable, pero esto se debe porque el conjunto KDD99 posee pocas muestras de este tipo de ataques, por consecuencia con un dataset más completo se podrá mejorar los parámetros de evaluación.

Tabla 7-3: Parámetros de evaluación por categoría KDD99

Categoría	Precisión	Sensibilidad o recall	Puntaje F1
DoS	100%	100%	100 %
U2R	98%	96%	97%
R2L	75%	60%	67%
PROBE	100%	99%	100%

Realizado por: Herrera, Jonathan 2021.

3.3.2. Resultados del entramiento del motor de machine learning con OPNids-eve.json

El log eve.json del IDS Suricata integrado en la plataforma OPNids fue descargado del servidor de la FIE, ya que este archivo posee todos los flujos de red del IDS en los dos meses del análisis de tráfico de datos los cuales fueron utilizados para el entramiento del motor de machine learning

con su algoritmo Random forest como se observan en la Figura 2-3. El procedimiento para obtener estos resultados fue explicado en el capítulo 2.

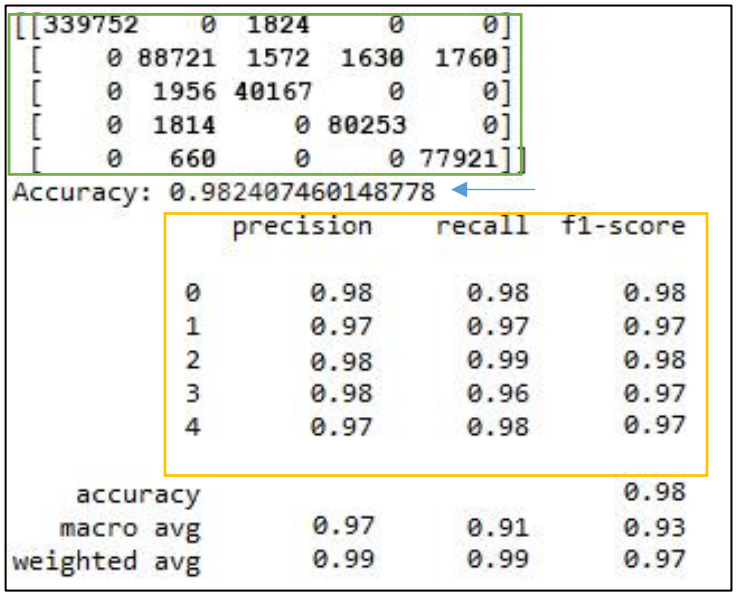


Figura 2-3: Resultados del entrenamiento del algoritmo con OPNids-eve.json

Realizado por: Herrera, Jonathan 2021.

La Figura 2-3 consta de tres partes, en primer lugar, dentro del recuadro de borde verde está la matriz de confusión del algoritmo, a través de ella se puede calcular todos los parámetros que necesitamos para la evaluación de este, a continuación, se observa la "Accuracy" o precisión y para finalizar en el recuadro de borde amarillo se observan los parámetros calculados por categoría.

La "Accuracy" o precisión del modelo calculada con el conjunto de datos OPNids-Suricata del log eve.json es de 98%, como se observa es un dato mucho más real al compáralo con el de KDD99 por varias razones como son: un conjunto de datos de mayor volumen, técnica de ensamble de árboles "Bootstrap", numero de estimadores (número de árboles) a implementar en el "bosque"; se utilizó un estimado de 100 árboles, valor por defecto; sin embargo, el numero dependerá en gran medida a los recursos informáticos que posea el computador que alberga a la plataforma OPNids, además se debe tener en cuenta el sobre ajuste del modelo, no por más estimadores que se usen va a mejorar el algoritmo y si lo hiciera la diferencia seria mínima cayendo en el error de sobre ajuste.

Tabla 8-3: Parámetros de evaluación por categoría OPNids-eve.json

Categoría	Precisión	Sensibilidad o recall	Puntaje F1
DoS	98 %	98 %	98 %
U2R	97 %	98 %	97 %
R2L	98 %	96 %	97 %
PROBE	98 %	99 %	98 %

Realizado por: Herrera, Jonathan 2021.

En la tabla 8-3 se observa los parámetros de precisión, sensibilidad o recall y puntaje F1 cada uno de ellos calculado por categoría. Al tratarse de una simulación con un conjunto de datos reales, filtrados, limpiado y categorizados se logra un modelo de machine learning muy preciso. En todas las categorías tenemos un promedio del 98% en general que nos da la certeza de un modelo preciso y con gran sensibilidad de clasificación. Con la categoría "R2L" tiene 96 % de sensibilidad es un valor aceptable, que puede mejorar con forme un mayor tiempo de entrenamiento y aumento del volumen de datos registrados, sin caer en los errores mencionados.

3.4. Resultados de la evaluación de la plataforma OPNids a través de la simulación de las amenazas detectadas mediante un escenario virtual

Con el entrenamiento del motor de machine learning DragonFly de la plataforma OPNids, se obtuvo un resultado de precisión y sensibilidad del 98%. Este resultado nos permite concluir que la plataforma OPNids a través del IDS Suricata y el motor de machine learning DragonFly realiza una efectiva detección de las amenazas informáticas en la intranet del edificio de la FIE, por esta razón para evaluar el desempeño y efectividad de la plataforma OPNids en la detección de tráfico de datos malicioso se procedió a realizar 4 ataques informáticos, uno por cada categoría; DoS, U2R, R2L y Probe, los cuales fueron escogidos por medio del registro de las amenazas del análisis de tráfico de datos de la FIE. El procedimiento de cada ataque se detalla en la etapa 3 del capítulo 2.

3.4.1. Resultados de la evaluación de la plataforma OPNids mediante ataque DoS Slowloris

Para evaluar el desempeño y efectividad de la plataforma OPNids se realizó un ataque informático de denegación de servicio (DoS) cuyo procedimiento se lo detallo en la etapa 3 del capítulo 2. A continuación, procedemos a revisar los resultados del ataque informático.

La plataforma OPNids estuvo analizando el tráfico de datos durante todo tiempo de ejecución del ataque informático, en la Figura 3-3 se presenta la alerta del ataque DoS generada a través del IDS Suricata comprobando el correcto desempeño de la plataforma OPNids en la detección de tráfico de datos malicioso en la red simulada del laboratorio de la FIE.

Figura 3-3: Alerta del ataque DoS en la plataforma OPNids

Realizado por: Herrera, Jonathan 2021.

Para evaluar la efectividad de la plataforma OPNids en la detección del tráfico de datos malicioso se ejecutó 40 pruebas seguidas del ataque con diferente número de paquetes y se registró los tiempos de detección como se observa en la tabla 9-3.

Tabla 9-3: Detección del ataque DoS 40 veces con OPNids

Prueba	Tiempo de envió de paquetes (segundos)	Paquetes	Tiempo de detección (segundos)
1	1	2000	10:18:39.374 hora inicial
2	1	2185	40.508
3	1	2370	41.868
4	1	2555	42.123
5	1	2740	43.937
6	1	2925	44.353
7	1	3110	45.653
8	1	3295	46.373
9	1	3480	47.579
10	1	3665	48.419

Tabla 9-3 (continuación)

11	1	3850	49.879
12	1	4035	50.125
13	1	4220	51.476
14	1	4405	52.298
15	1	4590	53.473
16	1	4775	54.506
17	1	4960	55.256
18	1	5145	56.367
19	1	5330	57.478
20	1	5515	58.589
21	1	5700	59.189
22	1	5885	01.675
23	1	6070	02.493
24	1	6255	03.438
25	1	6440	04.598
26	1	6625	04.976
27	1	6810	05.146
28	1	6995	05.479
29	1	7180	05.664
30	1	7365	05.873
31	1	7550	06.167
32	1	7735	06.317
33	1	7920	06.768
34	1	8105	06.918
35	1	8290	07.269
36	1	8475	07.520
37	1	8660	07.770
38	1	8845	07.921
39	1	9030	08.272
40	1	9215	08.422

Realizado por: Herrera, Jonathan 2021.

Una vez realizada las 40 pruebas se produce la detección del ataque informático por consecuencia se generan las alertas por parte del IDS suricata, el tiempo de detección en promedio es aproximadamente 1 segundo, sin embargo, el motor de machine learning produce un efecto de

solape ya que a partir de la prueba #26 se observa una detección dentro del mismo segundo y avanza progresivamente, entre las pruebas #27 al #40 se diferencia que el tiempo de detección del ataque se redujo en promedio aproximadamente a 255 ms. Como se observa en el grafico 1-3 a partir de la prueba #27 se evidencia una estabilidad de la detección del ataque informático, tomando en cuenta que se encuentra en un ambiente simulado la plataforma OPNids presenta mejoras en la detección de ataques de tipo DoS.

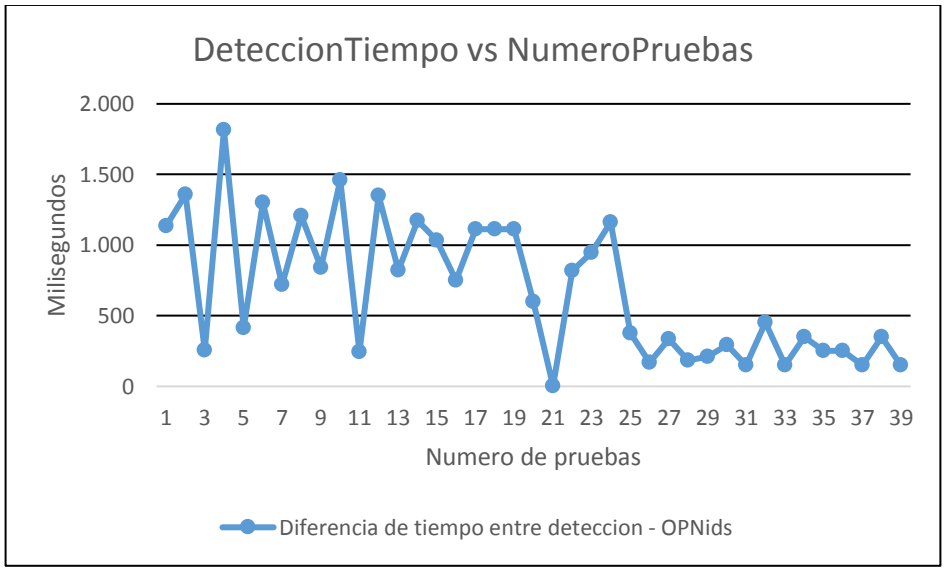


Gráfico 1-3: Diferencia de tiempo entre detecciones en ataque DoS

Realizado por: Herrera, Jonathan 2021.

