UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD INGENIERIA
CARRERA DE TELECOMUNICACIONES

Implementación de un sistema de detección de ataques de denegación de servicios en dispositivos IOT para hogar, utilizando herramientas open source.

Trabajo de Titulación para optar al título de Ingeniero en Telecomunicaciones

Autor:
Agualsaca Paca, Elvis Ismael

Tutor:
MgSc. Luis Gonzalo Santillán

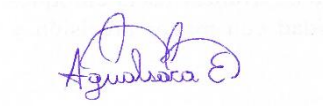
Riobamba, Ecuador. ***2026***

DECLARATORIA DE AUTORÍA

Yo, **Elvis Ismael Agualsaca Paca**, con cédula de ciudadanía **1726031881**, autor del trabajo de investigación titulado: **Implementación de un sistema de detección de ataques de denegación de servicios en dispositivos IOT para hogar, utilizando herramientas open source**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 24 de noviembre del 2025



Elvis Ismael Agualsaca Paca

C.I:1726031881

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

En la Ciudad de Riobamba, a los 24 días del mes de Noviembre de 2025, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por el estudiante **Elvis Ismael Agualsaca Paca** con CC: **1726031881**, de la carrera **TELECOMUNICACIONES** y dando cumplimiento a los criterios metodológicos exigidos, se emite el **ACTA FAVORABLE DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN** titulado **"Implementación de un sistema de detección de ataques de denegación de servicios en dispositivos IOT para hogar, utilizando herramientas open source"**, por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.

Ing. Luis Gonzalo Santillán, Mgs.
TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **"IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN DE ATAQUES DE DENEGACIÓN DE SERVICIOS EN DISPOSITIVOS IOT PARA HOGAR, UTILIZANDO HERRAMIENTAS OPEN SOURCE"**, por **ELVIS ISMAEL AGUALSACA PACA**, con cédula de identidad número **1726031881**, bajo la tutoría de **Mg. Luis Gonzalo Santillán Valdiviezo**; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 06 de enero de 2026.

Giovanny Cuzco, Mgs.
PRESIDENTE DEL TRIBUNAL DE GRADO

Yesenia Cevallos, PhD.
MIEMBRO DEL TRIBUNAL DE GRADO

Antonio Meneses, PhD.
MIEMBRO DEL TRIBUNAL DE GRADO

CERTIFICADO ANTIPLAGIO

Dirección
Académica
VICERRECTORADO ACADÉMICO

en movimiento
SGC
SISTEMA DE GESTIÓN DE LA CALIDAD
UNACH-RGF-01-04-08.15
VERSIÓN 01: 06-09-2021

CERTIFICACIÓN

Que, **Elvis Ismael Agualsaca Paca** con CC: **1726031881**, estudiante de la Carrera **TELECOMUNICACIONES**, Facultad de **INGENIERÍA**; ha trabajado bajo mi tutoría el trabajo de investigación titulado **"Implementación de un sistema de detección de ataques de denegación de servicios en dispositivos IOT para hogar, utilizando herramientas open source",** cumple con el 1% de similitud y 7 % de inteligencia Artificial, de acuerdo al reporte del sistema Anti plagio **COMPILATIO**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente, autorizo continuar con el proceso.

Riobamba, 15 de diciembre de 2025

Firmado electrónicamente por:
LUIS GONZALO
SANTILLAN
VALDIVIEZO
Validar únicamente con FirmaEC

Ing Luis Gonzalo Santillan, Mgs
TUTOR

DEDICATORIA

A Dios, por guiar cada paso de mi trayectoria y brindarme la fuerza y valentía para superar las dificultades.

A mi esposa, por su amor, paciencia y apoyo incondicional, y motivación han sido un pilar fundamental para alcanzar este logro académico.

A mi hija, quien es mi mayor fuente de inspiración. Su sonrisa y ternura me dieron la fuerza necesaria para no rendirme.

A mis padres, por su sacrificio, por su apoyo incondicional y por enseñarme que con esfuerzo y perseverancia todo es posible.

A mi hermano, por su acompañamiento, apoyo y palabras de aliento en cada etapa de este proceso.

Este trabajo es reflejo del esfuerzo compartido y del amor de mi familia, a quienes dedico este logro con profundo agradecimiento

AGRADECIMIENTO

Agradezco profundamente a mi esposa y a mi hija, por su amor, paciencia y comprensión a lo largo de este proceso.
A mis padres y a mi hermano, por su respaldo constante, consejos y palabras de aliento.
Finalmente, expreso mi más sincero agradecimiento a mi tutor de tesis, por su orientación, apoyo académico y valiosos aportes durante el desarrollo de este trabajo, los cuales fueron fundamentales para alcanzar los objetivos propuestos.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente estudio presenta la implementación de un sistema de detección de ataques de denegación de servicios (DoS) en dispositivos IoT para entornos domésticos, utilizando herramientas open source dado que el continuo desarrollo de soluciones prácticas y de bajo costo que permiten detectar, resulta fundamental para garantizar la seguridad y resiliencia de la red doméstica. Esto se desarrollará mediante el estudio de los diferentes tipos de ataques en dispositivos IoT y la manera de afrontarlos para así, llegar a implementar un prototipo funcional que detecte las vulnerabilidades. La metodología aplicada será de tipo mixto, empleando 3 tipos de investigación, como lo son aplicada, descriptiva y exploratoria de manera que permita abarcar diferentes ámbitos; además es de tipo cuasi-experimental ya que se realiza en un ambiente controlado dentro de un laboratorio doméstico. El trabajo presenta un enfoque mixto, en su mayoría cuantitativo, cuya implementación será el paso final luego de las fases de análisis de documentación y diseño del prototipo. Una vez se haya realzado la implementación se ejecutará la Fase de pruebas Controladas, donde se procede a enviar diferentes tipos de ataques DoS para verificar su correcta detección.

Para terminar con la investigación se mide la efectividad del sistema, mediante pruebas estadísticas básicas como media, moda, mediana, histogramas, matriz de Dispersión, matriz de Correlación de Pearson, complementada con el Modelo de Regresión Múltiple mediante el cual se asume que el modelo cumple parcialmente con los supuestos estadísticos fundamentales.

Palabras claves: DoS, IoT, MC Pearson, flood, Regresión Múltiple, supuestos.

ABSTRACT

This study presents the implementation of a denial-of-service (DoS) attack detection system in IoT devices for home environments, using open-source tools, given that the continuous development of practical, low-cost solutions for detection is essential to ensuring the security and resilience of home networks. This will be developed by studying the different types of attacks on IoT devices and how to address them, to implement a functional prototype that detects vulnerabilities. The methodology applied will be mixed, using three types of research: applied, descriptive, and exploratory, to cover different areas. It is also quasi-experimental, conducted in a controlled environment within a home laboratory. The work presents a mixed approach, mostly quantitative, whose implementation will be the final step after the documentation analysis and prototype design phases. Once implementation is complete, the Controlled Testing Phase will be executed, during which different types of DoS attacks will be sent to verify their correct detection. To conclude the research, the effectiveness of the system is measured using basic statistical tests, such as the mean, mode, median, histograms, and a dispersion matrix. Pearson correlation matrix, supplemented by a multiple regression model that partially complies with the fundamental statistical assumptions.

Keywords: DoS, IoT, Pearson MC, flood, Multiple Regression, assumptions.

Firmado electrónicamente por:
ANA ELIZABETH MALDONADO LEON
Validar únicamente con FirmaEC

Reviewed by:

Ms.C. Ana Maldonado León

ENGLISH PROFESSOR

C.I.0601975980

CAPITULO I INTRODUCCIÓN

1.1 Antecedentes

En los últimos años, la presencia de dispositivos del Internet de las Cosas (IoT) en los hogares ha crecido de forma sostenida, incrementando proporcionalmente los puntos vulnerables dentro de las redes domésticas. Diversos estudios señalan que las denominadas casas inteligentes no solo aportan comodidad y eficiencia, sino que también se encuentran expuestas a múltiples riesgos de seguridad [1].

No obstante, persiste un desafío crítico en la seguridad de la información: los ataques de denegación de servicio (DoS) y su variante distribuida (DDoS), cuyo propósito es saturar los recursos de un dispositivo o red para impedir su uso legítimo. En el contexto IoT doméstico, un ataque DoS puede inutilizar cámaras de seguridad, termostatos inteligentes, bombillas conectadas, routers, o incluso hacer caer el acceso a Internet de toda una vivienda, realizándolo mediante múltiples solicitudes a uno o varios servidores, como web, correo electrónico, bases de datos, proxy, etc., desde diferentes ubicaciones en el mundo, con el objetivo de sobrecargar el sistema hasta que se derrumba, o mediante ataques de fuerza bruta utilizando malware especializado que escanea, un ejemplo representativo es el malware Mirai, que demostró cómo los dispositivos IoT mal configurados pueden ser comprometidos y reclutados para conformar botnets capaces de ejecutar ataques DDoS a gran escala., pudiendo afectar no solo el ancho de banda sino también la latencia y las tablas de conmutación de flujo de datos [2].

En [3], se hizo una revisión de los conceptos de protección contra ataques DDoS en ambientes IoT. Los autores revisaron diferentes enfoques de defensa, tales como modelos basados en middleware, mecanismos de aprendizaje automático y otras soluciones complementarias, exponiendo sus limitaciones y vulnerabilidades.

Por su parte, [4] aborda los ataques DDoS describiendo sus diferentes modalidades y estrategias de detección y prevención, que incluyen desde el monitoreo continuo y el rastreo de direcciones IP hasta técnicas avanzadas como el marcado de paquetes y métodos de entropía. También se consideran soluciones prácticas como el uso de firewalls, ampliación de ancho de banda, equipos de red especializados y servicios de seguridad en la nube. Cada una de estas opciones aporta beneficios específicos, pero también tiene limitaciones que es importante tener en cuenta al implementarlas.

Asharf y colaboradores [5] se centraron en la detección de anomalías en redes utilizando inteligencia artificial y aprendizaje automático, diferenciando entre comportamientos normales y anómalos del tráfico de red. Se presentaron diversos métodos y modelos diseñados para fortalecer la seguridad de los sistemas; de igual manera, se evaluó el desempeño de estas técnicas empleando datos reales de tráfico de red, mostrando los resultados obtenidos en la detección de intrusiones en distintos entornos. Gracias al uso de estas tecnologías avanzadas, es posible identificar ataques nuevos y de gran magnitud, incluyendo aquellos denominados de día cero, aumentando significativamente la capacidad de respuesta ante amenazas emergentes.

En este sentido, implementar un sistema de detección de ataques de denegación de servicio en dispositivos IoT para el hogar mediante herramientas de código abierto se perfila como una solución práctica y necesaria. Esta solución puede apoyarse en algoritmos de aprendizaje automático o técnicas de detección de anomalías, integrándose con plataformas ya existentes como Suricata, Snort u otros IDS (ya sea a nivel de host o de red), e incluso con los gateways domésticos; siendo el objetivo el de construir una arquitectura que supervise el tráfico, identifique patrones de ataque, genere alertas y reaccione (bloqueando o mitigando) sin incurrir en un coste elevado, manteniendo al mismo tiempo la integridad, disponibilidad y rendimiento de los dispositivos IoT en el hogar [6].

1.2 Planteamiento del Problema

En la actualidad, es evidente que los dispositivos IoT presentes en los hogares están comprometidos en cuestión de minutos tras conectarse a la red, principalmente debido a que se establecen configuraciones por defecto y/o se utilizan credenciales/contraseñas débiles, es así que las investigaciones realizadas reportan que este tiempo puede llegar a ser demasiado corto (5 minutos), lo que los convierte en objetivos atractivos para los ciberdelincuentes, quienes aprovechan estas brechas para conformar redes botnets [7]. Esta situación se agravó en los últimos años; por ejemplo, en 2020 se registró un incremento significativo de ataques basados en variantes del malware Mirai, capaces de explotar vulnerabilidades comunes en dispositivos domésticos interconectados [8]. Estos incidentes son un claro ejemplo de que, en la búsqueda de un estilo de vida más interconectado, es imprescindible reforzar la ciberseguridad en el entorno doméstico.

Casos ampliamente documentados, como el ataque de la botnet Mirai, evidencian el impacto que pueden tener miles de dispositivos IoT comprometidos, utilizados como nodos en ataques masivos contra infraestructuras y servicios globales, dichos incidentes demuestran que los hogares inteligentes, al ser nodos de redes cada vez más interconectadas, pueden convertirse en vectores de ataque que afectan no solo a los propios usuarios, sino también a infraestructuras críticas más amplias [1].

A pesar de la magnitud de este riesgo, muchos dispositivos IoT en el hogar no tienen soluciones de seguridad sofisticadas; los sistemas de detección de intrusiones (IDS) que son comerciales tienden a ser caros o a necesitar configuraciones complicadas, las cuales no siempre son factibles en contextos domésticos, en este escenario, la literatura ha señalado a las herramientas de código abierto como Suricata, Snort o Zeek, así como a los algoritmos de aprendizaje automático y a las técnicas para detectar anomalías, como opciones 3

alentadoras para crear soluciones que sean asequibles, adaptables y económicas y que refuercen la seguridad en redes domésticas [9]

Por lo tanto, el problema central de esta investigación radica en determinar cómo desarrollar e implementar un sistema de detección de ataques de denegación de servicio (DoS) en entornos domésticos IoT que sea eficaz, asequible y de bajo consumo de recursos.

