UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TÍTULO DEL TRABAJO DE TITULACIÓN

IMPLEMENTACIÓN DE IDS/IPS BASADO EN SOFTWARE DE CÓDIGO ABIERTO PARA LA RED DEL CUERPO DE BOMBERO DE SALINAS.

AUTOR

Balón Malavé, Moisés Alexander

Examen Complexivo

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

Ing. Iván Alberto Coronel Suárez, MSIA.

Santa Elena, Ecuador

Año 2024

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Ing. José Sánchez Aquino, Mgt.
DIRECTOR DE LA CARRERA

Ing. Iván Coronel Suárez, Mgt.
TUTOR

Lsi. Daniel Quirumbay Yagual, MSIA.
DOCENTE ESPECIALISTA

Ing. Marjorie Coronel Suárez. Mgt.
DOCENTE GUÍA UIC

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por Balón Malavé Moisés Alexander, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 3 días del mes de Diciembre del año 2024

TUTOR

Firmado electrónicamente por:
IVAN ALBERTO
CORONEL SUAREZ

Ing. Iván Alberto Coronel Suárez,

MSIA.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
DECLARACIÓN DE RESPONSABILIDAD

Yo, Balón Malavé Moisés Alexander

DECLARO QUE:

El trabajo de Titulación, (Implementación de IDS/IPS basado en software de código abierto para la red del Cuerpo de Bombero de Salinas.) previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 3 días del mes de Diciembre del año 2024.

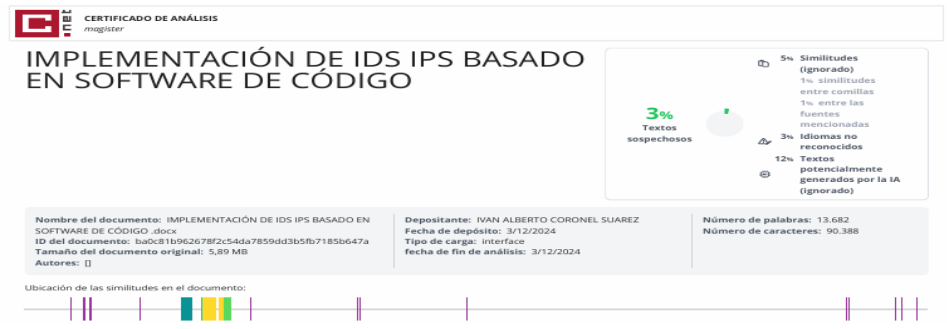
Moises Balón Malavé

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado (Implementación de IDS/IPS basado en software de código abierto para la red del Cuerpo de Bombero de Salinas), presentado por el estudiante, **Balón Malavé Moises Alexander** fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 3%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.


CERTIFICADO DE ANÁLISIS
magister

IMPLEMENTACIÓN DE IDS IPS BASADO EN SOFTWARE DE CÓDIGO

3% Textos sospechosos

5% Similitudes (ignorado)
1% similitudes entre comillas
1% entre las fuentes mencionadas
3% Idiomas no reconocidos
12% Textos potencialmente generados por la IA (ignorado)

Nombre del documento: IMPLEMENTACIÓN DE IDS IPS BASADO EN SOFTWARE DE CÓDIGO .docx
ID del documento: ba0c81b962678f2c54da7859dd3b5fb7185b647a
Tamaño del documento original: 5,89 MB
Autores: []

Depositante: IVAN ALBERTO CORONEL SUAREZ
Fecha de depósito: 3/12/2024
Tipo de carga: interface
fecha de fin de análisis: 3/12/2024

Número de palabras: 13.682
Número de caracteres: 90.388

Ubicación de las similitudes en el documento:

TUTOR

Firmado electrónicamente por:
IVAN ALBERTO CORONEL SUAREZ

Ing. Iván Alberto Coronel Suárez, MSIA.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, Balón Malavé Moises Alexander

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del trabajo de titulación con fines de difusión pública, dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 3 días del mes de Diciembre del año 2024

EL AUTOR

Moises Balón Malavé.

AGRADECIMIENTO

En primer lugar, quiero expresar mi profundo agradecimiento con Dios, por brindarme Fuerza, Salud y Esperanza necesarias a lo largo todo este proceso de estudio. Su guía y cuidado me acompañaron, especialmente en las noches de sabor amargo durante mi jornada de trabajo, a pesar de los desafíos y cansancio.

Agradezco a mi Familia, por el amor que me han brindado, a pesar de toda la circunstancia que hemos vivido, el amor de Padre y Madre siempre ha estado presente. El apoyo incondicional que me han brindado lo llevo en el corazón, y por su inmenso apoyo, este momento no hubiera sido posible.

Al Cuerpo de Bombero de Salinas, junto al Mayor (B) Ing. Cristhian Ramírez. y al Ing. Luiggi Villafuerte, que desde el inicio estuvieron dispuesto a apoyarme en esta fase final.

A mi querida Novia, gracias por apoyarme en los momentos difíciles, por apoyarme a no rendirme, por secar mis lágrimas de frustración, de desesperación, por creer en mí.

A mi hermano Edison Vergara y mis sobrinos, que a pesar de nos crecer juntos, la vida nos une nuevamente, por darnos esos consejos, por preocuparse, por estar pendiente de mamá cuando no estamos nosotros.

A mis compañeros de Universidad, Anahí, Andrea, Diana, David, Galo y Tunato, que han sido un gran apoyo durante toda esta etapa, los cuales me han

apoyado en esos días que llegaba a clases trasnochado después de salir de mi trabajo.

Y, por último, pero no menos importante al Ing. Iván Coronel, quien, a pesar de estar con el tiempo muy ocupado, aceptó ser parte de este trabajo final, eternamente agradecido con sus enseñanzas y correcciones.

Moises Alexander, Balón Malavé

DEDICATORIA

Dedico este trabajo con mucha emoción a Dios, por darme la fuerza para no rendirme, por guiarme en el buen camino y hacer las cosas bien y de bien.

A mi madre, Rebeca Malavé, por apoyarme en todo el camino, por seguir creyendo en mi a pesar de mi forma de ser, por sus consejos, por levantarse muy temprano a prepararme el desayuno, por preocuparse cuando solo llegaba a merendar e irme a trabajar. Es un pilar muy fundamental para mi vida.

A mi padre Luca Balon, que me ha apoyado también en todo camino, por estar pendiente de mí, por darme consejo, por cuidarme, por estar pendiente cuando salgo de la casa rumbo al trabajo, por esperarme al amanecer con la puerta abierta y preocuparse por si digerí algún alimento.

A mi hermano Smith Balon, que a pesar de su accidente nunca le falta esa sonrisa, que siempre se preocupó sus Hijas en ese momento difícil y que ahora estamos llegando a nuestra etapa de ser Profesionales. A mi cuñada Diana y mis sobrinas Bianca y Paulette, que en todo momento se han preocupado por mi cuando no regresaba a casa temprano.

A mi amada novia, Michelle Escalante, quien ha estado a mi lado en este proceso, el cual me ha demostrado su amor, su paciencia, su compresión y apoyo cuando me he sentido frustrado, desesperado. Tú compañía es muy especial para mí.

Y, por último, pero no menos importante, a mis amigos, Daker, Edward y Johan, que muchas veces, después de tener un día agotador han estado dispuesto para conversar y pasar el momento.

Simplemente gracias con todos y los llevo en el corazón.

Moises Alexander, Balón Malavé

ÍNDICE GENERAL

ÍNDICE DE TABLAS

INDICE DE FIGURAS

RESUMEN

El presente trabajo de implementación de sistema de detección y prevención de intrusos (IDS/IPS) basado en software de código en la red del Cuerpo de Bomberos de Salinas tiene como objetivo principal mejorar la seguridad de la infraestructura tecnológica de dicha institución. A través de cinco fases que incluyen el proyecto como son la selección de herramientas, pruebas de vulnerabilidad, implementación de herramientas, pruebas de validación y generación de reportes después de la implementación, se implementó un sistema de monitoreo en tiempo real y herramientas de detección y prevención de intrusos. Los resultados demuestran la efectividad del IDS/IPS en la identificación de alertas y bloqueos de ataques.

Palabras claves: IDS/IPS, Seguridad, Redes

ABSTRACT

The present work of implementing an intrusion detection and prevention system (IDS/IPS) based on code software in the network of the Salinas Fire Department has its main objective to improve the security of the technological infrastructure of said institution. Through five phases that include the project such as the selection of tools, vulnerability tests, implementation of tools, validation tests and generation of reports after implementation, a real-time monitoring system and intrusion detection and prevention tools were implemented. The results demonstrate the effectiveness of the IDS/IPS in identifying alerts and blocking attacks.

Keywords: IDS/IPS, Security, Networks

INTRODUCCIÓN

En este contexto, el Cuerpo de Bomberos de Salinas, es una institución cuyo objetivo principal se basa en la protección y seguridad ciudadana, en donde depende de sistemas de información y redes de comunicación para garantizar las respuestas a los ciudadanos ante algún incidente. Sin embargo, la ausencia de medidas de seguridad en su infraestructura de red expone a la institución a varios tipos de amenazas que pueden llegar a comprometer directamente sus datos y servicios.

El presente proyecto tiene como finalidad implementar herramientas de sistemas de detección y prevención de intrusos (IDS/IPS) mediante el uso de software de código abierto, entre los que se encuentran: Snort, Suricata y Netdata; las cuales permitirán monitorear en tiempo real el tráfico de red, identificar patrones de comportamientos inusuales y dar respuestas rápidas ante posibles ciberataques. La búsqueda de una solución escalable y adaptable para las necesidades de la organización sin costos elevados de soluciones comerciales; es el motivo principal para la elección de este tipo de software.

La metodología del proyecto se encuentra dividida en cinco fases. La primera fase comprende la investigación y selección de herramientas de seguridad que se adapten con la infraestructura actual del Cuerpo de Bomberos. La segunda, conlleva un análisis de vulnerabilidades utilizando técnicas de hacking ético, para identificar posibles puntos de acceso no autorizados. La tercera fase comprende la implementación de los sistemas IDS/IPS, configurando reglas de alerta y bloqueo específicas para la detección de ataques. En la cuarta fase, se ejecutan pruebas de seguridad simulando distintos tipos de ataques, evaluando la eficacia de las herramientas implementadas en la detección y prevención de amenazas. Finalmente, en la quinta fase, se elabora un reporte de los intentos de intrusos, para medir el desempeño del sistema de seguridad y realizar mejoras continuas.

CAPÍTULO 1.- FUNDAMENTACIÓN

1.1 Antecedentes.

La seguridad informática se ha convertido en uno de los aspectos fundamentales para garantizar integridad y confidencialidad en la información a nivel organizacional, el cual conlleva a que los ataques sean considerados problemas más frecuentes en el internet, el cual a su vez es considerado como uno de los principales causantes de la pérdida de información por las vulnerabilidades y/o brechas de seguridad que se tienen en un sistema sin previa revisión [1].

En el caso del Cuerpo de Bomberos de Salinas, su institución ubicada en el cantón Salinas, cuenta con dos sedes adicionales en el Municipio de Salinas y la Parroquia Santa Rosa. La comunicación que existe entre las sedes se realiza mediante una VPN adquirida a un proveedor de servicios de internet (Ver Anexo 1). La seguridad actual de la infraestructura presenta diversas debilidades, la cual se expone a diversos tipos de ataques, comprometiendo la integridad, confidencialidad y disponibilidad de la red. Con base en un análisis realizado junto con el responsable del área de TI, se optó por implementar herramientas de detección y prevención de intrusos, con el objetivo de fortalecer y mejorar la seguridad de la red y mitigar posibles ataques futuros.

Aunque los sistemas operativos cuentan con herramientas de seguridad como los firewalls, esto no siempre es suficiente para proteger completamente la infraestructura. Sin embargo, es crucial que estas herramientas sean completamente seguras, por lo tanto, la implementación de sistemas de detección de intrusos es fundamental para proteger servidores y redes de posibles ataques que puedan comprometer severamente la infraestructura, el cual esto conllevaría un gran costo para la recuperación de los datos [2].

Además, los sistemas de detección y prevención son esenciales para mitigar ataques en las redes y sistemas informáticos. Estas herramientas monitorean el tráfico entrante y saliente que se presentan en la red, permitiendo identificar y enfrentarse antes posibles amenazas de una forma pasiva o activa antes las amenazas. Aunque estas herramientas no están totalmente diseñadas para detener completamente los ataques, juegan un papel importante el cual es prevenir que las amenazas [3] .

Con el avanzar de los años, los ataques han ido aumentando, según estadísticas publicadas por CVE (*Common Vulnerabilities and Exposures*), en el 2023 se identificó más de 28.778 vulnerabilidades, el cual representa un aumento del 14% comparado al año anterior. Un reporte emitido por Verizon, el 74% de las brechas de seguridad se produce por el error humano, que a través de la ingeniería social pueden manipular a los empleados de una empresa para que les compartan datos privados [4].

Según la empresa Fortinet, entre uno de los ataques más comunes es el Ataque DoS Y DDoS, el cual son cada vez más frecuentes y de mayor volumen, además que la plataforma CloudFlare en 2022 detectó y mitigó ataques de más de 1tb/s. Según estudios realizados por Alejandro Santos uno de los mayores ataques fue por DDoS, de un tamaño de 2,5 Tb/s (Ver Figura 1), fue realizado por una variante de Botnet Mirai y cuyo objetivo era un servidor de un videojuego llamado Minecraft denominado *Wynncraft*. [5].

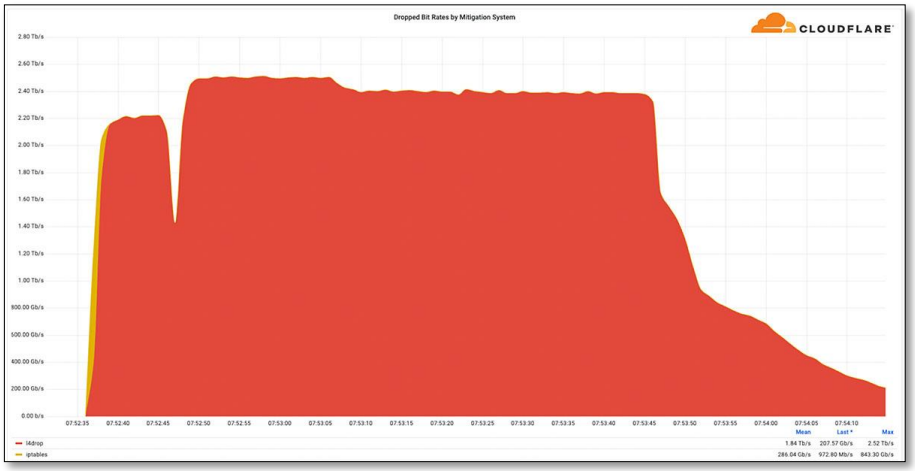


Figura 1Ataque al servidor Minecraft – Fuente CloudFlare [5]

Según las estadísticas publicada por la empresa Kaspersky revela que en Latinoamérica el phishing se sextuplico con el reinicio de la actividad económica, el cual también aumento a un 50% de ataques de troyanos a cuentas bancaria de las regiones, esto quiere decir que hay un aproximado de 5 ataques por minuto. Dicha empresa registro 286 millones bloqueos de intentos de *phishing*, en donde esto representa un aumento del 617% en comparación con el último año, dando un promedio de 544 ataques por minuto. Entre los países más afectados (Ver Figura 2) se encuentran Brasil, con 134 millones de intentos de ataque, México (43 millones), Perú (31,5 millones), Colombia (30,9 millones),

Ecuador (12,2 millones), Chile (10,5 millones) y Argentina (9,4 millones) [6].

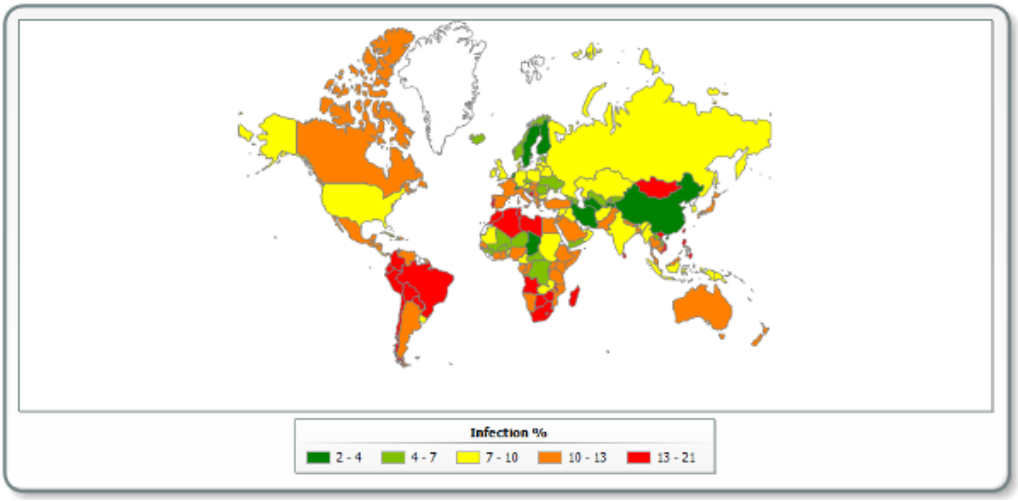


Figura 2 Mapa de los países más atacados por mensajes falsos (phishing) [6].

Además, recientemente la empresa CloudFlare revelo que llegó a mitigar un ataque de denegación de servicios distribuido mejor conocido como Ataque DDoS, el ataque logró alcanzar un máximo de 3,8 tbps de paquetes en un tiempo de 65 segundos (Ver Figura 3) y muchos de esos paquetes superaron 2000 millones de paquetes por segundo (Ver Figura 4). El ataque específicamente fue realizado en la capa 3 y 4 [7].

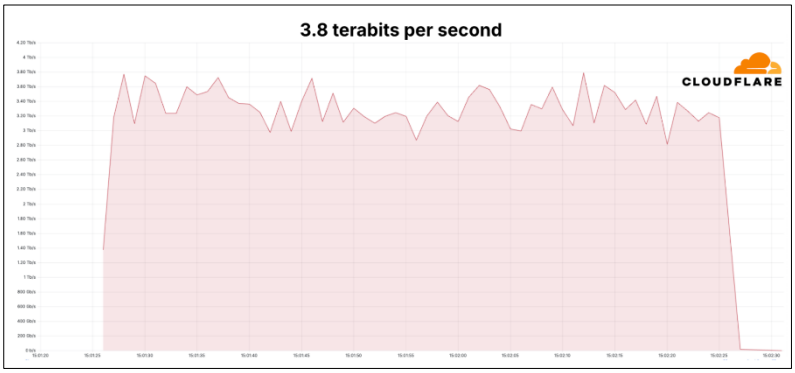


Figura 3 Ataque DDoS mitigado de 3,8 terabits por segundo [7]

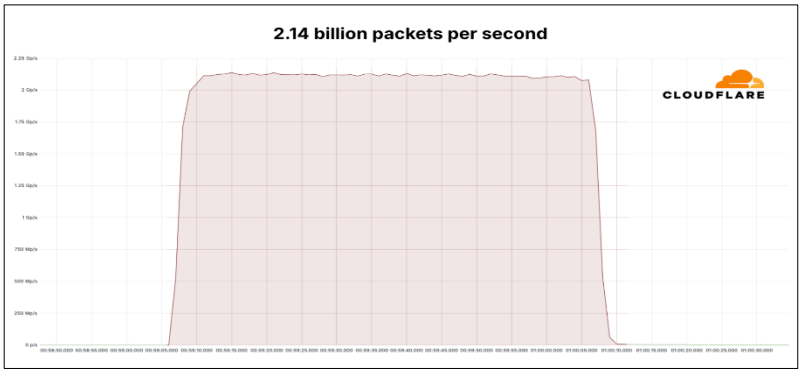


Figura 4 ataque DDoS mitigado de 2140 millones de paquetes por segundo [7]

Este nuevo ataque supera el récord anterior de Ataque DDoS volumétrico más grande que se alcanzó y fue de 3.74 Tbps en 2023 y fue dirigido a un cliente anónimo de la empresa Microsoft Azure en Asia. Estos ataques aprovechan el protocolo UDP en un puerto dijo y estos paquetes provienes de estos Rusia, Vietnam, Brasil, España, Estados Unidos entre otros, además usan dispositivos como Mikrotik, DVR y también servidores web que se vean comprometidos [7].

1.2 Descripción del proyecto.

Para el desarrollo de proyecto presente se ha propuesto la implementar las siguientes fases:

Fase I: Investigación y Selección de Herramientas de Seguridad

Investigar dos herramientas de código abierto el cual nos brinde una mejor seguridad para nuestra red y obtener información oficial de la red.

- Entrevista con el director de T.I
- Aplicar técnica de observación en el área de T.I
- Búsqueda de mejores herramientas de opensource.

Fase II: Pruebas de Vulnerabilidades y Escaneo de Red

A través de Kali Linux y script de Python se realizará la búsqueda y escaneo de posibles puertos y vulnerabilidades del servidor que podrían ser explotados para comprometer la seguridad de la información.

- Escaneo de puertos con nmap.
- Script en leguaje Python.

Los posibles ataques para las pruebas de vulneración de la red son:

- Denegación de Servicios.
- Ataque de Fuerza Bruta

Fase III: Implementación de herramientas de Seguridad

Llevaremos a cabo la instalación y configuración de cada una de las herramientas seleccionadas que se optó por usar en la fase anterior

- Instalación de Ubuntu 20.04

- Instalación de IDS Snort
- Instalación de IDS/IPS Suricata
- Instalación de NetData (Monitoreo en tiempo real)

Fase IV: Pruebas de Seguridad y Validación de la Implementación

Realizar ataques simulados bajo la supervisión del encargado del área de TI para evaluar el desempeño de las herramientas y a través de script de Python obtener datos relevantes sobre algún ataque existentes.

Fase V: Reporte De Ataques Post Implementación.

En esta última fase se llevará a cabo un reporte de los intentos de ataques que se han sido detectado después de la implementación del sistema de prevención y detección de intrusos. El objetivo principal de esta fase es evaluar el sistema frente a amenazas identificadas.

1.3 Objetivos del proyecto.

1.3.1 Objetivo General. -

- Integrar un Sistema de Detección y Prevención de Intrusos (IDS/IPS) para la prevención y detección de ataques, a través de software de código abierto minimizando la explotación de vulnerabilidades y proporcionando una respuesta rápida ante los ataques.

1.3.2 Objetivo específico. -

- Realizar el escaneo de puertos y servicios en la red utilizando herramientas de Kali Linux y script de Python para identificar vulnerabilidades existentes.
- Elaborar un manual técnico de la instalación y configuración de la herramienta IDS/IPS.
- Realizar un entorno de pruebas de seguridad utilizando script de Python y herramientas de Kali Linux para verificar y evaluar la efectividad del IDS/IPS implementado.

1.4 Justificación.

La seguridad informática juega un aspecto importante para garantizar la integridad y confidencialidad de toda la información que posee. Los crecientes ataques representan

una amenaza constante de aumentos significativos en incidente como son spam, malware, phishing, ataques de denegación de servicio, ransomware [8].

Dicha institución, al igual que cualquier otra institución no está exenta de tener estos riesgos, los servicios que manejan son fundamental para el flujo de toda la información, sin embargo, los sistemas de detección de intrusos, con el pasar del tiempo se hacen sumamente necesarios para las instituciones para evitar cualquiera de los tipos de ataques anteriormente mencionados, y estos permite contener los ataques a la infraestructura tecnológicas.

El presente proyecto tiene como finalidad implementar dos herramientas claves en la red del cuerpo de Bombero de Salinas: sistema de prevención y detección de intruso (IDS/IPS). Estas herramientas permitirán monitorear de manera continua las redes de comunicación para identificar posibles ataques. Además, su implementación nos facilitará identificar patrones de tráfico sospechosos en la red y emita alertas tempranas facilitando una respuesta rápida ante cualquier posible ataque que comprometa los activos de información del cuerpo de bombero.

Los sistemas IDS/IPS tiene la capacidad de identificar las actividades maliciosas en tiempo real, en donde permitirá que se reaccione de manera inmediata ante cualquier ataque detectado.

El presente proyecto esta alineado al Plan de Creación de oportunidades 2021-2025, el cual se basa específicamente en [9]:

Directriz 1: Soporte territorial para la garantía de derechos

Lineamiento Territorial: Acceso equitativo a servicios y reducción de brechas territoriales.

Según su política

A4: Fortalecer la conectividad y el acceso a las TIC como una vía para mejorar el acceso a otros servicios.

Objetivos del Eje Seguridad Integral.

Objetivo 10.- Garantizar la soberanía nacional, integridad territorial y seguridad del estado.

Según sus politicas:

10.1: Fortalecer al Estado para mantener la confidencialidad, integridad y disponibilidad de la información frente a amenazas provenientes del ciberespacio y proteger su infraestructura critica.

1.5 Alcance

El proyecto se basa específicamente en la implementación de sistemas de detección y prevención de intrusos, en el cual está basado en software de código abierto con el objetivo de mejorar la seguridad de la red del Cuerpo de bombero de Salinas, como anteriormente se mencionó, el proyecto se dividirá en 5 fases:

Fase I: Investigación y Selección de Herramientas de Seguridad.

Investigar dos herramientas de código abierto el cual nos brinde una mejor seguridad para nuestra red y obtener información oficial de la red. Además, se aplicará metodologías de investigación como la investigación documental y la investigación de campo, en donde se realizará entrevista al director del área de Tecnologías de información para obtener la mayor información real posible.

Fase II: Pruebas de Vulnerabilidades y Escaneo de Red.

A través de diferentes herramientas como Kali Linux y scripts de Python se realizará métodos de hacking ético para obtener información adicional de la red, estas prácticas de hacking se realizarán bajo autorización del encargado del área de TI, en donde toda información obtenida deberá ser de total confidencialidad y está no se podrá divulgar.

Fase III: Implementación de herramientas de Seguridad.

Con la selección de las herramientas de sistema de detección y prevención de intrusos se procederá a la instalación y configuración en la red. Esta fase incluirá personalización de reglas para identificar patrones de tráficos no autorizados o inusuales y también alertar sobre algún ataque para la respuesta inmediata del personal encargado del área.

Fase IV: Pruebas de Seguridad y Validación de la Implementación.

Se realizará ataques simulados bajo la supervisión del encargado del área de TI para evaluar el desempeño de las herramientas y a través de script de Python obtener datos relevantes sobre algún ataque existente.

Fase V: Reporte De Ataques Post Implementación

La quinta y última fase se centrará en la generación de reportes sobre los ataques detectados y mitigado tras la implementación del IDS/IPS en la red del Cuerpo de Bombero de Salinas, se enfocará principalmente en los ataques de Denegación de servicios y escaneo de red.

CAPÍTULO 2.- MARCO TEÓRICO Y METODOLOGÍA DEL PROYECTO

2.1 Marco Contextual

2.1.1 Cuerpo de Bombero de Salinas.

El Cuerpo de Bomberos de Salinas, fue fundada el 29 de julio de 1975, benemérita institución que está orientada a salvaguardar la vida, los bienes y el medio ambiente de la comunidad. Su labor consiste en prevenir y combatir con valentía tipo siniestros que puedan presentarse en cualquier momento del día, desde incendios [10].

Dicha institución está constituida por hombres y mujeres que velan por la vida de cada persona, además que se encuentra en constante capacitaciones y mejorando todas sus habilidades tantos como técnicas y operativas. El personal que brinda el apoyo a la comunidad ha permitido ganarse el respeto y admiración por arriesgar sus vidas para salvar las de otros. Su formación marcó un hito crucial en la seguridad y el bienestar del cantón Salinas y sus habitantes [10].