3.4.2. Resultados de la evaluación de la plataforma OPNids mediante ataque de Remote to Local (R2L)

Para evaluar el desempeño y efectividad de la plataforma OPNids se realizó un ataque informático de Remote to local (R2L) cuyo procedimiento se lo detallo en la etapa 3 del capítulo 2. A continuación, procedemos a revisar los resultados del ataque informático.

La plataforma OPNids estuvo analizando el tráfico de datos durante todo tiempo de ejecución del ataque informático, en la Figura 4-3 se presenta la alerta del ataque R2L generada a través del IDS Suricata comprobando el correcto desempeño de la plataforma OPNids en la detección de tráfico de datos malicioso en la red simulada del laboratorio de la FIE.

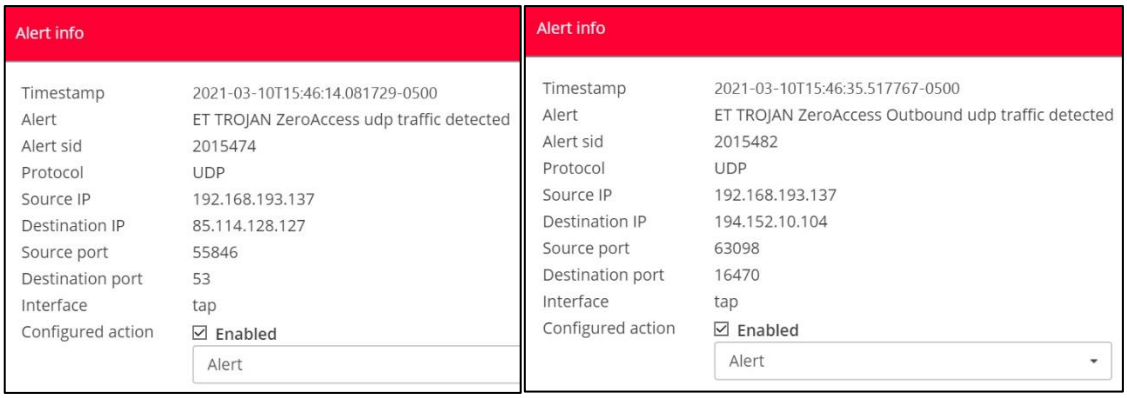


Figura 4-3: Alerta del ataque malware ZeroAccess en la plataforma OPNids

Realizado por: Herrera, Jonathan 2021.

Para evaluar la efectividad de la plataforma OPNids en la detección del tráfico de datos malicioso se ejecutó el ataque informático repetidamente durante 30 minutos y se registró el número de alertas con su respectiva hora inicial y final de detección como se observa en la tabla 10-3.

Tabla 10-3: Detección del ataque R2L durante 30 minutos con OPNids

Tiempo de ataque	# de alertas	Tiempo de detección (minutos)
10 minutos	50	15:46:14 (inicial) / 15:56:20 (final)
20 minutos	55	15:56:44 (inicial) / 16:06:36 (final)
30 minutos	65	16:06:54 (inicial) / 16:16:22 (final)

Realizado por: Herrera, Jonathan 2021.

Como se observa en la Tabla 10-3, durante los 30 minutos se realizaron aproximadamente 20 ataques informáticos simulados hacia la víctima, en la cual se registraron todas las alertas generadas por la plataforma OPNids evaluando el desempeño en la detección de tráfico de datos malicioso, el resultado confirma que mientras aumenta el tiempo del ataque las alertas de igual manera ya que existe una mejora en la detección, sin embargo en este tipo de ataques no se observa una mejora en la efectividad de la plataforma OPNids; ya que así el número de ataques aumente el resultado será el mismo con respecto al tiempo de detección, pero esto tiene una explicación y es que el vector de ataque de un malware envía una cantidad limitada de paquetes para la infección y no genera un patrón para que el motor de machine learning DragonFly sea más efectivo.

3.4.3. Resultados de la evaluación de la plataforma OPNids mediante ataque de User to Root (U2R)

Para evaluar el desempeño y efectividad de la plataforma OPNids se realizó un ataque informático de Usuario a Root (U2R) cuyo procedimiento se lo detallo en la etapa 3 del capítulo 2. A continuación, procedemos a revisar los resultados del ataque informático.

La plataforma OPNids estuvo analizando el tráfico de datos durante todo tiempo de ejecución del ataque informático, en la Figura 5-3 se presenta la alerta del ataque U2R generada a través del IDS Suricata comprobando el correcto desempeño de la plataforma OPNids en la detección de tráfico de datos malicioso en la red simulada del laboratorio de la FIE.

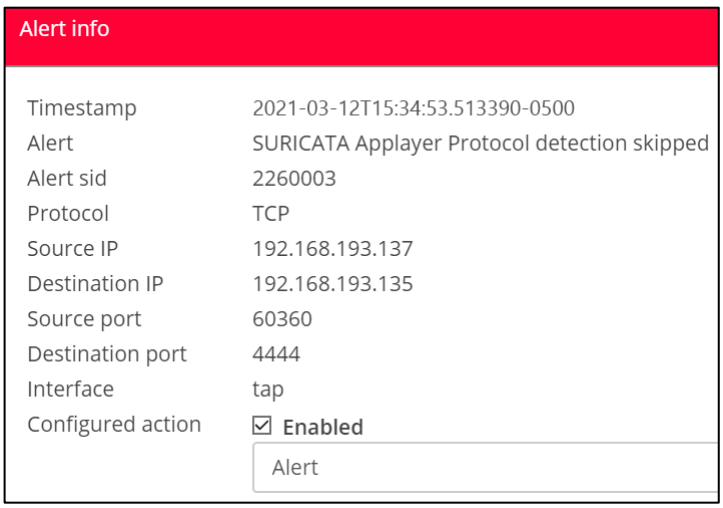


Figura 5-3: Alerta del ataque malware MSF Venom en la plataforma OPNids

Realizado por: Herrera, Jonathan 2021.

Para evaluar la efectividad de la plataforma OPNids en la detección del tráfico de datos malicioso se ejecutó el ataque informático repetidamente durante 30 minutos y se registró el número de alertas con su respectiva hora inicial y final de detección como se observa en la tabla 11-3.

Tabla 11-3: Detección del ataque U2R durante 30 minutos con OPNids

Tiempo de ataque	# de alertas	Tiempo de detección (minutos)
10 minutos	30	15:34:50 (inicial) / 15:44:30 (final)
20 minutos	33	15:44:48 (inicial) / 15:54:40 (final)
30 minutos	40	15:54:58 (inicial) / 16:04:20 (final)

Realizado por: Herrera, Jonathan 2021.

Como se observa en la Tabla 11-3, durante los 30 minutos se realizaron aproximadamente 20 ataques informáticos simulados hacia la víctima, en la cual se registraron todas las alertas generadas por la plataforma OPNids evaluando el desempeño en la detección de tráfico de datos malicioso, el resultado confirma que mientras aumenta el tiempo del ataque las alertas de igual manera ya que existe una mejora en la detección, sin embargo en este tipo de ataques no se observa una mejora en la efectividad de la plataforma OPNids; ya que así el número de ataques aumente el resultado será el mismo con respecto al tiempo de detección, pero esto tiene una explicación y es que el vector de ataque de un malware envía una cantidad limitada de paquetes para la infección y no genera un patrón para que el motor de machine learning DragonFly sea más efectivo.

3.4.4. Resultados de la evaluación de la plataforma OPNids mediante ataque de escaneo o probe

Para evaluar el desempeño y efectividad de la plataforma OPNids se realizó un ataque informático de Escaneo (Probe) cuyo procedimiento se lo detallo en la etapa 3 del capítulo 2. A continuación, procedemos a revisar los resultados del ataque informático.

La plataforma OPNids estuvo analizando el tráfico de datos durante todo tiempo de ejecución del ataque informático, en la Figura 6-3 se presenta la alerta del ataque DoS generada a través del IDS Suricata comprobando el correcto desempeño de la plataforma OPNids en la detección de tráfico de datos malicioso en la red simulada del laboratorio de la FIE.

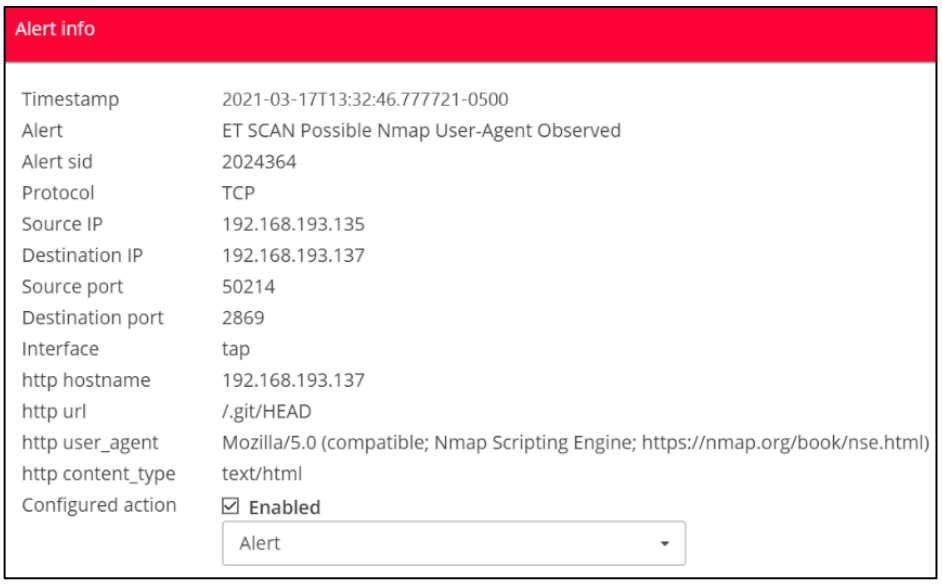


Figura 6-3: Alerta del ataque de escaneo Nmap en la plataforma OPNids

Realizado por: Herrera, Jonathan 2021.