En este contexto, surge la pregunta de investigación:

¿Cómo implementar un sistema de detección de ataques de denegación de servicio en dispositivos?

1.3 Justificación

El internet de las cosas es un fenómeno que se ha visto como una auténtica revolución tecnológica y que ha impactado de manera significativa el ámbito de las comunicaciones. En [10] mencionan que IoT o Internet de las Cosas puede comprenderse como una red inteligente que permite la conexión a Internet de distintos objetos y dispositivos, con el fin de posibilitar el intercambio de datos y la comunicación entre sí por medio de sensores, persiguiendo reglas y protocolos previamente establecidos. En este mismo contexto [11], destacan que el IoT representa una evolución constante de la red, en la que los objetos de uso diario adquieren la capacidad de comunicarse entre sí, transmitiendo y recibiendo información.

El fenómeno del crecimiento llevó a un aumento en la demanda de dispositivos "inteligentes" que pueden acceder a internet. Como publicó el análisis de [12], en ese año en el mundo había aproximadamente 7 mil millones de dispositivos IoT conectados, y para el 2025, se pronosticaba la cantidad total de dispositivos en 22 mil millones. Hoy en día es común encontrar sensores, cámaras, asistentes virtuales, electrodomésticos inteligentes y muchos otros aparatos que ya forman parte de la vida cotidiana, haciendo que las casas sean más cómodas, eficientes y automatizadas; sin embargo, junto con estos beneficios también

aparecen riesgos importantes relacionados con la seguridad. La mayoría de los dispositivos IoT poseen recursos limitados —como baja capacidad de procesamiento, memoria reducida y autonomía energética restringida—, lo que los hace vulnerables frente a ataques y explotación de sus debilidades de seguridad [13]. Así mismo, las actividades principales de los aparatos IoT abarcan la adquisición, gestión, tratamiento y conservación de datos, muchos de los dispositivos se vuelven cada vez más compactos, lo que los hace depender de puertas de enlace para comunicarse y llevar a cabo sus funciones.

Herramientas de código abierto como Suricata, Snort y Zeek han demostrado eficacia en la detección de intrusiones en redes IoT. Estas soluciones combinan detección basada en firmas y en anomalías, ofreciendo flexibilidad, bajo costo y adaptabilidad para entornos domésticos [14].

La incorporación de algoritmos de aprendizaje automático en sistemas IDS puede mejorar significativamente la detección de ataques DDoS al identificar patrones complejos y comportamientos anómalos en el tráfico de red; sin embargo, es crucial seleccionar técnicas que sean eficientes en términos de consumo de recursos, dado que los dispositivos IoT suelen tener capacidades limitadas. Modelos como Random Forest, Support Vector Machine (SVM) y K-Nearest Neighbor (KNN) han mostrado buenos resultados en la clasificación de tráfico legítimo y malicioso [15].

Por lo anterior, es que se justifica una investigación para proponer una arquitectura de detección de ataques DDoS en hogares IoT usando herramientas open source y así proveer una solución accesible, personalizable y adaptable a las necesidades de seguridad de cada hogar.

1.4 Objetivos

1.4.1 General

- Diseñar e implementar un sistema de detección de ataques de denegación de servicios (DoS) en dispositivos IOT para aplicaciones domésticas, utilizando herramientas open source.

1.4.2 Específicos

- Identificar los principales tipos de ataques de denegación de servicios que se presentan comúnmente en dispositivos IoT para conocer el funcionamiento de estos ataques y determinar cómo afrontar este tipo de ataques.
- Diseñar e implementar el dispositivo funcional empleando una herramienta open source.
- Evaluar el dispositivo aplicando pruebas controladas de ataques al dispositivo IoT en entornos privados para determinar el correcto funcionamiento y mediante pruebas estadísticas.

CAPITULO II MARCO TEÓRICO

2.1 Estado del Arte

2.1.1 Internet de las Cosas IoT

En sus inicios, Internet se concebía como una gran red global de computadoras diseñada para facilitar el intercambio de información y ofrecer múltiples servicios a los usuarios. Con el crecimiento exponencial de esta red surgió el concepto de Internet de las Cosas (IoT), una extensión natural que busca conectar objetos físicos al entorno digital. Esta nueva etapa representa una evolución en la que no solo las computadoras están interconectadas, sino también una amplia gama de dispositivos, lo que hace posible la conectividad en múltiples aspectos de la vida cotidiana. El Internet de las Cosas (IoT) se origina en la intersección entre los sistemas embebidos, las redes de comunicación y la computación ubicua, la noción de vincular objetos físicos con el ámbito digital se empezó a desarrollar a fines del siglo XX, cuando tecnologías como las redes de sensores inalámbricos y la identificación por radiofrecuencia (RFID) comenzaron a investigarse como métodos para supervisar y controlar objetos [16]. En 1982, se registró uno de los primeros ejemplos de IoT: una máquina expendedora de Coca-Cola que se muestra en la Figura 1 conectada a Internet en la Universidad Carnegie Mellon (CMU), la cual permitía monitorear la temperatura y disponibilidad de productos en tiempo real.

Fig. 1. Máquina de Coca Cola de la MCU

Nota: fuente https://www.ibm.com/think/topics/iot-first-device.

El avance del protocolo IPv6, junto con la miniaturización de sensores y componentes de hardware, impulsó la expansión del IoT a principios de la década del 2000. Estas innovaciones permitieron desarrollar sistemas más escalables y económicos, facilitando la interconexión de dispositivos en diversos sectores [17]. El ecosistema IoT se robusteció en la década del 2010 gracias al avance de la computación en la nube, a la implementación de la banda ancha y a que los teléfonos inteligentes se expandieron, es así que, de hecho, la ITU (International Telecommunication Union) lo identificó como un elemento clave para digitalizar el transporte, la salud, las ciudades y la industria [18].

En la actualidad, la arquitectura del IoT se orienta hacia esquemas híbridos donde la computación no se limita a la nube, sino que se complementa con el fog computing y el edge computing, reduciendo la latencia y mejorando la eficiencia energética al procesar datos cerca de la fuente. Un concepto para resaltar es el de Inteligencia artificial periférica, el mismo que se refiere a la combinación/ Integración del IoT con la Inteligencia artificial, permitió lograr obtener respuestas inmediatas, ahorro de ancho de banda y protección de la privacidad, todo esto mediante la ejecución de algoritmos de aprendizaje automático directamente en dispositivos locales [19].

La llegada del 5G fortaleció el ecosistema IoT al proporcionar velocidades de transmisión más altas, baja latencia y mayor densidad de conexión entre dispositivos, posibilitando nuevas aplicaciones en manufactura inteligente, vehículos conectados y salud digital. El despliegue (ya surgidas en países desarrollados) de tecnologías como 6G y las redes definidas por software (SDN), se colocan como las favoritas y claves para mejorar la interoperabilidad y escalabilidad de estos sistemas. Otros sectores también se han beneficiado del IoT, pues muestra avances significativos en la salud (IoMT), la industria (IIoT), las ciudades inteligentes y la logística. En el tema de salud, por ejemplo, los dispositivos permiten monitoreo remoto de pacientes, aunque presentan desafíos 8

regulatorios y de ciberseguridad. En el sector industrial, se puede optimizar procesos para de esta manera controlar y reducir los tiempos de inactividad gracias al IoT en conjunto con analítica predictiva [20].

La Figura 2 ilustra la evolución del número de dispositivos conectados a IoT entre 2019 y 2030, evidenciando una tendencia ascendente que proyecta la existencia de más de 30 mil millones de equipos inteligentes interconectados en el mundo.

Fig. 2. Evolución del uso de dispositivos conectados a IoT. Años 2019 - 2024, con previsiones de 2024 a 2030

Nota: Tomado de [13].

2.1.2 Ciberataques

Los ataques cibernéticos se presentan de muchas formas y se han convertido en una amenaza común a medida que nos volvemos más dependientes de la infraestructura y los dispositivos digitales, lo que afecta tanto a personas como a organizaciones; por ende, un ciberataque es un intento deliberado y malicioso por parte de una persona o grupo de irrumpir en los sistemas de información de organizaciones o individuos para robar, alterar o cambiar datos; sin embargo, con la rápida introducción de nuevas tecnologías, el número de ciberataques está aumentando. Para proteger la información personal y corporativa, es fundamental comprender los diferentes tipos de ciberataques a empresas o individuos y sus posibles consecuencias [21], a continuación, se presentan los tipos de ciberataques más comunes en diversas infraestructuras tecnológicas:

- **Ataque de malware**: se trata de software malicioso que abarca programas diseñados para interrumpir, dañar u obtener acceso no autorizado a los sistemas informáticos, esto incluye virus, gusanos, ransomware, spyware y adware; Una vez instalado, realiza actividades maliciosas como robo de datos, secuestro del sistema y desactivación de dispositivos. Puede operar de manera sigilosa para evitar ser detectado, explotar vulnerabilidades de software o utilizar tácticas de ingeniería social para engañar a los usuarios para que lo instalen accidentalmente, creando importantes riesgos de ciberseguridad y protección de datos. La eliminación de malware normalmente implica el uso de software antivirus especializado para escanear, detectar y poner en cuarentena o eliminar archivos o programas maliciosos y restaurar los dispositivos infectados a un estado seguro [21].

- **Phishing:** es un tipo de ciberataque que consiste en el envío de correos electrónicos genéricos por parte de ciberdelincuentes que se hacen pasar por legítimos. Estos correos electrónicos contienen enlaces fraudulentos para robar información privada del usuario. Los ataques de phishing son más eficaces cuando los usuarios no saben lo que están sucediendo.

- **Ataques de hombre en el medio (MitM):** por sus siglas en inglés, es un tipo de ataque que involucra un elemento malicioso que "escucha" las comunicaciones entre las partes y es una amenaza importante para las organizaciones. Estos ataques comprometen los datos enviados y recibidos, ya que los espías no sólo pueden acceder a la información, sino también inyectar sus propios datos; dada la importancia del flujo de información dentro y fuera de una organización, los ataques MiTM son una amenaza muy real y poderosa que los profesionales de TI deben poder afrontar [22].

- **Inyección SQL:** es un ataque que manipula ilegalmente una base de datos mediante la inyección de sentencias SQL no deseadas en una aplicación que tiene una base de datos relacional (RDBMS), dependiendo del método y el propósito, existen varios tipos de inyección SQL y, desde la perspectiva de un ciberatacante, pueden variar desde el robo de

información hasta la falsificación de datos y la detección de vulnerabilidades. Aunque se trata de un ataque antiguo, todavía causa mucho daño en la actualidad, lo que lo convierte en uno de los ataques de los que las organizaciones empresariales deberían tener especial cuidado [23].

- **Ataque de ransomware:** es un malware que cifra archivos importantes en el almacenamiento local y de red y exige un rescate para descifrarlos, aquí los piratas informáticos desarrollan este malware para ganar dinero mediante la extorsión digital. El ransomware está cifrado, por lo que la clave no se puede descifrar y la única forma de recuperar la información es mediante una copia de seguridad. Otros tipos de malware destruyen o roban datos, pero permiten otras opciones de recuperación, con el ransomware, si no tiene copias de seguridad, debe pagar un rescate para recuperar sus datos. Algunas empresas pagan el rescate y los atacantes no envían la clave de descifrado. Cuando el ransomware comienza a ejecutarse, escanea el almacenamiento local y de red en busca de archivos para cifrar. Esto es para archivos que cree que son importantes para su empresa o su gente. Esto incluye archivos de respaldo que pueden ayudar a restaurar la información [24].

- **Ingeniería social:** es un tipo de ataque que utiliza la interacción y manipulación humana para lograr los objetivos del atacante, esto a menudo implica persuadir a las víctimas para que comprometan su seguridad o violen las mejores prácticas de seguridad para obtener beneficios financieros o informativos del atacante. Los actores de amenazas utilizan la ingeniería social para disfrazarse a sí mismos y sus motivos, a menudo haciéndose pasar por personas de confianza; después de todo, el objetivo principal es influir, piratear la mente, no el sistema. Muchas de estas hazañas dependen del buen carácter de las personas o del miedo a situaciones negativas, la ingeniería social es popular entre los atacantes porque las personas son más fáciles de explotar que las vulnerabilidades de la red y del software [25].

- **Ataques basados en IoT:** El rápido crecimiento del IoT ha introducido nuevos desafíos de seguridad. Muchos dispositivos carecen de mecanismos de protección robustos, lo que facilita su incorporación a redes de bots (botnets) utilizadas para lanzar ataques DDoS o infiltrarse en redes más amplias [26].