A lo largo de los años, ha evolucionado con tecnologías avanzadas y métodos innovadores para combatir incendios y emergencias, demostrando un compromiso inquebrantable con la seguridad y la respuesta eficiente ante situaciones críticas. Su legado de valentía y servicio continúa siendo una inspiración para las generaciones presentes y futuras [10]. El cuerpo de bombero de Salinas está ubicado en el cantón salinas, tiene sede principal la Av. General Enríquez Gallo, y tres sedes más ubicadas en las parroquias Santa Rosa, José Luis Tamayo y Anconcito. (Ver Figura 5).

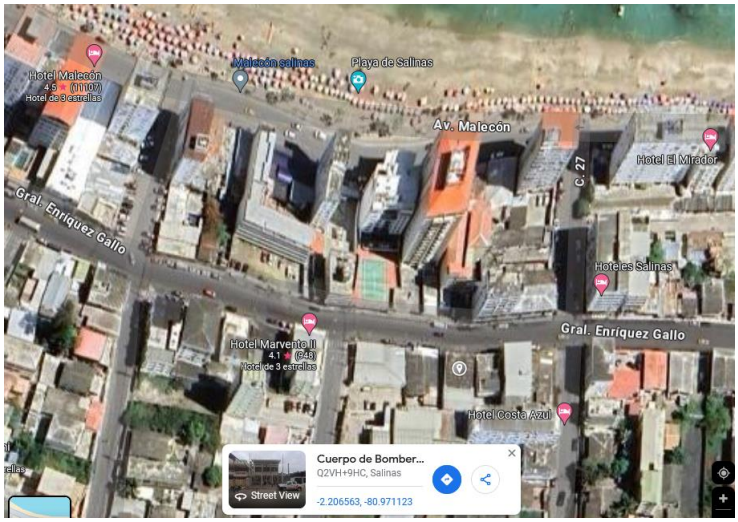


Figura 5 Ubicación del Cuerpo de Bomberos. Fuente Google Maps

2.1.2 Misión.

Intervenir oportunamente, para salvaguardar la vida y bienes de la comunidad, responder a las necesidades de los ciudadanos mediante un rápido, profesional y humanitario servicio, cumpliendo con el compromiso a través de la prevención, combate y extinción de incendios, servicios de emergencias médicas Pre-Hospitalarias, rescate, educación a la ciudadanía para la autoprotección, y cualquier otro evento producto de los fenómenos naturales, o sociales, con la preparación técnica de su personal para proporcionar el mejor servicio a la comunidad [12].

2.1.3 Visión

Disponer de una institución capacitada profesionalmente, con la máxima efectividad y eficacia en la prevención y atención de emergencias o desastres de su incumbencia, con el mejoramiento continuo de los equipos y el desarrollo técnico profesional, económico y social, de todo el personal rentado y voluntario, tanto hombres como mujeres que conforman la Institución para alcanzar máximos niveles de ejecución y operación, para la tranquilidad y satisfacción de la comunidad [12].

2.1.4 Valores.

Lealtad, espíritu de equipo, honestidad y cumplimiento de normas, solidaridad, rectitud de conciencia, sacrificio, honor, disciplina, abnegación, respeto a la dignidad humana, vocación de servicio e integridad moral. Estos valores los gerenciamos aún a riesgo de nuestra propia seguridad y bienestar [12].

2.1.5 Estructura Organizativa del Cuerpo de Bombero de Salinas.

A continuación, se observará la estructura organizativa de la institución (Ver Figura 6) [13].

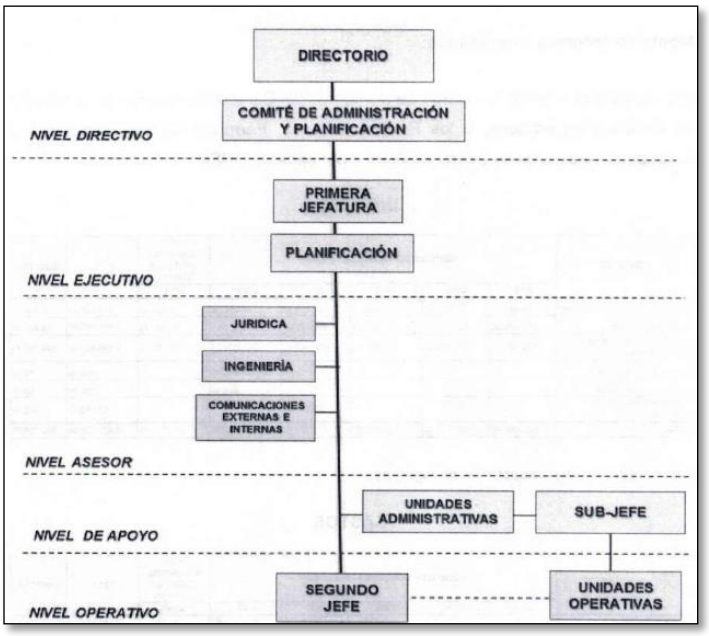


Figura 6 Estructura organizativa del cuerpo de bombero de Salinas [13]

2.1.6 Base Legal

Es fundamental tener claro el marco legal entorno a la seguridad en los sistemas de información y comunicaciones. El COIP establece disposiciones claves para proteger los activos digitales, imponiendo sanciones a conductas que vulneren la confidencialidad, integridad y disposición de la información.

2.1.6.1 Código Orgánico Integral Penal.

Sección Tercera

Delitos contra la seguridad de los activos de los sistemas de información y comunicación [14]:

"Art. 229.-revelación ilegal de base de datos. -La persona que, en provecho propio o de un tercero, revele información registrada, contenida en ficheros, archivos, bases de datos o medios semejantes, a través o dirigidas a un sistema electrónico, informático, telemático o de telecomunicaciones; materializando voluntaria e intencionalmente la violación del secreto, la intimidad y la privacidad de las personas, será sancionada con pena privativa de libertad de uno a tres años. [14]"

“art. 230.-interceptación ilegal de datos. -Será sancionada con pena privativa de libertad de tres a cinco años [14]:

1. La persona que, sin orden judicial previa, en provecho propio o de un tercero, intercepte, escuche, desvíe, grabe u observe, en cualquier forma, contenido digital en su origen, destino o en el interior de un sistema informático o dispositivo electrónico, una señal o una transmisión de datos o señales [14].
2. La persona que ilegítimamente diseñe desarrolle, ejecute, produzca, programe o envíe contenido digital, códigos de accesos o contraseñas, certificados de seguridad o páginas electrónicas, enlaces o ventanas emergentes o modifique el sistema de resolución de nombres de dominio de un servicio financiero o pago electrónico u otro sitio personal o de confianza, de tal manera que induzca a una persona a ingresar a una dirección o sitio de internet diferente al que quiere acceder [14].
3. La persona que posea, venda, distribuya o, de cualquier otra forma, disemine o introduzca en uno o más sistemas informáticos, dispositivos electrónicos, programas u otros contenidos digitales destinados a causar lo descrito en el número anterior [14].
4. La persona que a través de cualquier medio copie, clone o comercialice información contenida en las bandas magnéticas, chips u otro dispositivo electrónico que esté soportada en las tarjetas de crédito, débito, pago o similares [14].
5. La persona que produzca fabrique, distribuya, posea o facilite materiales, dispositivos electrónicos, o programas o sistemas informáticos destinados a la comisión del delito descrito en el inciso anterior [14].

Art. 234.-acceso no consentido a un sistema informático, telemático o de telecomunicaciones.

1. La persona que sin autorización acceda en todo o en parte a un sistema informático o sistema telemático o de telecomunicaciones o se mantenga dentro del mismo en contra de la voluntad de quien tenga el legítimo derecho sobre dicho sistema, será sancionada con la pena privativa de la libertad de tres a cinco años [14].

2. Si la persona que accede al sistema lo hace para explotar ilegítimamente el acceso logrado, modificar un portal web, desviar o redireccionar el tráfico de datos o voz u ofrecer servicios que estos sistemas proveen a terceros, sin pagarlos a las o los proveedores de servicios legítimos, será sancionada con la pena privativa de la libertad de tres a cinco años [14].

2.2 Marco Conceptual.

2.2.1 Redes

La computación en la red o conexión de red es quien se encarga de realizar el proceso de conectar dos o más dispositivos tecnológicos, como pueden ser computadoras, dispositivos moviles, router, servidores entro otros, esté se encarga de transmitir e intercambiar la información y recursos. [15].

2.2.2 Modelo OSI

Modelo de interconexión de sistemas abiertos, es u marco conceptual el cual separa las funciones de la comunicación de la red en 7 capas. Además, que el envío de datos a través de la red es algo complejo debido a que varias tecnologías como lo son hardware y software deben estar funcionando de manera correcta [16].

Las capas del modelo OSI se denomina más por su nombre y número, desde el más bajo al más alto y son los siguientes (Ver Tabla 1).

Grupo	#	Nombre	Tecnología y protocolos	Componentes comunes
Capas Superiores	7	Aplicación	DNS- DHCP – SNMP – FTP – POP3- HTTP- TELNET	Aplicación en red, correos electrónicos, navegadores, servidores web.
	6	Presentación	SSL – Shell -MIME	
	5	Sesión	NetBIOS	
Capas inferiores	4	Transporte	TCP & UDP	VoIP & Video – Firewall

	3	Red	IPV4 – IPV6 IPNAT – ARP – RARP – ICMP	Direccionamiento IP – Ruteo
	2	Enlace de datos	Frame Ethernet – WLAN – ATM	Interfaces de red y controladores - WAN
	1	Físico	Señales eléctricas – Ondas Luminosas	Medios Físicos, hubs y repetidores.

Tabla 1 Capas del modelo OSI

2.2.3 Firewall.

Firewall es un sistema de seguridad de las redes de computadoras, el cual permite restringir el tráfico del internet entrando o saliente de una red privada, esto es un software o puede ser también en unidad hardware que permite o bloquea los paquetes de datos de forma selectiva. [17].

2.2.4 Dirección IP

Es una dirección única el cual puede identificar a un dispositivo en internet o en una red local, la abreviatura IP significa “Protocolo de Internet”, este nos permite él envió de información entre uno o más dispositivos en una misma red. Esta dirección ip es una cadena de números separados por puntos. Cabe recalcar que estas direcciones IP no pueden ser aleatorias [18].

2.2.5 Sistema de Detección de intrusos.

Un sistema de detección de intrusiones (IDS) herramienta que monitorea el tráfico de la red e informa sobre las actividades sospechosas a los equipos con una respuesta a incidentes y sobre todo a las herramientas de ciberseguridad [19].

2.2.6 Tipos de sistema de detección de intrusos.

Según las fuentes de información existen dos tipos de clasificación de los IDS [19]:

- IDS basado en host (HIDS). - Pueden operar en toda la red, su función principal es proteger netamente a la maquina en la que se encuentra instalado, los datos que utilizan son los que se generan en los archivos logs del sistema. El HIDS tiene una gran ventaja, detecta los intentos fallidos de acceso o modificaciones de archivos [20].
- IDS basado en red (NIDS). - IDS más populares por utilizar el tráfico de la red (paquetes tcp/ip) como fuente oficial de la información. Buscan indicios de ataque a cualquier dirección o elemento que se encuentra en la red. La revisión de dicha información se puede realizar mediante los sniffer que permita censar y analizar los datos [20].

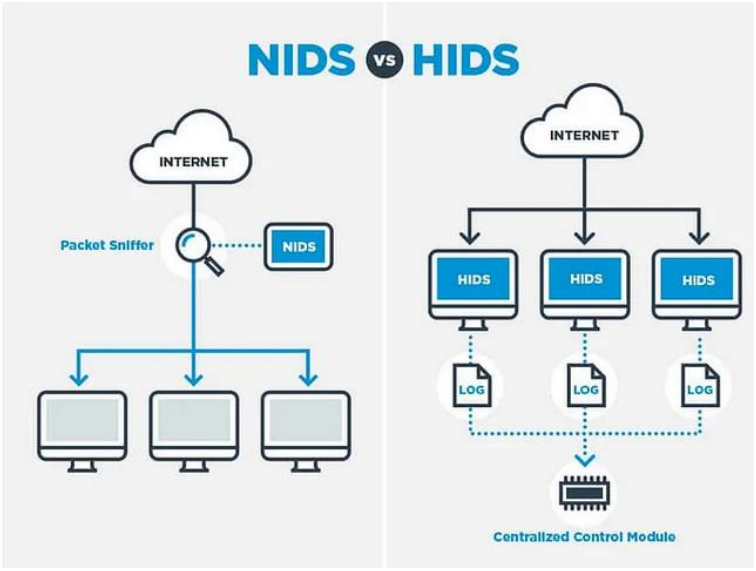


Figura 7 Sistema de detección de intruso basado en NIDS Y HIDS [20]

2.2.7 Tipos de detección.

La clasificación de los tipos IDS mencionados, también involucran dos tipos detecciones:

IDS basados en firmas: Detectan el acceso no autorizado a la red, comparando patrones de tráfico de red o archivos del sistema contra una base de datos de firmas o de ataques conocidos [21].

IDS basados en anomalías. Establecen una línea base del comportamiento del sistema o de la red sin ninguna actividad maliciosa y luego alertan cuando se detecta un comportamiento que se desvía del este modelo [21].

Sistema de prevención de intrusos IPS

Supervisa el tráfico existente en una red, el cual busca amenazas potenciales y las

bloqueas de manera automáticas mediante alertas al equipo de seguridad, finalizas las conexiones peligrosas, elimina el contenido malicioso [22].

Los IPS cuentan con tres métodos principales para las detecciones de amenaza, son de forma exclusiva o en combinación para lograr el análisis del tráfico en la red, los cuales son [23]:

Detección basada en firmas. – Analizan los paquetes que existen en la red en busca de firmas de ataque, en pocas palabras busca característica o comportamientos que están asociados a una sola amenaza específicas. Cuando el IPS encuentra un exploit o ataque, agrega las firmas en su base de datos que esta a su vez incrementa [23].

Detección basada en anomalías. - toma ejemplos de muestra al azar que se generan en el tráfico de la red y compara las muestras con las líneas base de nivel de rendimiento. Cuando se identifica que las muestras están por fuera de la línea base, el IPS desencadena una acción para prevenir un posible ataque y así emitir alertas [23].

2.2.8 Ciberseguridad

La ciberseguridad se encarga de brindar protección privacidad a toda la integridad de computadoras, servidores, dispositivos móviles, redes y ataques malicioso que puedan comprometer de toda la información que se posee en una empresa como puede ser: servidores, bases de datos, páginas web [24].

2.2.9 Ciberataques

Un ciberataque o un ataque cibernético es cualquier tipo de intento deliberado para acceder de manera no autorizada a una red, sistema informático con el fin de robar, modificar o extorsionar a la empresa con filtrar dicha información [25].

2.2.10 Tipos de ataques.

Los ataques más comunes que se presentan son:

- **DoS. - “**Un ataque de denegación de servicio (DoS) es un ataque a un sistema informático o red el cual envía varias solicitudes al recurso web que será atacado con la finalidad de desbordad la capacidad de sitio para administrar diversas solicitudes y evitar que funcione de una manera correcta [26].

- **Man in the middle. –** Conocido como hombre en el ataque de medio, es un término que se utiliza para espiar toda la información que se trafica en la red, el objetivo principal de este ataque es robar la información personal, contraseñas de inicio de sesión, detalles de tarjetas bancaria. Esta información robada durante el ataque tiene muchas finalidades, como falsificación de identidad, transacciones en línea o de fondos [27].
- **Fuerza Bruta**. – Método de prueba y error, conocido como ataque de fuerza bruta, consiste en intentar adivinar información de inicio de sesión, como son credenciales de alguna plataforma. Este tipo de ataque se caracteriza por el uso excesivo para forzar el acceso no autorizados de manera ilegal a cuentas privadas. A pesar de ser un ataque antiguo, sigue siendo efectivo y muy popular en la actualidad entre la comunidad de hackers [28].
- **Inyección SQL**. - Un ataque de inyección de SQL es un ataque que está dirigido a reemplazar la intención original de la solicitud mediante la presentación suministrada de un nuevo script por el atacante sentencias SQL directamente en la base de datos [29].
- **Phishing. –** Es una forma de obtener información confidencial a través de correos electrónicos mediante enlaces fraudulentos, que redirige a una página falsa incitando a rellenar la información personal, este método de robo es más usado para obtener las credenciales de redes sociales y cuentas bancarias **[**30].
- **Ingeniería Social. -** Técnica para manipular información privada que el atacante aprovecha por el error humano, el cual puede accedes a sistemas y objetos de valor, esto en ciberataque se obtiene esta información cuando el usuario de forma desprevenida expone datos sin percatarse, en donde se puede propagar infecciones de malware y llegar a estafas [31].

2.2.11 Kali Linux

Es una distribución de Linux que está basada en Debian, esta específicamente diseñada para el tema de seguridad informática como son: análisis de redes, ataques inalámbricos, análisis forense y más actividades relacionadas a la ciberseguridad. Esta distribución fue diseñada en base a la reescritura de BackTrap, que es otra distribución de Linux. Kali Linux es una de la seguridad de Linux más usada y una de las mejores [32].

2.2.12 Ubuntu

Ubuntu, es una de varias distribuciones de Linux, pero esta distribución es una de las más populares. Es un sistema operativo de código abierto, potente y sobre todo adaptables el cual también se utiliza para servidores, tanto en físico como virtual. Ubuntu es una alternativa a otros sistemas operativos como es Windows y MacOS [33].

2.2.13 Nmap

Nmap en su abreviatura en ingles Network Mapper, es una herramienta de línea de comando en Linux, también es de código abierto el cual se utiliza para escanea direcciones ip y puertos de una red para detectar posibles entradas de ataques o aplicaciones/servicios utilizados [34].

2.2.14 Wireshark

Es un analizador de paquetes de red, el cual tiene como finalidad lograr capturar toda la información que viaja en la conexión [35].

2.2.15 Snort

Snort es el principal Sistema de Prevención de Intrusiones de Código Abierto (IPS) en el mundo. Snort IPS utiliza una serie de reglas que ayudan a definir la actividad de la red maliciosa y utilizan esas reglas para encontrar paquetes que coinenen con ellos y genera alertas para los usuarios [36].

2.2.16 Suricata

Es un motor de IDS el cual tiene detección de amenazas de alta velocidad y bajo consumo de recurso, el cual tendrá un análisis de trafico de red buscando patrones para los accesos no autorizados [37].

2.2.17 NateData

Herramienta de código abierto que nos permitirá visualizar y monitorear métricas en tiempo real como procesos de cpu, red, disco y consultas sql [38].

2.3 Marco Teórico

2.3.1 La seguridad informática es estratégica.

Los incidentes de seguridad son eventos que se producen en la empresa, lo que lleva a tomar mayor conciencia sobre la importancia de la seguridad de la información. Los peligros en las redes cada vez aumentan constantemente y el internet es una puerta a posibles filtraciones y ataques, utilizando métodos que han ido evolucionando para comprometer el software de los sistemas de seguridad por eso se debe implementar un plan de contingencia o programa de seguridad para cualquier riesgo [39].

En una definición más amplia, un programa de seguridad de la información es un plan para mitigar los riesgos asociados con el procesamiento de la información y el uso de los recursos que lo soportan; existiendo tres elementos [39]:

- Integridad. – "Asegura que la información sea exacta, completa, sin alteraciones o modificaciones en su contenido".
- Confidencialidad. – "Tiene como propósito prevenir el uso no autorizado de la información, por personas no facultadas para tal efecto".
- Disponibilidad. - "asegura que los usuarios tengan acceso oportuno y fiable a sus recursos de información".

2.3.2 Sistemas de detección y prevención de intrusos: una taxonomía experimental basada en código abierto orientado a la industria 4.0

Un sistema de detección y prevención de intrusos es una aplicación o hardware que supervisa el tráfico que circula por una red, en donde se logra detectar cualquier tipo de actividad no autorizada que suponga un peligro de amenaza en la red. El sistema tiene capacidad en lograr bloquear cualquier tráfico no autorizado proveniente de origen maligno. Cabe recalcar que dependiendo de la ubicación del ids/ips en la topología, se podrá monitorear el tráfico de la red o subred o de un solo equipo [40].

Existen muchas herramientas, aplicaciones y sistemas, en donde ayudan a mejorar la seguridad de todo los dispositivos, programas e información de una red de datos, algunos ejemplos son los firewalls, redes privadas virtuales, antivirus, antimalware. El acompañamiento de un ids/ips [40].

Una de las herramientas de IDS en la actualidad es Snort, respaldado por la empresa

Cisco; el cual provee la base de datos de las reglas para la detección de tráfico malicioso, está disponible de forma libre y lo pueden diferentes IDS/IPS. Se ha convertido en un sistema estándar de firma de patrones en los IDS [40].

2.3.3 Evaluación del rendimiento de cortafuegos basados en software libre.

Según (Neupane, Hadd y Chen 2018) [41] "Los cortafuegos (firewalls) son sistemas de seguridad que controlan el tráfico de red mediante reglas prestablecidas". Estas herramientas de firewall están ubicadas directamente de cara a cara con el internet y son capaces de prevenir el acceso no autorizado hacia las rede internas de una organización. La implementación se realiza mediante herramientas basadas en software o dispositivos de hardware especializados.

Según (Konikiewicz & Markowski, 2017) "Los firewalls están basados en software y son implementados en un sistema operativo estándar y utilizan los recursos computacionales del ordenador donde este opera, por su parte, los firewalls están basado en hardware y constituyen a los dispositivos físicos fabricados exclusivamente para esta función que poseen su propia CPU, memoria RAM, almacenamiento interno y sistema operativo [41].

2.4 Metodología del proyecto

2.4.1 Metodología de la investigación

La investigación bibliográfica o documental consiste en la revisión de material bibliográfico existente con respecto al tema a estudiar. Se trata de uno de los principales pasos para cualquier investigación e incluye la selección de fuentes de información" [42]. Esta investigación se considera un paso esencial porque incluye un conjunto de fases que abarcan la observación, la indagación, la interpretación, la reflexión y el análisis para obtener bases necesarias para el desarrollo de cualquier estudio. Utilizamos como fuentes primarias, trabajos investigativos relacionados con el tema de estudio, así como, artículos, libros y revistas que proporcionaron información relevante para el desarrollo del proyecto.

La investigación de campo es aquella que se aplica extrayendo datos e informaciones directamente de la realidad a través del uso de técnicas de recolección (como entrevistas

o encuestas) con el fin de dar respuesta a alguna situación o problema planteado previamente [43].

2.4.2 Técnicas e instrumentos de recolección de datos

En la técnica de entrevista , se llevo a cabo un enfoque no estructurada, el cual fue dirigida al director de departamento TIC´S para así obtener información sobre la infraestructura de red de la institución.

Por otro lado, también se utilizó la técnica de observación para evaluar la cultura digital en el área de la seguridad informática. Esta técnica nos permitirá identificar el riesgo que se expone sobre la gestión de claves de seguridad.

2.4.3 Metodología de desarrollo del proyecto

Para el presente proyecto, la implementación de un sistema de seguridad como son prevención y detección de intrusos en el Cuerpo de Bombero de Salinas se ha optado por la metodología **Top-Down** [44] el cual se adaptarán a las siguientes fases (Ver Figura 8)**.** Con este enfoque nos va a permitir abordar la seguridad de la red desde un nivel general hacia detalles específico, en donde se asegurará que la información no se vea comprometida. Por el proyecto se dividirá en 5 fases:

Figura 8 Ciclo de metodología Top-Down

Fase I: Investigación y Selección de Herramientas de Seguridad

Se realizará una investigación de las herramientas de código abierto el cuál permita reforzar la seguridad de la red. Esta selección de dichas herramientas se basará en la capacidad para integrarse de manera eficaz con la infraestructura existente, su facilidad de uso y su adaptabilidad a las necesidades específicas del Cuerpo de Bomberos de Salinas. Por eso es muy importante destacar que la elección de las herramientas se fundamentará en los datos recopilados a través de entrevistas y técnicas de observación realizadas en el departamento de tecnologías de la información de la institución.

Fase II: Pruebas de Vulnerabilidad y Escaneo de Red

Para ello, se utilizarán herramientas de seguridad de Kali Linux y scripts personalizados en Python para ejecutar pruebas de escaneo de puertos, servicios y vulnerabilidades. Durante esta fase, se simularán posibles ataques a la red como, Ataque de Denegación de servicios (DoS), búsqueda de puertos abierto. Esto permitirá obtener un panorama claro de los puntos débiles de la infraestructura, preparando el entorno para la posterior implementación de las herramientas de seguridad.

Fase III: Implementación Progresiva de Herramientas de Seguridad

Se procederá a la implementación de las herramientas IDS/IPS seleccionadas, dicha implementación seguirá un enfoque progresivo, comenzando por la ubicación del equipo que almacenará las herramientas, consiguiente a la configuración de las herramientas las cuales incluirá la personalización de reglas de detección y bloqueos según las necesidades de seguridad identificadas en las fases anteriores.

Fase IV: Pruebas de Seguridad y Validación de la Implementación

Una vez que sido totalmente implementadas las herramientas en la infraestructura de la red, se realizarán pruebas de seguridad para validar su efectividad. Para esta fase se simularán diversos tipos de ataques para comprobar la capacidad del sistema IDS/IPS de detectar y responder a amenazas de manera adecuada. Los tipos de ataques serán controlados por los cuales serán: ataque de denegación de servicios, conexiones remotas al servidor, búsqueda de puertos abiertos entre otros.

Fase V: Reporte De Ataques Post Implementación

Esta fase se centrará en la generación de reportes sobre los ataques detectados y mitigado

tras la implementación del IDS/IPS en la red del Cuerpo de Bombero de Salinas, se enfocará principalmente en los ataques de Denegación de servicios y escaneo de red. Así mismo se analizará. Además, se realizará un análisis del comportamiento del tráfico de la red en tiempo real.

CAPITULO 3.- DISEÑO DE LA PROPUESTA

3.1 Diseño de la propuesta.

3.1.1 Análisis de la situación actual del Cuerpo de Bombero de Salinas.

En la actualidad, el Cuerpo de Bomberos del cantón Salinas cuenta con una infraestructura de red básica, el cual carece de seguridad, como lo es firewall y sistemas de detección de intrusos (Ver Figura 10). La ausencia de medidas de seguridad brinda una oportunidad para que los atacantes puedan vulnerar los puertos abiertos del servidor y dejen sin uso los servicio a través de un ataque de denegación de servicio. Además, la falta de segmentación de la red incrementa el riesgo de que sean atacados.

La infraestructura actual de la institución nos permitirá incorporar de a poco nuevas mejoras en las medidas de seguridad tecnológicas, lo que ayudará disminuir el riesgo de la exposición de la información. La ausencia de estas mejoras no solo pone en riesgo la integridad de los datos, sino que puede afectar la disponibilidad de los servicios internos, llegando a comprometer la operatividad de la institución. Es importante implementar medidas de seguridad para mitigar estos riesgos y proteger adecuadamente los activos de información.