Para evaluar la efectividad de la plataforma OPNids en la detección del tráfico de datos malicioso se ejecutó 40 pruebas seguidas del ataque y se registró los tiempos de detección como se observa en la tabla 12-3.

Tabla 12-3: Detección del ataque de escaneo 40 veces con OPNids

Prueba	Paquetes ICMP	Tiempo de detección (minutos)
1	1	13:32:46.778 hora inicial
2	1	34:26.876
3	1	36:30.537
4	1	38:27.483
5	1	40:10.829
6	1	42:35.497
7	1	44:45.748
8	1	46:33.452
9	1	48:55.416
10	1	50:29.183
11	1	52:38.475
12	1	54:06.631
13	1	56:22.963
14	1	58:52.482
15	1	14:00:05.834
16	1	02:54.536
17	1	04:35.941
18	1	06:56.183
19	1	08:41.237
20	1	10:18.842
21	1	12:38.189
22	1	14:01.539
23	1	15:25.768
24	1	15:58.956
25	1	16:15.679
26	1	16:55.997
27	1	17:05.146
28	1	17:50.883
29	1	18:10.137
30	1	18:21.537

Tabla 12-3 (continuación)

31	1	18:45.732
32	1	18:58.258
33	1	19:08.674
34	1	19:15.697
35	1	19:35.159
36	1	19:48.748
37	1	20:05.749
38	1	20:26.498
39	1	20:40.714
40	1	20:58.963

Realizado por: Herrera, Jonathan 2021.

Una vez realizada las 40 pruebas se produce la detección del ataque informático por consecuencia se generan las alertas por parte del IDS suricata, el tiempo de detección en promedio es aproximadamente 2:27 minutos, sin embargo, el motor de machine learning produce un efecto de solape ya que a partir de la prueba #23 se observa una detección dentro del mismo minuto y avanza progresivamente, entre las pruebas #24 al #40 se diferencia que el tiempo de detección del ataque se redujo en promedio aproximadamente a 31 segundos. Como se observa en el grafico 2-3 a partir de la prueba #24 se evidencia una estabilidad de la detección del ataque informático, tomando en cuenta que se encuentra en un ambiente simulado la plataforma OPNids presenta mejoras efectivas en la detección de ataques de tipo escaneo.

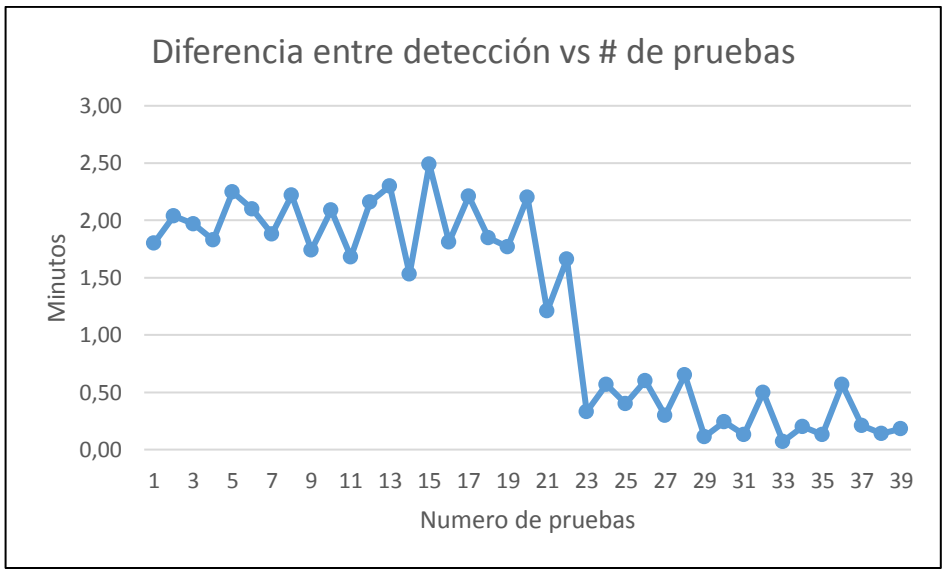


Gráfico 2-3: Diferencia de tiempo entre detecciones en ataque de escaneo
Realizado por: Herrera, Jonathan 2021.

CONCLUSIONES

- La detección de tráfico de datos malicioso en redes corporativas se ve reflejado hacia el nivel de seguridad interna como externa de la red y de los mecanismos para monitorear la misma, en la actualidad las técnicas actuales de detección señalan la implementación de un Sistema de Detección de Intrusos con machine learning como lo es la plataforma OPNids con su motor DragonFly capaz de crear un punto de inflexión en la detección de tráfico de datos malicioso y mejorar la autonomía de los IDS.

- La plataforma OPNids presenta varias ventajas en las que destacan su IDS Suricata, motor de machine learning DragonFly e interfaz web de control versus plataformas tradicionales como Snort y Zeek que únicamente son IDS basados en reglas/firmas en consola, pero si bien tienen la capacidad de obtener resultados similares, la plataforma OPNids brinda un conjunto combinado de herramientas capaces de automatizar los procesos de control de la seguridad de red del edifico de la FIE.

- La implementación de la plataforma OPNids en la Facultad de Informática y Electrónica (FIE) brinda a los estudiantes como docentes la posibilidad de interactuar con un sistema integrado a la seguridad informática de la red, lo que permite comprender los peligros de navegar por redes sin ningún control, es decir, que no son administradas, además de comprender las técnicas de machine learning con su algoritmo de clasificación Random forest y la detección de intrusos a través del IDS Suricata.

- La plataforma OPNids implementada cumple con el objetivo planteado ya que por un periodo aproximado a 3 meses y medio analizo más de 77 Gb de información proveniente de todo el tráfico de datos de la Facultad de Informática y Electrónica; en ese tráfico detecto aproximadamente 3 mil alertas de posibles ataques a través del IDS Suricata, a su vez enviados hacia el motor de machine learning DragonFly para su entrenamiento en conjunto con el srcipt programado adicionalmente, como resultado se obtuvo una precisión y sensibilidad del modelo de 98%.

- Se realizó la simulación de 4 ataques informáticos para evaluar el desempeño y efectividad de la plataforma OPNids en los cuales las categorías de DoS y Escaner (Probe) presentaron mejor desempeño en la detección de los ataques informáticos con una efectividad respecto al tiempo de 255 ms y 31 s respectivamente, al contrario, sucedió con los ataques de “U2R” y “R2L” en los cuales no existió mejoras en la efectividad, pero por motivos del vector de ataque, sin embargo, si existió mejora en el desempeño a través de la detección.

RECOMENDACIONES

- Se recomienda realizar el análisis de tráfico de datos del edificio de la FIE por un periodo de tiempo mayor al ejecutado en este trabajo de titulación, esto ayudará a determinar de una mejor manera las amenazas y posibles ataques informáticos, además de mejorar en el entramiento del motor de machine learning DragonFly.

- La plataforma OPNids en ocasiones presenta errores de programación ya que al dejar de tener soporte a partir de inicios de enero de 2020 no fueron solucionados los mismos, por ese motivo se recomienda mantener un control sobre el uso de los recursos informáticos utilizados como: procesador, ram y almacenamiento; verificarlos periódicamente aseguran un correcto funcionamiento.

- Para futuras pruebas del motor de machine learning Dragonfly se recomienda utilizar diferentes algoritmos de clasificación programados de manera correcta en el script adicional para obtener mejoras en los resultados y poder detectar de mejor manera otro tipo de ataques.

- El IDS suricata es capaz de manejar reglas propias, es decir, se puede implementar reglas personalizadas en la FIE para ejecutar un control de tráfico de datos como bloqueo de paquetes de redes sociales, descargas, VPN, entre otros.

- El servidor en que fue implementado la plataforma OPNids presentó problemas de almacenamiento y cortes de alimentación, se debe asegurar y monitorear la infraestructura física en la que se va a desplegar la plataforma.

- OPNids es una plataforma compleja al momento de querer aplicar algún cambio, ya que se encuentra basado en una distribución de freebsd y con un núcleo OPNsense modificado.

BIBLIOGRAFÍA

ABDULLAH, B.; et al. "Performance Evaluation of a Genetic Algorithm Based Approach to Network Intrusion Detection System". *International Conference on Aerospace Sciences and Aviation Technology* [en línea], 2009, vol. 13, no. AEROSPACE SCIENCES, pp. 1-2. [Consulta: 25 julio 2019]. Disponible en: https://www.researchgate.net/publication/303363543_Performance_Evaluation_of_a_Genetic_Algorithm_Based_Approach_to_Network_Intrusion_Detection_System/link/5ba13fd945851574f7d572b3/download

AGUIRRE, Cristobal. *ML part1: Introducción a los arboles de decisión* [blog]. 2019a. [Consulta: 12 enero 2020]. Disponible en: https://www.cristobal-aguirre.com/arboles-de-decision

AGUIRRE, Cristobal. *ML part2: Random Forests* [blog]. 2019b. [Consulta: 12 enero 2020]. Disponible en: https://www.cristobal-aguirre.com/random-forests

AHMAD, H.; et al. "An Overview of Intrusion Detection System (IDS) along with its Commonly Used Techniques and Classifications". *International Journal of Computer Science and Telecommunications* [en línea]. 2014, vol. 5, no. 2, pp. 22. [Consulta: 8 agosto 2019]. Disponible en: https://pdfs.semanticscholar.org/ad2e/6f2ddf676cbe22ddaffab23973ce263f7f41.pdf?_ga=2.75296337.294603642.1569359883-584894126.1569359883.