2.1.3 Ataques de Denegación de Servicio DoS

Los ataques de denegación de servicio (DoS) son una de las amenazas más comunes para la seguridad de redes y sistemas informáticos. Busca saturar un servidor o una red con tráfico malicioso, bloqueando el acceso legítimo.

La Denegación de Servicio Distribuido (DDoS) viene siendo la versión mejorada y sofisticada de DoS, en donde varios dispositivos (llamados bots), actúan de manera coordinada para generar un flujo masivo de solicitudes maliciosas. DDos emplea una estrategia distribuida cuya consecuencia es la amplificación del impacto, llegando a afectar múltiples objetivos simultáneamente. En conclusión, DoS se diferencia principalmente del DDoS, debido que este último aprovecha una red de dispositivos para ejecutar el ataque de forma conjunta. La Tabla 1 resume las diferencias de ambos tipos de ataques [13].

TABLA I
DIFERENCIAS ENTRE DOS Y DDOS

Ataques DoS	**Ataques DDos**
No distribuidos	Distribuidos
Impacto de calidad baja	Impacto de calidad severa
Relativa lentitud Proceso inmediato	Relativa lentitud Proceso inmediato
Origen único Origen múltiple	Origen único Origen múltiple

Nota. Tomado de [13].

Como se observa, los ataques DDoS superan a los DoS tradicionales en términos de impacto y rapidez, lo cual se explica principalmente por la ventaja que ofrece la distribución de los dispositivos que ejecutan el ataque.

La primera incidencia de un ataque a gran escala registrada se remonta a 1996, cuando el proveedor de servicios de Internet Panix fue víctima de un ataque denominado SYN flood, o inundación de mensajes de sincronización. Dicho ataque logró con el objetivo propuesto que era la interrupción de los servicios durante aproximadamente 36 horas, después de que se saturara la red con un alto volumen de mensajes de sincronización. La Figura 3 refleja esquemáticamente la dinámica de este ataque, donde queda en evidencia el impacto y la magnitud del evento en la infraestructura de la red [13].

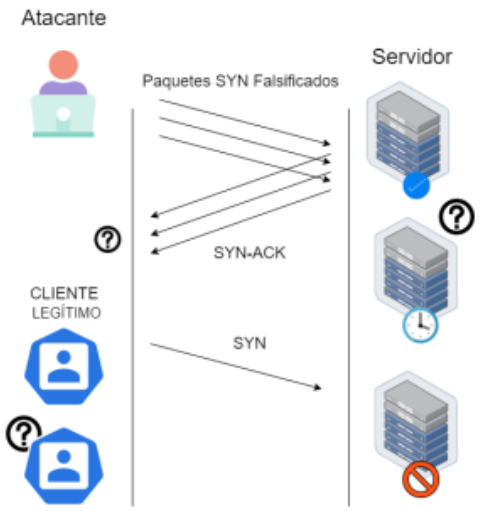


Fig. 3. Ataque SYNC Flood

Nota: Tomado de [13].

Con el paso del tiempo, se logró evidenciar la evolución y ejecución de los ataques de DoS, donde un gran número de dispositivos IoT fueron transformados en bots con el único fin de generar tráfico malicioso, incrementando exponencialmente el volumen de datos de GB a TB. A la par, el desarrollo de tecnologías de comunicación de alta velocidad como las redes 5G trajo consigo la posibilidad de realizar intrusiones más rápidas y efectivas, al mismo tiempo que los atacantes logran mantenerse prácticamente imperceptibles dentro de la infraestructura de la red. Es así como formalmente, el concepto de DoS/DDoS surgió en la década de los noventa, cuando los sistemas conectados a internet empezaron a masificarse.

En el año de 1999, ocurrió uno de los primeros incidentes documentados, cuando un ataque DDoS dirigido contra la Universidad de Minnesota interrumpió su red académica [27]; desde entonces, la evolución de este tipo de ataques ha venido dando tanto en complejidad como escalabilidad, ya no llegando a afectar solamente a servidores individuales, sino que comprometiendo seriamente a infraestructuras sensibles y servicios masivos en línea.

Los primeros enfoques de detección se basaban en sistemas tradicionales de Intrusion Detection Systems (IDS) como Snort y Suricata, herramientas open source que analizaban patrones de tráfico para identificar firmas conocidas de ataques [28]. No obstante, estas soluciones resultaban insuficientes frente a los ataques modernos que usan tráfico camuflado, técnicas de baja frecuencia o combinaciones de múltiples vectores.

Con la evolución de la investigación, surgieron métodos basados en anomalías y aprendizaje automático, los cuales permiten diferenciar comportamientos normales y anormales en la red doméstica. El sistema DÏoT es un claro ejemplo donde demostraron que era posible construir perfiles de comportamiento por dispositivo y detectar desviaciones en tiempo real, incluso en dispositivos con recursos limitados, utilizando un enfoque de aprendizaje federado para agregar perfiles de comportamiento de forma eficiente. Hasta donde se sabe, es el primer sistema que emplea un enfoque de aprendizaje federado para la detección de intrusiones basada en la detección de anomalías. En consecuencia, DÏoT puede hacer frente a ataques emergentes, nuevos y desconocidos [29].

Las arquitecturas de seguridad comenzaron a usar herramientas de código abierto (open source) como ELK Stack (Elasticsearch, Logstash y Kibana), con el fin verificar el tráfico, llegar a correlacionar eventos y de esta manera generar alertas automáticas [30]. Estas herramientas, sumadas a NIDS como Suricata y motores de correlación, han hecho posible la implementación de sistemas de detección adaptados al ámbito doméstico, con costos accesibles y escalabilidad.

El enfoque del tema de detección de ataques actualmente es lograr la integración de modelos híbridos, los mismos que combinan técnicas tradicionales de inspección con enfoques Machine Learning y Deep Learning. Estos modelos tienen como objetivo de que buscan mejorar la precisión, reducir falsos positivos y anticipar comportamientos de ataques emergentes [31]. Es así como, además, se exploran esquemas distribuidos y de federated learning (aprendizaje federado) para que el procesamiento pueda realizarse parcialmente en el hogar sin comprometer privacidad ni rendimiento.

2.2 Marco Teórico

2.2.1 Seguridad en IoT

La seguridad en el Internet de las Cosas (IoT) no solo constituye un desafío técnico, sino también un compromiso con la protección de la privacidad y los entornos personales. Cada dispositivo conectado recopila y transmite información sensible, lo que convierte la seguridad en un componente esencial de la vida digital moderna [32].

El empleo de métodos de autenticación deficientes (no MFA), software obsoleto y/o cifrado ineficaz se convierten en peligro potente para la seguridad en IoT debido a las constantes amenazas, riesgos y ataques cibernéticos. Es por ello, que es menester adoptar una perspectiva de seguridad en todos los niveles de la arquitectura, para lo mismo, se han creado protocolos de seguridad, como el Protocolo de Seguridad de Capa de Protección (SSL) teniendo en cuenta las capas de aplicación y transporte; sin embargo, las medidas de protección se basarán en la tecnología, el estándar Wi-Fi, por ejemplo, representa un riesgo significativo para la seguridad [13].

2.2.2 Federated learning (FL)

El entrenamiento, las pruebas y la validación de modelos de aprendizaje automático requieren datos, éstos a veces se encuentran dispersos entre muchos, incluso millones de, participantes (dispositivos). El aprendizaje federado es una forma relativamente nueva de

desarrollar modelos de aprendizaje automático, donde cada dispositivo federado comparte los parámetros de su modelo local en lugar de compartir todo el conjunto de datos utilizado para entrenarlo. La topología de aprendizaje federado define cómo se comparten los parámetros, en una topología centralizada, los participantes envían los parámetros de su modelo a un servidor central que los utiliza para entrenar un modelo central, el cual, a su vez, les envía parámetros actualizados. En otras topologías, como la peer-to-peer o la jerárquica, los participantes comparten sus parámetros con un subconjunto de sus pares. El aprendizaje federado es una posible solución para desarrollar modelos de aprendizaje automático que requieren conjuntos de datos enormes o muy dispersos. Sin embargo, no es una solución universal para todos los escenarios de aprendizaje automático [33], actualmente presenta problemas abiertos que los científicos e ingenieros trabajan arduamente para resolver, algunos de los cuales se mencionan a continuación:

- **Eficiencia de la comunicación:** FL implica numerosas transferencias de datos, por lo tanto, el servidor central o las partes que reciben los parámetros deben ser resilientes a fallos y retrasos en la comunicación, esto con el fin de garantizar una comunicación y sincronización eficientes entre los dispositivos federados, tema que sigue siendo aún explotado y relevante.
- **Heterogeneidad de dispositivos:** las capacidades de computación de las partes federadas suelen ser heterogéneas y, en ocasiones, desconocidas para las demás partes o el servidor central. Sigue siendo difícil garantizar que las tareas de entrenamiento funcionen en un conjunto heterogéneo de dispositivos.
- **Heterogeneidad de los datos:** Los conjuntos de datos de los partidos federados pueden ser muy heterogéneos en términos de cantidad, calidad y diversidad. Es difícil medir de antemano la heterogeneidad estadística de los

conjuntos de datos de entrenamiento y mitigar los posibles impactos negativos que dicha heterogeneidad podría tener.

- **Privacidad:** es necesario implementar de manera eficiente tecnologías que mejoren la privacidad para evitar fugas de información de los parámetros del modelo compartido.

2.2.3 Vulnerabilidad

Hay muchas formas de definir una vulnerabilidad, pero, en esencia, es como una herida abierta en un sistema, una grieta en la armadura que protege nuestros activos y controles. Según [28], una vulnerabilidad es una debilidad que puede ser aprovechada por una o más amenazas, como si el sistema dejara al descubierto su lado más frágil, su punto débil; también se puede entenderla como la capacidad que tiene una red o infraestructura de verse afectada negativamente por diversas amenazas, como si, pese a su complejidad y fortaleza, pudiera sucumbir ante un ataque inesperado.

2.2.4 Hping3

Hping3 es una herramienta de análisis y generación de paquetes personalizada que admite protocolos ICMP, TCP y UDP, que funciona de manera parecida a un ping, pero con la diferencia que tiene la capacidad de manipular los paquetes a nivel más granular, además permite diseñar paquetes ajustando el tamaño, contenido y velocidad de transmisión, lo que facilita pruebas más específicas. También Hping3, puede usarse para evaluar la seguridad de redes, como pruebas contra firewalls e IDS (Sistemas de Detección de Intrusos), así como para medir el rendimiento de la red con distintos protocolos. Entre sus funcionalidades, incluye una función avanzada de traceroute para mapear la ruta de los paquetes. Hping3 se maneja desde la línea de comandos (terminal/cmd) y ofrece parámetros detallados para controlar todas sus funcionalidades y opciones [35].

2.2.5 Raspberry Pi

Raspberry Pi es una computadora económica con un tamaño compacto, del tamaño de una tarjeta de crédito, que se puede conectar a un monitor de computadora o TV y usar con un mouse y teclado normal, está integrada con sistema operativo Linux que permite a personas de todas las edades explorar la informática y aprender lenguajes de programación como Scratch y Python, además puede manejar la mayoría de las tareas típicas de escritorio, desde navegar por la web, reproducir vídeos de alta definición, manipular documentos de Office hasta jugar. También, Raspberry Pi tiene la capacidad de interactuar con el mundo exterior, puede usarse en una amplia variedad de proyectos digitales, desde reproductores de música y video, detectores de padres, estaciones meteorológicas hasta cajas para pájaros con cámaras infrarrojas.[36].

Hoy, aproximadamente 10 años tras su lanzamiento, la Raspberry Pi es más potente, se presenta superior vinculada y resulta más versátil. La nueva Raspberry Pi 5 cuenta con un procesador más rápido, puertos USB 3.0, mejor conectividad (Wi-Fi 5 y Bluetooth 5.0), dos salidas HDMI 4K a 60 Hz y soporte para PCIe. Se compara al rendimiento de una computadora de escritorio, pero en un tamaño pequeño y a bajo costo. Esta evolución ha aumentado mucho sus aplicaciones: ya no solo se usa en el aula o en pasatiempos, sino que ahora impulsa servidores en casa, paneles de control, quioscos interactivos, sistemas embebidos, automatización industrial, cuadros IoT y estaciones multimedia.

La familia Raspberry Pi ha crecido, no se trata solo del "modelo de tarjeta de crédito": hoy también existen Raspberry Pi Pico, microcontroladores ultraeconómicos que permiten crear proyectos de bajo consumo, rápidos y programables en MicroPython o C/C, ideales para robótica, wearables, automatización y productos integrados. Su tamaño aún más pequeño y su precisión de control de hardware los convierten en una excelente alternativa para proyectos que no requieren un sistema operativo completo, pero sí potencia y

flexibilidad. Raspberry Pi no es sólo un dispositivo, sino un universo abierto de posibilidades tecnológicas que crece con una comunidad global de creadores, educadores y profesionales [36].

Fig. 4. Raspberry Pi

Nota: fuente https://raspberrypi.c

2.2.6 Ataque de denegación de servicio (DoS)

Según [30], un ataque a la seguridad de red consiste en el acceso no autorizado o la alteración de los servicios legítimos de un sistema, lo que lo expone a vulnerabilidades que comprometen su funcionamiento.