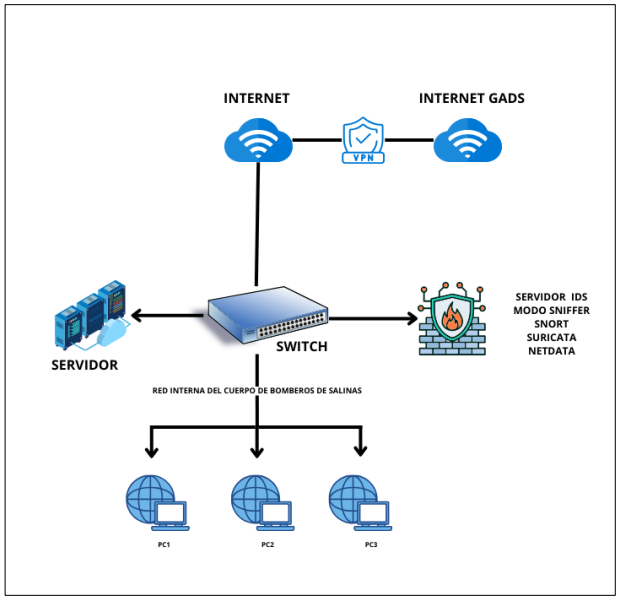


Figura 9 Arquitectura actual de la infraestructura del cuerpo de bombero de salinas

3.1.2 Diseño de la solución.

La implementación del sistema de detección y prevención de intrusos re realizara desde una computadora adicional, el cual su sistema base será Ubuntu 20.04.6, el cual analizara las anomalías o ataques que se presente en toda la red. Además de verificar los puertos que estarán abiertos y serán de servicios de la red.

El diseño del sistema está alineado con las prioridades definidas por el director de TI, configurándose como un **Packet Sniffer**. Su función principal será detectar, alertar y, en caso necesario, bloquear los ataques identificados en tiempo real, garantizando la seguridad y estabilidad de la red.

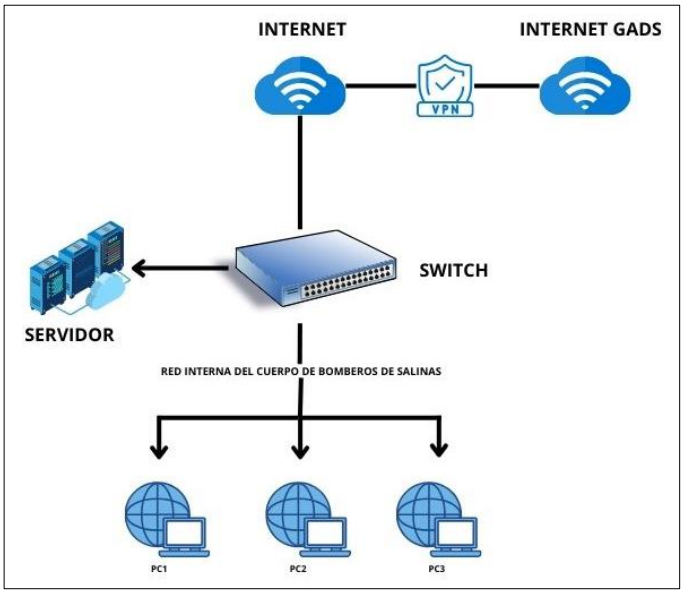


Figura 10 Arquitectura con la implementación del sistema de detección de intruso.

3.2 Desarrollo y pruebas.

3.2.1 Fase I: Investigación y Selección de Herramientas de Seguridad

Para la recolección de datos se aplicará la metodología de campo, el cual involucra la entrevista con el director del departamento de TIC´S del cuerpo de bombero de salinas (Ver Anexo 1 y 2). A través de esta entrevista se obtuvo los siguientes resultados.

- Servidor Principal:

  Dell Tower Power DC T320 (Xeon 2.2 GHz, 72GB RAM PC3, 8 x 3TB 7200 RPM).

- Servidor biométrico.

Computadora HP 260 Mini Desktop.

- Equipos de Red:

  Switch TP-Link (TL-SG1024D no gestionable, 24 puertos Gigabit, rackeable).

  Router Huawei HG8245H ON

Una vez recolectada toda la información de la infraestructura de la red, se realiza la búsqueda y análisis de las mejores herramientas de código abierto para la implementación de los sistemas de detección y prevención de intrusos (IDS/IPS). El análisis se enfoca en que las soluciones a implementar ofrezcan una facilidad de integración y la capacidad de una respuesta antes incidentes.

Con base en el análisis, se ha elaborado la siguiente tabla (Ver Tabla 2) el cual presente diferentes tipos de sistemas de detección y protección de intrusos y monitoreo en tiempo real.

Herramienta	Ventajas	Desventajas	Beneficios
Snort (IDS)	• Amplia comunidad de soporte. • Reglas muy fáciles de configurar. • Detección basada en firmas.	• Compleja configuración inicial. • Genera alertas falsas si la regla no está bien configurada. • Bajo rendimiento de grandes redes.	• Es compatible con diferentes sistemas. • Es herramienta de OpenSource. • Puede ser usado como sniffer, logger paquetes.
Suricata (IDS/IPS)	• Detección basada en firmas y anomalías. • Registro completo del tráfico.	• Bajo nivel de detección para nuevos ataques. • Carece de una interfaz administrativa.	• Maneja grandes volúmenes de datos en tráfico de red. • Es una herramienta de Open Source

			• Suricata como ips bloquea los ataques a través de sus reglas
Zeek (Bro) IDS	• Análisis detallado de los protocolos. • Detección de tráfico malicioso. • Facilidad de integración	• No proporciona prevención de intrusos. • No posee interfaz gráfica nativa. • Configuración de reglas compleja.	• Es de código abierto, de una manera ideal para monitoreo pasivo. • Integración con herramientas forenses.
Osssec (IDS)	• Está basado en fuerte detección en host. • Posee amplia información en la web • Soporta monitoreo para la integridad de archivos.	• No posee un enfoque en las redes. • No es el más factible para verificar el tráfico en tiempo real. • Se limita a eventos basado en host.	• Es herramienta de código abierto, pero soporta una versión premium. • Ideal para redes con varios servidores. • Es una buena elección para detección de intrusos en servidores.
NetData (Monitoreo)	• Monitoreo en tiempo real. • Es fácil de implementar y usar.	• No es ideal para grandes redes. • Soporte muy limitado para entornos grandes.	• Es una herramienta de código abierto. • Ideal para el monitoreo en tiempo real.

	• Posee una interfaz gráfica web. • Posee un amplio	• No es recomendable para un análisis a largo plazo.	• Monitoreo de varias interfaces del equipo.
Zabbix (Monitoreo en tiempo real)	• Soporte más robusto para varios dispositivos. • Almacena grandes cantidades de datos largo plazo.	• Su implementación es compleja. • Requiere de recursos más grande.	• Plataforma de código abierto, el cual se extiende a una versión de paga según la implementación a utilizar.

Tabla 2 Herramientas de IDS/IPS y Monitoreo en tiempo real

Las herramientas seleccionadas luego de la obtención de datos son las siguientes:

Herramienta 1: SNORT

Snort Open Source, es un sistema de intrusos en red, de una manera libre y gratuita, este sistema de detección de intruso basado en red implementa un motor de detección de ataques y barrido de puerto el cual permite registrar, alertar y responder cualquier anomalía en tiempo real [45]. Snort realiza registros de paquetes en tiempo real, análisis de tráfico, análisis de protocolos y finalmente análisis de contenido.

Snort es un sistema que puede integrarse fácilmente con otros tipos de sistemas y dentro de ellas incluyes las siguientes características [46]:

- **Registros de paquetes.**

Snort permite registrar los paquetes de datos en modo de captura, este almacenando toda la información en directorios organizados por la dirección IP del host de la red. Este sistema de detección de intrusos recopila cada paquete que circula por la red y los guarda para análisis posterior, lo que facilita identificar patrones sospechosos y posibles amenazas [46].

- **Análisis de protocolo**

Puede adaptarse y analizar cualquier tipo de protocolo, este proceso de rastreo de red captura datos en capas de protocolo para un análisis adicional, el análisis recopila en Protocolo de control de transmisión / IP (TCP / IP) [46].

- **Monitoreo de tráfico en tiempo real.**

Monitorea todo el tráfico de red que entra y sale de la misma, monitorea en tiempo real y emite alertas a los usuarios para que analicen los paquetes o amenazas maliciosas en las redes IP [46].

- **Reglas fáciles de implementar.**

Estás reglas son muy fáciles de implementar, además permite que la supervisión y protección de la red este en total funcionamiento, el lenguaje de configuración de las reglas es muy flexible y bastantes simples [46].

Herramienta 2: SURICATA

Suricata es el núcleo que reside con su alta capacidad para actuar como un sistema de detección y prevención de intrusos. Uno de los aspectos que destaca a este ids como uno de los mejores es el rendimiento, este ha sido diseñado para que se aproveche al máximo el hardware moderno, se puede procesar grandes cantidades de trafico de red sin la necesidad de ralentizarse, debido a su motor eficiente [47].

Uno de los desafíos para cualquier herramienta de seguridad de red es muy compleja y es aquí donde suricata realmente brilla y es muy fácil de integrarse con otros tipos de sistemas, facilita la tarea de monitorear y gestionar las alertas [47].

Suricata posee las mismas características mencionada con Snort, la única diferencia es que reduce el consumo de cpu, ambos IDS son eficiente y efectivos

Herramienta 3: NetData

NetData es un monitor de red en tiempo real, de código abierto orientada a distribuciones de Linux, es muy sencillo de implementarlo y fácil de configurarlo, esto nos permite monitorear el estado de la red antes un posible ataque y va de la mano con los IDS/IPS, también es muy fácil de integrarse con cualquier sistema [48]. NetData puede monitorear gran cantidad de elementos y componente del sistema como son [49]:

- Cpu
- Ram
- Cargas
- Interfaces de red
- Lectura/escritura de disco
- N° de interrupciones
- N° de procesos

Adicional también monitorear software como servidores Apache/Nginx, métodos http, códigos de respuesta http y posee un sistema de alarmas que se puede personalizar [49].

3.2.2 Fase II: Pruebas de vulnerabilidad y escaneo de red.

3.2.2.1 Escaneo de red con script de Python.

El escaneo de red mediante un script de Python permitirá analizar la seguridad de la infraestructura identificando puertos. Este proceso ayuda a detectar posibles amenazas vulnerables que podrían ser explotados por hackers. El script realiza una conexión a ciertos puertos y verifica si está abierto, registrando cualquier puerto disponible para recibir conexiones (Ver Figura #11), para verificar el código (Ver Anexo #3).

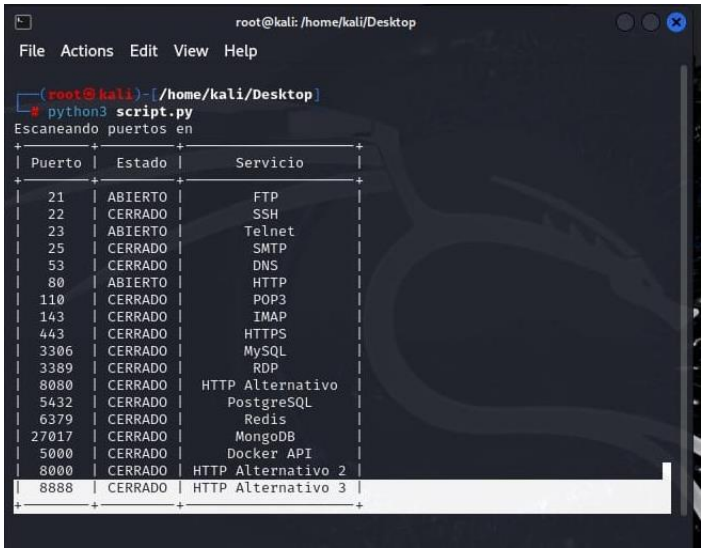


Figura 11 Escaneo de red a través de script de Python.

3.2.2.2 Análisis de la red con Wireshark

Para garantizar la seguridad de la red del cuerpo de bomberos de Salinas, se realiza una captura del tráfico de red, antes de implementar las herramientas y reglas de los sistemas de detección y prevención de intrusos. Este análisis es fundamental para identificar posibles vulnerabilidades y asegurar que la red esté funcionando correctamente sin ningún tipo de amenaza (Ver Figura #12).

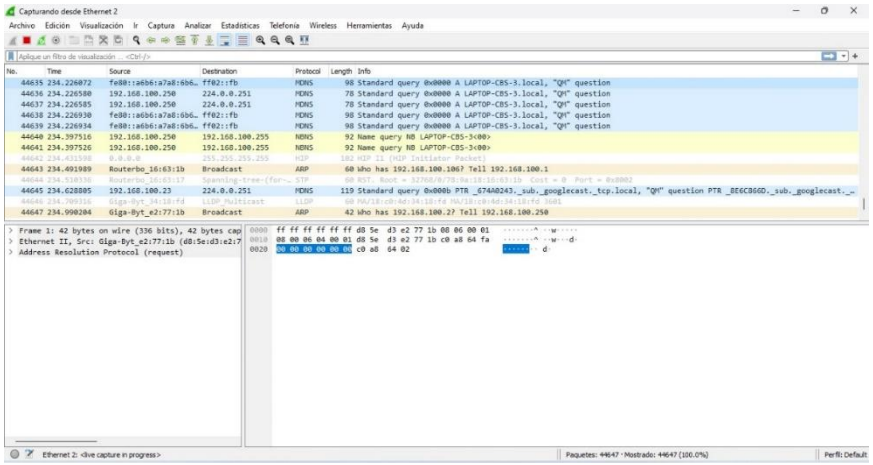


Figura 12 Captura de datos de la red del CBS

Durante la captura de datos, se pudo visualizar diversos tipos de tráfico que se encontraban en la red, el cuales están incluidos los siguientes protocolos comunes:

- MDNS. - Multicast o protocolo Dns que resuelve nombre de host en direcciones ip [50].
- NBNS. - NetBIOS es parte de la suite del protocolo tcp, tiene el mismo propósito del MDNS, pero se puede ejecutar encima de varios protocolos de redes diferentes [51].
- ARP. - Protocolo de resolución de direcciones, el cual lleva a cabo la resolución de direcciones ipv4 [52].
- STP. - Protocolo de red que se utiliza para evitar los bucles que se producen en la red el cual se crean por enlaces redundantes [53].

Al verificar los paquetes se visualizó que era un comportamiento normal, esto quiere decir que no revelo anomalías que pongan en riesgo nuestra red.

3.2.2.3 Ataque de fuerza bruta.

Dentro de las pruebas de vulnerabilidad, uno de los ataques más comunes es de Ataque de Fuerza bruta, el cual consta de tener un diccionario inmenso, en donde almacena la mayor parte de login y passsword más comunes en todo el mundo.

Una vez obtenido el diccionario, el escaneo de puerto, el ataque será lanzado en el puerto 22 “SSH” con la herramienta Hydra que nos brinda Kali Linux (Ver Figura 13,14 y 15)

Figura 13 Lista de datos de Rockyou.txt

Una vez tengamos listo el diccionario de datos, con hydra procederemos a ejecutar el ataque de fuerza bruta, con el siguiente comando:

hydra 1xx.1xxx.1x.xxx ssh -l rockyou.txt -P rockyou.txt -s 22

L: es el login que debe buscar en el diccionario o si en este caso conocemos cuál es podemos remplazar.

P: es la contraseña, como desconocemos esta será buscada en el diccionario rockyou. Esto llevará su tiempo, si la contraseña está ahí nos arrojará el resultado, caso contrario no habrá respuesta

S: el servicio que queremos comprometer para el ataque.

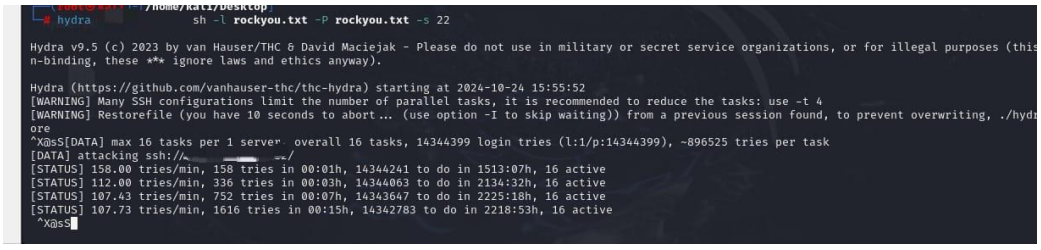


Figura 14 Ataque de fuerza bruta con diccionario.

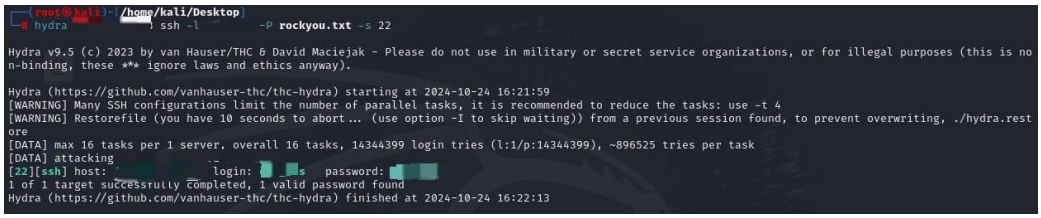


Figura 15 Resultados del ataque de fuerza bruta

3.2.3 Fase III: Implementación de herramientas de Seguridad.

3.2.3.1 Instalación de Ubuntu 20.04.6

Para la implementación del sistema de detección y prevención de intrusos (IDS/IPS), es necesario realizar la instalación del sistema operativo Ubuntu en la versión 20.04.6. La instalación es muy sencilla y requiere de recursos mínimos de hardware, además que su manejo es muy fácil.

A continuación, se detallará la instalación de Ubuntu:

PASO 1: Configuración del idioma y teclado del sistema operativo en el cual vamos a trabajar (Ver Figura 16).

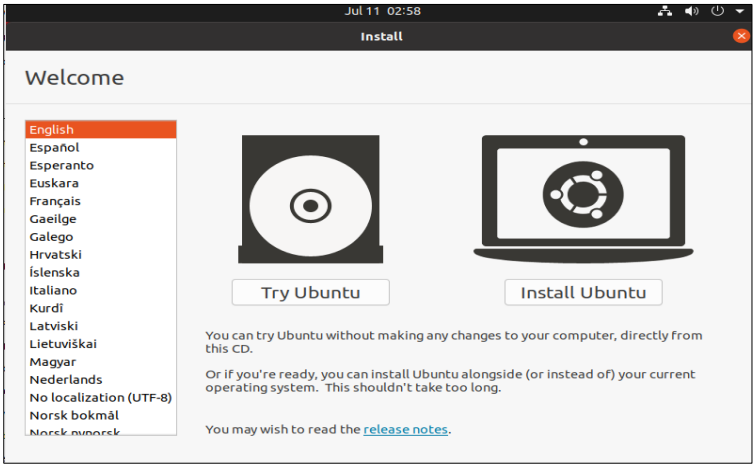


Figura 16 Configuración del idioma y teclado.

PASO 2: En el siguiente paso, se nos pedirá que selecciones el tipo de instalación. Elegiremos la opción “Instalación mínima” (Ver Figura 17)

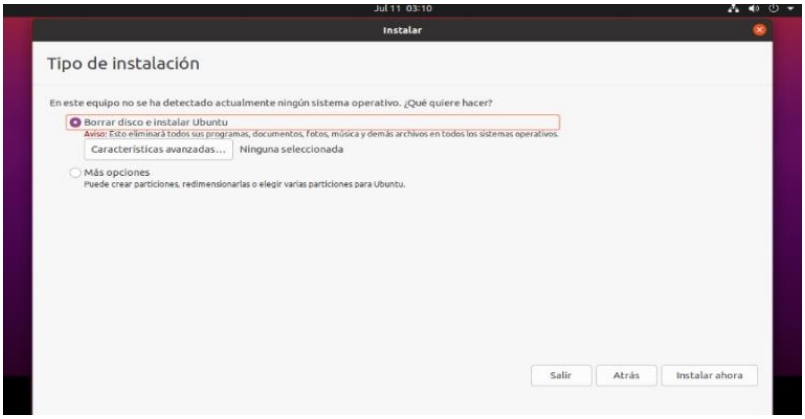


Figura 17 Elección de tipo de instalación

PASO 3: Una vez realizado el paso anterior, debemos elegir en que disco duro se almacenara el sistema, si en caso de que el disco a usar tenga contenido, este lo borrara automáticamente. (Ver Figura 18)

Figura 18 Ubicación del S.O

PASO 4: Es muy importante elegir nuestra zona horaria correctamente, debido a que nos permitirá verificar la hora exacta en la que ocurrió un incidente. (Ver Figura 19).

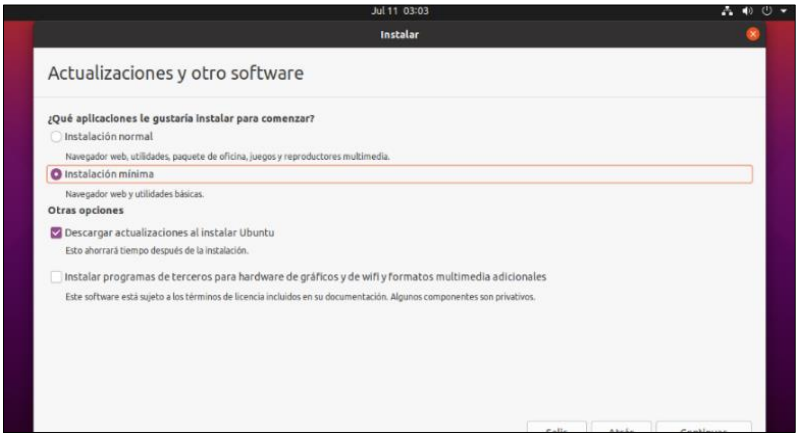


Figura 19 Zona Horaria del S.O

PASO 5: Después de la elegir de la zona horaria, se debe configurar el equipo con el usuarios y contraseña segura. Una vez realizado este paso se comenzará con la instalación (Ver Figura 20 y 21)

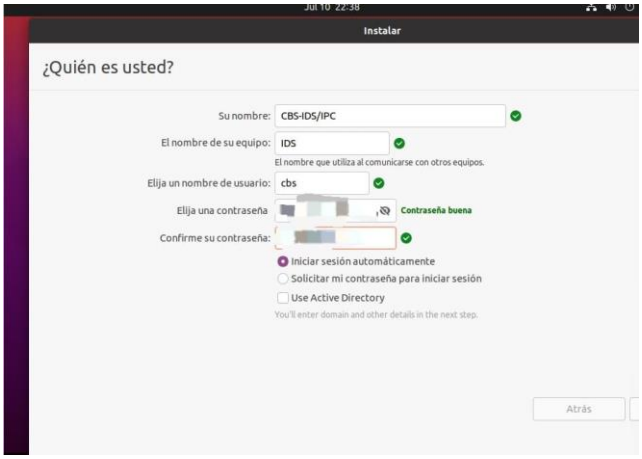


Figura 20 Configuración de usuario y contraseña

Figura 21 Instalación final de Ubuntu.

3.2.3.2 Instalación y configuración del sistema de detección de intruso (Snort)

La información detallada sobre la instalación del sistema de prevención de intruso Snort puede consultarse en el Anexo #4. Para la correcta configuración del motor IDS, se describen los siguientes pasos con un enfoque en las siguientes reglas:

Paso Opcional: Si desea trabajar de una manera remota, debe instalar o habilitar el puerto SSH a través de este comando:

cbs_ids@cbs:~$ sudo apt install openssh-server

Antes de empezar con las configuraciones de las reglas, es fundamental comprender el formato adecuado que se debe seguir para garantizar una buena implementación.

En la estructura de las reglas de Snort se divide netamente en dos partes: **Rule Header y Rule Option** que significa Cabecera y opción de las reglas, para comprender un poco más el Rule Header debemos tomar en cuenta estos datos que se deben ingresar:

- Action (Acción -> Alert, Pass, Drop, Reject).
- Protocol (protocolo).
- Source address (Dirección IP de la fuente).
- Source port (Puerto de la fuente).
- Direction (Dirección).
- Destination address (Dirección de destino).
- Destination port (Puerto de destino).

Ejemplo del Rule Header se vería así:

alert icmp 000.000.0.00 any -> any any

En el encabezado puede variar según el tipo de regla que necesitemos en nuestro sistema de detección de intrusos.

En Rule Option corresponde a que tipo de respuesta debe emitir Snort, siempre y cuando se cumplan las condiciones dadas. Ejemplo:

(msg: "ICMP Attempt Attack"; sid:0000000)

- El comando "msg" es quien estará a cargo de emitir un mensaje de alerta de acuerdo con la regla implementada.

- El texto indicara en pantalla que tipo de ataque o incidente se está registrando.

- El "sid" es el número de identificación que se da para que se registre en el ids Snort.

<acción> <protocolo> <IP-origen> <Puerto-origen> <dirección> <IP-destino> <Puerto-destino> [(<opción-1>; …; <opción-n>;)

Tabla 3 Estructura de reglas de Snort

Paso #1: Una vez se haya realizado la configuración de la interfaz de red (Ver Anexo #4),

se empezará a implementar las reglas a través de este comando que nos redirige a la carpeta Snort, específicamente donde se almacenan las reglas:

root@cbsIDS:/etc/snort# nano /etc/snort/rules/local.rules

Inmediatamente se abrirá una ventana (Ver figura 22), en donde comenzaremos a trabajar en la prevención de intrusos.

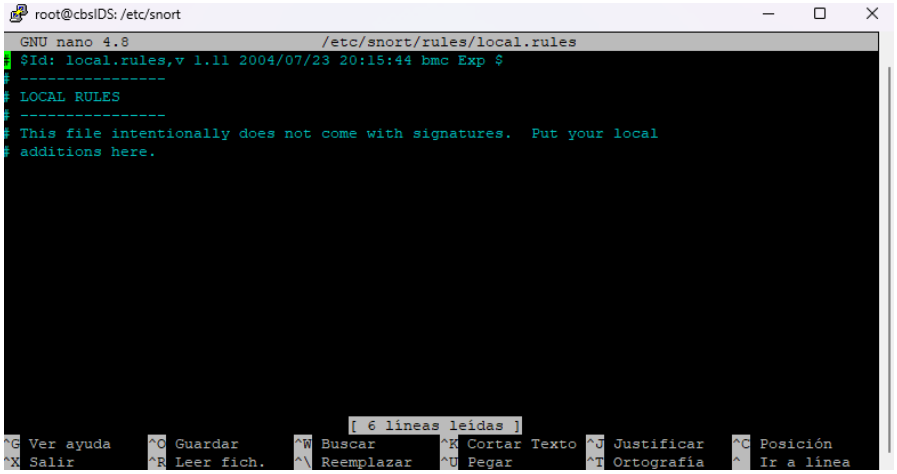


Figura 22 Ventana de configuración de reglas.

Nota: no olvidar que la variable $HOME_NET almacera nuestra dirección ip.

Paso #2: Dentro del archivo local.rules, procederemos a redactar las reglas que permitirán el correcto funcionamiento del motor IDS, siguiendo el formato adecuado para que el sistema detecte y responda a posibles amenazas (Ver Figura 22).

Regla # 1: alert icmp 1x.1x.x.x/xxx any -> any any (msg:"Alguien está haciendo un ping con nuestra red."; sid:19910316; rev:1;)

Esta regla nos alerta que se está haciendo un ping a nuestra red, además nos demostrara de que dirección IP se está haciendo

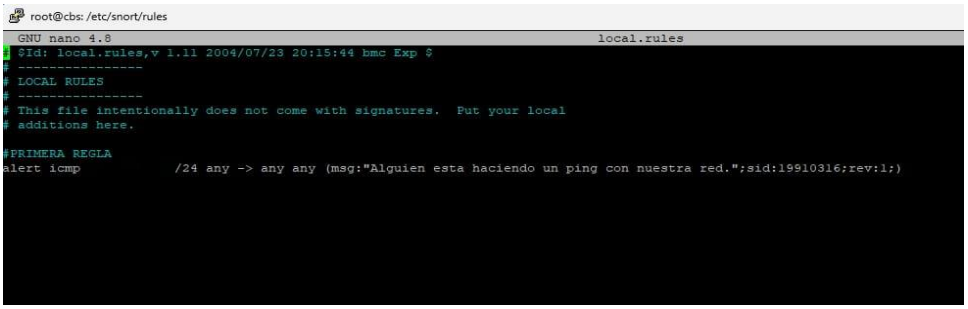


Figura 23 Configuración de la primera regla de Snort

Regla #2: Esta regla tiene como objetivo monitorear el tráfico en el puerto 80, el cual está asociado al servicio HTTP, para detectar posibles actividades sospechosas o no autorizadas.