AL-SUBAIE, M. y ZULKERNINE, M. "Efficacy of Hidden Markov Models over neural networks in anomaly intrusion detection". *Proceedings - International Computer Software and Applications Conference* [en línea], 2006, vol. 1, pp. 328. [Consulta: 5 agosto 2019]. Disponible en: https://ieeexplore.ieee.org/abstract/document/4020093.

ÁLVAREZ, Jana. *Machine Learning y Support Vector Machine: porque el tiempo es dinero* [blog]. 2016. [Consulta: 12 enero 2020]. Disponible en: https://www.analiticaweb.es/machine-learning-y-support-vector-machines-porque-el-tiempo-es-dinero-2/

ASHOOR, A. y GORE, S. "Importance of Intrusion Detection System (IDS)". *International Journal of Scientific and Engineering Research 2011* [en línea], 2011, vol. 2, no. 1, pp. 3-6. [Consulta: 28 julio 2019]. Disponible en: http://www.ijser.org/researchpaper%5CImportance_of_Intrusion_Detection_System.pdf.

ASTUDILLO, J.; et al. Adaptacion del ids/ips suricata para que se pueda convertir en una solucion empresarial [en línea] (Tesis). Escuela Superior Politecnica del Litoral, Facultad de Ingenieria en Electricidad y Computacion -Ecuador. 2011 pp. 1-149. [Consulta: 15 enero 2020]. Disponible en: https://www.dspace.espol.edu.ec/bitstream/123456789/19502/2/tesina_seminario0.6.pdf.

BBC News. *Cuáles fueron los peores hackeos informáticos de la historia* [blog]. [Consulta: 18 julio 2019]. Disponible en: https://www.bbc.com/mundo/noticias-46426990

BEAL, Vangie. *Intrusion Detection (IDS) and Prevention (IPS) Systems* [blog]. [Consulta: 27 septiembre 2019]. Disponible en: https://www.webopedia.com/DidYouKnow/Computer_Science/intrusion_detection_prevention.asp

BENCHIMOL, D. *Hacking desde cero* [en línea]. 1era. Buenos Aires: s.n, 2011. [Consulta: 25 junio 2019]. Disponible en: http://www.tugurium.com/docs/HakingCero.pdf.

BELCIC, Ivan. *Rootkits: qué hacen, cómo funcionan y cómo eliminarlos* [blog]. [Consulta: 10 enero 2020]. Disponible en: https://www.avast.com/es-es/c-rootkit

BHATTACHARYYA, Indresh. *Support Vector Regression or SVR* [blog]. [Consulta: 10 enero 2020]. Disponible en: https://medium.com/coinmonks/support-vector-regression-or-svr-8eb3acf6d0ff

BOSAGH, R. "On the Evolution of Machine Learning: from Linear Models to Neural Networks". [en línea], 2006, pp. 2. [Consulta: 8 junio 2019]. Disponible en: https://stanford.edu/~rezab/papers/neuralinterview.pdf.

BRUNEAU, G. "The History and Evolution of Intrusion Detection". *SANS Institute Information Security Reading Room* [en línea], 2019. pp. 3. [Conslta: 31 julio 2019]. Disponible en: https://www.sans.org/reading-room/whitepapers/detection/history-evolution-intrusion-detection-344.

CAÑEDO, R. "Aproximaciones para una historia de Internet". *Acimed* [en linea], 2004, (Cuba) vol. 12, no. 1, pp. 2-17. [Consulta: 15 junio 2019]. ISSN 10249435. Disponible en: http://scielo.sld.cu/pdf/aci/v12n1/aci05104.pdf

CARAZO, G. Análisis del tráfico de red para la deteccion de ataques informáticos [En linea] (Trabajo de titulacion). Universidad de Jaen, Jaen, España. 2017. pp. 5-6. [Consulta realizada: 07 Junio 2019]. Disponible en: http://tauja.ujaen.es/bitstream/10953.1/6432/1/Documentacion.pdf.

CARRANZA, M. y NARANJO, R. Modelo basado en las técnicas de minería de datos aplicada a la detección de ataques en las redes de datos de la Facultad de Informática y Electrónica [en línea] (Trabajo de titulación). ESPOCH, FIE, Sistemas, Ecuador. 2014. pp. 177. [Consulta:12 junio 2019] Disponible en: http://dspace.espoch.edu.ec/handle/123456789/3545.

CHABLÉ, Hilda. Herramientas de monitoreo y detección de intrusos en servidores Linux. [en línea] (Trabajo de titulacion). Centro de Investigacion y Estudios Avanzados del Instituto Politécnico Nacional, Departamento de computación. Distrito Federal, Mexico, 2007. pp. 21-23. [Consulta: 1 agosto 2019]. Disponible en: https://www.cs.cinvestav.mx/TesisGraduados/2007/tesisHildaChable.pdf.

CHANALUISA, Darwin; et al. Implementación del sistema de gestión y administración de seguridad para la Dirección de Tecnologías de la Universidad Central del Ecuador (DTIC) [en línea] (Trabajo de titulación). UCE, Ciencias fisicas y matematicas, Informatica, Ecuador. 2012. pp. 127-128. [Consulta: 15 agosto 2019]. Disponible en: http://www.dspace.uce.edu.ec/bitstream/25000/365/1/T-UCE-0011-19.pdf.

CLOUDFLARE. *Que es un ataque DDoS?*. [blog]. [Consulta: 10 noviembre 2019]. Disponible en: https://www.cloudflare.com/es-la/learning/ddos/what-is-a-ddos-attack/

CHIO, C. y FREEMAN, D. *Machine Learning & Security Protecting systems with data and algorithms* [en línea]. United States of America: O´Reilly, 2018. [Consulta: 12 agosto 2019]. Disponible en: https://www.amazon.com/Machine-Learning-Security-Protecting-Algorithms/dp/1491979909.

COUNTERFLOW, AI. *Dragonfly-mle* [blog]. 2018. [Consulta: 15 enero 2020]. Disponible en: https://github.com/counterflow-ai/dragonfly-mle

DECISO BV. *Inline Intrusion Prevention System* [blog]. [Consulta: 30 diciembre 2019]. Disponible en: https://docs.opnsense.org/manual/ips.html

DE LA HOZ, E.; et al. "Modelo de detección de intrusiones en sistemas de red, realizando selección de características con FDR y entrenamiento y clasificación con SOM". *Inge-Cuc* [en linea], 2012, vol. 8, no. 1, pp. 91. [Consulta: 10 diciembre 2019]. Disponible en: https://dialnet.unirioja.net/descarga/articulo/4869005.pdf.

DE LA TORRE, Diego. *Los ciberataques más famosos de la historia* [blog]. [Consulta: 18 julio 2019]. Disponible en: https://blogthinkbig.com/ciberataques-famosos-historia

DELOITTE. "Ataque masivo de Ransomware" [en línea]. 2017, pp. 1-49. [Consulta: 20 julio 2019]. Disponible en: https://www2.deloitte.com/content/dam/Deloitte/ec/Documents/deloitte-analytics/Estudios/Informe-detallado-Ataque masivo-Ransomware-WannaCry-finales.pdf.

FARNAAZ, N. y JABBAR, M.A., Random Forest Modeling for Network Intrusion Detection System. Procedia Computer Science [en línea], 2016, vol. 89, pp. 213-217. ISSN 18770509. DOI 10.1016/j.procs.2016.06.047. [Consulta: 19 de marzo 2021] Disponible en: http://dx.doi.org/10.1016/j.procs.2016.06.047.

EQUIPO DE EXPERTOS UIV. *Principios fundamentales de la seguridad de redes* [blog]. [Consulta: 09 julio 2019]. Disponible en: https://www.universidadviu.com/principios-fundamentales-de-la-seguridad-en-redes/

EPICALSOFT INSTANCE. *Tipos de problemas en Machine Learning* [blog]. [Consulta: 23 diciembre 2019]. Disponible en: http://epicalsoft.blogspot.com/2018/11/azure-machine-learning-algoritmos-de.html

FreeBSD. *Instalando FreeBSd* [blog]. [Consulta: 28 diciembre 2019]. Disponible en: https://www.freebsd.org/doc/handbook/bsdinstall-start.html

GANDHI, Rohith. *Support Vector Machine – Introduction to Machine Learning Algorithms* [blog]. 2018. [Consulta: 15 enero 2020]. Disponible en: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

GOPALKRISHNA N. PRABHU; et al. "Network Intrusion Detection System". *International Journal of Engineering Research and Applications* [en línea], 2014. vol. 4, no. 4, pp. 69-72. [Consulta: 5 agosto 2019]. ISSN 2248-9622. Disponible en: https://www.academia.edu/7548146/Network_Intrusion_Detection_System.