DoS busca hacer que un recurso específico de un servidor quede completamente fuera de nuestro alcance. Imagina querer visitar un lugar especial, pero encontrar las puertas cerradas con un cartel que dice "No disponible", eso es, en esencia, lo que logra un ataque DoS. Los ciber atacantes, empleando herramientas que envían una avalancha de solicitudes y datos, saturan el servidor con tanta información que éste, al verse abrumado, ya no puede responder a quienes desean acceder a su contenido, así, un servicio que debería estar al alcance de todos se convierte en un espacio vacío e inaccesible [38].

2.2.6.1 Ataques de Denegación de Servicio Distribuido (DDoS)

A diferencia de los ataques DoS tradicionales, los ataques DDoS (Distributed Denial of Service) emplean múltiples equipos comprometidos conocidos como bots que actúan simultáneamente desde diferentes ubicaciones, generando un tráfico masivo hacia el objetivo y dificultando su defensa. Este tipo de ataque inunda el sistema con un flujo incesante de

solicitudes sin sentido, dificultando que se atiendan las verdaderas necesidades de quienes dependen de esos recursos. La escala de estos ataques puede ser abrumadora; algunos bots pueden trabajar juntos utilizando miles de máquinas, lo que convierte la respuesta en un verdadero reto. Por eso, es esencial que las empresas dediquen tiempo y recursos a reforzar sus redes con medidas de seguridad adecuadas, cuidando así su confianza y el bienestar de sus operaciones [3].

2.2.6.2 Inundación SYN (Sincronización TCP)

Un ataque de inundación SYN (SYN Flood) explota la vulnerabilidad del proceso de conexión TCP. El atacante envía un gran número de solicitudes de sincronización (SYN) sin completar el protocolo de enlace, provocando que el servidor mantenga abiertas las conexiones y agote sus recursos. En consecuencia, el sistema víctima queda inoperativo al no disponer de recursos suficientes para atender solicitudes legítimas, generando pérdida de conectividad y servicio [39].

2.2.6.3 Inundación UDP

Este tipo de ataque DDoS afecta en profundidad a la infraestructura de la red, provocando una "inundación" mediante una avalancha de paquetes de protocolo de datagramas de usuario (UDP); a diferencia de TCP, UDP carece de conexión y no necesita un protocolo de enlace, lo que facilita a los atacantes generar enormes volúmenes de tráfico con un esfuerzo mínimo. En este escenario, el atacante lanza una multitud de paquetes UDP hacia el servidor o red objetivo, con frecuencia empleando direcciones IP falsas para dificultar el rastreo. Debido a la ausencia de mecanismos en UDP para asegurar la entrega de paquetes o verificar destinatarios, la infraestructura debe lidiar con cada paquete que llega, aquello resulta en una inundación devastadora de paquetes UDP que consume el ancho de banda, la capacidad de procesamiento y otros recursos provocando congestión, lentitud

en la red, intermitencias y hasta interrupciones en el servicio para los usuarios legítimos que buscan comunicarse con la infraestructura afectada [40].

2.2.6.4 Inundación ICMP

Las inundaciones ICMP (Ping Flood) son ataques DDoS que envían grandes volúmenes de solicitudes echo request del protocolo ICMP (Internet Control Message Protocol) a un dispositivo o red, con el fin de saturar su capacidad de respuesta y provocar pérdida de conectividad. Este tipo de tráfico masivo provoca lentitud, alta latencia y, en casos extremos, la interrupción total del servicio. [40]

2.2.7 Sistema de detección de intrusos (IDS)

El sistema de detección de intrusos (IDS) es una aplicación que monitorea el tráfico de la red y está buscando algunas amenazas y actividades sospechosas o maliciosas, cuando se detectan riesgos y amenazas de seguridad, los IDS envían advertencias a los equipos de Ti y de seguridad

Hay varios tipos de IDS: NIDS (observa la red en busca de tráfico sospechoso), HIDS (analiza un host en busca de intrusiones) e IPS (IDS activo que intenta detener ataques en tiempo real). Los IDS hacen uso de métodos básicos de seguridad, registrando todo movimiento y guardándolo en bases de datos, verificando dispositivos y alertando al usuario de cualquier suceso sospechoso. Los IDS además se enfocan en comparar paquetes con reglas preestablecidas para identificar amenazas, equipados con sondas distribuidas por la red, operan sin interrumpir la comunicación. Su mayor fortaleza radica en su capacidad de monitorear toda la infraestructura, lo que permite detectar ataques y patrones sin ser percibidos, brindándonos la calma en un mundo lleno de incertidumbres [15].

2.2.8 Herramientas de capturas de datos

La efectividad de los IDS se basa en su capacidad de capturar y analizar el tráfico, para ello, se requieren programas que desempeñan un papel crucial en la detección y

monitorización de la seguridad, protegiendo nuestro espacio digital. Existen numerosas herramientas de captura de tráfico, cada una con sus propias ventajas y desventajas, las más exploradas y valiosas son de código abierto y accesibles económicamente, haciendo que gracias a sus diseños ligeros puedan ser procesadas por una Raspberry Pi, ofreciendo una interfaz amigable que facilita su uso y configuración.

Snort

Es un IDS/IPS de código abierto para detectar y prevenir intrusiones en la red en tiempo real, analizando el tráfico en busca de patrones maliciosos y alertando sobre amenazas potenciales, como accesos no autorizados o malware.

Snort es más que un simple sistema de detección de intrusos, se trata de un guardián digital valiente que protege los espacios en línea, el mismo que fue creado en 1998 de la mano de Martin Roesch y aún su esencia sigue viva bajo el ala de Cisco, reflejando un compromiso profundo con la seguridad. En su lenguaje de reglas, se encuentra una forma de comunicar lo que queremos proteger, permitiendo que cada usuario personalice su entorno y se sienta seguro. La última versión, Snort 3.0, ofrece versatilidad: puede actuar como un sniffer, un registrador de paquetes o un NIDS, su diseño accesible permite que cualquiera, sin importar el sistema operativo, pueda adoptar esta herramienta [23]. Con cada modo de operación y opción configurable, Snort se convierte en un aliado cercano, siempre listo para defender nuestro mundo digital [41].

Suricata

Suricata constituye un protector en la navegación digital, un motor de red que asegura conexiones con eficiencia, creado por la comunidad Open Information Security Foundation (OISF); es de código abierto. Su diseño escalable y capaz de distribuir la carga entre procesadores hace que cada byte importe, alcanzando hasta 10 gigabits por segundo sin perder rendimiento.

Suricata cumple con el objetivo de alertar a los administradores ante cualquier amenaza gracias a un conjunto de reglas que actúa como su modelo motivo por el cual se lo describe como un guardián de red potente.

El diseño de Suricata admite la integración con otras herramientas de seguridad dado que está en constante evolución, incluso en su última versión (4.0), siempre en miras a mejorar sus capacidades para detectar intrusos y adaptarse a un mundo en constante cambio. Es así como, Suricata no solo se trata de tecnología, sino que de una promesa de un entorno digital más seguro [42].

2.2.9 Características de Suricata

Las reglas de Suricata más que simples directivas son el corazón palpitante de un sistema que vela por la seguridad en el vasto campo de datos. Cada regla actúa como un guardián, especificando qué tráfico debe ser monitoreado y qué pasos deben seguirse cuando algo sospechoso cruza nuestras fronteras digitales. La estructura de Suricata se compone de tres partes fundamentales que son: la acción, que decide la respuesta; la cabecera, que identifica el tráfico; y las opciones, que brindan detalles específicos. En conjunto forman una “Signature” que refleja la esencia de lo que se quiere proteger.

El formato de las reglas de Suricata se establece de la siguiente manera:

Acción: Define la acción que se ejecuta cuando hay una coincidencia con la firma.

Header: Especifica el protocolo como direcciones IP, puertos de origen y destino de la regla.

Options: Contiene opciones adicionales para ajustar la regla acorde cómo sea el caso. El parámetro "nocase" es un ejemplo en donde se busca una coincidencia sin distinguir entre mayúsculas y minúsculas.

A continuación, se detallan cada uno de los parámetros aplicables a las reglas de Suricata, con descripciones y ejemplos de algunos de ellos.

Action:

- **alert:** se registra una notificación en los eventos y se permite el tráfico.
- **drop:** se elimina el tráfico que coincide con la regla.
- **reject:** se impide el tráfico que coincide con la regla, enviando un mensaje de error ICMP.
- **pass:** se deja pasar el tráfico que coincide con la regla sin registrar una alerta en los eventos.
- **sdrop:** se elimina el tráfico que coincide con la regla sin registrar alerta alguna en los eventos.
- **ignore:** se pasa por alto el tráfico que coincide con la regla sin registrar una alerta.
- **replace:** se modifica el contenido que coincide con la regla antes de permitirlo o descartarlo.
- **log:** es el registro de eventos del tráfico que coincide con la regla sin generar alerta ni bloquear el tráfico.
- **user:** el usuario aplica una acción específica definida en la configuración.

Header:

- **Protocolo:** pueden emplear protocolos como TCP, UDP, ICMP, ICMPv6, IP y ARP en la capa de transporte.
- **Dirección IP** de origen o destino: indican las direcciones IP que marcan el inicio y el destino del tráfico.
- **Puerto de origen o destino:** establece los puertos de origen y destino que guían el tráfico.
- **Contenido:** alude al contenido particular que se desea encontrar en el tráfico, como cadenas de texto o secuencias de bytes.

Options:

- **Distance**: Restringe la búsqueda a un rango específico al establecer la distancia desde la coincidencia más reciente.
- **Offset:** se utiliza para indicar desde qué punto del paquete se debe comenzar a buscar cierta información, omitiendo una cantidad determinada de bytes al inicio.
- **Depth:** sirve para hacer más eficiente la búsqueda dentro de un paquete, limitándola a una cantidad específica de bytes en lugar de revisar todo su contenido.

A continuación, en la Tabla 2 se presentan las principales diferencias encontradas en las dos herramientas de IDS.

TABLA II
SNORT VS SURICATA

Característica	Snort	Suricata
Tipo de Software	Open Source	Open Source
Desarrollador	Martin Roesch / Cisco	Open Information Security Foundation (OISF)
Modo de Operación	Sniffer, Packet Logger, NIDS	IDS, IPS, seguridad en red
Escalabilidad	Limitada	Alta
Manejo de Procesadores	Monohilo	Multihilo
Rendimiento	Menor en ancho de banda	Hasta 10 Gbps
Configuración de reglas	Lenguaje fácil de usar	Soporta reglas externas
Alertas y Notificaciones	Genera alertas	Alertas en tiempo real
Soporte de Protocolos	TCP, UDP, ICMP, HTTP, FTP, DNS, SMB, RDP	TCP, UDP, ICMP, HTTP, FTP, DNS, SMB, RDP, SIP, MQTT, IPv6

Nota: Fuente https://www.stationx.net/suricata-vs-snort/

Tras analizar los sistemas de detección de intrusos, encontramos que Snort y Suricata son más que herramientas; son aliados en la lucha por la seguridad digital, ambas comparten raíces, utilizando las reglas básicas, sin embargo, su principal diferencia radica en que Snort es monohilo y más simple de configurar, mientras que Suricata es multihilo, por lo que

aprovecha mejor los procesadores modernos y ofrece mayor rendimiento en redes de alto tráfico, por ende, Suricata avanza con la rapidez de un río en crecimiento. Para este caso, elegirá Suricata, convencidos de que su capacidad multihilo y escalabilidad aportarán un rendimiento excepcional a nuestra Raspberry Pi 4 Modelo B, esta placa de IoT dispone de un procesador ARM de cuatro núcleos que podrán ser utilizados de manera concurrente por este motor.

2.2.10 IDS/IPS con rol de Gateway

La creación de este esquema es un esfuerzo por proteger una red, dado que, al situar la sonda en el medio de la conexión entre los dispositivos y el acceso de Internet, el sistema se convierte en un guardián silencioso cuyo principal objetivo es detectar el tráfico no deseado, que amenaza la integridad del entorno, y eliminarlo de manera transparente, casi como un susurro que pasa desapercibido, pero resguardando la armonía de toda la infraestructura.

Presentar esta topología implica considerar un búnker: un router que es la puerta al mundo exterior y una Raspberry Pi simple pero poderosa que se convierte en Access Point y sonda IDS. Combinadas, estas herramientas forman un ecosistema donde cada dato se protege. Este diseño no es solo sobre tecnología; es sobre la responsabilidad de proteger lo creado, garantizando que cada experiencia en línea sea segura y considerada.