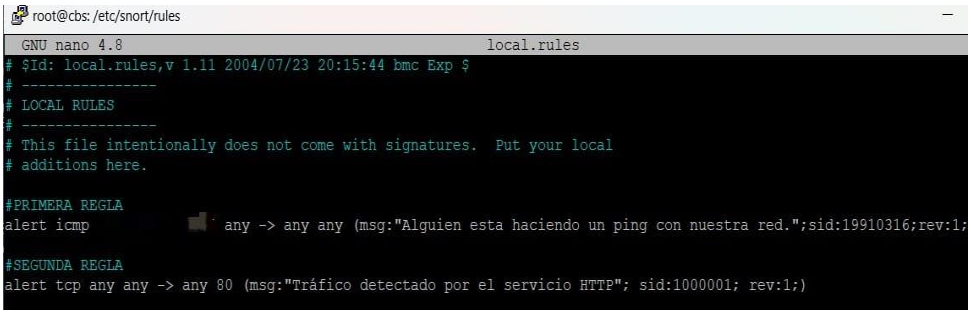


Figura 24 Segunda regla, Servicio HTTP

Regla #3: Esta regla tiene como propósito verificar si existe alguna conexión establecida a través del servicio SSH, para identificar posibles accesos no autorizados o intentos de intrusión.

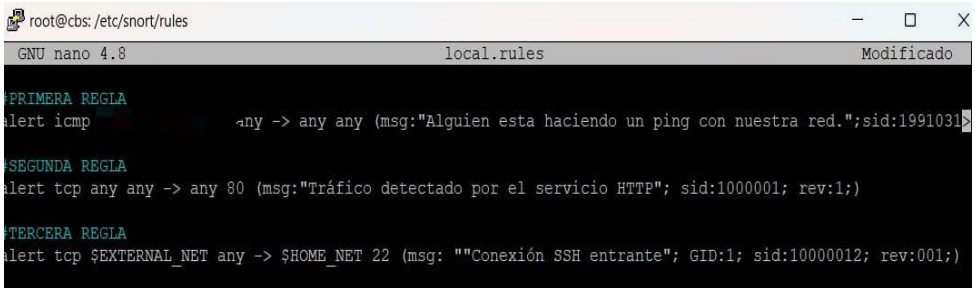


Figura 25 Regla de conexión por SSH

Regla #4: Este conjunto de reglas se utiliza para escanear puertos mediante herramientas como Nmap, con el objetivo de identificar servicios potencialmente vulnerables (Ver Figura 26).

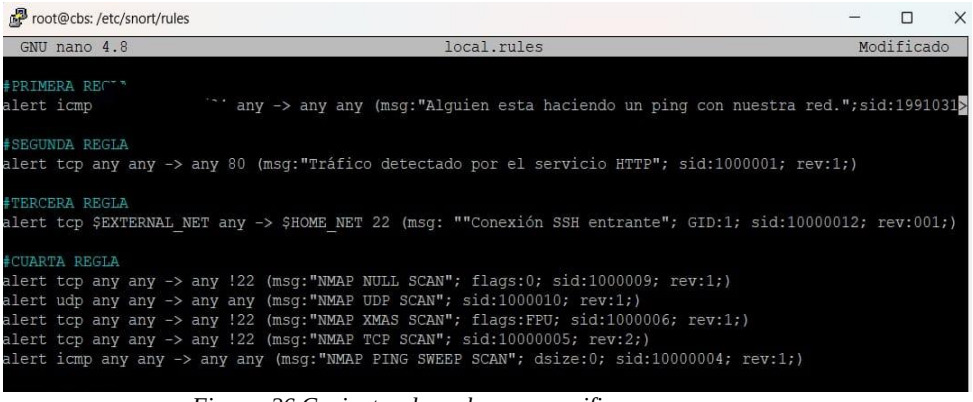


Figura 26 Conjuntos de reglas para verificar escaneo.

Regla 5: Esta regla registra el tráfico anómalo (Bad Traffic) generado en la red, identificando patrones inusuales que podrían indicar intentos de intrusión o actividad maliciosa.

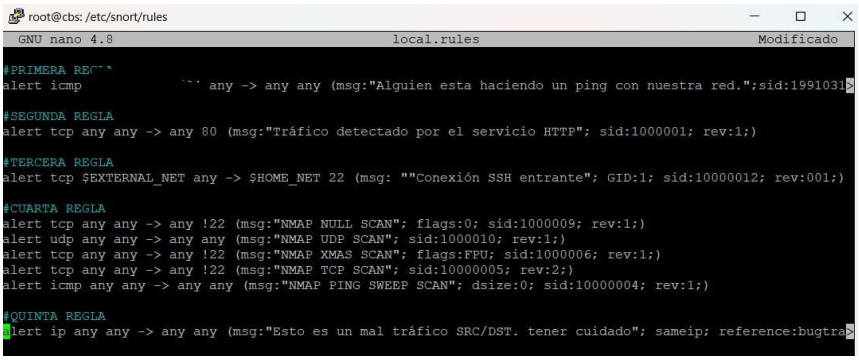


Figura 27 Regla de Bad Traffic

Regla 6: Alerta de posible ataque de denegación de servicio (DoS) detectado en la red. Esta actividad podría comprometer la disponibilidad de los servicios.

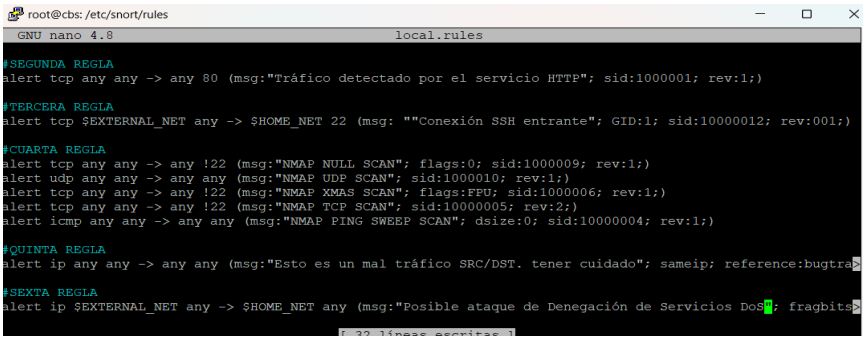


Figura 28 Regla Denegación de Servicios.

Nota: Las siguientes reglas puede consultarla en el Anexo #5, donde encontrara más reglas.

Paso #3: Una vez que se hayan implementado todas las reglas, debemos configurar el siguiente archivo "**Snort.conf**", el cual almacenara la dirección ip de nuestro servidor.

root@cbs:/# nano /etc/snort/snort.conf

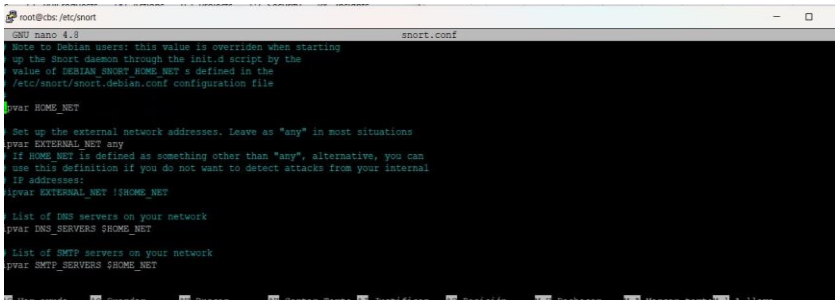


Figura 29 Configuración dirección ip en snort.conf

Una vez nos abra la ventana de Snort.conf, deberemos buscar el apartado **ipvar HOME_NET any,** en esta parte cambiaremos **any** por nuestra dirección ip.

Paso #4: El último paso es dirigirnos a la carpeta Snort, y poner el producción nuestro ids para que muestre las alertas de posibles ataques.

snort -A console -c snort.conf -i enp2s0

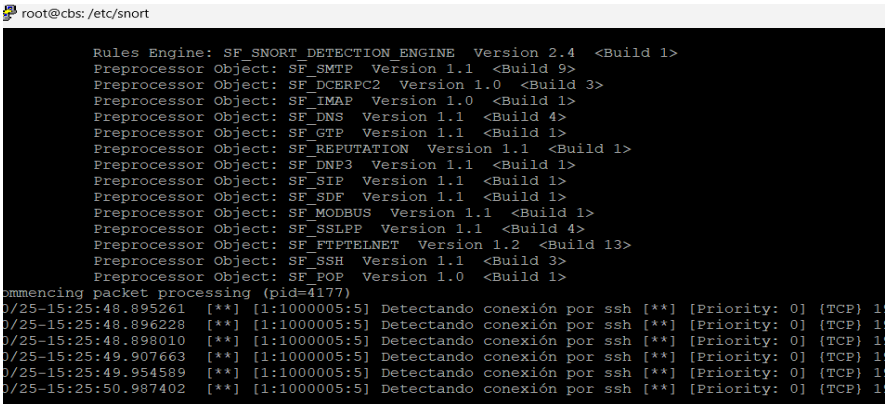


Figura 30 Producción del IDS Snort

Nota: La interfaz de red puede cambiar dependiendo de cada equipo. En este caso nuestra interfaz es **enp2s0**

3.2.3.3 Instalación y configuración del sistema de detección de intruso (Suricata).

La información detallada sobre la instalación del sistema de prevención de intruso Suricata puede consultarse en el Anexo #6. Para la correcta configuración del motor IDS, se describen los siguientes pasos con un enfoque en las siguientes reglas.

Antes de empezar con la configuración, es fundamental comprender el propósito y funcionamiento de las reglas de Suricata y cómo se diferencian de las reglas empleadas en otros IDS previos. Las reglas en Suricata son fragmentos de código que especifican condiciones bajo las cuales el sistema debe generar alertas o prevenir acciones sospechosas.

Las reglas de suricata están compuestas por tres partes principales: **Action, Header y Options**. Esta estructura general para las reglas de este motor es conocida como "Signature", en donde esta tiene una serie de campos que describen las características de las reglas y esta puede ser aplicada en base a las necesidades de la organización.

Su formato es la siguiente:

Action: Es la que determina que suceso se está ejecutando y/o coincide con algunas firmas. Además, suricata generar Alertas (alert), Descartar paquetes (drop), rechazar conexión (Reject) y omitir el tráfico (pass).

Header: Se define que tipo de protocolos se va a utilizar, que dirección ip es la que se protege, cual es el puerto de origen y destino de la dirección de la regla.

Options: Nos proporciona una serie de opciones que se pueden definir en la regla base, como mostrar un mensaje, que parámetros usar, limitar la profundidad de una búsqueda (depth), una distancia que se especifica la última coincidencia de contenido.

<acción> <protocolo> <IP-origen> <Puerto-origen> <dirección> <IP-destino> <Puerto-destino> (<opción-1>; <opción-2>; …; <opción-n>;)

Tabla 4 Estructura de regla Suricata.

Un ejemplo más claro sobre la estructura de suricata es la siguiente:

ACTION | HEADER | OPTIONS

alert tcp any any -> any any (msg:" Mensaje de Prueba"; flow:stateless; flags:S; recon; sid:#; priority:#; rev:#;)

- A**lert:** Se genera una alerta si se cumple la condición que se está ejecutando.
- **Tcp:** es el protocolo que se utiliza para verificar si es el ataque con el protocolo mencionado.
- **any any -> any any:** Definición de las direcciones IP, puertos de origen y destino en el cual se buscarán los paquetes.
- **Msg** “ ”: definirá el mensaje a mostrar de acuerdo con la regla estructurada.
- **flow:stateless**: Nos indica que la regla implementada aplica a flujos sin estado , no espera el seguimiento de tcp.
- F**lags:S**: Buscará paquetes que hayan establecido el bit SYN (S) activo.
- **sid**:#: Identificador único de la regla establecida.
- **priority**:5#: Prioridad asignada a la regla, en una escala del 1 al 5.
- **rev**:#: Número de revisión de la regla establecida**.**

Una vez que tenemos claro la estructura de las reglas vamos a configurar Suricata para que funcione como IDS/IPS, para esto seguiremos los siguientes pasos como IDS:

Paso #1: Para configurar Suricata, se debe crear un archivo de reglas denominado “my”. Este archivo es fundamental en la detección de intrusiones, y aunque el nombre puede variar, debe conservar la extensión .rules al final para que el sistema lo reconozca adecuadamente como un archivo de regla.

root@cbsIDS:/etc/suricata# touch /etc/suricata/rules/my.rules

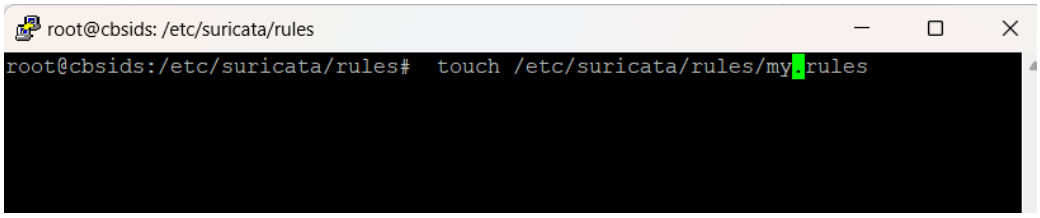


Figura 31 Creación del fichero de reglas de Suricata.

Paso #2: Dentro del archivo creado “my.rules” vamos a configurar las alertas de nuestro motor, cabe recalcar que este funcionara como detector y prevención de intrusos. Con el siguiente comando vamos a modificar el archivo.

root@cbsids:/etc/suricata/rules$ nano my.rules

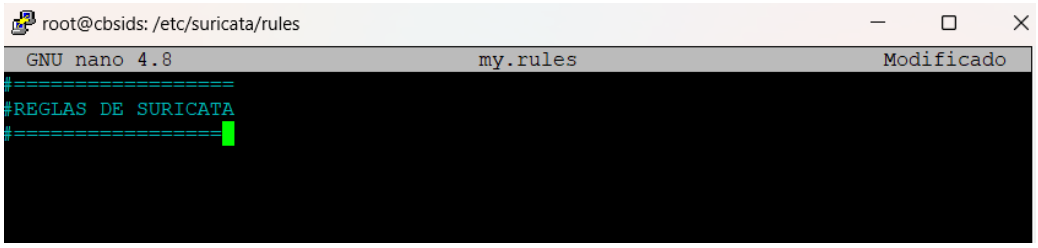


Figura 32 Archivo my.rules de Suricata.

Regla #1: Esta regla incluirá tres subreglas adicionales, ya que su propósito es detectar cualquier solicitud de ping y, además, identificar el sistema operativo desde el cual se realiza. Esto proporcionará información valiosa sobre el origen del posible ataque.

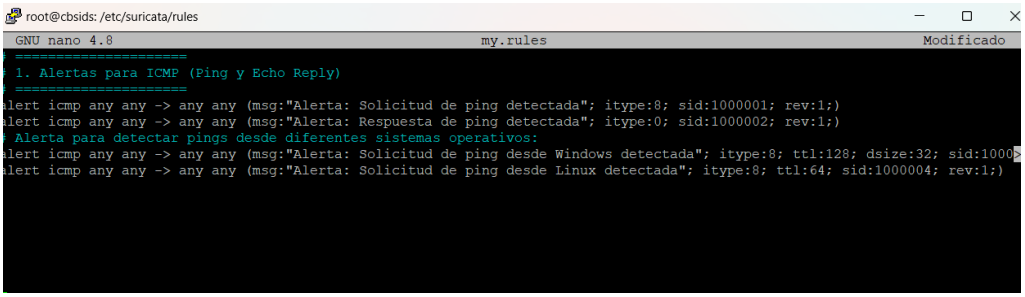


Figura 33 Reglas para la detección ICMP (Ping).

Nota: Las diferentes reglas se mostrarán de acuerdo con el sistema operativo en el cual se realizó la solicitud de ping.

Regla #2: La segunda regla tiene como objetivo verificar si existe alguna conexión externa no autorizada a través del servicio SSH.

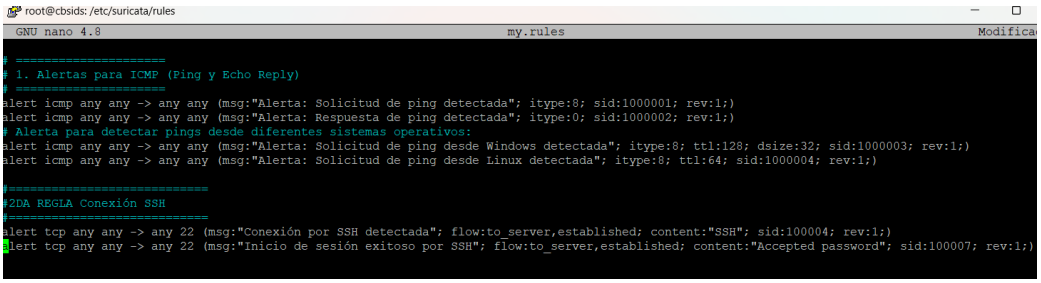


Figura 34 Regla detección de conexiones ssh

Regla #3: Esta regla tiene como objetivo identificar posibles ataques SYN flood, en el que el atacante envía repetidamente paquetes SYN para agotar los recursos del servidor, generando una sobrecarga que puede hacer que el servidor no pueda responder correctamente.

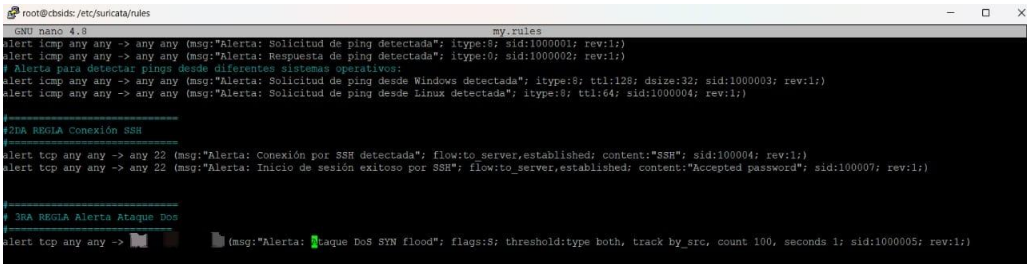


Figura 35 Regla De Ataque de Denegación de Servicios (DoS).

Regla #4: Esta regla está diseñada para detectar el tráfico HTTP legítimo, buscando paquetes TCP en el puerto 80 con los Flags **SYN** y **ACK**. La presencia de estos flags indica el establecimiento de una conexión TCP normal, que es común en las solicitudes HTTP.

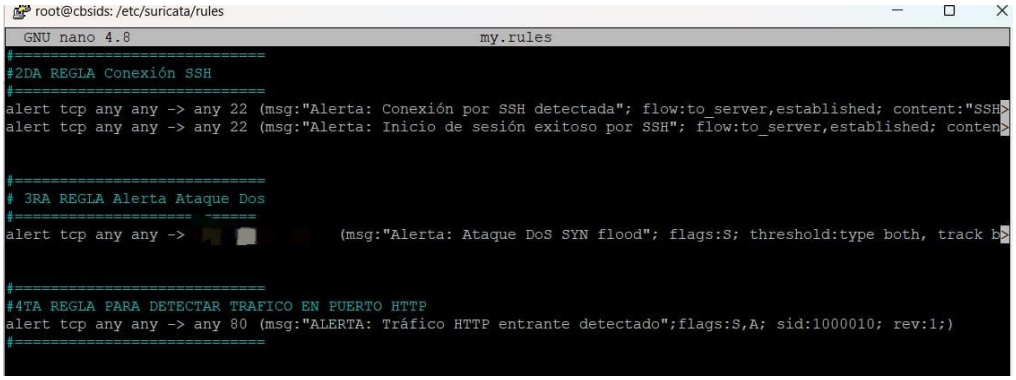


Figura 36 Regla de Tráfico normal por HTTP

Regla #5: Para utilizar un monitor en tiempo real, configuraremos una dirección IP y puerto específicos para su funcionamiento. Además, es importante implementar alertas que detecten tanto el tráfico habitual como posibles ataques DoS dirigidos a esta dirección y puerto.

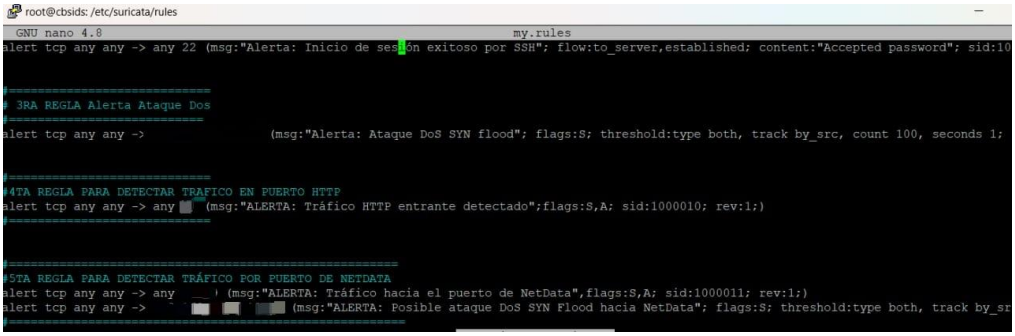


Figura 37 Regla De ataque DoS por puerto de Netdata

Regla #6: Una de las reglas más importantes es la relacionada con las conexiones FTP, por lo que es fundamental implementar una regla que nos genere una alerta al detectarlas.

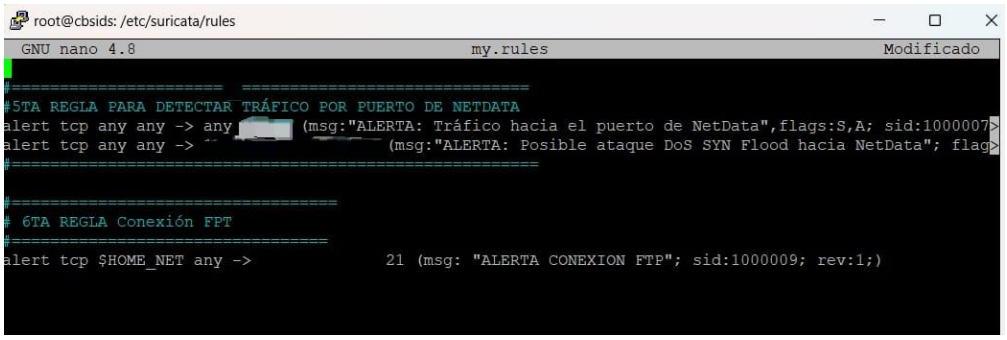


Figura 38 Regla de Alerta por FTP.

Regla #7: Es crucial estar atentos a los escaneos de puertos en nuestra red, ya que obtener esta información nos hace vulnerables a posibles ataques o al robo de datos. Por esta razón, implementamos la siguiente regla**.**

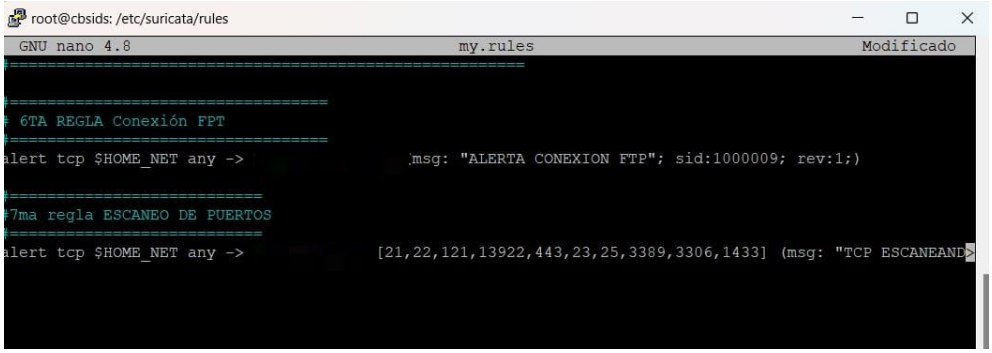


Figura 39 Regla para el escaneo de puertos.

REGLAS DE IPS SURICATA

Nota: Para habilitar el modo IPS de Suricata, realizaremos la configuración a través del siguiente comando. Es fundamental evitar posibles conflictos con las reglas ya existentes, por eso crearemos y trabajaremos en un archivo nuevo llamado“my2.rules”. En este nuevo archivo se guardarán las nuevas reglas de IPS.

Para que suricata funcione en modo IPS, deberemos ejecutar el siguiente comando que se relaciona con la iptables, el cual redirecciona el tráfico ICMP hacia una cola de Netfilter (NFQUEUE).

sudo iptables -I INPUT -p icmp -j NFQUEUE --queue-num 0

Es necesario actualizar el módulo nfnetlink_queue, esté habilitará la comunicación entre el iptables y el sistema de colas de Netfilter. Esto se logra con el siguiente comando:

sudo modprobe nfnetlink_queue

Regla #1: En esta primera regla, nos enfocaremos en bloquear los paquetes ICMP (ping), implementando una medida a través de la acción “drop” que permita bloquear este tipo de tráfico, tanto si proviene de sistemas Windows como Linux.

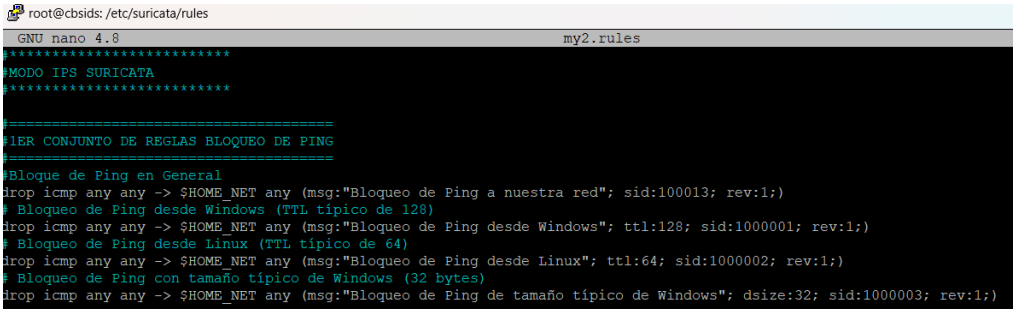


Figura 40 Conjuntos de reglas de bloqueo de Ping

Regla #2: Se configurarán reglas que bloqueen ataques DoS. Estas reglas estarán enfocadas en los protocolos SYN, ICMP y UDP. Los ataques basados en SYN intentan saturar los puertos con conexiones incompletas. Los ataques ICMP y UDP pueden sobrecargar la red o los servicios. Con estas medidas, se mejorará la protección contra este tipo de tráfico malicioso (Ver Figura 41).

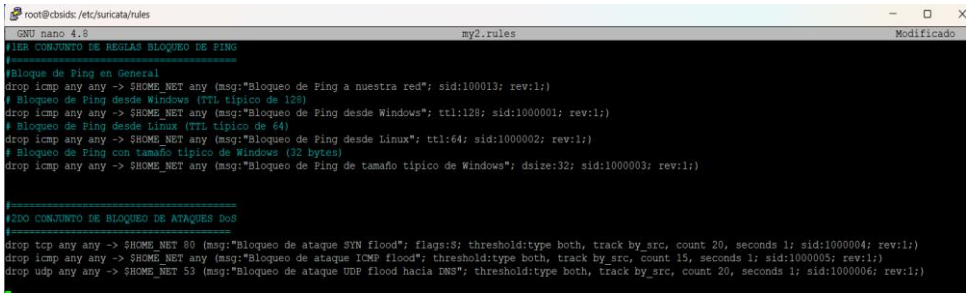


Figura 41 Conjunto de reglas para ataque DoS.