GONZÁLES, Ligdi. Aprendizaje Supervisado: Logistic Regression [blog]. Venezuela, 2018a. [Consulta: 25 diciembre 2019]. Disponible en: http://ligdigonzalez.com/aprendizaje-supervisado-logistic-regression/

GONZÁLES, Ligdi. Regresion Lineal Simple - Teoría [blog]. Venezuela, 2018b. [Consulta: 30 diciembre 2019]. Disponible en: http://ligdigonzalez.com/algoritmo-regresion-lineal-simple-machine-learning/

GONZÁLES, Ligdi. Aprendizaje supervisado: Linear Regression [blog]. Venezuela, 2018c [Consulta: 5 enero 2019]. Disponible en: http://ligdigonzalez.com/aprendizaje-supervisado-linear-regression/

GONZÁLES, Ligdi. Aprendizaje supervisado: Polynomial Regression [blog]. Venezuela, 2018d [Consulta: 8 enero 2019e]. Disponible en: http://ligdigonzalez.com/aprendizaje-supervisado-polynomial-regression/

GONZÁLES, Ligdi. Aprendizaje supervisado: Support Vector Regression [blog]. Venezuela, 2018e [Consulta:12 enero 2019]. Disponible en: http://ligdigonzalez.com/aprendizaje-supervisado-support-vector-regression/

HALL, M. y SMITH, L. "Practical Feature Subset Selection for Machine Learning". [en linea], 1998. (New Zealand), pp. 1. [Consulta:]. Disponible en: https://researchcommons.waikato.ac.nz/bitstream/handle/10289/1512/Practical%20feature%20subset%20selection%20for%20machine%20learning.pdf?sequence=1&isAllowed=y

HARRISON, Onel. *Machine Learning Basics with the K-Nearest Neighboors Algorithm* [blog]. 2018. [Consulta: 15 enero 2020]. Disponible en: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

HENDRICKX, I. *Local Classification and Global Estimation Explorations of the k-nearest neighbor algorithm* [en línea]. 2005. [Consulta: 15 agosto 2019]. Disponible en: https://www.researchgate.net/publication/237124275_Local_Classification_and_Global_Estimation_Exploration_of_the_k-nearest_neighbor_algorithm.

IBM, X.-F. "IBM X-Force Threat Intelligence Index 2018 Notable security events of 2017, and a look ahead". *IBM Security* [en línea], 2018. no. March, pp. 7. [Consulta: 15 julio 2019]. Disponible en: https://www.ibm.com/support/knowledgecenter/SSB2MG_4.6.0/com.ibm.ips.doc/concepts/wap_buffer_overflow.htm.

IBM, Knowledge center. *Web application protection categories, Buffer overflow attacks*. [blog], 2018a. [Consulta: 20 julio 2019]. Disponible en: https://securityintelligence.com/2018-ibm-x-force-report-shellshock-fades-gozi-rises-and-insider-threats-soar/.

IBM, Knowledge center. *Web application protection categories, Buffer overflow attacks*. [blog], 2018b. [Consulta: 20 julio 2019]. Disponible en: https://www.ibm.com/support/knowledgecenter/SSB2MG_4.6.0/com.ibm.ips.doc/concepts/wap_injection_attacks.htm.

INSTITUTO NACIONAL DE CIBERSEGURIDAD. "Diseño y Configuración de IPS, IDS y SIEM en Sistemas de Control Industrial ". *Certsi* [en línea], 2017. vol. V1, pp. 7-31. [Consulta: 28 julio 2019]. Disponible en: https://www.incibe-cert.es/sites/default/files/contenidos/guias/doc/certsi_diseno_configuracion_ips_ids_siem_en_sci.pdf.

JACKSON, T.R. , et al. *An Investigation of a Compromised Host on a Honeynet Being Used to Increase the Security of a Large Enterprise Network* [en línea]. 2004.[Consulta:28 julio 2019]. Disponible en: https://ieeexplore.ieee.org/document/1437791.

JIMÉNEZ, Carlos. Diseño y Optimización de un Sistema de Detección de Intrusos Híbrido [en línea] (Trabajo de titulacion). Universidad de Almeria, España, Almeria, 2009, pp. 17. [Consulta: 26 julio 2019]. Disponible en: http://www.adminso.es/images/8/88/PFC_carlos.pdf.

JETT, Justin. *IMAP Attacks* [blog]. [Consulta: 20 julio 2019]. Disponible en: https://www.informationsecuritybuzz.com/expert-comments/imap-attacks/

JUNIPER, Networks. *IP Address Sweep and Port Scan* [blog]. 2020a. [Consulta: 21 julio 2019]. Disponible en: https://www.juniper.net/documentation/en_US/junos/topics/topic-map/security-ip-sweep-and-port-option.html

JUNIPER, Networks. *IP Address Sweep and Port Scan* [blog]. 2020b. [Consulta: 21 julio 2019]. Disponible en: https://www.juniper.net/documentation/en_US/junos/topics/topic-map/security-ip-sweep-and-port-option.html

JUNIPER, Networks. *IP Address Sweep and Port Scan* [blog]. 2020c. [Consulta: 21 julio 2019]. Disponible en: https://www.juniper.net/documentation/en_US/junos/topics/topic-map/security-ip-sweep-and-port-option.html

LAN, K.; et al. "Effect of Malicious Traffic on the Network". [en línea], 2009, pp. 1. [Consulta: 10 de junio de 2019]. Disponible en: https://www.researchgate.net/publication/228611227_The_Effect_of_Malicious_Traffic_on_the_Network.

LIPPINCOTT, Elisa. Trend Micro TippingPoint Named a Leader in 2017 Gartner Magic Quadrant for Intrusion Detection and Prevention Systems (IDPS) [blog]. 2017. [Consulta: 19 enero 2020]. Disponible en: https://blog.trendmicro.com/trend-micro-tippingpoint-named-a-leader-in-2017-gartner-magic-quadrant-for-intrusion-detection-and-prevention-systems-idps/

LÓPEZ, Raúl. Libro online de IAAR [blog]. [Consulta: 10 diciembre 2019]. Disponible en: https://iaarbook.github.io/machine-learning/

MAMAMI, D. "Fases de un Ataque Hacker" [en línea], 2013, pp. 1-2. [Consulta: 20 junio 2019]. Disponible en: http://www.revistasbolivianas.org.bo/pdf/rits/n8/n8a29.pdf.

MANZO, Héctor. *Cronología de los virus informáticos* [blog]. [Consulta: 18 julio 2019]. Disponible en: https://www.timetoast.com/timelines/cronologia-de-los-virus-informaticos

MARR, Bob. (2016). "*A Short History of Machine Learning - Every Manager Should Read*" [blog]. [Consulta: 7 junio 2019]. Disponible en: https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learningevery-manager-should-read/2c80954a15e7

MARTINEZ, Jose. *Error cuadratico medio para regresion* [blog]. [Consulta: 10 enero 2020]. Disponible en: https://iartificial.net/error-cuadratico-medio-para-regresion/

MCHUGH, J.; et al. *Defending Yourself: The Role of Intrusion Detection Systems* [en línea]. S.l.: s.n., 2000. [Consulta: 30 julio 2019]. Disponible en: https://ieeexplore.ieee.org/document/6156710.

MIERES, J. "Ataques informáticos Debilidades de seguridad comúnmente explotadas". Evil fingers [en línea], 2009, pp. 3. [Consulta: 8 junio 2019]. Disponible en: https://www.evilfingers.net/publications/white_AR/01_Ataques_informaticos.pdf.

MIRA, Emilio., 2001. Implantación de un Sistema de Detección de Intrusos en la Universidad de Valencia [en línea] (Trabajo de titulación). Universidad de Valencia, Informatica, España. 2001. pp. 13-14. [Consulta: 14 junio 2019]. Disponible en: https://www.rediris.es/cert/doc/pdf/ids-uv.pdf.

MITCHELL, T. "Really Work ?". *AI Magazine* [en línea], 1997. vol. 18, no. 3, pp. 11-12. [Consulta: 9 agosto 2019] Disponible en: https://pdfs.semanticscholar.org/451f/86130daf5632b87cd4c0f417b245c6ebb582.pdf.

NMAP. *Introduction* [blog]. [Consulta: 20 julio 2019]. Disponible en: https://nmap.org/

NORTON, LifeLock. *What is a computer worm, and how does it work?* [blog]. [Consulta: 20 julio 2019]. Disponible en: https://us.norton.com/internetsecurity-malware-what-is-a-computer-worm.html

O´CONNOR, Fred. *What you need to know about powershell attacks* [blog]. 2017. [Consulta: 10 junio 2020]. Disponible en: https://www.cybereason.com/blog/fileless-malware-powershell

OISF. *Suricata Rules* [blog]. 2016a. [Consulta: 10 enero 2020]. Disponible en: https://suricata.readthedocs.io/en/suricata-4.1.3/rules/intro.html

OISF. *Action Order* [blog]. 2016b. [Consulta: 10 enero 2020]. Disponible en: https://suricata.readthedocs.io/en/suricata-4.1.3/configuration/suricata-yaml.html#suricata-yaml-action-order

OISF. *Runmodes* [blog]. 2016c. [Consulta: 10 enero 2020]. Disponible en: https://suricata.readthedocs.io/en/suricata-4.1.3/performance/runmodes.html

ORELLANA, Johanna. *Arboles de decisión y Random forest* [blog]. 2018. [Consulta: 18 enero 2020]. Disponible en: https://bookdown.org/content/2031/ensambladores-random-forest-parte-i.html

OPNids. Documentation of OPNids [blog]. 2018. [Consulta: 5 agosto 2019]. Disponible en: https://docs.opnids.io/intro.html

PADILLA R., et al. *Estado De Las Tecnologías De Información Y Comunicación (Tic) En El Sistema Universitario Ecuatoriano – Uetic 2018* [en línea]. 2019. S.l.: s.n. ISBN 9789942852755. [Consulta: 20 enero 2021]. Disponible en: www.creativecommons.org/licences/by-nc/4.0EstedocumentosepuededescargarenformatoPDFdesdehttps://www.cedia.edu.ec/es/publicaciones/libros.