2.2.11 Métodos para el procesamiento y la visualización de datos

Hasta ahora se ha investigado el funcionamiento de los sistemas IDS y su capacidad para capturar tráfico de red. La información recopilada se guarda de manera desorganizada, en archivos de tipo log que son difíciles de leer y analizar, por ende, es necesario desarrollar una aplicación que transforme estos datos en información útil, actualizada y accesible para quienes no están familiarizados con los sistemas de información. Es así como, la implementación de un IDS frecuentemente exige la instalación de un SIEM, cuyo término

es una combinación de las siglas SIM (Gestión de Información de Seguridad) y SEM (Gestión de Eventos de Seguridad) [43] . La tecnología SIEM proporciona un análisis en tiempo real de las alertas de seguridad generadas por el hardware y software de red vinculados al IDS, también proporcionan una variedad de funcionalidades, incluyendo el análisis de información de red y dispositivos de seguridad, la gestión de identidades y accesos, la supervisión de vulnerabilidades y amenazas, así como el manejo de registros y bases de datos. El propósito es vigilar/gestionar los privilegios de usuarios/servicios, controlar inventarios y cambios de configuración; proporcionar datos necesarios para auditorías, revisiones y repuestas a incidentes. Las soluciones SIEM están diseñadas para consolidar eventos de seguridad de diversas fuentes organizativas, incluyendo aplicaciones web y servidores; una vez que el SIEM recoge los eventos, procesa los datos para estandarizarlos, interpreta la información normalizada, genera alertas ante anomalías y produce informes necesarios para los administradores, guardando esta información y las reglas del IDS en los distintos logs generados por Suricata.

Elastic Stack

Según [37], Elastic Stack también conocido como ELK, es más que un conjunto de herramientas es un conjunto de herramientas de gestión de seguridad de información que ofrece soluciones gratuitas y eficientes. Se encuentra conformada por los siguientes componentes, mostrados también en la Figura 5:

Elasticsearch: este producto es un motor de búsqueda y análisis RESTfull distribuido que es flexible, fácil de usar, escalable y especialmente rápido, en donde los datos se almacenan en formato JSON. Además, este motor de búsqueda es fácil de aprender e integrar, permitiendo realizar una visualización de datos en tiempo real, pudiendo combinarse con el aprendizaje automático (machine learning) y otorgando compatibilidad con REST, JSON y API.

Kibana: actúa como la interfaz gráfica para consultar, visualizar, crear paneles de control, reseñas, mapas y controlar los datos, facilitando la interpretación de la información, la principal ventaja de Kibana es que es muy fácil de entender para los usuarios novatos y le permite personalizar los gráficos con múltiples colores y también le permite comparar datos con elementos anteriores.

Logstash: este producto funciona como el nexo que unifica todos los flujos de datos, permitiendo un manejo eficiente de los datos, una de las mayores ventajas es que puede recopilar y procesar datos de múltiples fuentes, datos estructurados, no estructurados o parcialmente estructurados, incluidos sistemas, noticias web y servidores de aplicaciones.

Los componentes del Elastic Stack se integran para ofrecer una solución de SIEM más robusta que utilizar Elasticsearch de forma aislada. Su facilidad de configuración y versatilidad lo hacen accesible, incluso para quienes inician en el camino del análisis de datos. La elección de este sistema se basa en su habilidad para analizar datos generados por Suricata, además de recolectar y procesar registros de diversas fuentes. Ofrece una interfaz intuitiva que se adapta a las necesidades del usuario, como la notable su capacidad de instalación en una Raspberry Pi, un dispositivo económico y accesible. La sólida comunidad en línea respalda este ecosistema, proporcionando guías y recursos útiles para los usuarios.

Fig. 5. Gestión de ELK

Nota: Tomado de [44].

CAPITULO III METODOLOGIA

El presente trabajo investigativo está conformado por la siguiente metodología.

3.1 Tipo de Investigación

El presente trabajo combina distintos enfoques metodológicos debido a su naturaleza teórico-práctica, lo que permite abordar el problema desde una perspectiva integral, en este sentido, se desarrolla una investigación exploratoria, ya que permite identificar y analizar herramientas open source potencialmente útiles para el desarrollo del sistema propuesto; explorar el panorama de soluciones disponibles favorece la selección óptima del sistema de detección de intrusos (IDS) a implementar.

Además, se usa una evaluación descriptiva, para medir y caracterizar el desempeño del sistema en términos de detección de ataques, tiempos de respuesta y eficiencia. Esta perspectiva es crucial en las fases de implementación.

Finalmente, la investigación es aplicada, ya que pretende solucionar un problema real creando una solución práctica: un sistema efectivo de detección de ataques DoS para dispositivos IoT domésticos.

3.2 Diseño de Investigación

La investigación adopta un diseño cuasi-experimental debido a que los experimentos se ejecutarán en un entorno cerrado (laboratorio doméstico), garantizando seguridad y control total de las pruebas durante el proceso de manipulación de variables controladas como intensidad del ataque, número de dispositivos, tipo de tráfico, etc, esto con el fin de observar el comportamiento del sistema.

3.3 Enfoque

La investigación adopta un enfoque mixto, predominantemente cuantitativo, dado que se recopilan y analizan datos numéricos como la tasa de detección, tiempos de respuesta, eficiencia del sistema y tráfico de red para evaluar el desempeño del prototipo.

Mientras que el enfoque cualitativo se dará en elementos como la observación (en el comportamiento del sistema), análisis comparativo entre herramientas open source y la evaluación de facilidad de implementación. Este enfoque mide el rendimiento técnico y a la par permite valorar la usabilidad, eficiencia y replicabilidad del sistema.

3.4 Técnicas de Recolección de Datos

Se basa principalmente en la captura, registro y análisis del tráfico de red creado por los dispositivos IoT en un entorno doméstico imitado. Esta técnica le permite recopilar información cuantitativa relacionada con el comportamiento habitual y anómalo de la red para alimentar y evaluar el sistema de detección de ataque DOS.

Durante todo el proceso de implementación se utilizarán herramientas open source de monitoreo y análisis de red (Wireshark, tcpdump o Zeek) para recolectar paquetes y generar logs de tráfico. Los datos obtenidos incluirán información de cabecera como direcciones IP, puertos, protocolos, tiempos de envío, tamaño de paquetes, etc; y así mismo se tendrán métricas estadísticas (tasa de paquetes, conexiones simultáneas, volumen de datos).

3.5 Población y Muestra

La población de estudio corresponde a los dispositivos IoT presentes en redes domésticas.

La muestra se conforma por dispositivos simulados en un entorno de laboratorio, que representan cámaras IP, sensores de movimiento y otros equipos inteligentes conectados.

3.6 Métodos de Análisis

Para la recolección de información se hizo una revisión bibliográfica de varias sociedades y publicaciones científicas como: IEEE Xplorer, Scopus, Google Scholar, revistas, tesis, dedicadas a vulnerabilidades a dispositivos IoT. De este modo se selecciona

artículos con información relevantes para el diseño de un sistema de detección de ataques de denegación de servicios

El desarrollo del trabajo se realizó por fases (Figura 6), las mismas que se describen a continuación. Cabe aclarar que, durante la práctica, el procedimiento puede sufrir modificaciones ocasionadas por los distintos escenarios de prueba:

La Primera Fase, denominada Investigación y Análisis de ataques DoS en dispositivos IoT, consistió en revisar literatura científica, casos de estudio y reportes de seguridad relacionados con ataques DoS en dispositivos IoT.

En la Fase 2, Diseño del Dispositivo, se identificaron y seleccionaron herramientas open source para la detección de vulnerabilidades, se desarrolló e implementó el dispositivo con estas herramientas.

Fase 3: Pruebas controladas y evaluación del dispositivo; se realizaron pruebas para verificar la capacidad del sistema para identificar y bloquear diferentes tipos de ataques. Aquí se documentaron los hallazgos, desde ataques identificados hasta intervalos de respuesta y cualquier anomalía detectada.

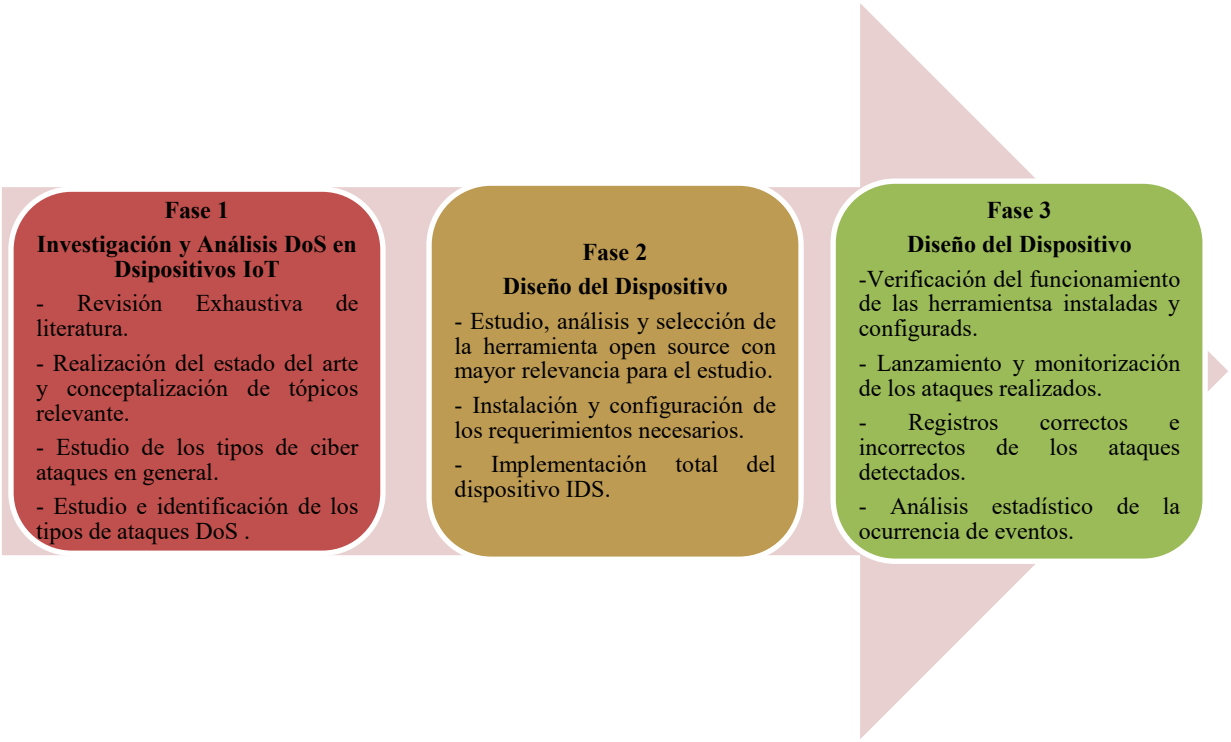


Fig. 6. Diagrama de Flujo-Metodología

3.7 Procesamiento de Datos

El procesamiento de los datos obtenidos se realizó mediante el software estadístico R-4.5.1, debido a su capacidad para manejar grandes volúmenes de información, ejecutar análisis exploratorios, aplicar técnicas de limpieza y transformar datos para modelos de detección.

3.8 Operacionalización de las variables.

Para la variable dependiente Efectividad del dispositivo, que describió el porcentaje entre el número de ataques efectuados y el número de ataques detectados, se presenta lo mostrado en la Tabla 3.

TABLA III
OPERACIONALIZACIÓN VARIABLE DEPENDIENTE

Variable	Dimensiones	Indicadores	Instrumento	Escala
Efectividad del Dispositivo	**Precisión en la detección de ataques DoS**	Porcentaje entre el número de ataques efectuados y el	Pruebas en Laboratorio Logs de Suricata	Cuantitativa (%)/Numérico

		número de ataques detectados.		

Para las variables independientes se analizarán el tipo de ataque que se define como la clasificación de los diferentes métodos utilizados por un atacante para interrumpir o degradar el funcionamiento normal de los dispositivos IoT o de la red doméstica, mediante la segunda variable independiente que es los paquetes enviados (de tráfico malicioso) pudiendo aprovecharse de vulnerabilidades del sistema, y esto se lo presenta en la tabla 4.

TABLA IV
OPERACIONALIZACIÓN VARIABLES INDEPENDIENTES

Variables	Dimensiones	Indicadores	Instrumento	Escala
Tipo de ataque	**Ataques por inundación**	Tipo de Protocolo empleado (TCP, UDP, ICMP) Patrón de tráfico anómalo	Capturas de tráfico Logs del IDS	Categórica/Cuantitativa
Paquetes enviados	**Número de paquetes transmitidos**	Cantidad de paquetes de red transmitidos por un dispositivo IoT o por la red doméstica hacia Internet en un intervalo de tiempo determinado.	Captura de tráfico con tcpdump o Wireshark.	Cuantitativa, Paquetes por segundo

CAPITULO IV RESULTADOS Y DISCUSIÓN

En el presente capítulo se describirá el procedimiento llevado a cabo para la parte práctica del trabajo, desde la instalación del SO hasta la implementación del IDS empleando servicios de FTP, DHCP y DNS.

Debido a la capacidad para realizar las funciones descritas de manera eficiente se realizó la elección de la Raspberry Pi, debido a que proporciona una solución compacta, versátil y sobre todo que el empleo esta herramienta en proyectos de red, combina la simplicidad técnica con un enfoque práctico y económico.