Regla #3: Para mejorar la supervisión de ataques DoS, configuraremos Netdata para monitorear el tráfico en tiempo real. Netdata detectará patrones de ataques SYN, ICMP y UDP, alertando sobre cualquier anomalía. Esto permitirá identificar rápidamente posibles intentos de saturación de servicios. Además, proporcionará métricas detalladas sobre el uso de recursos durante los ataques. Con esta configuración, se optimiza la detección y respuesta ante amenazas."

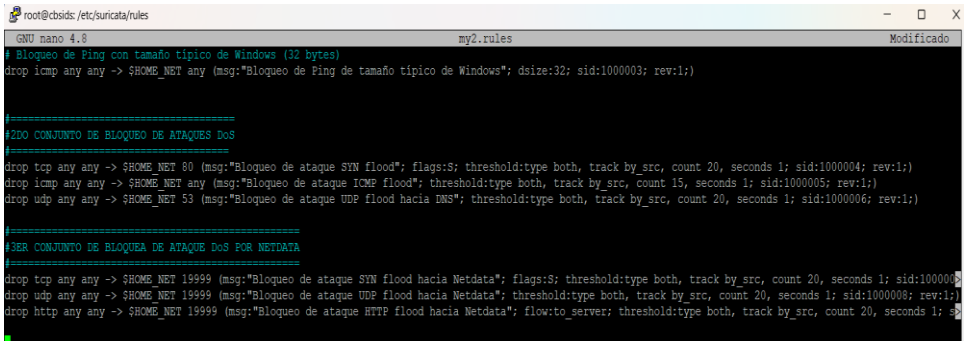


Figura 42 Conjunto de regla para ataque en NetData.

Regla #4: Es fundamental bloquear los accesos no autorizados en las conexiones remotas, ya sea por SSH o FTP, para proteger la información sensible de la institución y prevenir posibles riesgos de seguridad**.**

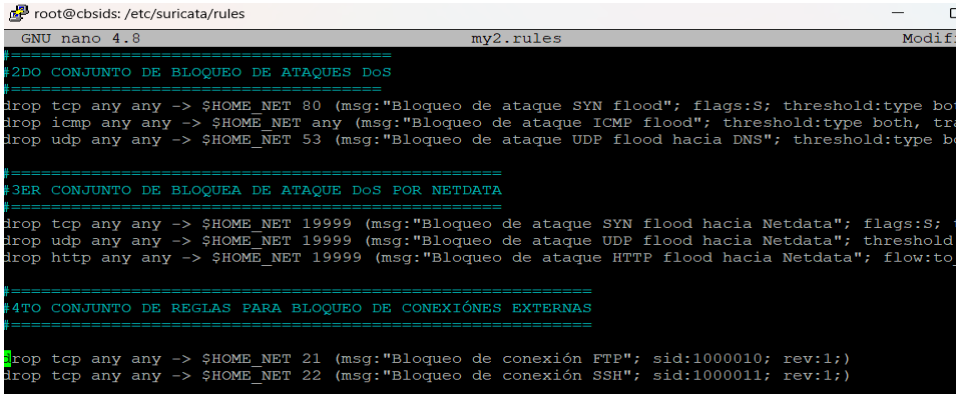


Figura 43 Conjunto de Reglas de bloques de servicios.

Regla #5: El bloque de accesos no autorizado en conexiones remota es importante, pero debemos tener en cuenta que el personal de TIC´S debe tener acceso a las misma para realizar algún tipo de trabajo, mantenimiento o revisar posibles ataques. Por esta razón es importante permitir el acceso únicamente a direcciones IP autorizadas.

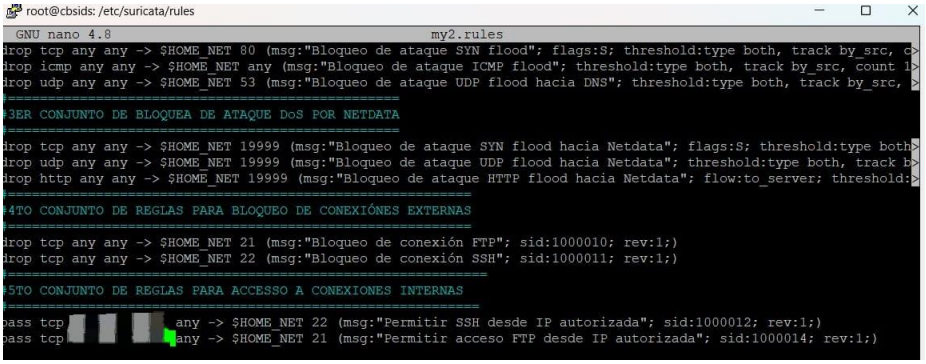


Figura 44 Conjunto de reglas para acceso autorizados de conexiones externas.

Nota: Todas estas reglas las podrán visualizar en el Anexo #7.

3.2.3.4 Instalación y configuración del monitoreo en Tiempo real (NetData).

El monitoreo en tiempo real es muy fundamental, está diseñada para proporcionar una visualización detallada y comprensible del rendimiento de nuestra red, esta es una herramienta muy útil acompañado de los IDS.

Paso #1: Abrimos el terminal para comenzar con la instalación y ejecutamos el siguiente comando (ver Figura #)45:

root@cbsIDS:/home/cbs_ids# apt-get install netdata

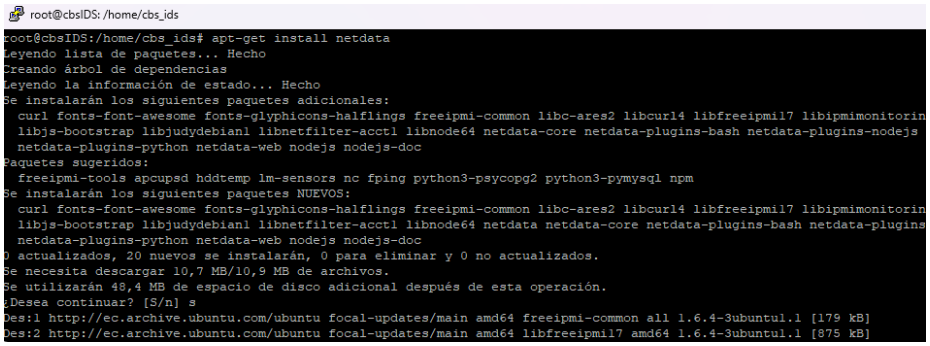


Figura 45 Instalación de Netdata

Paso #2: Configuramos el archivo de NetData el cual debe tener la dirección ip de nuestro servidor para habilitar el monitoreo en tiempo real. Esto nos permite verificar el estado del sistema a través de la interfaz web de NetData, utilizando el siguiente comando:

root@cbsIDS:/home/cbs_ids# nano /etc/netdata/netdata.conf

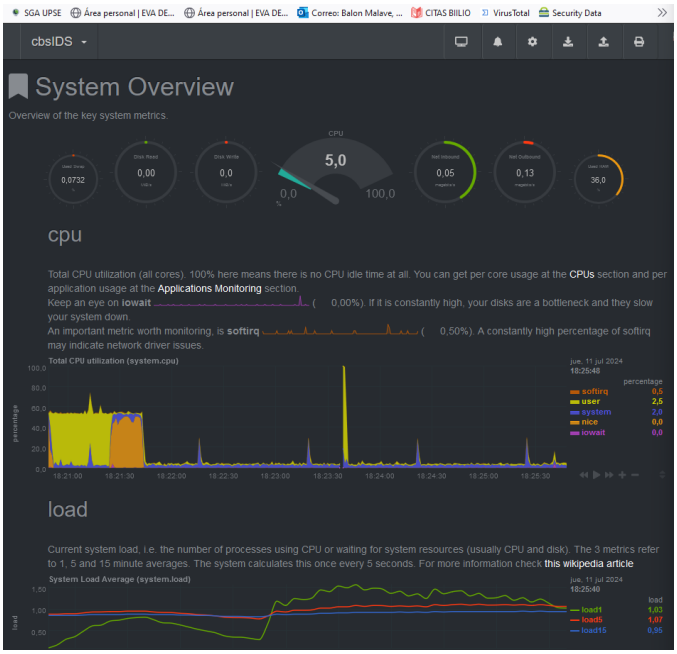


Figura 46 Configuración de dirección IP de NetData

Paso #3: A través de la dirección IP asignada, podemos acceder al monitoreo en tiempo real desde un navegador, lo que nos permite visualizar el estado del sistema de manera interactiva (ver Figura #47).

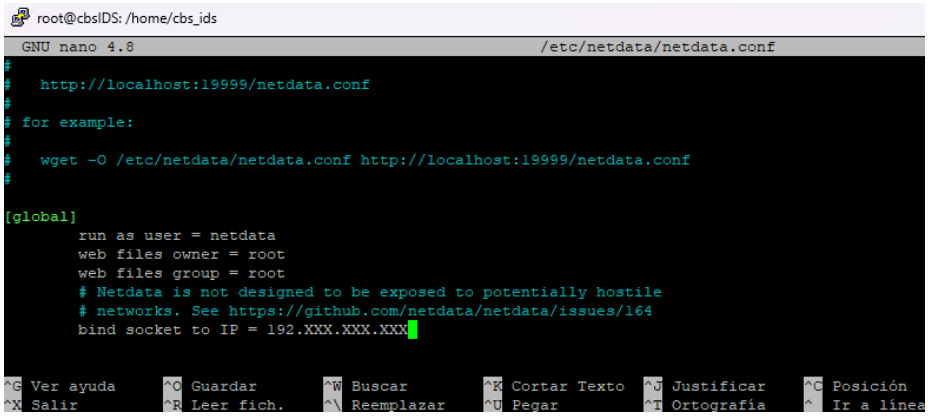


Figura 47 Configuración de dirección IP para monitoreo.

Una vez completados los pasos para configurar la interfaz web de nuestro monitor en tiempo real, es momento de integrar el motor del sistema de detección de intrusos, que en este caso será Suricata. Es fundamental que, para garantizar una vigilancia eficiente del tráfico de red.

A continuación, se describen en detalle los pasos necesarios para enlazar Suricata, asegurando una integración óptima y aprovechando al máximo sus capacidades de análisis y detección.

Paso #4: Primero, hay que asegurar de que Suricata esté configurado para generar logs en el archivo fast.log, para verificar vamos a usar el siguiente comando:

root@cbsids:/etc/suricata# nano /etc/suricata/suricata.yaml

dentro de ese archivo, buscaremos Outputs y añadiremos esto:

```
# Extensible Event Format (nicknamed EVE) event log in JSON format
- eve-log:
   # enabled: yes
   # filetype: regular #regular|syslog|unix_dgram|unix_stream|redis
   # filename: eve.json
    enabled: yes
    filetype: regular
    filename: /var/log/suricata/eve.json
    types:
      - alert:
          enabled: yes
```

Figura 48 Enlace de Suricata con Netdata

|**Paso #5:** una vez hecho la configuración anterior, vamos al directorio de NetData a realizar la última configuración en el archivo log.conf.

root@cbsids:# cd /etc/netdata/go.d

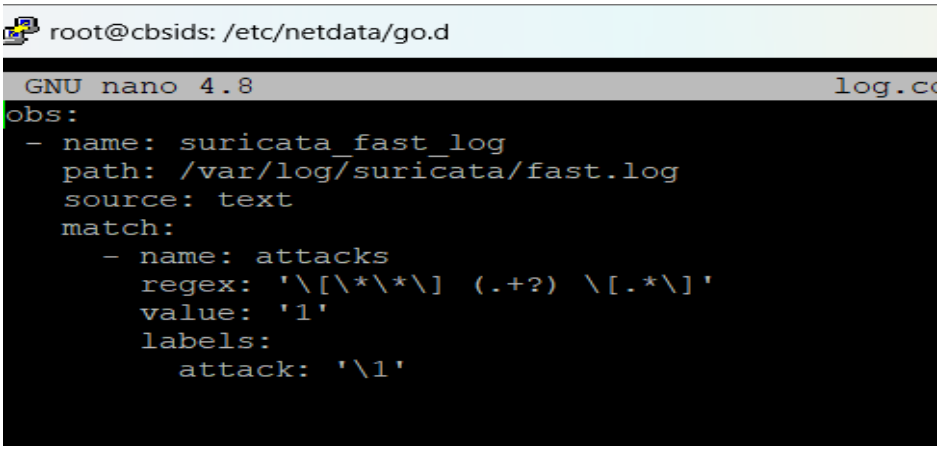


Figura 49 Configuración del archivo log.conf (NetData)

Como último paso reiniciamos el servicio de NetData y este nos mostrara las alertas de cuando se ejecute un ataque.

sudo systemctl restart netdata

sudo journalctl -u netdata | grep suricata (Integramos ambos motores)**.**

3.2.4 Fase IV: Pruebas de Seguridad y Validación de la Implementación

3.2.4.1 Pruebas de validación de implementación de sistema de prevención de intrusos Snort.

Prueba #1: Para realizar las pruebas con el sistema de detección de intrusos debemos poner en escucha nuestra herramienta, por lo consiguiente debemos abrir la consola a través del siguiente comando:

root@cbsIDS:/etc/snort# snort -A console -c snort.conf -i enp2s0

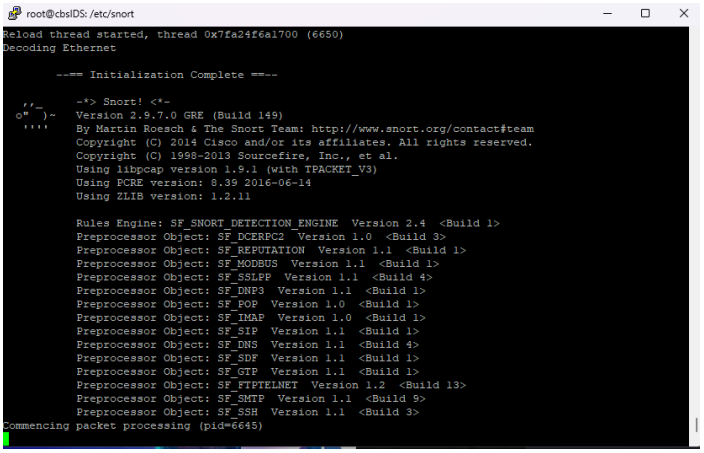


Figura 50 IDS Snort en producción

Prueba #2 Ejecutaremos la detección de ping con la ayuda de las reglas configuradas, este debe mostrar la ip de la máquina, la ip de al atacante, el mensaje predeterminado, hora y fecha del ataque

- Realizamos el ping a la dirección ip del servidor.

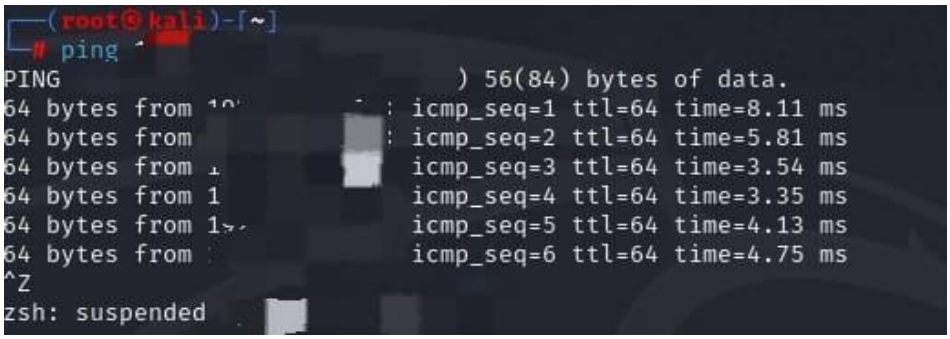


Figura 51 Ping desde Máquina Linux

- Si se realiza un ping de un sistema operativo diferente a Windows podremos diferencia porque nos mostrara un ping *NIX, mientras que si es de Windows nos dirá.

Figura 52 IDS Snort en producción, alerta de ping desde Linux

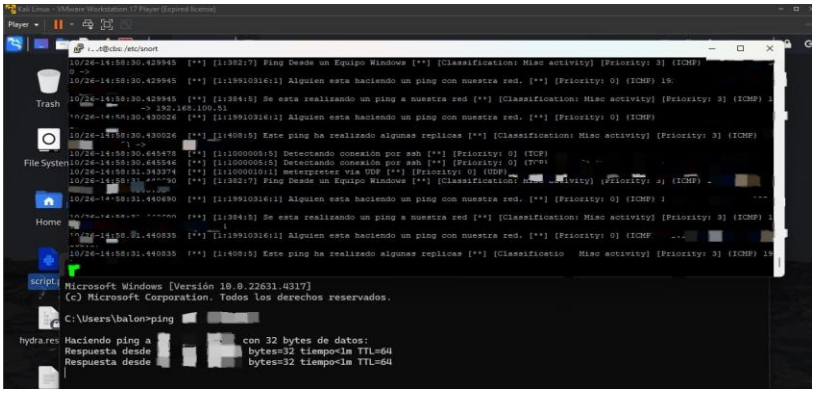


Figura 53 IDS Snort en producción, alerta ping desde Windows

Prueba #3: Para verificar esta regla realizaremos un escaneo de red utilizando Nmap desde Kali Linux.

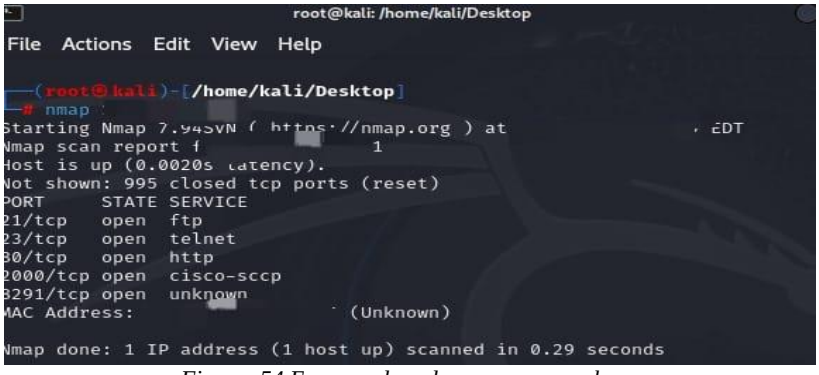


Figura 54 Escaneo de red con nmap, pruebas

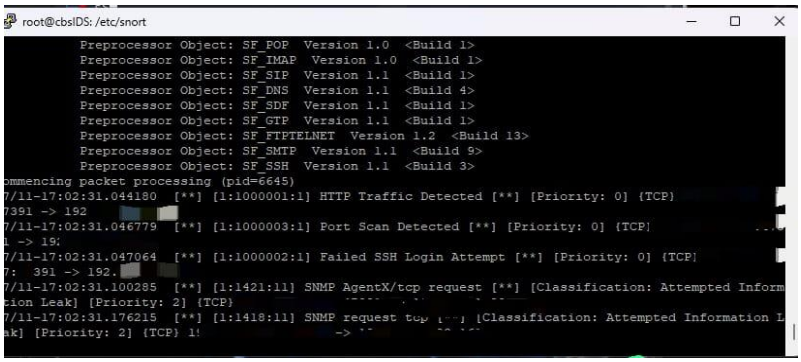


Figura 55 IDS en producción, detección de escaneo de red.

Prueba #4 : Procederemos a verificar la actividad de posibles conexiones SSH no autorizadas para asegurar la integridad y seguridad de los accesos al sistema. Esta verificación es crucial para detectar intentos de acceso no autorizado mediante SHH (Ver Figura 54).

- Intentamos conectar a Servidor mediante putty, intentaremos realizar accesos incorrectos.

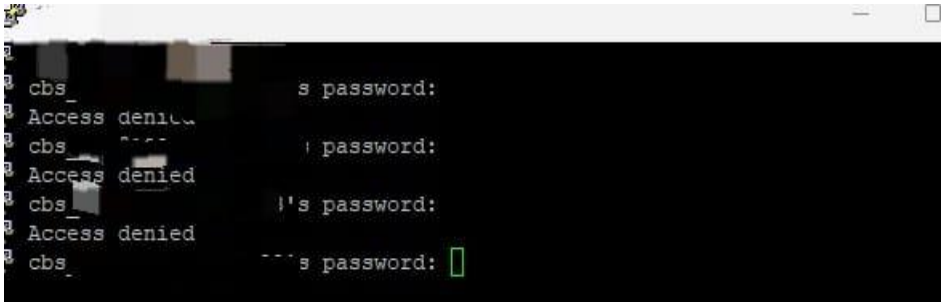


Figura 56 Intentos de acceso incorrecto por SSH

- Realizaremos la verificación de las alertas generadas por el IDS implementado, analizando tanto los intentos de conexión fallidos como los exitosos

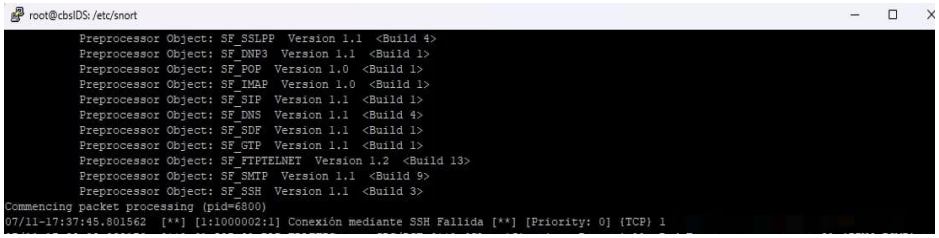


Figura 57 IDS Snort en producción, intento fallidos por SSH

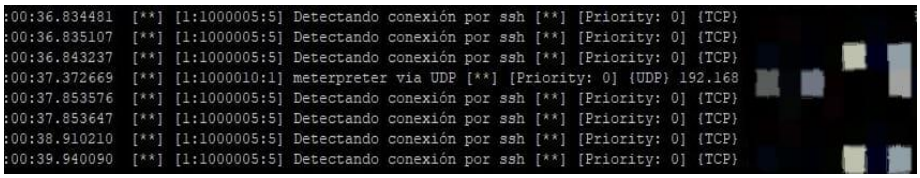


Figura 58 IDS Snort en producción, intento exitoso por SSH

Prueba #5: Esta regla nos permite ver si estamos siendo atacados en la red o específicamente en algún puerto de nuestro servidor.

- Desde Kali Linux, realizaremos un ataque de denegación de servicio utilizando la herramienta Hydra para llevar a cabo un ataque de fuerza bruta. Este ataque generará múltiples intentos de conexión a un servicio específico (como SSH) en un corto periodo de tiempo, simulando una sobrecarga en el sistema objetivo.

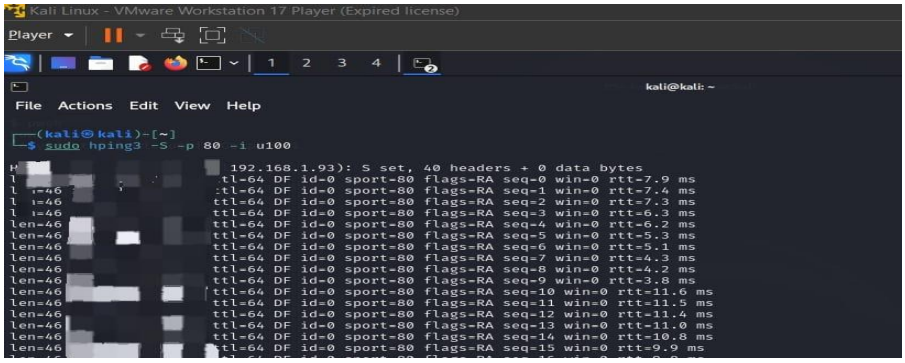


Figura 59 Ataque de Denegación de Servicios.

- La generación de alertas ante ataques de Denegación de Servicio (DoS) permite identificar rápidamente intentos de sobrecargar el servidor, asegurando su disponibilidad

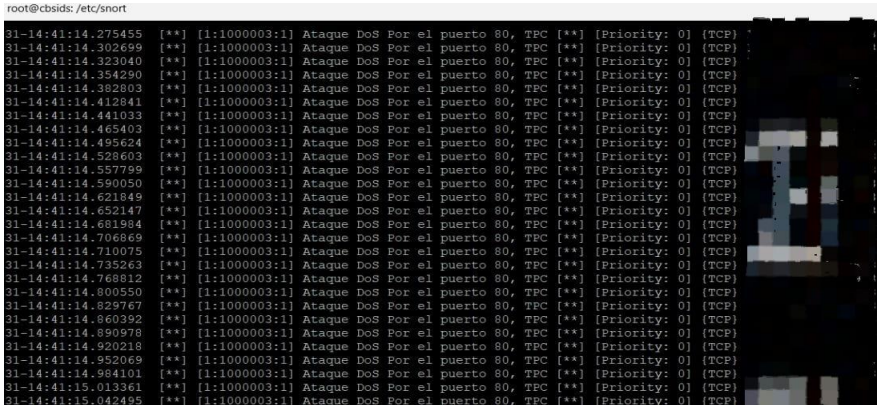


Figura 60 IDS Snort en producción, alerta de ataque DoS.

Regla #6: Está diseñada para detectar y registrar intentos de conexión al protocolo FTP. Esta alerta se activará cada vez que se identifique una conexión establecida en el puerto FTP (puerto 21), permitiendo monitorear accesos no autorizados o actividad sospechosa en este servicio (Ver Figura 61 y 62).

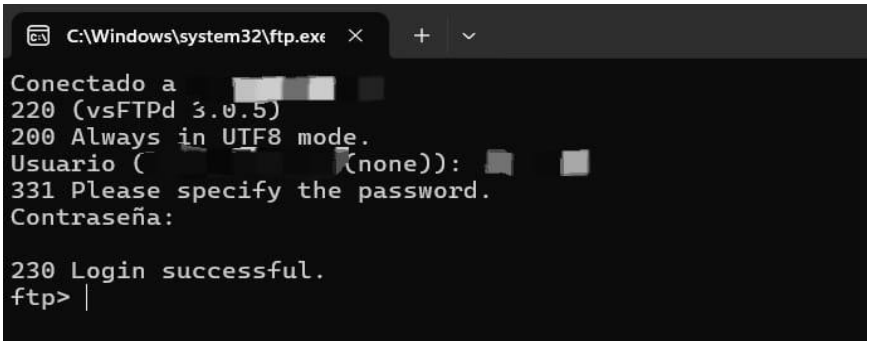


Figura 61 Intento de conexión por FTP

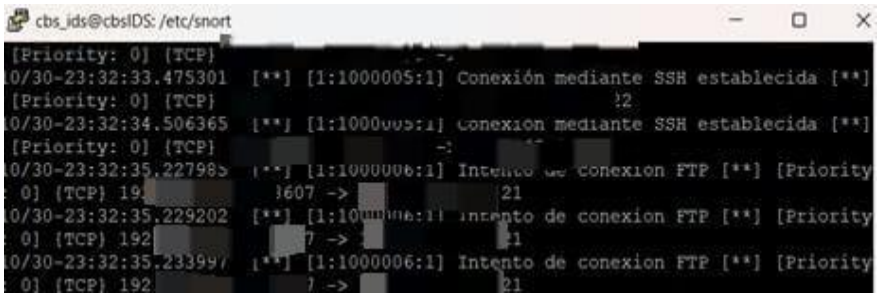


Figura 62 IDS Snort en producción, Generación de alertas por intento de conexión FTP.

3.2.4.2 Pruebas de Validación de Implementación de sistema de prevención de intrusos Suricata.

Se realizará las pruebas de validación en un entorno controlado, esta será en el modo de IDS.

Prueba #1: Este conjunto de reglas está diseñado principalmente para detectar paquetes ICMP, lo que permite identificar los paquetes conocidos como *ping* en la red. Además, proporcionará información sobre el sistema operativo desde el cual se originan.