PARDO, Andrea. *Tipos de ataques* [blog]. [Consulta: 22 julio 2019]. Disponible en: https://blogseguridadandrea.wordpress.com/2016/11/13/4-1-tipos-de-ataques/

PALO ALTO NETWORKS. *What is a denial of service attack (DoS)?* [blog]. [Consulta: 05 agosto 2019]. Disponible en: https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos

PELUFFO, I.; et al. "Machine Learning aplicado en Sistemas de Detección de Intrusos". *Workshop de Investigadores en Ciencias de la Computación* [en línea], (Argentina), 2014, pp. 3-4.[Consulta: 10 junio 2019]. Disponible en: http://sedici.unlp.edu.ar/bitstream/handle/10915/43264/Documento_completo.pdf?sequence=1&isAllowed=y.

POLYAKOV, Alexander. *Machine Learning for Cibersecurity 101* [blog]. [Consulta: 28 noviembre 2019]. Disponible en: https://towardsdatascience.com/machine-learning-for-cybersecurity-101-7822b802790b

RODRÍGUEZ, Daniel. La regresión logística [blog]. [Consulta: 25 diciembre 2019]. Disponible en: https://www.analyticslane.com/2018/07/23/la-regresion-logistica/

ROESCH, M. Snort – Lightweight Intrusion Detection for Networks. *Proceedings of LISA '99: 13th Systems Administration Conference* [en línea], 1999, USA, pp. 229-238. [Consulta: 18 enero 2020]. Disponible en: https://www.usenix.org/legacy/publications/library/proceedings/lisa99/full_papers/roesch/roesch.pdf

RUIZ, Sergio. *El algoritmo K-NN y su importancia en el modelado de datos* [blog]. 2017.[Consulta: 10 enero 2020]. Disponible en: https://www.analiticaweb.es/algoritmo-knn-modelado-datos/

REYES, R.; et al. *Estado De Las Tecnologías De Información Y Comunicación (Tic) En El Sistema Universitario Ecuatoriano – Uetic 2019* [en línea]. 2019. S.l.: s.n. ISBN 9789942852755. [Consulta: 20 enero 2021] Disponible en: https://www.cedia.edu.ec/dmdocuments/publicaciones/Libros/UETIC_2019.pdf.

RUSSELL, S. y NORVIG, P. *Artificial Intelligence - A modern approach* [en línea]. New Jersey-USA: Prentice Hall, 2009. [Consulta:8 junio 2019]. Disponible en: https://www.ics.uci.edu/~rickl/courses/cs-171/aima-resources/Artificial Intelligence A Modern Approach (3rd Edition).pdf.

SABHNANI, M. y SERPEN, G. "KDD feature set complaint heuristic rules for R2L attack detection". *Proceedings of the International Conference on Security and Management* [en línea], 2003, (United State of America), vol. 1, pp. 310-316. [Consulta: 9 junio 2019]. Disponible en: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.2289&rep=rep1&type=pdf.

SAMUEL, A. "Some Studies in Machine Learning Using the Game of Checkers". *IBM Journal of Research and Development* [en línea], 1959, (United States of America) vol. 3, no. 3, pp. 211-212. [Consulta: 6 junio 2019]. ISSN 0018-8646. Disponible en: https://doi.org/10.1147/rd.33.0210

SARLE, W. "Neural Networks and Statistical Models". [en línea], 1994, pp. 1. [Consulta: 7 junio 2019]. Disponible en: https://pdfs.semanticscholar.org/b138/2af52d580fb3db34118434fc731453d03fc6.pdf.

SCHOOTT, Madison. *Random Forest Algorithm for Machine Learning* [blog]. 2019. [Consulta: 18 enero 2020]. Disponible en: https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb

SHAFIQ M.; et al. "Network Traffic Classification Techniques and Comparative Analysis Using Machine Learning Algorithms". [en linea]. 2016, pp. 2451. [Consulta: 11 junio 2019]. Disponible en: https://www.researchgate.net/publication/316900016_Network_Traffic_Classification_techniques_and_comparative_analysis_using_Machine_Learning_algorithms

STALLINGS, W. *Fundamentos de seguridad en redes : aplicaciones y estándares* [en linea]. España-Madrid: Pearson Prentice Hall, 2004. [Consulta: 17 junio 2019]. Disponible en: https://www.academia.edu/22539038/Fundamentos_de_seguridad_en_redes_Aplicaciones_y_estándares_2da_Edición

TARAZONA, C. "Amenazas informáticas y seguridad de la Información" [en línea], 2007, (Colombia), pp. 137-138 [Consulta: 3 julio 2019]. Disponible en: https://revistas.uexternado.edu.co/index.php/derpen/article/download/965/915/.

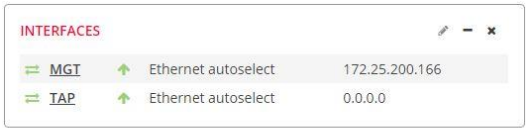
THURAISINGHAM, B.; et al. "Data mining for security applications". *Proceedings of The 5th International Conference on Embedded and Ubiquitous Computing, EUC 2008* [en línea]. 2008. vol. 2, no. December, pp. 585. [Consulta: 8 agosto 2019] Disponible en: https://www.researchgate.net/publication/221452043_Data_Mining_for_Security_Applications/link/0c96052a93b46c6edc000000/download.

VALLEJOS, O. *Introducción a internet* [en línea]. S.l.: s.n., 2003. [Consulta: 15 junio 2019]. Disponible en: http://ing.unne.edu.ar/pub/internet.pdf.

YÚBAL, Francisco. *La historia de Creeper, el primer virus jamás programado* [blog]. [Consulta: 18 julio 2019]. Disponible en: https://www.xataka.com/historia-tecnologica/la-historia-de creeper-el-primer-virus-informatico-jamas-programado

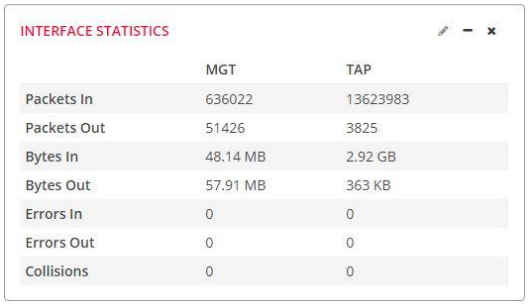
ANEXOS

ANEXO A: Estadísticas de paquetes in/out de la Plataforma OPNids en la Vlan estudiantes



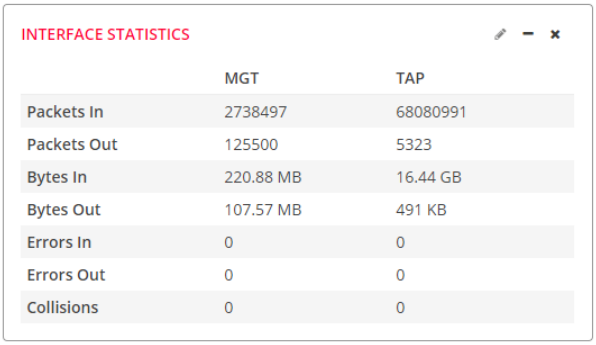
INTERFACES

MGT	Ethernet autoselect	172.25.200.166
TAP	Ethernet autoselect	0.0.0.0



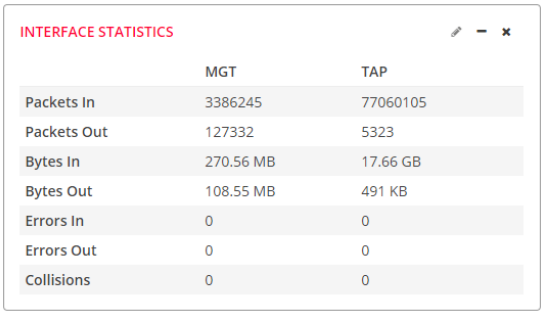
INTERFACE STATISTICS

	MGT	TAP
Packets In	636022	13623983
Packets Out	51426	3825
Bytes In	48.14 MB	2.92 GB
Bytes Out	57.91 MB	363 KB
Errors In	0	0
Errors Out	0	0
Collisions	0	0



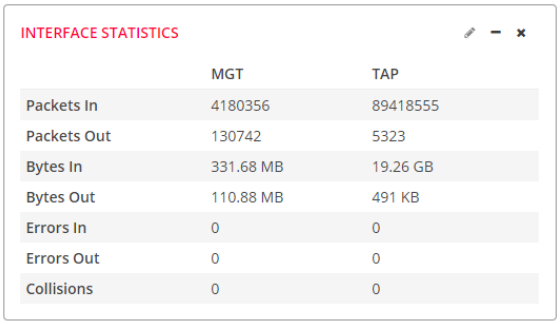
INTERFACE STATISTICS

	MGT	TAP
Packets In	2738497	68080991
Packets Out	125500	5323
Bytes In	220.88 MB	16.44 GB
Bytes Out	107.57 MB	491 KB
Errors In	0	0
Errors Out	0	0
Collisions	0	0



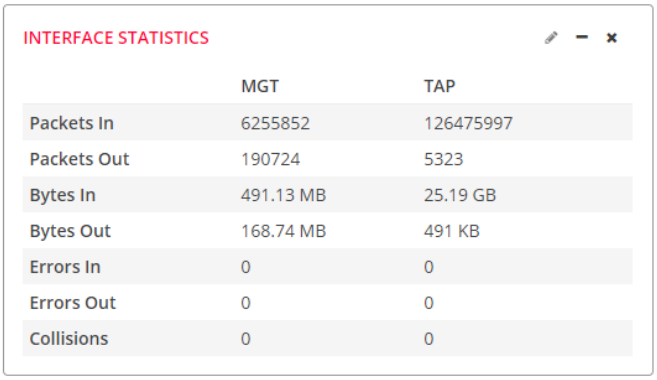
INTERFACE STATISTICS

	MGT	TAP
Packets In	3386245	77060105
Packets Out	127332	5323
Bytes In	270.56 MB	17.66 GB
Bytes Out	108.55 MB	491 KB
Errors In	0	0
Errors Out	0	0
Collisions	0	0