4.1 Instalación y Configuración de Raspberry Pi

4.1.1 Descarga del Software de Raspberry Pi OS

El primer elemento necesario para comenzar la instalación fue el software Raspberry Pi OS v1.8.5. Este sistema operativo se descargó de la página oficial de Raspberry Pi, en la cual se pueden encontrar versiones tanto actuales como antiguas. "Raspberry Pi OS con escritorio" fue la recomendada para la mayoría de los usuarios. Este paso fue crucial, ya que 33 suministró la imagen que luego se grabó en una tarjeta microSD, la misma que sirvió como almacenamiento para el sistema operativo y las aplicaciones.

4.1.2 Grabación del Software en la Tarjeta microSD

Una vez completada la descarga, el siguiente paso consistió en grabar la imagen del sistema operativo en la tarjeta microSD, como lo indica la Figura 7. Este proceso fue más sencillo al utilizar la herramienta Raspberry Pi Imager. Tras instalar el Imager, se seleccionó el sistema operativo correspondiente y la tarjeta de almacenamiento, para después proceder con la escritura de la imagen.

Fig. 7. Sistema Operativo y microSD

Otro aspecto favorable de Raspberry Pi Imager es la facilidad que ofrece para hacer ajustes del sistema antes de la grabación de la ISO, es así como permite a los usuarios establecer un nombre de host, modificar el usuario y la contraseña por defecto, configurar automáticamente la conexión a la red Wi-Fi, además presenta la opción de activar SSH que permite habilitar el acceso remoto al dispositivo, lo que es crucial para su gestión eficiente.

4.1.3 Configuración de la red para la activación de puntos de acceso

El primer paso consistió en instalar el software hostapd, encargado de convertir la interfaz de red en un punto de acceso y servidor de autenticación. Para ello, fue necesario ejecutar la instalación con privilegios de administrador. Adicionalmente, se instaló el paquete isc-dhcp-server, el cual proporcionó el servicio DHCP encargado de asignar direcciones IP a los dispositivos conectados. Este servicio permitió que cada equipo obtuviera configuraciones de red automáticamente, garantizando la conectividad y el funcionamiento adecuado del punto de acceso. La adecuada configuración de ambos servicios es esencial para un funcionamiento óptimo del punto de acceso. Esto se lo realizará mediante el comando:

sudo apt install hostapd isc-dhcp-server

Posteriormente, se editó el archivo de configuración de red para definir los parámetros correspondientes a la interfaz wlan0, con el siguiente comando:

sudo ifconfig wlan0 down

A continuación, se debe editar el archivo de configuración de red en etc/network/interfaces para definir la configuración de wlan0. Si ya existe una sección para wlan0, se modifica; de lo contrario, se agregó lo que muestra la Figura 8. De igual manera, se configuró la interfaz cableada eth0 para que obtuviera su dirección IP automáticamente mediante DHCP. Con estas configuraciones, la sonda IoT obtuvo una topología de red estable y adecuada para las pruebas. La configuración final quedó:

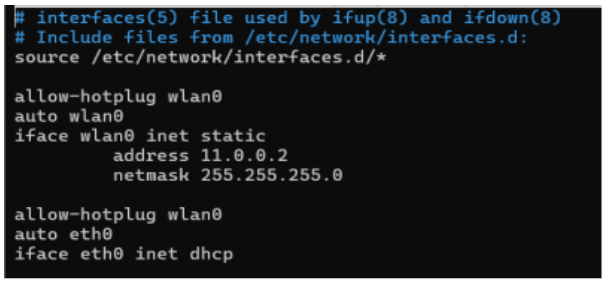


Fig. 8. Creación WLAN

Para configurar el servicio DHCP en la sonda, se editó el archivo de configuración: /etc/dhcp/dhcpd.conf, que gestiona el daemon DHCP. La Figura 9 muestra las modificaciones que se realizó en el fichero:

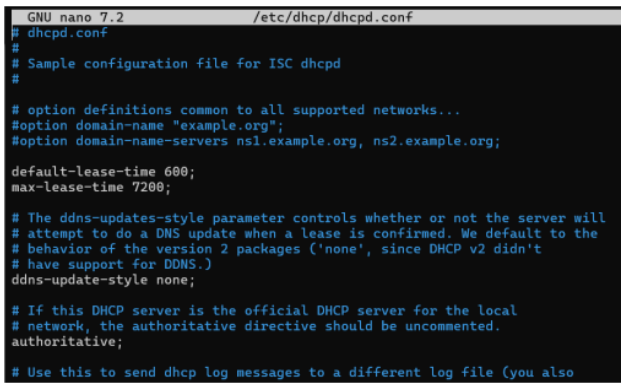


Fig. 9. Creación DHCP

Se comentó las líneas relacionadas con domain-name y option domain-name-servers, después se descomentó la línea authoritative, lo que estableció que la sonda actúe como el

servidor DHCP principal en la red. Se agregó la siguiente configuración para definir el servicio:

```
subnet 11.0.0.0 netmask 255.255.255.0 {
range 11.0.0.10 11.0.0.20;
option broadcast-address 11.0.0.255;
option routers 11.0.0.1;
default-lease-time 600;
max-lease-time 7200;
option domain-name "local";
option domain-name-servers 8.8.8.8;
}
```

Los elementos que fueron configurados indican:

- **Range (Rango de IP):** El servidor asignará direcciones IP en el segmento 11.0.0.10 al 11.0.0.20, un total de 11 direcciones, en la subred 11.0.0.0 con máscara 255.255.255.0
- **Option broadcast-address (Dirección de broadcast):** La dirección de broadcast para la red es 11.0.0.255.
- **Option routers (Puerta de enlace):** Indica la puerta de enlace predeterminada, que es 11.0.0.1, a donde se dirigirá el tráfico externo.
- Default-lease-time (tiempo estándar de asignación): fijado en 600 s, es el tiempo de reserva de una IP para un cliente,
- **Max-lease-time (Tiempo máximo de asignación):** es el tiempo máximo de reserva de la dirección IP, independientemente de la petición del cliente, establecido en 7200 segundos.
- **Option domain-name (Nombre de dominio):** que define el nombre de dominio del servidor DHCP, en este caso, "local".

Inmediatamente se procedió a la configuración del Access Point o punto de acceso, el mismo, que se muestra en el Anexo A.

Se utilizó comandos específicos de iptables para configurar la NAT entre la interfaz de acceso a Internet eth0 y la interfaz del punto de acceso inalámbrico wlan0.

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT

La primera línea habilitó una regla MASQUERADE, la cual permite a los paquetes de la red local salir a través de eth0, preservando la privacidad de las direcciones IP internas.

En la segunda que trata sobre el reenvío de conexiones establecidas, permitió el reenvío de paquetes desde eth0 hacia wlan0 solo si el estado de la conexión es RELATED o ESTABLISHED, asegurando que solo las sesiones iniciadas desde la red local puedan recibir respuestas, anteponiendo la seguridad.

La tercera línea que trata sobre el reenvío desde wlan0, configuró el reenvío de paquetes desde wlan0 hacia eth0 sin restricciones lo cual permite el acceso a Internet para cualquier conexión fomentando la apertura y accesibilidad.

Este script con comandos de iptables es recomendable almacenarlo en el fichero /etc/firewall-probe.sh, con permisos de ejecución mediante chmod a fin de promover la confiabilidad del sistema, como lo indica la Figura 10.

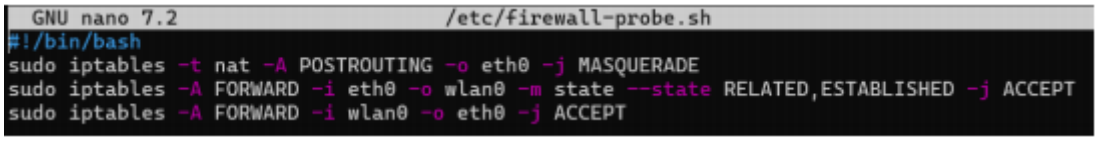


Fig. 10. Fichero firewall-probe.sh

Es necesario que las reglas de iptables se carguen automáticamente durante el arranque del sistema, por ello para asegurarse que esto se cumpla, se añadió una línea específica en el archivo */etc/rc.local*. Este archivo actúa como un script que se ejecuta al finalizar el proceso de arranque en modo multiusuario, que es el modo predeterminado del sistema operativo. La línea que se agregó es:

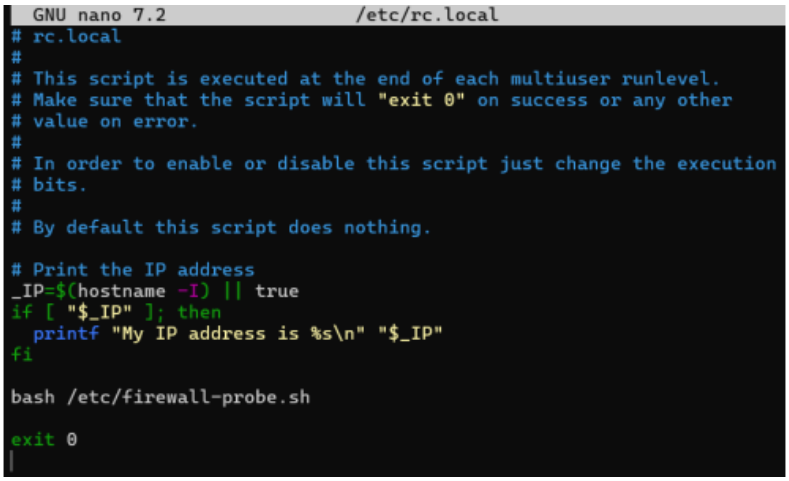


Fig. 11. Fichero rc.local

La configuración realizada en la Figura 11 garantizó que el script responsable de configurar las reglas del firewall se ejecute de manera automática al inicio del sistema, promoviendo la integridad y seguridad de la red desde el comienzo de la operación del sistema.

El paso final en la configuración del dispositivo encargado de recolectar tráfico es crucial e implica el inicio automático de los servicios hostapd e isc-dhcp-server al comienzo de la carga del SO, lo cual se logra mediante la ejecución de los siguientes comandos específicos:

sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl enable isc-dhcp-server

Luego de ejecutados estos comandos, se garantizó que, tras reiniciar el sistema, el punto de acceso inalámbrico esté completamente configurado y en pleno funcionamiento, asegurando así una operación eficiente y sin interrupciones desde el inicio, lo que promueve la fiabilidad y disponibilidad del servicio, como lo indica la Figura 12.

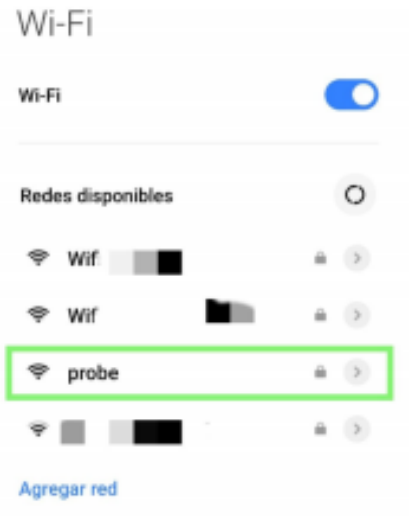


Fig. 12. Prueba Sonda

4.1.4 Topología de Red

La arquitectura final empleada en esta implementación se muestra en la Figura 13

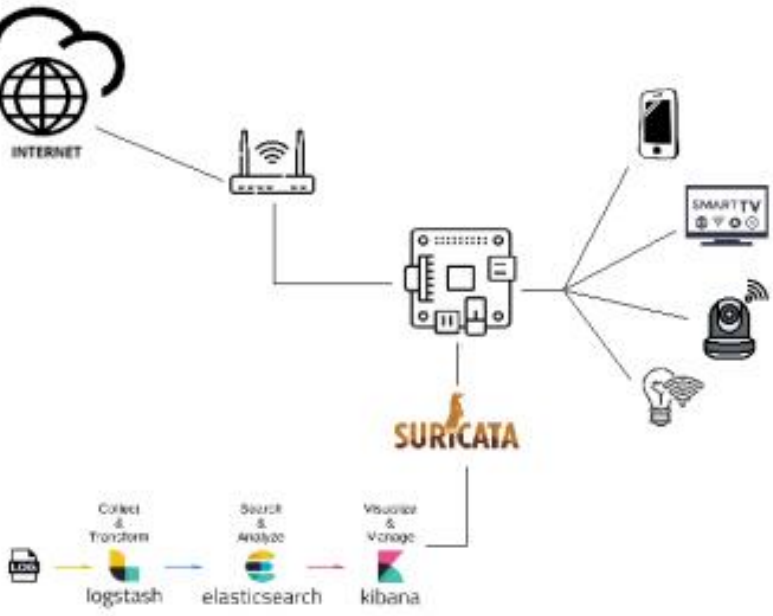


Fig. 13. Topología de Red

En este trabajo, se llevó a cabo la instalación de un sistema de detección y prevención de intrusiones: Suricata, en conjunto con la pila ELK (Elasticsearch, Logstash, Kibana) en una Raspberry Pi 3. La integración de estas herramientas mejoró la eficiencia operativa y refuerzo la seguridad y la gestión de los datos recopilados, la información obtenida a través

de este sistema es fácilmente accesible a través de un navegador web, permitiendo un análisis detallado y una respuesta rápida ante cualquier incidencia de seguridad.