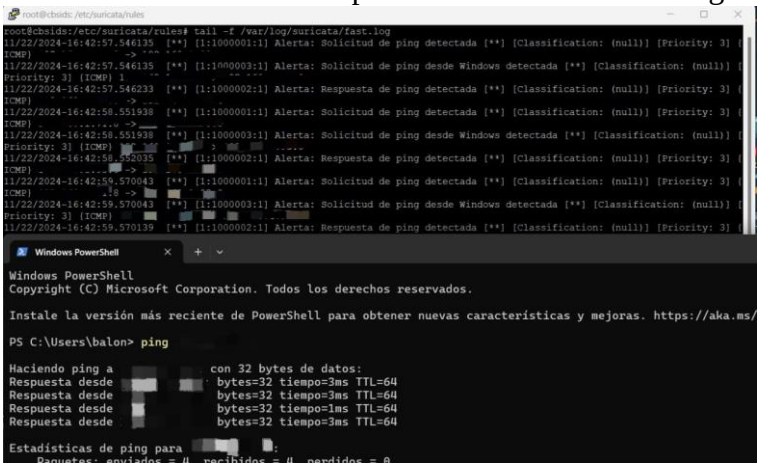


Figura 63 IDS Suricata en producción, Detección de Ping, Windows

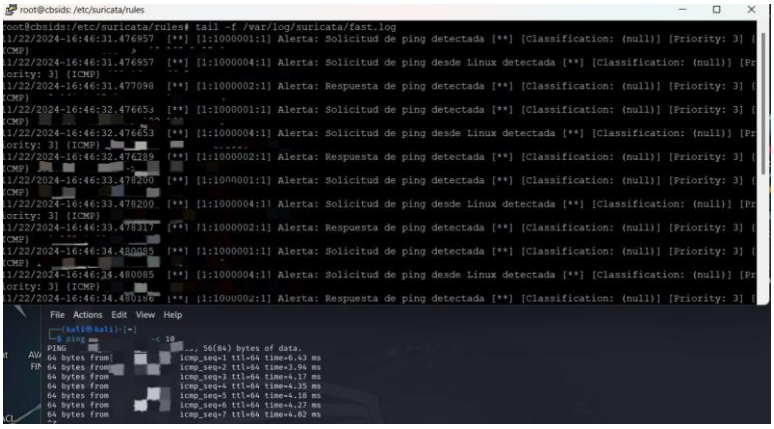


Figura 64 IDS Suricata en producción, Detección de Ping, Linux

Regla #2: También a comparación de la regla de Snort, en suricata también optaremos por la detección de conexiones ssh, en este caso tuvimos un tráfico real en nuestra página web desde una dirección ip no autorizada en el entorno controlado.

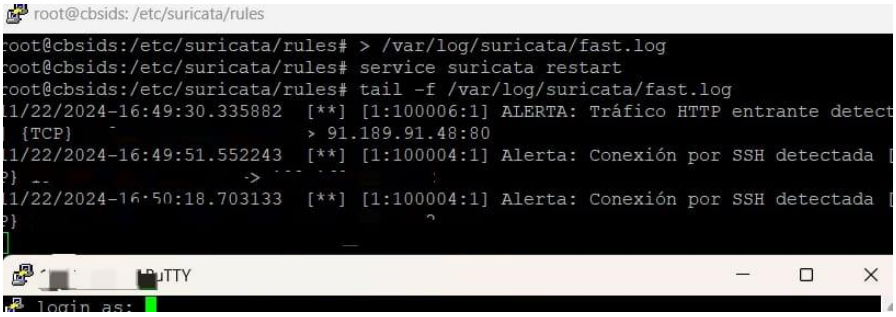


Figura 65 IDS Suricata en producción, Detección de conexiones SSH.

Regla #3: Un ataque común que afecta a diversos sistemas es la Denegación de Servicios. Este tipo de ataque puede generar confusión al analizar el tráfico de red, resulta complicado diferenciar entre un ataque y tráfico legítimo entrante. Por ello, se decidió redactar cuidadosamente las reglas de detección para minimizar los falsos positivos.

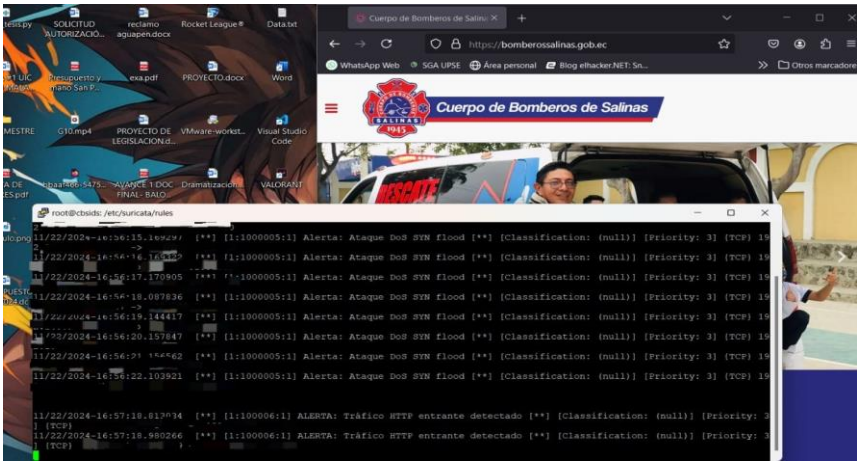


Figura 66 IDS Suricata en producción, , Tráfico legitimo en 80.

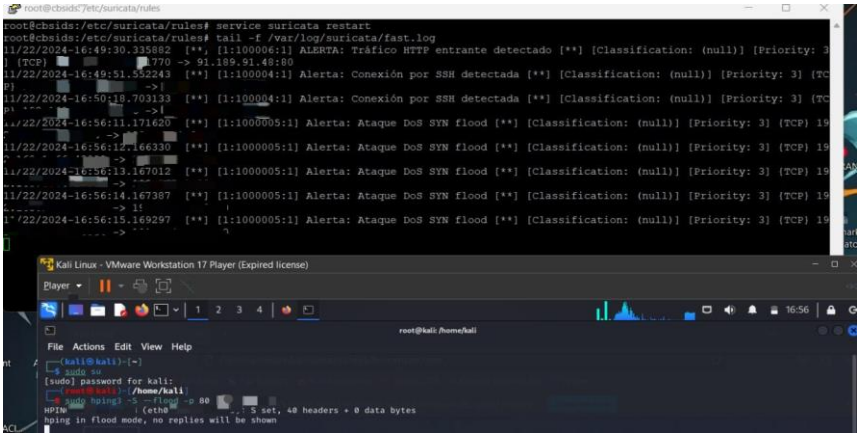


Figura 67 IDS Suricata en producción, Detección de ataque DoS Puerto 80.

Regla #3: Un ataque común que afecta a diversos sistemas es la Denegación de Servicios en nuestro sistema de monitoreo en tiempo real. Este mismo tipo de ataque también puede generar confusión al analizar el tráfico de red real con un ataque.

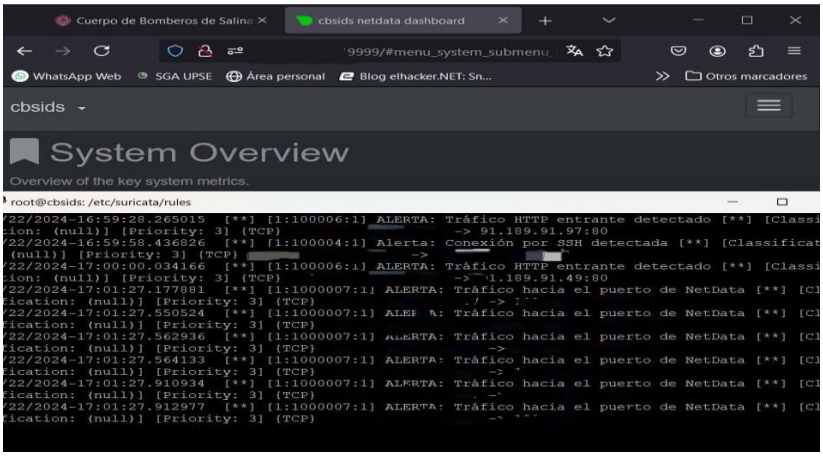


Figura 68 IDS Suricata en producción, Tráfico legitimo en Puerto Netdata.

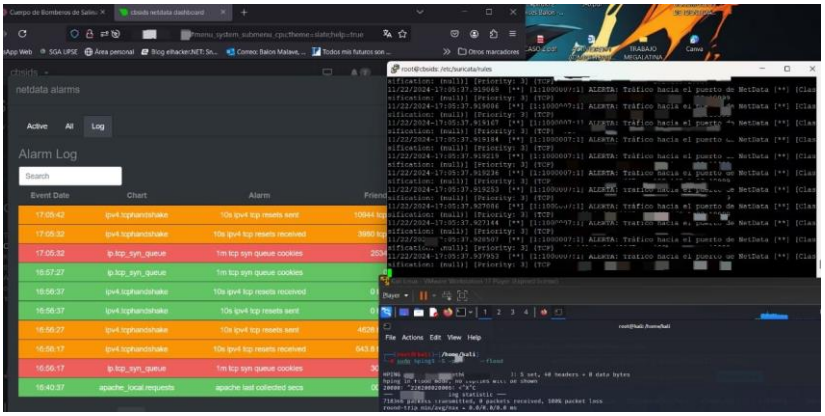


Figura 69 IDS Suricata en producción, Ataque DoS en Puerto Netdata.

En este ataque podemos visualizar como realmente nos alerta nuestro monitor en tiempo real y queda registrado en los logs el ataque que se está realizando.

Regla #4: La alerta de conexión FTP es importante, ya que permite identificar y verificar si se están llevando a cabo intentos de acceso no autorizados. Esto contribuye prevenir posibles vulneraciones al sistema.

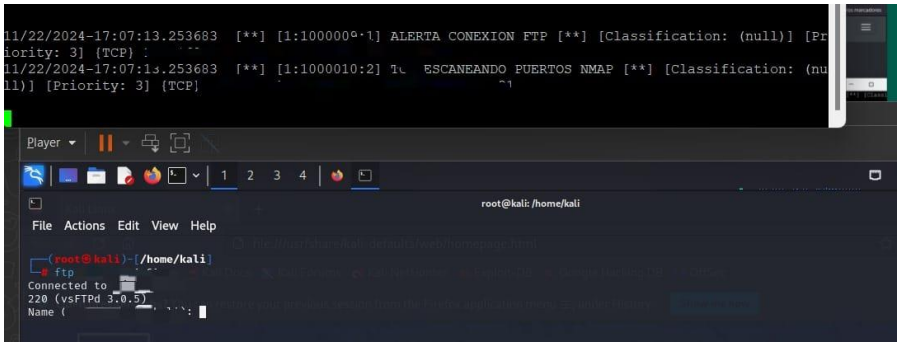


Figura 70 IDS Suricata en producción, Intento de conexión FTP, Linux.

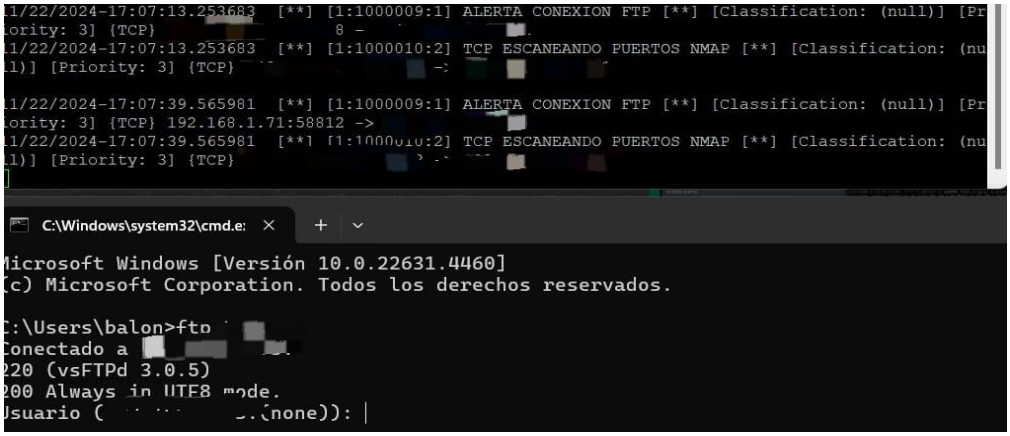


Figura 71 Suricata en producción, Intento de conexión FTP, Windows.

Prueba controlada en modo de prevención de intrusos (Bloqueos y acceso autorizado)

Prueba #1: La primera regla incluye un conjunto de subreglas que nos permitirá bloquear los paquetes ICMP (ping) dirigidos a la dirección IP de nuestro servidor, previniendo ataques de denegación de servicio a través de este protocolo. Además, se configurará para verificar el sistema operativo desde el cual se han enviado las solicitudes.

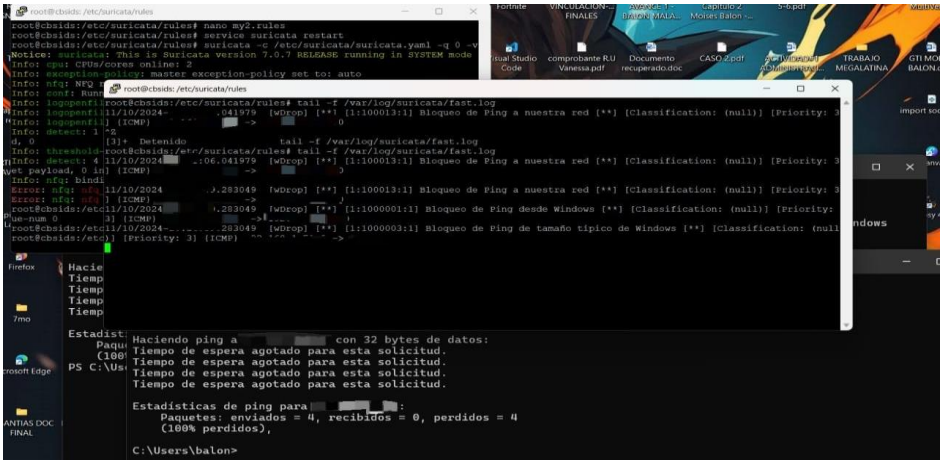


Figura 72 IPS Suricata en producción, bloqueo de ping general de Windows.

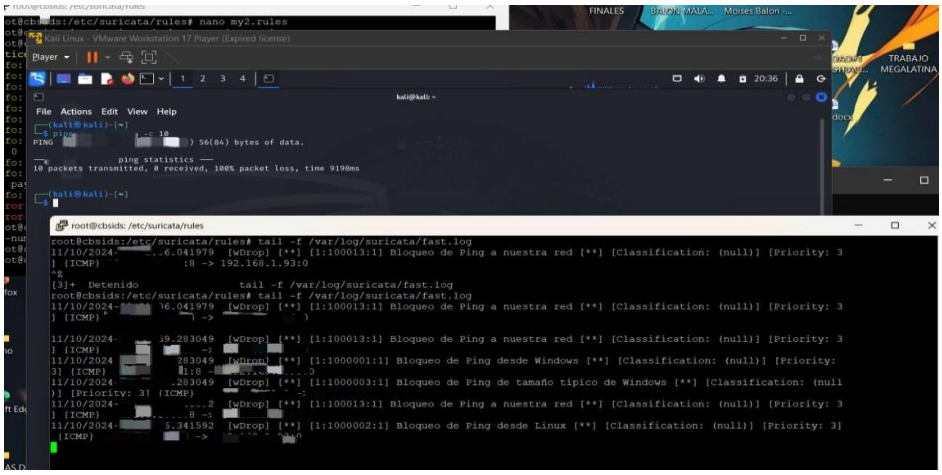


Figura 73 IPS Suricata en producción, bloqueo de ping de Linux.

Prueba #2: El siguiente conjunto de reglas está diseñado para mitigar ataques de denegación de servicio (DoS) que utilizan los protocolos **SYN** y **UDP**. Estas reglas tienen como objetivo bloquear o filtrar los paquetes maliciosos dirigidos a la dirección IP de nuestro servidor, evitando su sobrecarga y garantizando que solo el tráfico legítimo pueda acceder al sistema (Ver Figura #61 y #62).

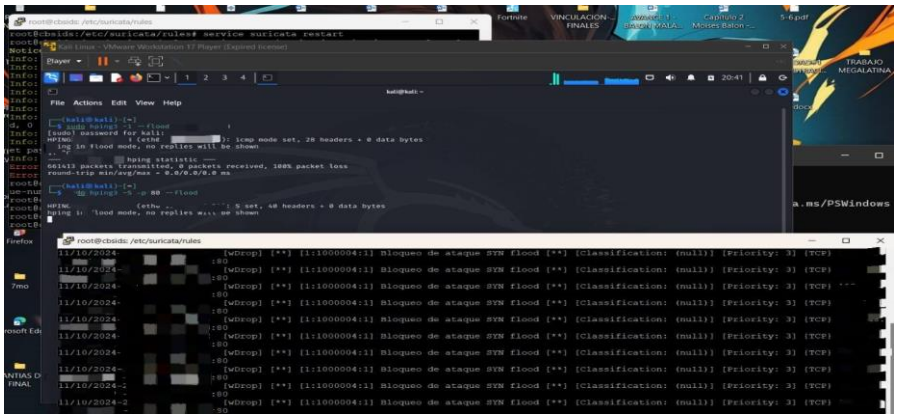


Figura 74 IPS Suricata en producción, Bloque de ataque SYN flood, DoS .

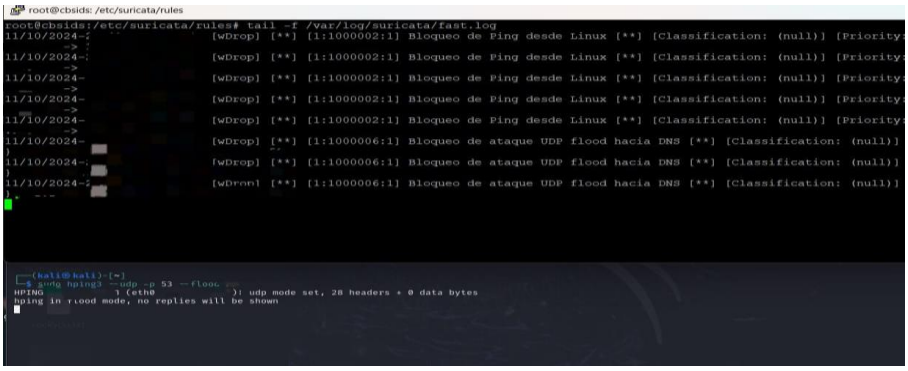


Figura 75 IPS Suricata en producción, Bloque de ataque UDP, DoS .

Prueba #3: Para implementar esta regla de bloqueo de conexiones por SSH y FTP, es necesario realizar una configuración adicional mediante algunos comandos específicos:

sudo iptables -A INPUT -p tcp --dport 22 -j REJECT

El comando proporcionado agrega una regla al firewall para bloquear conexiones entrantes al puerto 22 (usado por SSH).

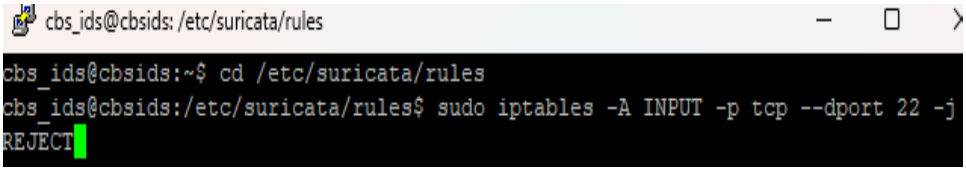
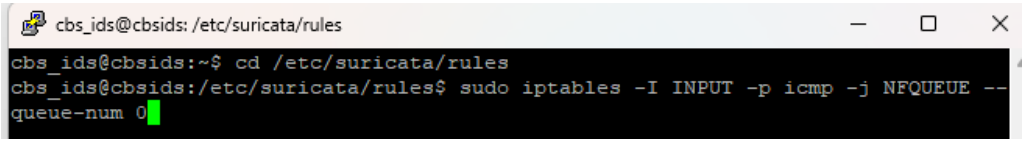


Figura 76 Configuración para bloqueo por SSH.

sudo iptables -I INPUT -p icmp -j NFQUEUE --queue-num 0

Permite bloquear y registrar el tráfico ICMP basándose en reglas personalizadas definidas en el programa que maneja la cola NFQUEUE.

Figura 77 Actualización de tablas para IPS Suricata.



sudo modprobe nfnetlink_queue

Cuando usas una regla de iptables con la acción -j NFQUEUE, los paquetes se colocan en una cola NFQUEUE. Sin el módulo nfnetlink_queue, no sería posible procesar esa cola desde una aplicación de espacio de usuario, es decir pone en producción en el bloqueo de cada servicio con las reglas.

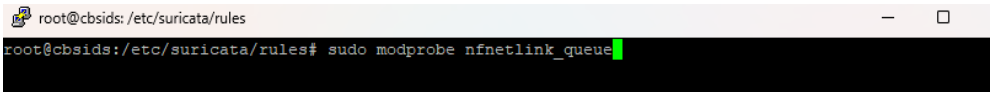


Figura 78 En producción reglas de bloqueo en Suricata.

Pruebas de rendimiento y alerta de Netdata.

Al ejecutar un ataque de Denegación de Servicio (DoS), nuestro monitor en tiempo real evidencia una saturación significativa en la red, lo que afecta su rendimiento normal. Además, se observa que el CPU del servidor está completamente comprometido, funcionando al 100% de su capacidad. Este comportamiento es característico de este tipo de ataque, ya que el servidor intenta procesar una cantidad masiva de solicitudes maliciosas, lo que puede llevar a la interrupción total de los servicios y afectar la disponibilidad del sistema.

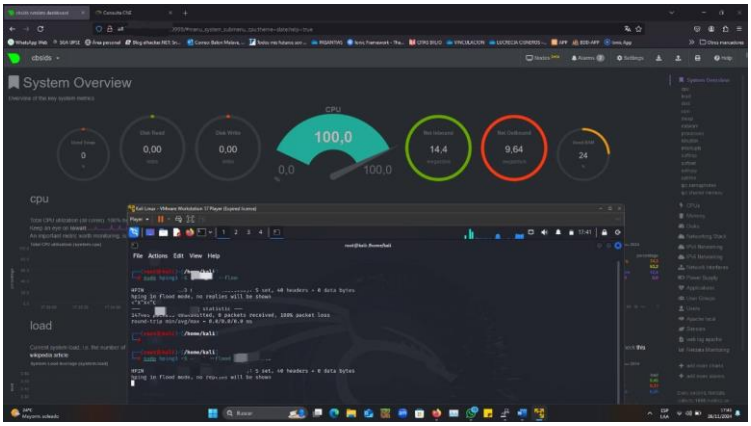


Figura 79 Saturación del sistema con monitoreo en tiempo real.

En la prueba realizada, Netdata registró alertas relacionadas con un posible ataque. Se identificaron eventos como un alto número de TCP resets (WARNING) y un exceso de cookies TCP SYN (CRITICAL), indicando un posible ataque de tipo SYN Flood.

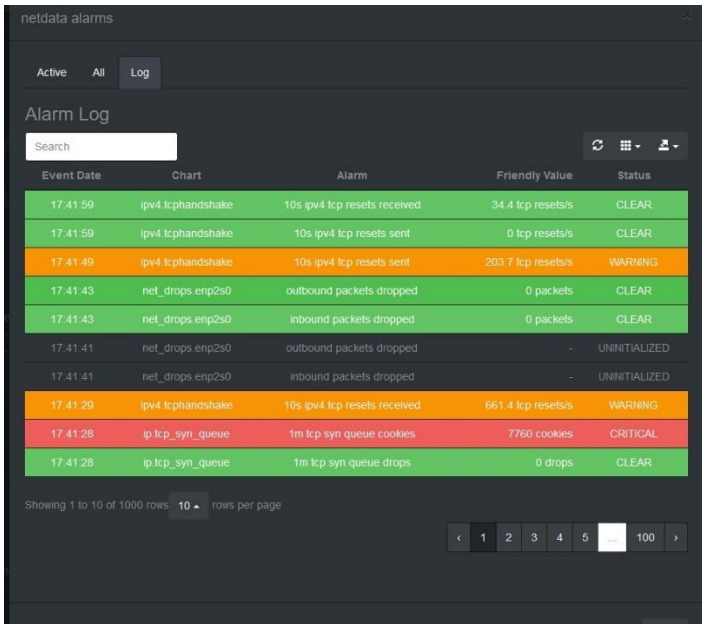


Figura 80 Log de paquetes tcp indicando posible ataque DoS

3.2.5 Fase V: Reporte De Ataques Post Implementación.

Está última fase, se generarán reportes de los ataques en el entorno de prueba. En dicho entorno se detectaron y mitigaron ataques tras la implementación de los sistemas de detección y prevención de intrusos. Además, se detectó la presencia de tráfico con direcciones IP fuera del rango permitido para el ambiente de prueba.

El objetivo principal de esta etapa es validar la efectividad de la solución, además, se analizarán las tendencias en los intentos de ataque detectados, destacando las mejoras obtenidas en la seguridad de la red y recomendando ajustes o nuevas estrategias para fortalecer la infraestructura tecnológica.

El script de Python proporcionado (ver Anexo #8) será utilizado para realizar un análisis conciso del ataque simulado en el entorno controlado. Este proceso permitirá identificar y contabilizar las ocurrencias (paquetes) asociadas al ataque, determinar el tipo específico de ataque llevado a cabo y generar información clave para evaluar su impacto.

El script analizará los datos capturados durante la simulación, procesando la información de manera eficiente para detectar patrones anómalos. Esto no solo facilita una evaluación

rápida, sino que también optimiza el tiempo requerido para la identificación de vulnerabilidades y el fortalecimiento de las medidas de seguridad.

El análisis generado servirá como base para validar los resultados esperados del experimento y documentar posibles mejoras en el entorno controlado o en los sistemas de detección utilizados.

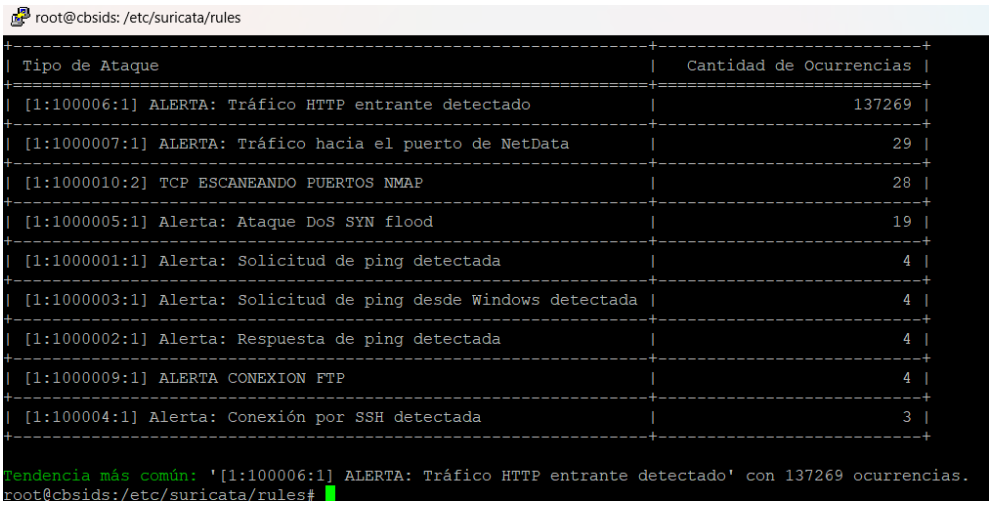


Figura 81 Reportes de ataques post implementación.