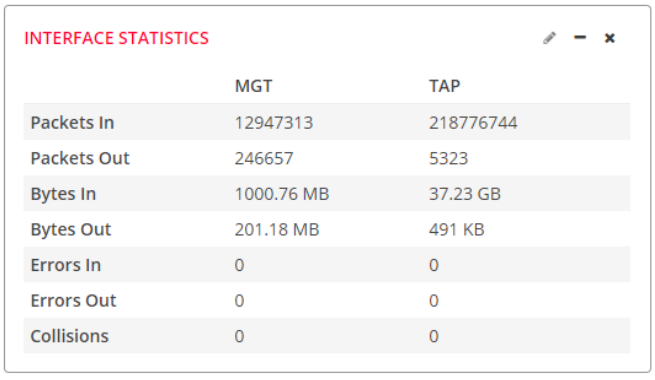
INTERFACE STATISTICS

	MGT	TAP
Packets In	4180356	89418555
Packets Out	130742	5323
Bytes In	331.68 MB	19.26 GB
Bytes Out	110.88 MB	491 KB
Errors In	0	0
Errors Out	0	0
Collisions	0	0



INTERFACE STATISTICS

	MGT	TAP
Packets In	6255852	126475997
Packets Out	190724	5323
Bytes In	491.13 MB	25.19 GB
Bytes Out	168.74 MB	491 KB
Errors In	0	0
Errors Out	0	0
Collisions	0	0



INTERFACE STATISTICS

	MGT	TAP
Packets In	12947313	218776744
Packets Out	246657	5323
Bytes In	1000.76 MB	37.23 GB
Bytes Out	201.18 MB	491 KB
Errors In	0	0
Errors Out	0	0
Collisions	0	0

ANEXO B: Registro de alertas en los dos periodos de la Vlan estudiantes

Timestamp	Alerta	SID	Protocolo	IP origen	IP destino	Puerto origen	Puerto destino	Observaciones
2019-12-12T17:59:29	SURICATA STREAM ESTABLISHED packet out of window	2210020	TCP	172.25.200.4	172.25.200.132	80	61503	
2019-12-12T17:59:29	SURICATA STREAM SHUTDOWN RST invalid ack	2210046	TCP	172.25.200.132	172.25.200.4	61503	80	
2019-12-12T17:59:29	SURICATA STREAM Packet with invalid ack	2210045	TCP	172.25.200.132	172.25.200.4	61503	80	
2019-12-12T17:59:29	SURICATA STREAM FIN invalid ack	2210030	TCP	172.25.200.132	172.25.200.4	61503	80	
Viernes 13-12-2019								
2019-12-13T14:11:45	SURICATA zero length padN option	2200094	IPV6-ICMP	fe80:0000:0000:0000:78db:7c13:d59c:fe12	ff02:0000:0000:0000:0000:0000:0016			Se repitio la misma alerta desde las 6pm/12-12-2019 hasta las 2:12pm/13-12-2019
2019-12-13T14:13:40	SURICATA STREAM CLOSEWAIT invalid ACK	2210017	TCP	172.25.203.223	172.25.200.166	55741	443	
2019-12-13T14:13:40	SURICATA STREAM Packet with invalid ack	2210045	TCP	172.25.203.223	172.25.200.166	55741	443	
2019-12-13T14:13:40	SURICATA STREAM CLOSEWAIT ACK out of window	2210015	TCP	172.25.200.166	172.25.203.223	443	55741	
2019-12-13T14:13:40	SURICATA STREAM SHUTDOWN RST invalid ack	2210046	TCP	172.25.203.223	172.25.200.166	55741	443	
2019-12-13T16:31:20	SURICATA STREAM ESTABLISHED invalid ack	2210029	TCP	172.25.201.82	192.188.46.165	42616	443	
2019-12-13T16:31:20	SURICATA STREAM Packet with invalid ack	2210045	TCP	172.25.201.82	192.188.46.165	42616	443	
2019-12-13T16:31:20	SURICATA STREAM ESTABLISHED packet out of window	2210020	TCP	192.188.46.165	172.25.201.82	443	42616	
sabado 14 domingo 15-2019								
2019-12-14T16:31:25	SURICATA STREAM ESTABLISHED invalid ack	2210029	TCP	2800:0068:000a:5200:1e98:ecff:fe22:27cb	2800:0068:0000:bebe:0000:0000:0003	50952	443	Alertas generadas por miles
2019-12-14T16:31:25	SURICATA STREAM Packet with inva	2210045	TCP	2800:0068:000a:5200:1e98:ecff:fe22:27cb	2800:0068:0000:bebe:0000:0000:0003	50952	443	
2019-12-14T16:31:25	SURICATA STREAM FIN out of window	2210038	TCP	2800:0068:0000:bebe:0000:0000:0003	2800:0068:000a:5200:1e98:ecff:fe22:27cb	443	50952	
2019-12-14T16:42:20	ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management	2013504	TCP	172.25.203.161	23.219.145.146	50816	80	
2019-12-15T06:15:20	ET POLICY GNU/Linux APT User-Agent Outbound likely related to	2013504	TCP	172.25.203.161	23.219.145.146	50828	80	
2019-12-15T16:31:20	ET POLICY GNU/Linux YUM User-Agent Outbound likely related to package management	2013505	TCP	172.25.201.82	66.109.26.212	38506	80	
2019-12-15T16:31:21	ET POLICY GNU/Linux YUM User-Agent Outbound likely related to package management	2013505	TCP	172.25.202.169	66.187.230.28	45952	80	
2019-12-15T16:31:21	ET POLICY GNU/Linux YUM User-Agent Outbound likely related to package management	2013505	TCP	172.25.201.176	66.187.230.28	37164	80	
2019-12-15T16:31:24	SURICATA STREAM ESTABLISHED packet out of window	2210020	TCP	2800:0068:0000:bebe:0000:0000:0003	2800:0068:000a:5200:1e98:ecff:fe22:27cb	443	33610	
2019-12-15T16:31:24.	SURICATA STREAM ESTABLISHED invalid ack	2210029	TCP	2800:0068:000a:5200:1e98:ecff:fe22:27cb	2800:0068:0000:bebe:0000:0000:0003	33610	443	
Lunes 16-12-2019								
2019-12-16T04:15:30	ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management	2013504	TCP	172.25.203.161	23.219.145.146	50842	80	
2019-12-16T11:57:39	SURICATA STREAM ESTABLISHED invalid ack	2210029	TCP	172.25.200.4	172.25.203.223	80	61160	
2019-12-16T11:57:39	SURICATA STREAM Packet with invalid ack	2210045	TCP	172.25.200.4	172.25.203.223	80	61160	
2019-12-16T11:57:39	SURICATA STREAM Packet with invalid ack	2210045	TCP	172.25.203.223	172.25.200.4	61160	80	
2019-12-16T11:57:39	SURICATA STREAM SHUTDOWN RST invalid ack	2210046	TCP	172.25.203.223	172.25.200.4	61160	80	
2019-12-16T11:58:22	SURICATA ICMPv4 unknown code	2200025	ICMP	172.25.202.180	172.25.200.4			
2019-12-16T13:56:51	SURICATA STREAM excessive retransmissions	2210054	TCP	172.25.200.166	172.25.201.134	443	50631	
2019-12-16T14:22:08	SURICATA HTTP unable to match response to request	2221010	TCP	172.25.200.4	172.25.203.223	80	50345	
Martes 17-12-2019								
2019-12-17T08:10:22	ET SCAN Behavioral Unusual Port 445 traffic Potential Scan or Infection	2001569	TCP	172.25.203.151	172.25.201.238	64731	445	
2019-12-18T16:57:13	ET POLICY Possible Kali Linux hostname in DHCP Request Packet	2022973	UDP	172.25.201.13	172.17.102.223	68	67	
2019-12-18T17:53:21	ET DROP Dshield Block Listed Source group 1	2402000	TCP	185.156.73.42	172.25.201.62	53986	61840	
2019-12-18T17:31:47.	ET INFO DNS Query for Suspicious .icu Domain	2026888	UDP	172.25.201.62	172.17.102.39	58166	53	
2019-12-18T17:53:21	ET DROP Dshield Block Listed Source group 1		TCP	185.156.73.42	172.25.201.62	53986	61840	
Del jueves 19 de diciembre de 2019 al Domingo 05 de Enero - receso por festividades								
2020-01-08T08:13:26	ET DOS Possible SSDP Amplification Scan in Progress	2019102	UDP	172.25.200.80	172.25.201.100	57397	1900	
2020-01-09T16:28:58	ET CINS Active Threat Intelligence Poor Reputation IP group 39	2403338	UDP	50.200.136.108	172.25.201.252	57599	6881	
2020-01-09T16:32:52	ET INFO WinHttp AutoProxy Request wpad.dat Possible BadTunnel	2022913	TCP	172.25.201.252	83.248.108.95	57945	80	
2020-01-09T16:44:37	ET POLICY External IP Check myexternalip.com	2019980	TCP	172.25.201.252	216.239.34.21	63589	80	
2020-01-09T16:46:17	ET POLICY External IP Lookup - checkip dyndns.org	2021378	TCP	172.25.201.252	131.186.113.70	51135	80	
2020-01-09T16:48:18	ET POLICY DynDNS Checkip External IP Address Server Response	2014932	TCP	131.186.113.70	172.25.201.252	80	51135	
2020-01-09T17:08:00	ET DROP Spamhaus DROP Listed Traffic Inbound group 24	2400023	UDP	196.196.192.38	172.25.201.252	59207	6881	
2020-01-10T09:03:07	ET DOS Possible SSDP Amplification Scan in Progress	2019102	UDP	172.25.202.117	172.25.200.1	58119	1900	
2020-01-10T11:04:01	ET DOS Possible SSDP Amplificatior	2019102	UDP	172.25.202.117	172.25.200.56	62051	1900	
2020-01-10T13:02:08	ET DROP Spamhaus DROP Listed Traffic Inbound group 24	2400023	UDP	196.194.211.122	172.25.203.252	64849	24941	

ANEXO C: Estadísticas de paquetes in/out de la Plataforma OPNids en la Vlan docentes

INTERFACE STATISTICS

	MGT	TAP
Packets In	283515	7393336
Packets Out	26812	2936
Bytes In	23.20 MB	1.90 GB
Bytes Out	28.43 MB	276 KB
Errors In	0	0
Errors Out	0	0
Collisions	0	0

INTERFACE STATISTICS

	MGT	TAP
Packets In	1157210	23200341
Packets Out	54532	4761
Bytes In	87.81 MB	4.30 GB
Bytes Out	61.31 MB	454 KB
Errors In	0	0
Errors Out	0	0
Collisions	0	0

ANEXO D: Registro de alertas de un mes de la Vlan docentes.