La arquitectura de la pila ELK, mostrada en la Figura 14 permitió la recopilación de datos de múltiples fuentes, aunque inicialmente se centre en la información provista por la sonda IoT especialmente creada para este proyecto; la flexibilidad y escalabilidad de este enfoque permiten la expansión futura para incluir más fuentes de datos, ampliando así el alcance y la efectividad del sistema de seguridad.

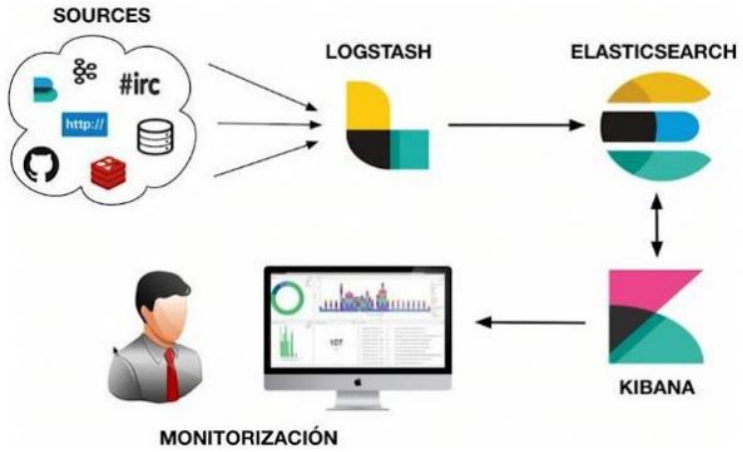


Fig. 14. Arquitectura Pila ELK

Si bien, la interpretación de logs puede ser desafiante para usuarios sin experiencia dado a su complejidad, se utilizó una solución integral que recolecta datos de diversas fuentes y los envía a Logstash, herramienta que procesa y enriquece los datos antes de almacenarlos en Elasticsearch. Kibana facilitó el análisis de esta información, ofreciendo interfaces visuales intuitivas que mejoran la accesibilidad en el manejo de datos.

4.1.5 Instalación de Suricata

Suricata fue la herramienta seleccionada debido a su eficiencia en el uso de recursos, característica que la hizo adecuada para dispositivos como la Raspberry Pi. Antes de instalarla, fue necesario actualizar el sistema operativo. Una vez completada la instalación, se verificó el estado del servicio mediante el comando “sudo systemctl status”, lo que permitió confirmar su correcto funcionamiento:

sudo apt-get install suricata

Una vez realizada la instalación, antes de realizar la configuración, se verificó el estado del servicio mediante el comando "sudo systemctl status", lo cual permite confirmar su actividad y correcto funcionamiento, como se presenta en la Figura 15.

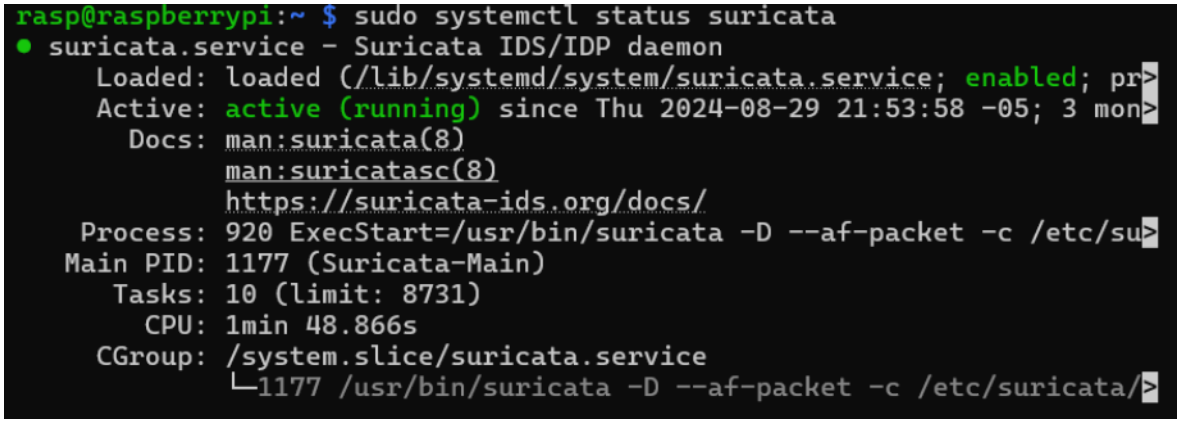


Fig. 15. Funcionamiento de Suricata

4.1.5.1 Integración de Reglas Personalizadas en Suricata

Luego de crear reglas en Suricata, es crucial asegurar su carga correcta al inicio, esto implica añadir las reglas al archivo de configuración en ***/etc/suricata***. Una vez creada la regla, se agregó al fichero ***/etc/suricata/suricata.yaml*** para su implementación (Figura 16), lo hecho garantiza la efectividad de las reglas.

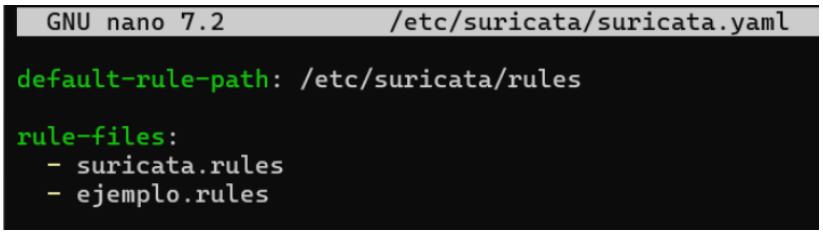


Fig. 16. Regla personalizada

4.1.6 Instalación de Pila Elk

4.1.6.1 Elascticsearch

La instalación de Elasticsearch fue un proceso fundamental debido a su capacidad para analizar grandes volúmenes de datos en tiempo real. La instalación consistió en importar la clave pública, agregar el repositorio oficial e instalar el paquete mediante los

comandos correspondientes. Posteriormente, se editó el archivo de configuración elasticsearch.yml (Figura 17), donde se ajustaron parámetros como la dirección IP del servidor, el puerto y los nodos de descubrimiento:

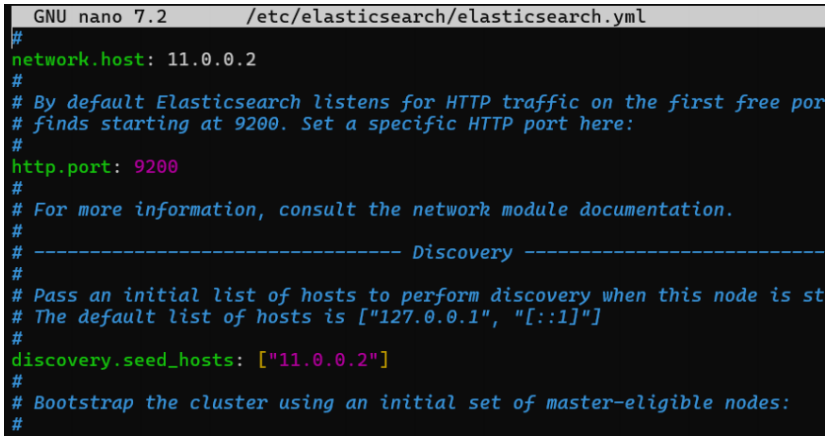


Fig. 17. Configuración Elasticsearch

Una vez realizados estos cambios, se reinició el servicio de Elasticsearch:

sudo systemctl start elasticsearch.service

Para consultar el estado de Elasticsearch, simplemente se introdució la dirección IP más el puerto 9200 en un navegador, como lo muestra la Figura 18.

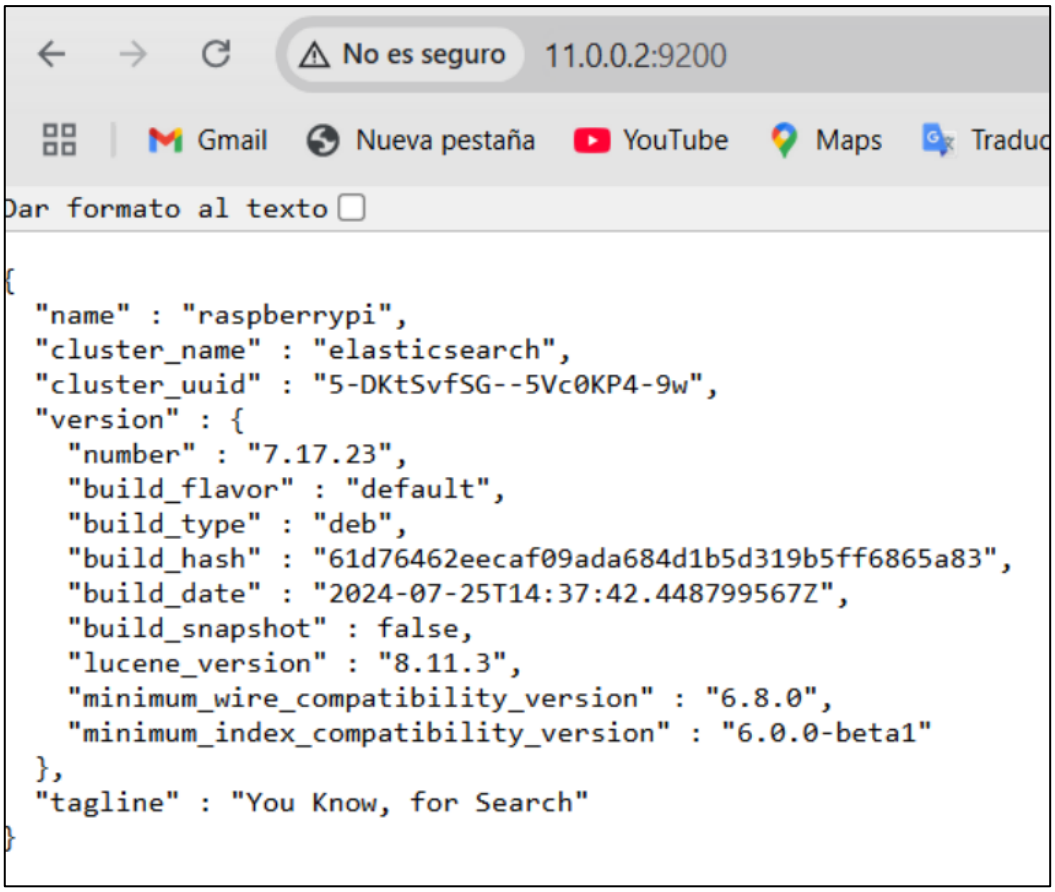


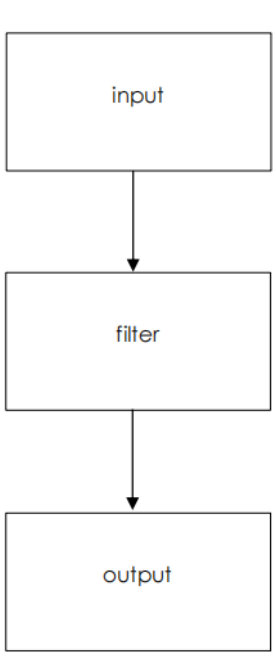
Fig. 18. Respuesta Elasticsearch

4.1.6.2 Logstash

Esta herramienta que permite la recepción, transformación y envío de datos independientemente de su formato o complejidad; así mismo simplifica el análisis de datos no estructurados utilizando Grok para extraer estructuras legibles, una vez actualizados los paquetes de Elasticsearch, la instalación de Logstash es sencilla, empleando el comando:

sudo apt-get update && sudo apt-get install logstash

Se editó el archivo /etc/logstash/conf.d/logstash.conf para configurar Logstash de forma que pueda leer los logs generados por Suricata y analizar los datos procedentes de la sonda. Logstash se configura en secciones (input, output, filter) donde la sección filter es de vital importancia debido que allí es donde se realizan procesamientos avanzados de los datos obtenidos. La figura 19 muestra un formato básico de configuración:




```
input {
  file {
    path => "/var/log/suricata/eve.json"
    start_position => "beginning"
  }
}

filter {
  json {
    source => "message"
  }
  geoip {
    source => "src_ip"
  }
}

output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "suricata-logs-%{+YYYY.MM.dd}"
  }
}
```

Fig. 19. Ejemplo Logstash.conf

Este archivo configuró Logstash para que lea los registros JSON generados por SURICATA, los procese para extraer la información de geolocalización de las direcciones IP y, a continuación, los envíe a Elasticsearch para su indexación.