Esta es una manera de identificar los tipos de ataques que se han generado y, por supuesto, que nuestro sistema IDS/IPS ha podido detectar

CONCLUSIONES.

El desarrollo e implementación de reglas personalizadas en herramientas como **Snort** permitió identificar y prevenir amenazas comunes en las infraestructuras, entre las que se destacan según el reporte de la fase V, son los tráficos detectados en la red, ataques de denegación de servicios, escaneo de puertos y tráfico en el puerto de la herramienta de monitoreo en tiempo real. Durante el proyecto, se lograron diseñar reglas más específicas para minimizar falsos positivos, mejorando la precisión del sistema de detección de intrusos (IDS).

Adicionalmente, se exploraron y compararon otras herramientas y la selección de **Suricata**, que se destaca por su rendimiento en sistemas debido a su capacidad para aprovechar múltiples núcleos de CPU. Aunque Suricata también comparte una gran similitud con Snort, también demostró ser una solución robusta para la detección y

prevención de intrusos, Por otro lado, el uso de **Netdata** nos permitió obtener una visualización en tiempo real del rendimiento de los sistemas y de los recursos de red.

En el ambiente de pruebas controlada se pudo demostrar la combinación de herramientas como Snort y Suricata con buenas soluciones y el monitoreo como Netdata resulta en un enfoque integral para la ciberseguridad. Sin embargo, la generación de falsos positivos nos conllevo a la necesidad de realizar ajustes de manera debido a los nuevos posibles ataques que se generan. Estos resultados refuerzan la importancia de mantener una vigilancia activa, implementar políticas de seguridad en capas y garantizar la actualización continua de las configuraciones del IDS/IPS.

Finalmente, este proyecto afirma que herramientas como Snort, Suricata y Netdata, cuando se configuran y utilizan adecuadamente, pueden formar un ambiente seguro para evitar exponer la información. No obstante, estas requieren monitoreo constante, ajustes regulares y personalización para adaptarse a las características únicas de cada red, garantizando así una detección precisa de amenazas y una mejor respuesta ante posibles incidentes.

RECOMENDACIONES.

- Considerar este punto importante, tener en cuenta que la seguridad informática se puede ver comprometida por diversos tipos de ataques, lo que podría poner en riesgo la integridad de los datos. Por eso es recomendable que se particione la red mediante Vlans, segmentando la red según los departamentos mencionados en la estructura organizativa.

- Fomentar la cultura digital dentro de la institución, donde el personal se vea involucrado para evitar que las claves de seguridad se expongan a las vistas de todos. Además, es indispensable que el departamento de TIC’S estén preparados para cualquier tipo de amenaza que se presente.

- Es fundamental que las reglas se estén actualizando de manera constante para evitar falsos positivos e ir añadiendo nuevas reglas que aborden posibles amenazas emergentes, pero también es necesario implementar un Firewall para proporcionar una capa adicional de protección y garantizar una seguridad robusta.

REFERENCIAS

[1] G. Arango y D. Oscar, “El ABC de la seguridad informática: guía práctica para entender la seguridad digital”, 2023.

[2] C. A. Ocampo, Y. V. Castro Bermúdez, y G. R. Solarte Martínez, “Sistema de detección de intrusos en redes corporativas”, *Universidad Tecnológica de Pereira Pereira, Colombia*, 03-ene-2017.

[3] Europa Press, “Sistemas IDS, IPS y SIEM: qué son y por qué son importantes para la seguridad de la red de las empresas”, *portaltic*, 12-sep-2020. [En línea]. Disponible en: https://www.europapress.es/portaltic/ciberseguridad/noticia-sistemas-ids-ips-siem-son-son-importantes-seguridad-red-empresas-20200912113036.html. [Consultado: 01-dic-2024].

[4] J. G. Martinez, “Estadísticas de Ciberseguridad: Pronóstico para el 2024”, *Deltaprotect.com*, 14-mar-2024. [En línea]. Disponible en: https://www.deltaprotect.com/blog/estadisticas-de-ciberseguridad-pronostico-2024. [Consultado: 01-dic-2024].

[5] A. S. [Hispasec], “Overview de ataques DDoS a lo largo de 2022”, *Una Al Día*, 10-nov-2022. [En línea]. Disponible en: https://unaaldia.hispasec.com/2022/11/uad-ddos-cloudflare-trimestre-2022.html. [Consultado: 01-dic-2024].

[6] “Nueva epidemia: el phishing se sextuplicó en América Latina con el reinicio de la actividad económica y el”, *Kaspersky*, 23-ago-2023. [En línea]. Disponible en: https://latam.kaspersky.com/blog/panorama-amenazas-latam-2023/26586/. [Consultado: 01-dic-2024].

[7] M. Arora, “Cómo Cloudflare mitiga automáticamente un ataque DDoS de 3,8 Tb/s, el mayor registrado”, *Blog de Cloudflare*, 02-oct-2024. [En línea]. Disponible en: https://blog.cloudflare.com/es-es/how-cloudflare-auto-mitigated-world-record-3-8-tbps-ddos-attack/. [Consultado: 01-dic-2024].

[8] T. I. Canales, “La importancia de contar con un IDS”, *Canales TI*, 03-sep-2018. .

[9] “Plan de Creación de Oportunidades 2021-2025 – Secretaría Nacional de Planificación”, *Gob.ec*. [En línea]. Disponible en: https://www.planificacion.gob.ec/plan-de-creacion-de-oportunidades-2021-2025/. [Consultado: 01-dic-2024].

[10] “Historia – Cuerpo de Bomberos de Salinas”, *Gob.ec*. [En línea]. Disponible en: https://bomberossalinas.gob.ec/antecedentes/. [Consultado: 01-dic-2024].

[11] “Quienes Somos – Cuerpo de Bomberos de Salinas”, *Gob.ec*. [En línea]. Disponible en: https://bomberossalinas.gob.ec/quienessomos/. [Consultado: 01-dic-2024].

[12] “Misión y Visión – Cuerpo de Bomberos de Salinas”, *Gob.ec*. [En línea]. Disponible en: https://bomberossalinas.gob.ec/misionvision/. [Consultado: 01-dic-2024].

[13] “Rendición de Cuentas – 2021 – Cuerpo de Bomberos de Salinas”, *Gob.ec*. [En línea]. Disponible en: https://bomberossalinas.gob.ec/rendicion-de-cuentas-2021. [Consultado: 01-dic-2024].

[14] L. 0. R. O. S. 180 De, “CÓDIGO ORGÁNICO INTEGRAL PENAL, COIP”, *Gob.ec*. [En línea]. Disponible en: https://www.defensa.gob.ec/wp-content/uploads/downloads/2021/03/COIP_act_feb-2021.pdf. [Consultado: 01-dic-2024].

[15] ¿Qué son las redes informáticas?”, *Ibm.com*, 29-oct-2024.

[16] AWS, “¿Qué es el modelo OSI?”, *Amazon.com*. [En línea]. Disponible en: https://aws.amazon.com/es/what-is/osi-model/. [Consultado: 01-dic-2024].

[17] “¿Qué es un firewall? Funcionamiento de los firewalls y tipos de firewalls”, /, 13-may-2019. [En línea]. Disponible en: https://latam.kaspersky.com/resource-center/definitions/firewall. [Consultado: 01-dic-2024].

[18] “¿Qué es una dirección IP y qué significa?”, /, 07-nov-2020. [En línea]. Disponible en: https://www.kaspersky.es/resource-center/definitions/what-is-an-ip-address. [Consultado: 01-dic-2024].

[19] “¿Qué es un sistema de detección de intrusiones (IDS)?”, *Ibm.com*, 15-jul-2024. .

[20] E. Morales, “INTEGRACIÓN DE UN IDS/IPS AL CONTROLADOR SDN PARA LA PREVENCIÓN Y DETECCIÓN DE ATAQUES DE SEGURIDAD (DoS) EN UN ESCENARIO DE REDES DEFINIDAS POR SOFTWARE”, ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, Chimborazo, 2018.

[21] M. Escalante, “Qué es un Sistema de Detección de Intrusiones (IDS)”, *abcXperts*, 27-jun-2023. [En línea]. Disponible en: https://abcxperts.com/que-es-un-sistema-de-deteccion-de-intrusiones-ids/. [Consultado: 01-dic-2024].

[22] “¿Qué es un sistema de prevención de intrusiones (IPS)?”, *Ibm.com*, 16-jul-2024. .

[23] “¿Qué es un IPS (Sistema de Prevención de Intrusiones)?”, *Fortinet*. [En línea]. Disponible en: https://www.fortinet.com/lat/resources/cyberglossary/what-is-an-ips. [Consultado: 01-dic-2024].

[24] “¿Qué es la ciberseguridad?”, /, 25-may-2020. [En línea]. Disponible en: https://latam.kaspersky.com/resource-center/definitions/what-is-cyber-security [Consultado: 01-dic-2024].

[25] “¿Qué es un ataque cibernético?”, *Ibm.com*, 05-oct-2023. .

[26] “¿Qué son los ataques DDoS y cómo evitarlos?”, /, 23-nov-2017. [En línea]. Disponible en: https://latam.kaspersky.com/resource-center/threats/ddos-attacks [Consultado: 01-dic-2024].

[27] “Man in the middle (MITM) attack”, *Imperva.com*. [En línea]. Disponible en: https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/. [Consultado: 01-dic-2024].

[28] “Ataques de fuerza bruta: protección con contraseña”, /, 07-nov-2018. [En línea]. Disponible en: https://latam.kaspersky.com/resource-center/definitions/brute-force-attack?srsltid=AfmBOoqpfDuSkWE7kwOJbvUAkD8ZzhEzBarXPByrPNh3Hn6Bkw6vKlN6. [Consultado: 01-dic-2024].

[29] M. M. De La QuintanaIllanes, “SQL INYECTION”.

[30] “Phishing”, *Malwarebytes*, 16-ago-2023. [En línea]. Disponible en: https://www.malwarebytes.com/es/phishing. [Consultado: 01-dic-2024].

[31] “¿Qué es la ingeniería social?”, /, 06-dic-2017. [En línea]. Disponible en: https://latam.kaspersky.com/resource-center/definitions/what-is-social-engineering?srsltid=AfmBOoqN_Bo5rpSggvjPQ7qwFkkuZy8v9mzngYqEsiDJjFcq8BECTmpE. [Consultado: 01-dic-2024].

[32] R. Altube, “Kali Linux: Qué es y características principales”, *Openwebinars.net*, 05-nov-2021. .

[33] M. Medina, “¿Qué es Ubuntu y para qué Sirve? 【Guía Principiantes】”, *Hostinet*, 11-jun-2023. .

[34] “Guía de referencia de Nmap (Página de manual)”, *Nmap.org*. [En línea]. Disponible en: https://nmap.org/man/es/index.html. [Consultado: 01-dic-2024].

[35] “UCM-Proyecto de Innovación Software libre para ciencias e ingenierías”, *Ucm.es*. [En línea]. Disponible en: https://www.ucm.es/pimcd2014-free-software/wireshark. [Consultado: 01-dic-2024].

[36] H. Patilla, H. Huamani, y Y. Conislla, “Snort Open Source como detección de intrusos para la seguridad de la infraestructura de red”, 2021.

[37] “Suricata, la herramienta para detección de tráfico malicioso en tu red”, *Scassi.com*. [En línea]. Disponible en: https://es.scassi.com/es/temas-de-actualidad/suricata-la-

herramienta-para-deteccion-de-trafico-malicioso-en-tu-red. [Consultado: 01-dic-2024].

[38] “Monitor your entire infrastructure in high-resolution and in real-time”, *Netdata.cloud*. [En línea]. Disponible en: https://www.netdata.cloud/. [Consultado: 01-dic-2024].

[39] C. A. Rocha Haro, “La Seguridad Informática”, 2011.

[40] J. C. Gómez Castaño, N. J. Castaño Pérez, y L. C. Correa Ortiz, “Sistemas de detección y prevención de intrusos: Una taxonomía experimental basada en código abierto orientada a la industria 4.0”, *Cienc. Ing. Neogranadina*, vol. 33, núm. 1, pp. 75–86, 2023.

[41] R. Perdigón, “Evaluación del rendimiento de cortafuegos basados en software libre”, *NOVASINERGIA REVISTA DIGITAL DE CIENCIA, INGENIERÍA Y TECNOLOGÍA*, vol. 5, núm. 1, pp. 31–42, 2022.

[42] “Investigación Bibliográfica: Definición, Tipos, Técnicas”, *Lifeder.com*. [En línea]. Disponible en: https://www.lifeder.com/investigacion-bibliografica/. [Consultado: 01-dic-2024].

[43] “Definición de Investigación de Campo”, *Enciclopedia.net*. [En línea]. Disponible en: https://enciclopedia.net/investigacion-campo/. [Consultado: 01-dic-2024].

[44] J. C. Saavedra, “Diseño de Red con Top-Down”, *Juancarlossaavedra.net*, 30-ene-2015. [En línea]. Disponible en: https://juancarlossaavedra.net/2015/01/diseno-de-red-con-top-down/. [Consultado: 01-dic-2024].

[45] J. C. Santos, *Seguridad y Alta Disponibilidad (Grado Superior)*. 2011.

[46] “Analizando Snort: sistema de detección de intrusiones”, *Ciberseguridad*, 02-jul-2021. [En línea]. Disponible en: https://ciberseguridad.com/servicios/sistema-deteccion-intrusos-ids/snort/. [Consultado: 01-dic-2024].

[47] S. Mendoza, “Suricata IDS IPS: Rendimiento y Seguridad de la Red”, *Tecnetone.com*. [En línea]. Disponible en: https://blog.tecnetone.com/suricata-ids-ips-rendimiento-y-seguridad-de-la-red. [Consultado: 01-dic-2024].

[48] R. Perdigón-Llanes, “Evaluación de Snort y Suricata para la detección de sondeos de redes y ataques de denegación de servicio”, *Rev. cient. sist. inform.*, vol. 2, núm. 2, p. e363, 2022.

[49] “Netdata, navaja suiza para monitorizar el sistema”, *El array de Jota*, 04-dic-2017. [En línea]. Disponible en: https://www.elarraydejota.com/netdata-navaja-suiza-para-monitorizar-el-sistema/. [Consultado: 01-dic-2024].

[50] “Multicast DNS: la resolución de nombres para redes locales”, *IONOS Digital Guide*, 10-ago-2020. [En línea]. Disponible en: https://www.ionos.com/es-us/digitalguide/servidores/know-how/multicast-dns/. [Consultado: 01-dic-2024].

[51] “NetBIOS/NBNS - wireshark wiki”, *Wireshark.org*. [En línea]. Disponible en: https://wiki.wireshark.org/NetBIOS/NBNS. [Consultado: 01-dic-2024].

[52] “¿Qué es el ARP (Address Resolution Protocol)?”, *IONOS Digital Guide*, 11-sep-2023. [En línea]. Disponible en: https://www.ionos.com/es-us/digitalguide/servidores/know-how/arp-resolucion-de-direcciones-en-la-red/. [Consultado: 01-dic-2024].

[53] M. Escalante, “Qué es el Spanning Tree Protocol (STP)”, *abcXperts*, 30-may-2023. [En línea]. Disponible en: https://abcxperts.com/que-es-el-spanning-tree-protocol-stp/. [Consultado: 01-dic-2024].

ANEXOS

Anexo #1: Ficha de entrevistas.

“Implementación de sistemas de detección y prevención de intrusos (ids/ips) basado en software de código abierto para la red del cuerpo de bombero de salinas.”

RESPONSABLE: MOISES BALÓN MALAVÉ	**Entrevista con el director del departamento de TIC´S.**

Objetivo:

Recopilar la mayor información posible de la infraestructura de red del Cuerpo de bombero de salinas.

Técnicas aplicadas.

En esta fase con la ayuda de las técnicas aplicadas como es la entrevista y técnica de observación se logró obtener los siguientes resultados.

- Impresoras hp.
- Red no segmentada.
- Claves expuestas en escritorios.

En la entrevista se pudo obtener los siguientes datos:

- Configuración por ip estática.
- Página web basada en WordPress
- 2 vpn para su comunicación con el departamento ubicado en el GADS
- Su comunicación es mediante script.
- Servidor Dell Tower Power DC T320 (Xeon 2.2 GHz, 72gb ram pc3, 8 x 3 Tb 7200 rpm).
- Computadora hp 260 mini desktop (Servidor para biométrico)
- Switch Tp-link (TL-SG1024D No gestionable, 24 puertos Gigabyte rackeable).
- Router Huawei HG8245H ONT.

Tabla 5 Reporte de fase - Reconocimiento e investigación

Anexo #2: Ficha de reporte de escaneo.

“Implementación de Sistemas de Detección y Prevención de intrusos (IDS/IPS) basado en software de código abierto para la red del Cuerpo de Bombero de Salinas.**”**

FICHA DE REPORTE DE ESCANEO	
RESPONSABLE:	Ing. Luiggi Villafuerte

1. Información General

- **Nombre del Servidor:**
- **IP del Servidor:** 19x16x.1xx.xxx
- **Fecha de E+++valuación:** 05/JUL/2024
- **Evaluador:** Moises Balón

2. Resumen de la Evaluación

- **Objetivo de la Evaluación:**

❖ Evaluar la seguridad del servidor mediante la identificación de puertos abiertos y posibles vulnerabilidades.

- **Descripción General del Sistema:**

❖ Sistema Operativo:

Windows

❖ Servicios Críticos:

Privado

3. Resultados de la Evaluación

3.1 Escaneo de Puertos (Nmap)

Comando para utilizarse: nmap -sS -sV -oN

Puerto	Servicio
21	FTP
22	SSH
23	Telnet
25	SMTP
53	DNS
80	HTTP
110	POP3
143	IMAP
443	HTTPS
3306	MySQL
3389	RDP
8080	HTTP Alternativo
5432	PostgreSQL
6379	Redis
8000	HTTP Alternativo
8888	HTTP Alternativo

Tabla 6 Ficha de reporte de escaneo.

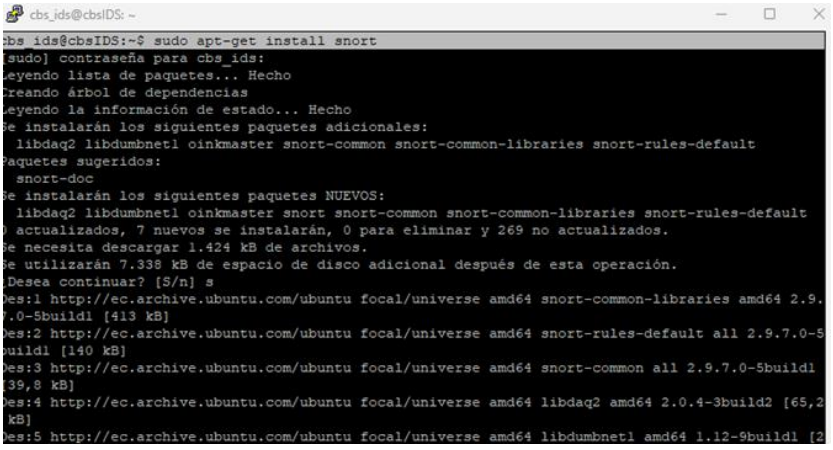
Anexo #3: Script python para el escaneo de red en el Cuerpo de Bomberos de Salinas

```python
import socket
from prettytable import PrettyTable
# IP de destino
target_ip = "192.168.100.1"
# Puertos a escanear y sus servicios
port_services = {
    21: "FTP",
    22: "SSH",
    23: "Telnet",
    25: "SMTP",
    53: "DNS",
    80: "HTTP",
    110: "POP3",
    143: "IMAP",
    443: "HTTPS",
    3306: "MySQL",
    3389: "RDP",
    8080: "HTTP Alternativo",
    5432: "PostgreSQL",
    6379: "Redis",
    27017: "MongoDB",
    5000: "Docker API",
    8000: "HTTP Alternativo 2",
    8888: "HTTP Alternativo 3"
}
# Crear una tabla para los resultados
table = PrettyTable()
table.field_names = ["Puerto", "Estado", "Servicio"]
def scan_port(ip, port):
    try:
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sock.settimeout(1)  # Tiempo de espera de 1 segundo
        result = sock.connect_ex((ip, port))
        service = port_services.get(port, "Desconocido")
        if result == 0:
            table.add_row([port, "ABIERTO", service])
        else:
            table.add_row([port, "CERRADO", service])
        sock.close()
    except Exception as e:
        table.add_row([port, f"ERROR: {e}", "Desconocido"])
def scan_ports(ip, port_services):
    print(f"Escaneando puertos en {ip}...")
    for port in port_services.keys():
        scan_port(ip, port)
    print(table)
if __name__ == "__main__":
    scan_ports(target_ip, port_services)
```

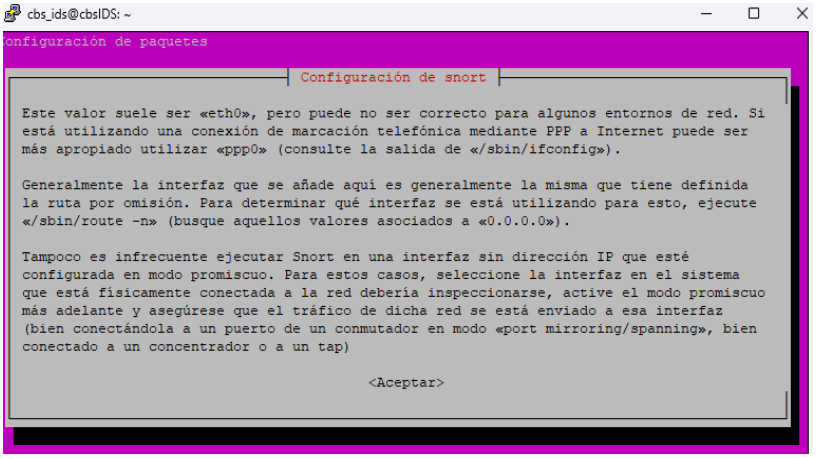
Anexo #4 Instalación y Configuración detallada de la instalación de Snort.

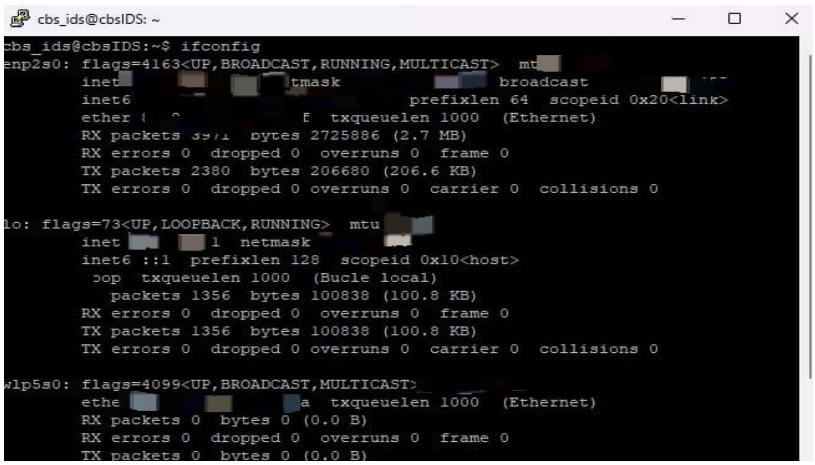
Paso #1: Para la instalación de Snort, debemos abrir nuestra terminal, en este caso utilizaremos putty, una vez establecida la conexión instalaremos Snort con el siguiente comando.

cbs_ids@cbsIDS:~$ sudo apt-get install Snort

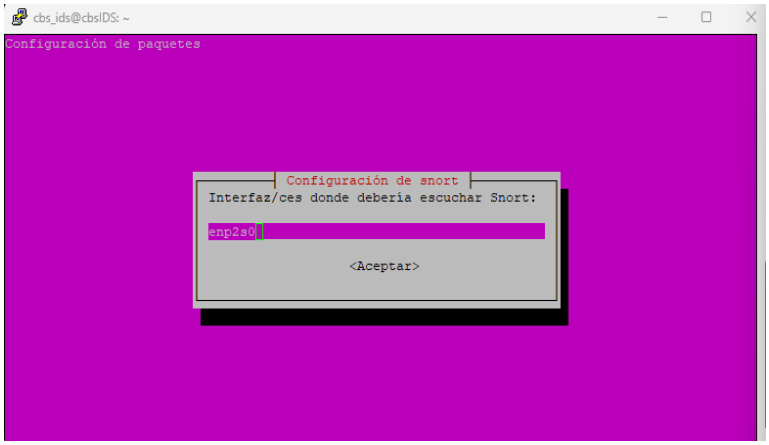


Paso #2 Una vez instalado Snort, automáticamente se nos arrojara una ventana para comenzar la configuración, adicional deberemos abrir otra terminal para verificar en que interfaz se está trabajando.

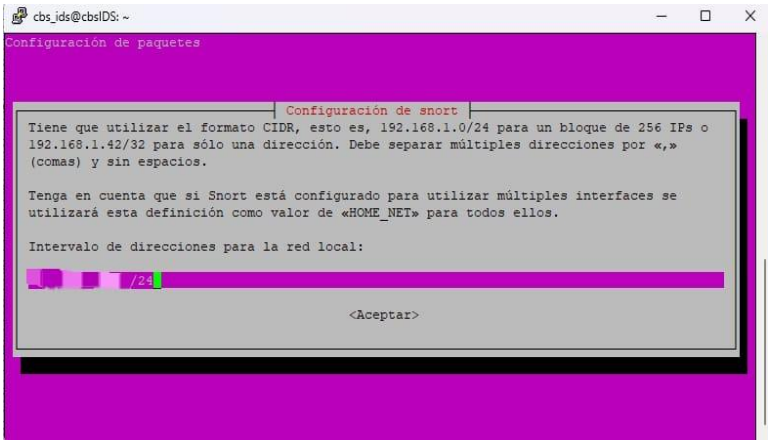




Paso #3: Obtenida la dirección ip e interfaz de red debemos volver a la configuración de Snort y en esta vez nos pedirá la interfaz de red a trabajar.



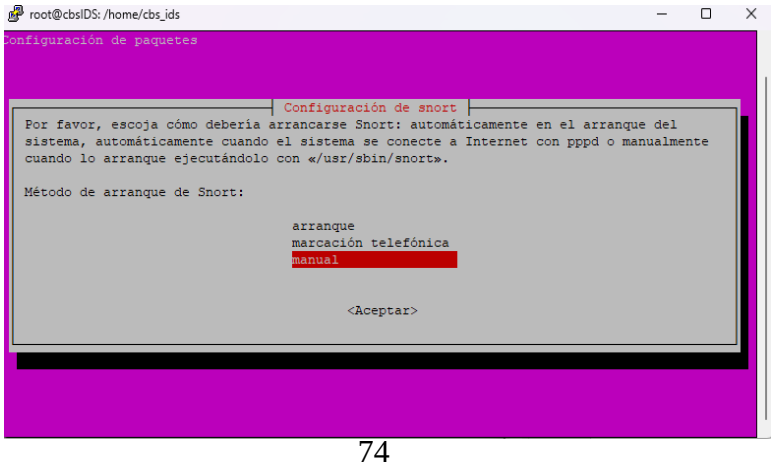
Paso #4: Nos pedirá que coloquemos el intervalo o dirección ip de la red local, en este caso colocaremos la dirección Ip del servidor y seguirá con la instalación de paquetes.