2020-02-07T10:38:57	ET SCAN NMAP -sA (1)	2000538	TCP	172.217.0.165	172.25.201.134	443	55696	
2020-02-07T12:24:05.548387-0500	ET SCAN NMAP -sA (2)	2000540	TCP	172.217.8.138	172.25.201.134	443	59343	
2020-02-08T15:32:41	ET POLICY SSH session in progress on Expected Port	2001978	TCP	172.25.202.91	172.25.201.82	34480	22	35 ALERTAS
2020-02-08T18:09:27	ET SHELLCODE Rothenburg Shellcode	2009247	TCP	172.25.201.176	172.25.202.89	989	2049	
2020-02-08T18:22:36	ET POLICY Credit Card Number Detected in Clear (16 digit spaced)	2001375	TCP	172.25.202.89	172.25.201.176	2049	989	3 alertas
2020-02-08T19:18:59	ET NETBIOS DCERPC DCOM ExecuteShellCommand Call - Likely Lateral Movement	2027189	TCP	172.25.201.176	172.25.202.89	989	2049	2 <lertas
2020-02-08T21:36:36	SURICATA Applayer Detect protocol only one direction	2260002	TCP	172.25.200.4	172.25.201.199/172.25.201.214/172.25.201.51/172.25.200.3/172.25.201.184/172.25.201.192/172.25.201.76/172.25.200.243	51764/55893	554	13 alertas
2020-02-11T16:58:47	SURICATA STREAM ESTABLISHED SYNACK resend	2210022	TCP	93.190.142.167/85.17.84.77	172.25.202.71	80	54455	4 alertas
2020-02-11T18:15:28	ET EXPLOIT Malformed HeartBeat Request	2018372	TCP	172.25.201.176	172.25.202.89	989	2049	2 alertas

ANEXO E: Script adicional a la plataforma OPNids para visualización de parámetros del algoritmo Random forest.

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

#Descargamos el dataset kdd99

url= 'http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data_10_percent.gz'

df= pd.read_csv(url, header=None)

df.head()

#Cargamos el log eve.json desde el path, este proceso de lo realiza en forma diferenciada

with open('eve.json') as file:

data = json.load(file)

#Etiquetamos las columnas

df.columns= [ 'duration','protocol_type', 'service', 'flag',
'src_bytes','dst_bytes','land','wrong_fragment','urgent','hot','num_failed_logins','logged_in',

'num_compromised', 'root_shell', 'su_attempted', 'num_root', 'num_file_creations', 'num_shells',
'num_access_files', 'num_outbound_cmds', 'is_host_login',
'is_guest_login','cnt','srv_count','serror_rate','srv_serror_rate','rerror_rate','srv_rerror_rate','same
_srv_rate','diff_srv_rate','srv_diff_host_rate','dst_host_count','dst_host_srv_count','dst_host_sam
e_srv_rat','dst_host_diff_srv_rate','dst_host_same_src_port_rate','dst_host_srv_diff_host_rate','d
st_host_serror_rate','dst_host_srv_serror_rate','dst_host_rerror_r
te','dst_host_srv_rerror_rate','outcome']

print(df.describe())

df['outcome'].unique()

#Categorizando variables

df=df.replace(to_replace =["ipsweep.","portsweep.","nmap.","satan."],  value ="probe")

df=df.replace(to_replace =["ftp_write.", "guess_passwd.","imap.","multihop.","phf.","spy.",
"warezclient.","warezmaster."],  value ="r2l")
```

```
df=df.replace(to_replace =["buffer_overflow.","loadmodule.","perl.", "rootkit."],  value ="u2r")

df=df.replace(to_replace =["back.", "land.","neptune.", "pod.","smurf.","teardrop."],value
="dos")

#Contamos los datos de cada categoría o código asignado

from collections import Counter

Counter(df['outcome'])

#Preprosesamos los datos de entrada

from sklearn import preprocessing

label_encoder = preprocessing.LabelEncoder()

df['protocol_type']= label_encoder.fit_transform(df['protocol_type'])

df['service']= label_encoder.fit_transform(df['service'])

df['flag']= label_encoder.fit_transform(df['flag'])

df['flag']= label_encoder.fit_transform(df['flag'])

df['outcome']= label_encoder.fit_transform(df['outcome'])

#Limpeza de datos correlacionados

corr = df.corr()

plt.figure(figsize=(15,12))

sns.heatmap(corr)

plt.show()

print('{:>30} {:>30} {:>30}'.format(*["Feature 1","Feature 2","Correlation"]))

x=[]

for i in corr:

    for j in corr:

        if((corr[i][j]>0.97) and i!=j and (i not in x)):

            l=len(i)+len(j)

            print('{:>30} {:>30} {:>30}'.format(*[i,j,corr[i][j]]))
```

```
            x.append(i)
for i in x:
    if(i in df.columns):
        df.drop(i,axis=1,inplace=True)
#Elimiamos datos correlacionados
df.drop('is_host_login',axis = 1, inplace=True)
df.drop('num_outbound_cmds',axis = 1, inplace=True)
#Entrenamos el modelo con una semilla del 20%
X= df.drop(['outcome'], axis=1)
Y=df['outcome']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test= train_test_split(X, Y, test_size=0.2, random_state=8)
#Ejecutamos el algoritmo de clasificación Random Forest
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification
rf = RandomForestClassifier(random_state=8,n_estimators=8)
rf.fit(X_train, y_train)
#Imprimimos los parámetros de evaluación
y_preds=rf.predict(X_test)
print(confusion_matrix(y_test, y_preds))
print("Accuracy:",metrics.accuracy_score(y_test,y_preds))
print(metrics.classification_report(y_test,y_preds))
```

ANEXO F: Encuesta realizada para la comparación de características IDS/IPS Open Source

Comparación de características de IDS/IPS Open Source

La presente encuesta busca obtener los resultados para realizar una comparativa de los mejores IDS/IPS Open Source en el mercado.
Las opciones para escoger van desde: 1. No es importante 2. Poco importante 3. Ni importante ni poco importante 4. Importante 5. Muy importante

*Obligatorio

Considera usted que el soporte brindado por las empresas desarrolladoras es importante para la evolución de las herramientas IDS/IPS *

	1	2	3	4	5	
No es importante	○	○	○	◉	○	Muy importante

Cree usted que los recursos informáticos no son importantes al momento de mejorar el rendimiento de los IDS/IPS *

	1	2	3	4	5	
No es importante	○	◉	○	○	○	Muy importante

Considera usted que la implementación de técnicas machine learning mejoran la detección de amenazas en los IDS/IPS *

	1	2	3	4	5	
No es importante	○	○	○	○	◉	Muy importante

Considera usted que la detección automática de protocolos no es una función principal de los IDS/IPS *

	1	2	3	4	5	
No es importante	○	○	◉	○	○	Muy importante

Considera que la utilización de hardware de video para la aceleración de análisis de red por parte de los IDS/IPS es importante, aunque su tecnología continúe en desarrollo *

	1	2	3	4	5	
No es importante	○	○	◉	○	○	Muy importante

Considera importante la capacidad de funcionar en diferentes plataformas al momento de elegir un IDS/IPS *

	1	2	3	4	5	
No es importante	○	○	○	◉	○	Muy importante

Considera que la detección de nuevas anomalías en la red durante un análisis es un proceso importante para los IDS/IPS *

	1	2	3	4	5	
No es importante	○	○	○	◉	○	Muy importante

Cree usted que los IDS/IPS no deben estar disponibles para el análisis del tráfico de datos permanente de una infraestructura de red *

	1	2	3	4	5	
No es importante	○	◉	○	○	○	Muy importante

Considera que el escalamiento de una infraestructura de red debe estar acorde a las capacidades de las herramientas de análisis de red como lo son los IDS/IPS *

	1	2	3	4	5	
No es importante	○	○	○	◉	○	Muy importante

Los administradores de red no consideran que una interfaz gráfica de administración sea importante dentro de los IDS/IPS *

	1	2	3	4	5	
No es importante	○	○	◉	○	○	Muy importante

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL APRENDIZAJE

UNIDAD DE PROCESOS TÉCNICOS Y ANÁLISIS BIBLIOGRÁFICO Y DOCUMENTAL

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: *27/04/2021*

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: Jonathan Patricio Herrera Silva
INFORMACIÓN INSTITUCIONAL
Facultad: Informática y Electrónica
Carrera: Ingeniería en Electrónica, Telecomunicaciones y Redes
Título a optar: Ingeniero en Electrónica, Telecomunicaciones y Redes
f. Analista de Biblioteca responsable: Lic. Luis Caminos Vargas Mgs.

LUIS ALBERTO CAMINOS VARGAS

Firmado digitalmente por LUIS ALBERTO CAMINOS VARGAS
Nombre de reconocimiento (DN): c=EC, l=RIOBAMBA, serialNumber=0602766974, cn=LUIS ALBERTO CAMINOS VARGAS
Fecha: 2021.04.27 10:35:07 -05'00'

0850-DBRAI-UTP-2021