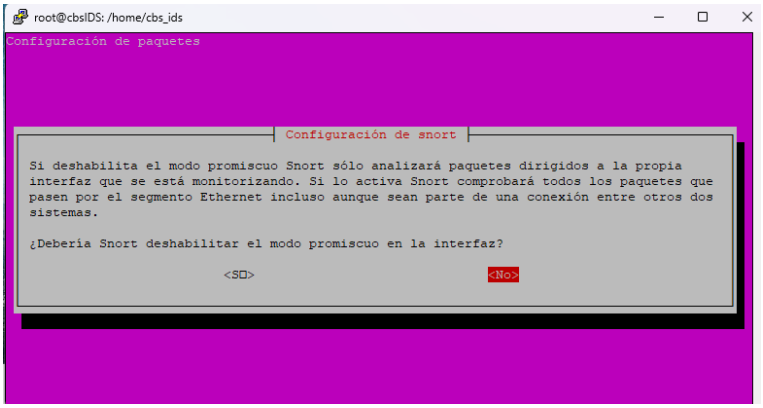
Paso #5: Una vez hecho la configuración se debe realizar de forma manual otras configuraciones con esta siguiente línea:

root@cbsIDS:/home/cbs_ids# dpkg-reconfigure snort

En esta configuración debemos elegir el método de arranque manual.

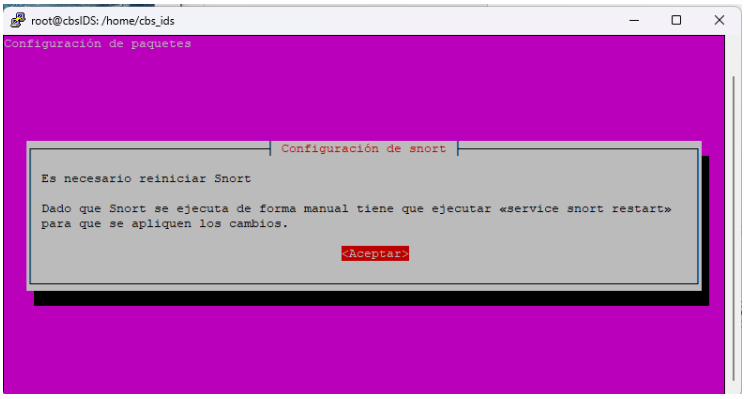


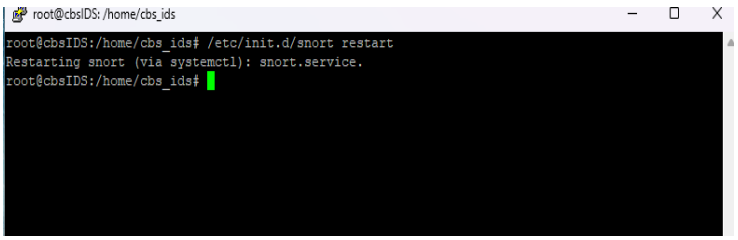
Paso #6: Una vez realizado el paso anterior, se nos pedirá desactivar la interfaz en el modo promiscuo; seleccionaremos “no”.



- **Paso #7:** Este último paso nos pedirá reiniciar el servicio de Snort, el comando a utilizar es:

root@cbsIDS:/home/cbs_ids# /etc/init.d/snort restart





Anexo #5: Reglas de sistema de detección de intrusos en Snort.

#==

Detecta un escaneo masivo de puertos TCP, activando una alerta si se detectan 20 paquetes en 60 segundos desde la misma fuente.

alert tcp any any -> $HOME_NET any (msg: " Escaneo de puertos tcp masivo detectado"; flags: S; threshold: type both, track by_src, count 20, seconds 60; sid:10000009; rev:1)

#==

Alerta sobre tráfico inusual en la red, con referencias a vulnerabilidades conocidas.

alert ip any any -> any any (msg:" Está habiendo un mal tráfico en la red"; sameip; reference:bugtraq,2666; reference:cve,1999-0016; reference:url,www.cert.org/advisories/CA-1997-28.html; classtype:bad-unknown; sid:527; rev:8;)

#==

Detecta intentos de conexión FTP provenientes de direcciones externas hacia el puerto 21 de la red interna.

alert tcp !$HOME_NET any -> $HOME_NET 21 (msg: "Inttento de conexión FTP externa"; flags: S; sid:10000010; rev:1)

#==

Alerta sobre pings realizados desde sistemas operativos Linux, identificando el tipo de contenido en el paquete.

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:" Ping Realizado desde S.O Linux"; itype:8; content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|"; depth:32; classtype:misc-activity; sid:366; rev:7;)

#==

Detecta pings replicados, lo que puede indicar un comportamiento anómalo o potencialmente malicioso.

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:" Este ping se ha replicado, cuidado"; icode:0; itype:0; classtype:misc-activity; sid:408; rev:5;)

#==

Alerta sobre pings realizados desde sistemas operativos Windows, proporcionando un indicador del sistema origen.

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:" Ping realizado desde S.O Windows"; itype:8; content:"abcdefghijklmnop"; depth:16; reference:arachnids,169; classtype:misc-activity; sid:382; rev:7;)

#==

Detecta conexiones SSH al servidor, alertando si se establece una nueva conexión.

alert tcp any any -> any 22 (msg:"Conexión SSH detectada"; flow:to_server; threshold:type both, track by_src, count 1, seconds 60; sid:100001; rev:1;)

#==

Alerta sobre accesos exitosos a través de SSH, lo que indica que se ha establecido una sesión SSH.

alert tcp any any -> any 22 (msg:" Acceso exitoso por SSH"; flow:established; content:"SSH"; sid:100004; rev:1;)

#==

Detecta intentos fallidos de inicio de sesión por SSH, lo que puede indicar un ataque de fuerza bruta.

alert tcp any any -> any 22 (msg:" Intento de inicio de sesión fallido por SSH"; flow:to_server; content:"failed password"; http_header; sid:100002; rev:1;)

#==

Alerta sobre accesos exitosos por FTP, identificando la respuesta de éxito.

alert tcp any any -> any 21 (msg:" Acceso exitoso por FTP"; flow:established; content:"230"; sid:100005; rev:1;)

#==

Detecta intentos fallidos de inicio de sesión por FTP, indicando problemas de autenticación.

alert tcp any any -> any 21 (msg:" Intento de inicio de sesión fallido por FTP"; flow:to_server; content:"530 Login incorrect"; sid:100003; rev:1;)

#==

Detecta intentos de escaneo de red en el puerto 705, identificando posibles herramientas de recolección de datos.

alert tcp $EXTERNAL_NET any -> $HOME_NET 705 (msg:" Se está requiriendo datos para un escaneo de Red"; flow:stateless; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; reference:cve,2002-0013; classtype:attempted-recon; sid:1421; rev:11;)

#==

Alerta sobre escaneos Nmap realizados por UDP, identificando referencias a vulnerabilidades conocidas.

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:" Nmap realizado por UDP"; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; reference:cve,2002-00002-0013; classtype:attempted-recon; sid:1417; rev:9;)

#==

Alerta sobre escaneos Nmap realizados por TCP, identificando referencias a vulnerabilidades conocidas.

alert tcp $EXTERNAL_NET any -> $HOME_NET 161 (msg:" Nmap realizado por TCP"; flow:stateless; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; reference:cve,2002-0013; classtype:attempted-recon; sid:1418; rev:11;)

#==

Detecta traps SNMP por el puerto UDP 162, indicando intentos de recolección de información.

alert udp $EXTERNAL_NET any -> $HOME_NET 162 (msg:" SNMP trap udp"; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; classtype:attempted-recon; sid:1419; rev:9;)

#==

Alerta sobre ataque DoS ICMP dirigido a la IP 1xx.1xx.xxx.xxx, activándose si hay 50 paquetes en 10 segundos.

alert icmp any any -> 1xx.1xx.xxx.xxx any (msg:" Ataque DoS ICMP, por el puerto 80"; threshold: type threshold, track by_src, count 50, seconds 10; sid:1000002; rev:1;)

#==

Detecta ataques DoS en el puerto 80 por TCP hacia la IP 1xx.1xx.xxx.xxx, activándose bajo ciertas condiciones de tráfico.

alert tcp any any -> 1xx.1xx.xxx.xxx any (msg:" Ataque DoS Por el puerto 80, TPC"; threshold: type threshold, track by_src, count 50, seconds 10; sid:1000003; rev:1;)

#==

Detecta ataques DoS en toda la red a través de conexiones TCP.

alert tcp $EXTERNAL_NET any <> $HOME_NET 0 (msg:" Ataque DoS en toda la red, TPC"; flow:stateless; classtype:misc-activity; sid:524; rev:8;)

#==

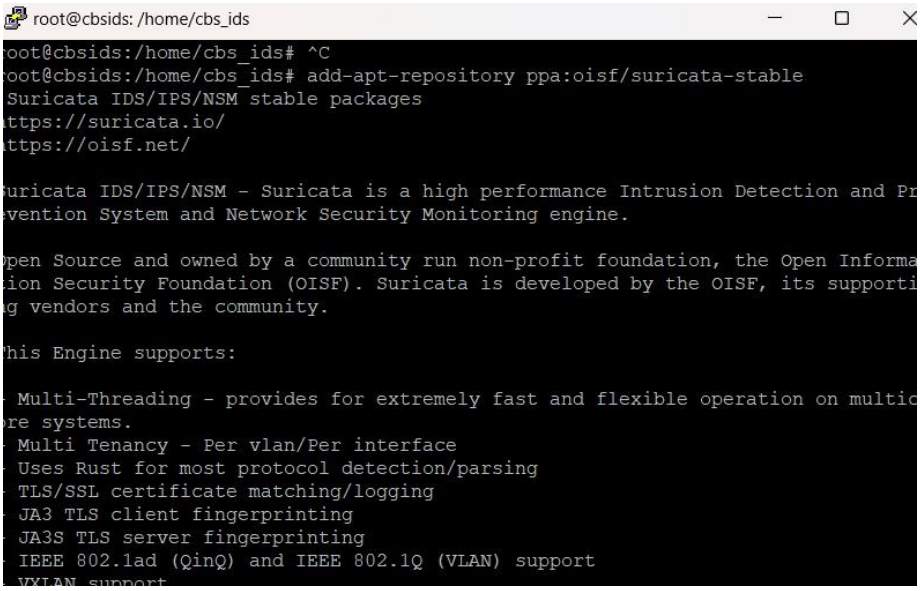
Detecta intentos de acceso fallido por fuerza bruta SSH a la IP 1xx.1xx.xxx.xxx.

alert tcp any any -> 1xx.1xx.xxx.xxx 22 (msg:" Intento de acceso fallido por fuerza bruta SSH"; flow:to_server,established; content:"Failed password"; http_method; threshold: type threshold, track by_src, count 5, seconds 60; sid:1000004; rev:1;)

Anexo #6: Instalación y Configuración detallada de la instalación de Suricata.

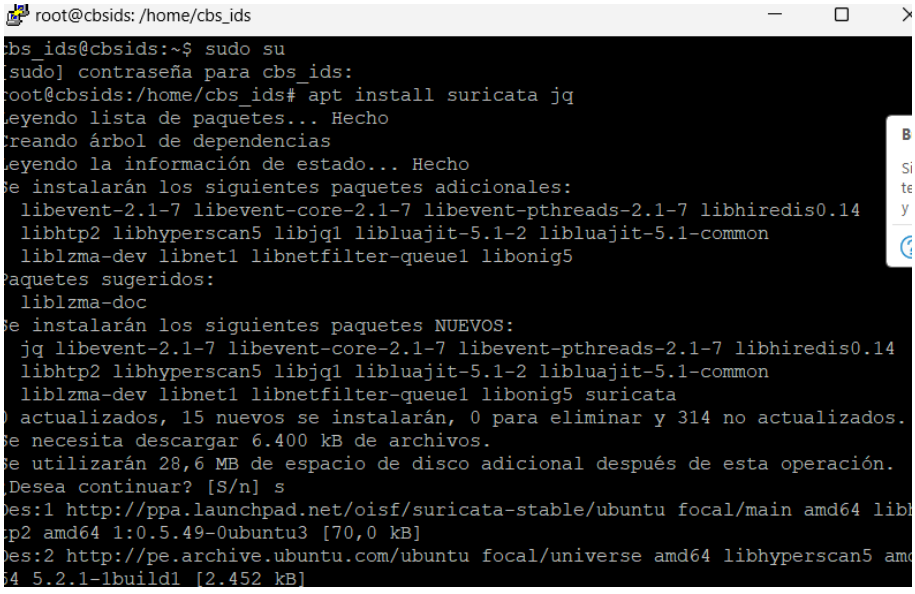
Paso #1: Para la instalación de Suricata, debemos abrir nuestra terminal, en este caso utilizaremos putty, una vez establecida la conexión descargaremos el repositorio de Suricata con el siguiente comando.

root@cbsids:/home/cbs_ids# add-apt-repository ppa:oisf/suricata-stable



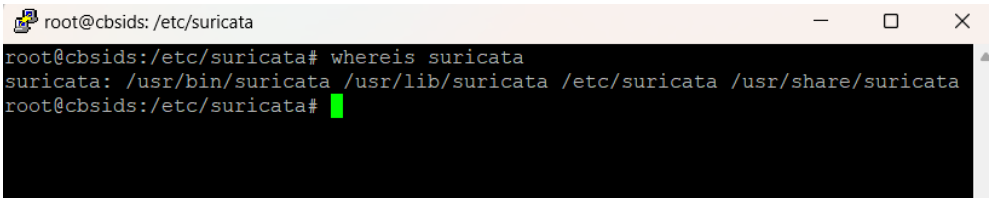
Paso #2: Una vez descargado el repositorio, procedemos a instalar Suricata acompañado con jq que nos ayudara a leer datos de archivos json que se generan.

root@cbsids:/home/cbs_ids# apt install suricata jq



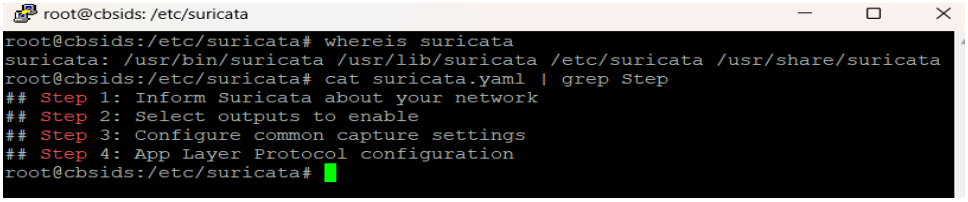
Paso #3: Para verificar en que directorio se ha instalado suricata usaremos el siguiente comando:

root@cbsids:/etc/suricata# whereis suricata



Paso #4: Con el Step podemos verificar que apartado nos incluyes suricata para poder trabajar con su motor IDS/IPS y así poder configurar en el siguiente paso.

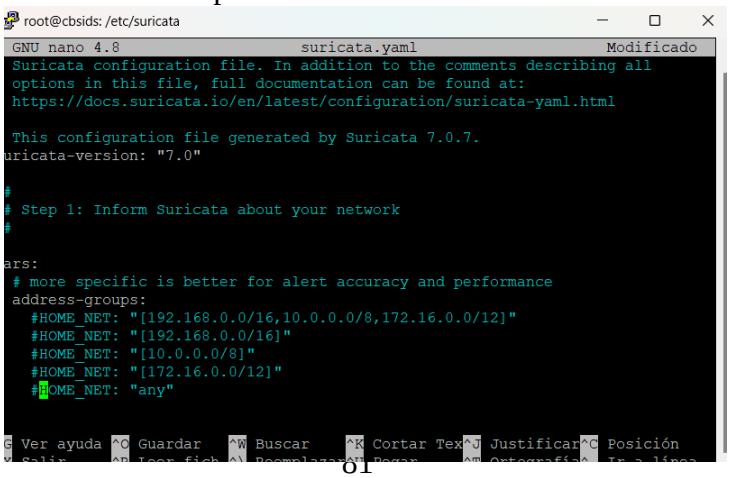
root@cbsids:/etc/suricata# cat suricata.yaml | grep Step



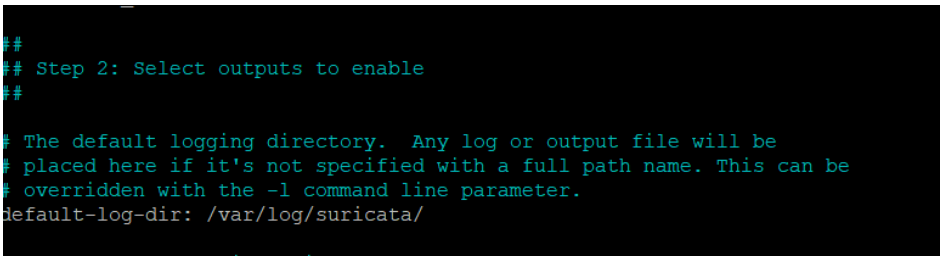
Paso #5: A continuación, procederemos a configurar la interfaz de red y asignar la dirección IP para que Suricata pueda monitorear el tráfico de nuestra red de manera eficiente. Esta configuración permitirá que Suricata escuche y analice el tráfico en la interfaz seleccionada, detectando patrones sospechosos y posibles amenazas de seguridad en tiempo real.

root@cbsids:/etc/suricata# nano suricata.yaml

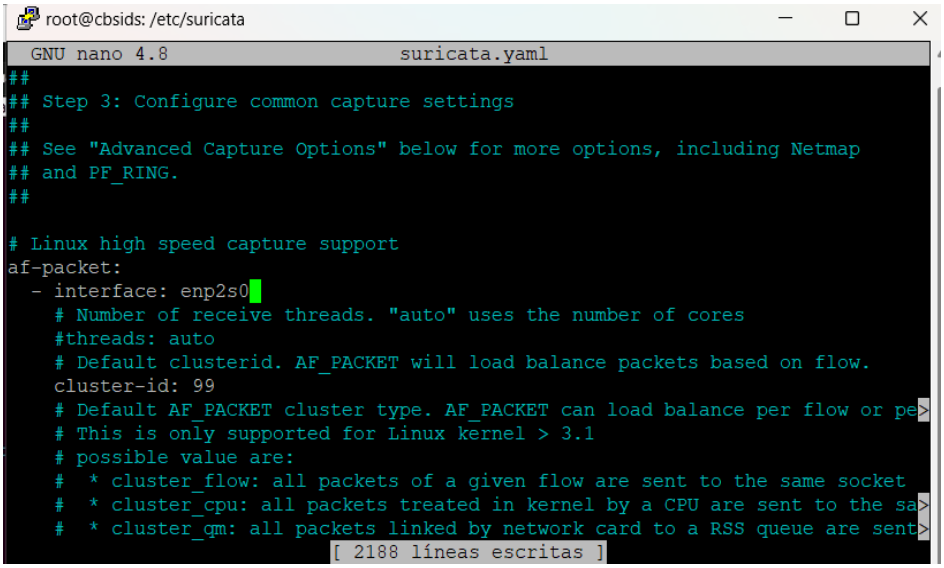
En el Step 1 vamos a configurar con que dirección Ip vamos a trabajar, por defecto suricata trae algunas direcciones para modificar.



En el Step 2 podemos cambiar o verificar en que directorio se registraran los logs que se generen durante la detección y prevención de intrusos obtenidos por Suricata.



En el Step 3, configuraremos la interfaz de red en la que nuestro estará en modo escucha. Esto significa que asignaremos a Suricata la interfaz adecuada para capturar y analizar todo el tráfico de la red en tiempo real, permitiéndole detectar actividades sospechosas o posibles amenazas de manera continua.



Nota: Una vez realizada la configuración de la interfaz de red, podemos ajustar las reglas de alerta y de prevención de intrusos en Suricata. Estas reglas son esenciales para identificar patrones de tráfico inusuales o sospechosos y responder ante posibles amenazas de seguridad. Las reglas de alerta permiten que Suricata detecte y notifique actividades anómalas, mientras que las reglas de prevención permiten bloquear activamente el tráfico malicioso.

Anexo #7: Reglas de sistema de detección y prevención de intrusos Suricata.

Regla en modo IDS

#==

 # 1. ALERTAS PARA ICMP (PING Y ECHO REPLY)

alert icmp any any -> any any (msg:"Alerta: Solicitud de ping detectada"; itype:8; sid:1000001; rev:1;)

alert icmp any any -> any any (msg:"Alerta: Respuesta de ping detectada"; itype:0; sid:1000002; rev:1;)

#Alerta para detectar pings desde diferentes sistemas operativos:

alert icmp any any -> any any (msg:"Alerta: Solicitud de ping desde Windows detectada"; itype:8; ttl:128; dsize:32; sid:1000003; rev:1;)

alert icmp any any -> any any (msg:"Alerta: Solicitud de ping desde Linux detectada"; itype:8; ttl:64; sid:1000004; rev:1;)

#==

#2DA REGLA CONEXIÓN SSH

alert tcp any any -> any 22 (msg:"Alerta: Conexión por SSH detectada"; flow:to_server,established; content:"SSH"; sid:100004; rev:1;)

alert tcp any any -> any 22 (msg:"Alerta: Inicio de sesión exitoso por SSH"; flow:to_server,established; content:"Accepted password"; sid:100007; rev:1;)

#==

 # 3RA REGLA ALERTA ATAQUE DOS

alert tcp any any -> 1XX.1XXX.X.X 80 (msg:"Alerta: Ataque DoS SYN flood"; flags:S; threshold:type both, track by_src, count 100, seconds 1; sid:1000005; rev:1;)

#==

#4TA REGLA PARA DETECTAR TRAFICO EN PUERTO HTTP

alert tcp any any -> any 80 (msg:"ALERTA: Tráfico HTTP entrante detectado";flags:S,A; sid:100006; rev:1;)

#==

#5TA REGLA PARA DETECTAR TRÁFICO POR PUERTO DE NETDATA

alert tcp any any -> any 19999 (msg:"ALERTA: Tráfico hacia el puerto de NetData"; flags:S,A; sid:1000007; rev:1;)

alert tcp any any -> 1XX.1XXX.X.X 19999 (msg:"ALERTA: Posible ataque DoS SYN Flood hacia NetData"; flags:S; threshold:type both, track by_src, count 50, seconds 1; sid:1000008; rev:1;)

#==

 # 6TA REGLA CONEXIÓN FTP

alert tcp $HOME_NET any -> 1XX.1XXX.X.X 21 (msg: "ALERTA CONEXION FTP"; sid:1000009; rev:1;)

#==

#7MA REGLA ESCANEO DE PUERTOS

alert tcp $HOME_NET any -> 1XX.1XXX.X.X [21,22,121,13922,443,23,25,3389,3306,1433] (msg: "TCP ESCANEANDO PUERTOS NMAP"; sid:1000010; rev:2;)

#==

MODO IPS SURICATA

#==

#1ER CONJUNTO DE REGLAS BLOQUEO DE PING

drop icmp any any -> $HOME_NET any (msg:"Bloqueo de Ping a nuestra red"; sid:100013; rev:1;)

drop icmp any any -> $HOME_NET any (msg:"Bloqueo de Ping desde Windows"; ttl:128; sid:1000001; rev:1;)

drop icmp any any -> $HOME_NET any (msg:"Bloqueo de Ping desde Linux"; ttl:64; sid:1000002; rev:1;)

drop icmp any any -> $HOME_NET any (msg:"Bloqueo de Ping de tamaño típico de Windows"; dsize:32; sid:1000003; rev:1;)

#==

#2DO CONJUNTO DE BLOQUEO DE ATAQUES DoS

drop tcp any any -> $HOME_NET 80 (msg:"Bloqueo de ataque SYN flood"; flags:S; threshold:type both, track by_src, count 20, seconds 1; sid:1000004; rev:1;)

drop icmp any any -> $HOME_NET any (msg:"Bloqueo de ataque ICMP flood"; threshold:type both, track by_src, count 15, seconds 1; sid:1000005; rev:1;)

drop udp any any -> $HOME_NET 53 (msg:"Bloqueo de ataque UDP flood hacia DNS"; threshold:type both, track by_src, count 20, seconds 1; sid:1000006; rev:1;)

#==

#3ER CONJUNTO DE BLOQUEA DE ATAQUE DoS POR NETDATA

drop tcp any any -> $HOME_NET 19999 (msg:"Bloqueo de ataque SYN flood hacia Netdata"; flags:S; threshold:type both, track by_src, count 20, seconds 1; sid:1000007; rev:1;)

drop udp any any -> $HOME_NET 19999 (msg:"Bloqueo de ataque UDP flood hacia Netdata"; threshold:type both, track by_src, count 20, seconds 1; sid:1000008; rev:1;)

drop http any any -> $HOME_NET 19999 (msg:"Bloqueo de ataque HTTP flood hacia Netdata"; flow:to_server; threshold:type both, track by_src, count 20, seconds 1; sid:1000009; rev:1;)

#==

#4TO CONJUNTO DE REGLAS PARA BLOQUEO DE CONEXIÓNES EXTERNAS

#==

drop tcp any any -> $HOME_NET 21 (msg:"Bloqueo de conexión FTP"; sid:1000010; rev:1;)

drop tcp any any -> $HOME_NET 22 (msg:"Bloqueo de conexión SSH"; sid:1000011; rev:1;)

ANEXO #8 Script para reportes post implementación.

```python
import re
from collections import Counter
from tabulate import tabulate
from colorama import Fore, Style
import matplotlib.pyplot as plt

# Ruta al archivo de log de Suricata
log_file_path = '/var/log/suricata/fast.log'

# Función para analizar el archivo de log y contar los ataques
def analyze_suricata_logs(file_path):
    # Expresión regular mejorada para detectar el tipo de ataque
    attack_pattern = re.compile(r'\[\*\*\]\s(.*?)\s\[.*?\]')  # Captura el mensaje del ataque dentro de
[**] y antes de [sid]

    # Lista para almacenar los ataques detectados
    attack_list = []

    try:
        # Abrir el archivo de log de Suricata
        with open(file_path, 'r') as log_file:
            for line in log_file:
                match = attack_pattern.search(line)
                if match:
                    attack_type = match.group(1).strip()  # Captura y limpia el tipo de ataque
                    attack_list.append(attack_type)
    except FileNotFoundError:
        print(f"{Fore.RED}Error: El archivo {file_path} no se encontró.{Style.RESET_ALL}")
        return
    except Exception as e:
        print(f"{Fore.RED}Error al analizar los logs: {e}{Style.RESET_ALL}")
        return

    # Contar la frecuencia de cada ataque detectado
    attack_counter = Counter(attack_list)

    # Si no se detectaron ataques
    if not attack_counter:
        print(f"{Fore.YELLOW}No se detectaron alertas en el archivo de logs.{Style.RESET_ALL}")
        return

    # Crear la tabla de resultados
    table = [[attack, count] for attack, count in attack_counter.most_common()]
    headers = ["Tipo de Ataque", "Cantidad de Ocurrencias"]
    print(tabulate(table, headers, tablefmt="grid"))

    # Mostrar el ataque más común
    most_common_attack, most_common_count = attack_counter.most_common(1)[0]
    print(f"\n{Fore.GREEN}Tendencia más común:{Style.RESET_ALL} '{most_common_attack}' con
{most_common_count} ocurrencias.")

    # Generar gráfica del top 3 ataques
    plot_top_attacks(attack_counter)

# Función para generar gráfica del top 3 ataques
def plot_top_attacks(attack_counter):
    top_3 = attack_counter.most_common(3)
    if not top_3:
        print(f"{Fore.YELLOW}No hay suficientes datos para generar una gráfica.{Style.RESET_ALL}")
        return

    labels, values = zip(*top_3)
    colors = ['#FF5733', '#33FF57', '#3357FF']  # Colores personalizados para la gráfica

    plt.bar(labels, values, color=colors)
    plt.title("Top 3 Tipos de Ataques")
    plt.xlabel("Tipos de Ataques")
    plt.ylabel("Cantidad de Ocurrencias")
    plt.xticks(rotation=15, ha="right")
    plt.tight_layout()
    plt.grid(axis="y", linestyle="--", alpha=0.7)
    plt.show()

# Ejecutar el análisis de logs
analyze_suricata_logs(log_file_path)
```