4.1.6.3 Kibana

Finalmente, se instaló Kibana, una aplicación front-end de open source en Elastic Stack, que facilitó la visualización y búsqueda de datos almacenados en Elasticsearch. Kibana se considera la herramienta de representación para Elastic Stack (anteriormente ELK Stack, que incluía Elasticsearch, Logstash y Kibana) y actuó como una interfaz para monitorear, administrar y asegurar los clústeres de Elastic Stack. También se convierte en un punto central para soluciones integradas del Stack; tras actualizar los paquetes, la instalación de Kibana se realizó con el siguiente comando:

sudo apt-get update && sudo apt-get install kibana

En cuanto a la configuración, se edita el archivo ***/etc/kibana/kibana.yml*** con los parámetros de la Figura 20:

```
  GNU nano 7.2                  /etc/kibana/kibana.yml
# Kibana is served by a back end server. This setti
server.port: 5601

# Specifies the address to which the Kibana server
# The default is 'localhost', which usually means r
# To allow connections from remote users, set this
server.host: "11.0.0.2"
```

```
# The URLs of the Elasticsearch instances to use for all your queries.
elasticsearch.hosts: ["http://11.0.0.2:9200"]
```

Fig. 20. Configuración de Kibana

Para acceder a la interfaz de Kibana, ingresamos la IP seguida de :5601 en el navegador, como indica la Figura 21.

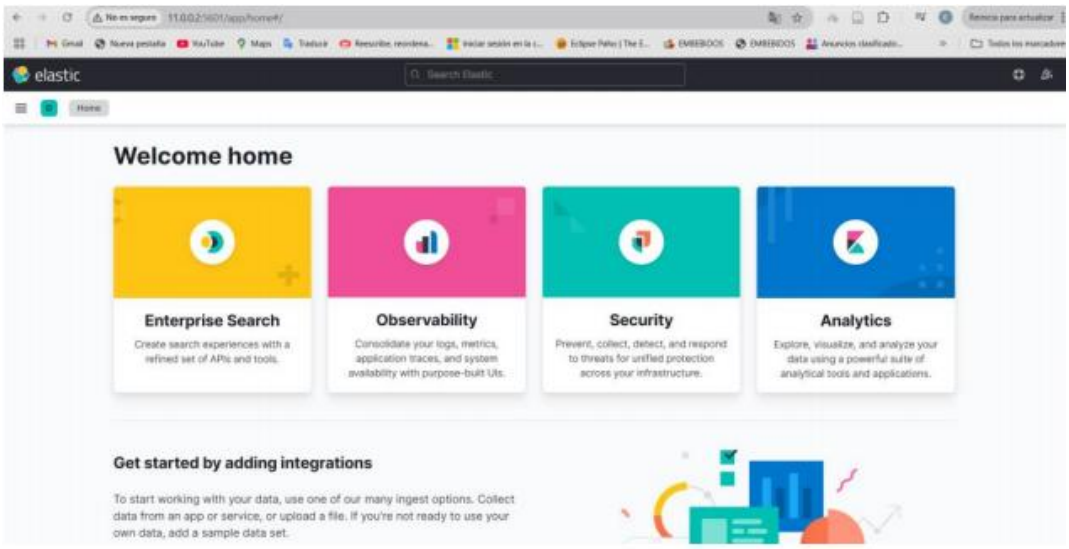


Fig. 21. Interfaz de Kibana

4.2 Resultados

Una vez finalizada la configuración de la pila ELK y tras acceder a la interfaz, se pudo que los ataques DoS ejecutados fueron registrados correctamente, tal como se evidencia en la fecha correspondiente como se lo muestra la figura 22.

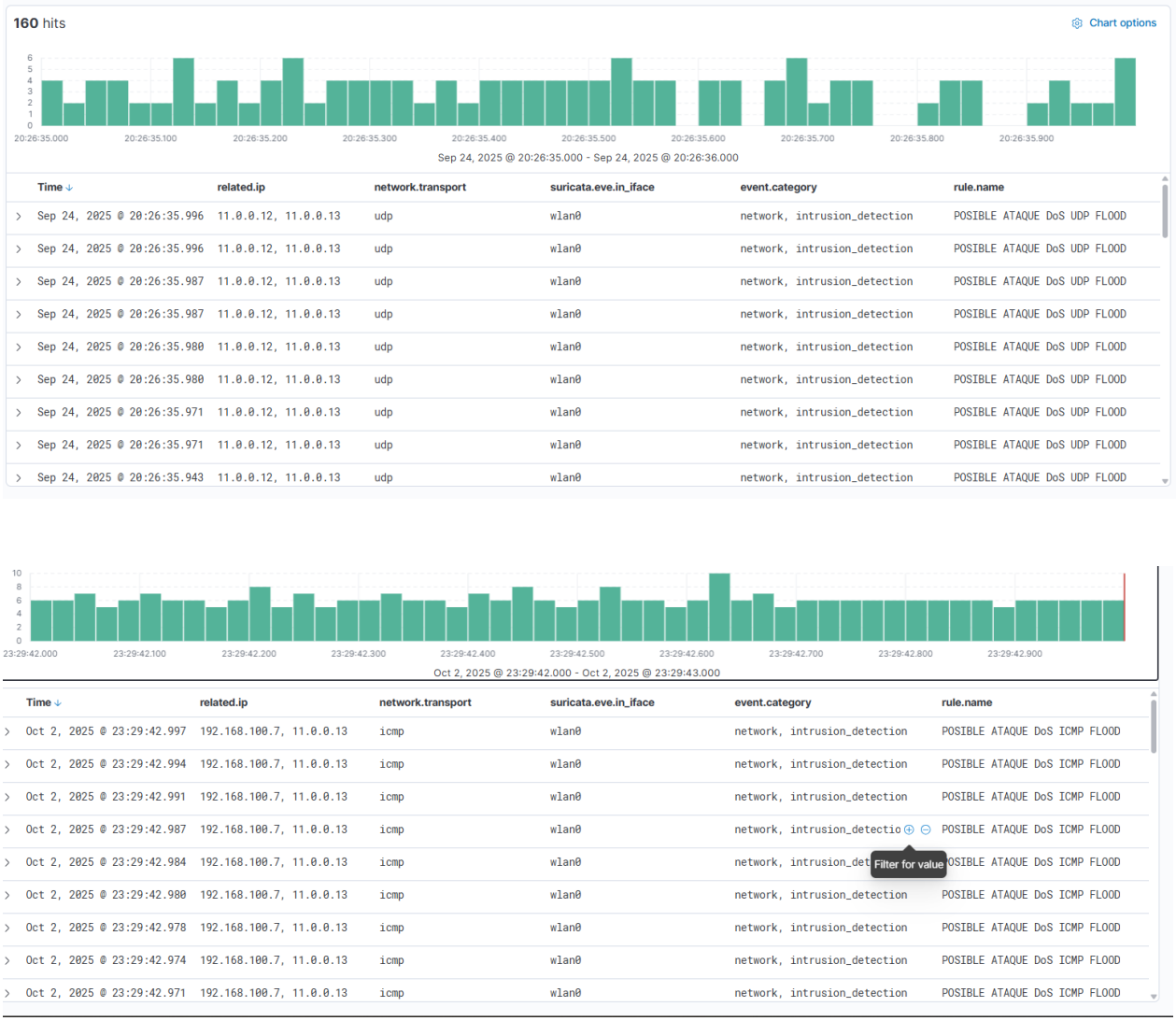


Fig. 22. Interfaz de ELK

En la Figura 21 se observa el ataque TCP enviado, mientras que en la Figura 22 se aprecia su detección inicial.

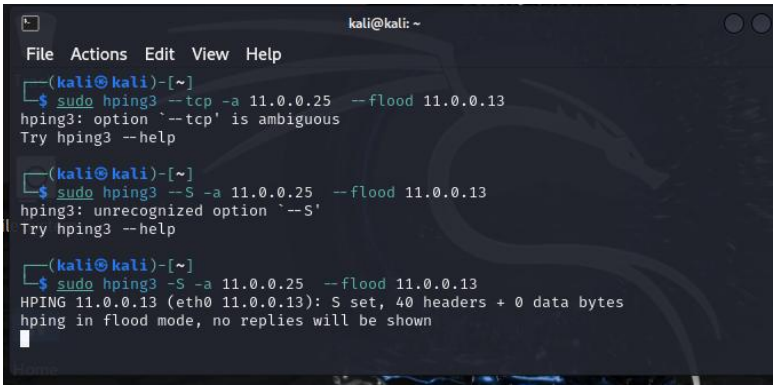
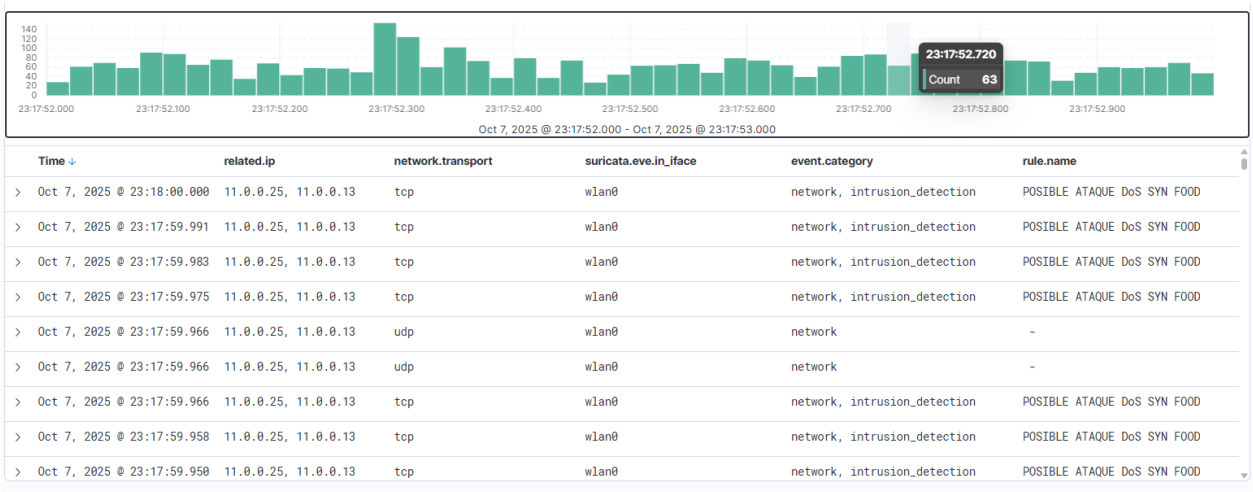


Fig. 23. Ataque TCP enviado

```
*] [Classification: (null)] [Priority: 3] {TCP} 11.0.0.25:48019 -> 11.0.0.13
:0
10/07/2025-23:18:09.746240  [**] [1:100001:1] POSIBLE ATAQUE DoS SYN FOOD [*
*] [Classification: (null)] [Priority: 3] {TCP} 11.0.0.25:48164 -> 11.0.0.13
:0
10/07/2025-23:18:09.755158  [**] [1:100001:1] POSIBLE ATAQUE DoS SYN FOOD [*
*] [Classification: (null)] [Priority: 3] {TCP} 11.0.0.25:49408 -> 11.0.0.13
:0
10/07/2025-23:18:09.764184  [**] [1:100001:1] POSIBLE ATAQUE DoS SYN FOOD [*
*] [Classification: (null)] [Priority: 3] {TCP} 11.0.0.25:49854 -> 11.0.0.13
:0
10/07/2025-23:18:09.770545  [**] [1:100001:1] POSIBLE ATAQUE DoS SYN FOOD [*
*] [Classification: (null)] [Priority: 3] {TCP} 11.0.0.25:50591 -> 11.0.0.13
:0
10/07/2025-23:18:09.791977  [**] [1:100001:1] POSIBLE ATAQUE DoS SYN FOOD [*
*] [Classification: (null)] [Priority: 3] {TCP} 11.0.0.25:51324 -> 11.0.0.13
:0
client_loop: send disconnect: Connection reset
PS C:\Users\asus>
```

Fig. 24. Detección el Ataque TCP

La figura 23 confirma, a través de la pila ELK, que el ataque UDP fue registrado correctamente, y en la Figura la Figura 24 puede notarse la saturación generada en la red durante la prueba.



Time	related.ip	network.transport	suricata.eve.in_iface	event.category	rule.name
Oct 7, 2025 @ 23:18:00.000	11.0.0.25, 11.0.0.13	tcp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS SYN FOOD
Oct 7, 2025 @ 23:17:59.991	11.0.0.25, 11.0.0.13	tcp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS SYN FOOD
Oct 7, 2025 @ 23:17:59.983	11.0.0.25, 11.0.0.13	tcp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS SYN FOOD
Oct 7, 2025 @ 23:17:59.975	11.0.0.25, 11.0.0.13	tcp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS SYN FOOD
Oct 7, 2025 @ 23:17:59.966	11.0.0.25, 11.0.0.13	udp	wlan0	network	-
Oct 7, 2025 @ 23:17:59.966	11.0.0.25, 11.0.0.13	udp	wlan0	network	-
Oct 7, 2025 @ 23:17:59.966	11.0.0.25, 11.0.0.13	tcp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS SYN FOOD
Oct 7, 2025 @ 23:17:59.958	11.0.0.25, 11.0.0.13	tcp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS SYN FOOD
Oct 7, 2025 @ 23:17:59.950	11.0.0.25, 11.0.0.13	tcp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS SYN FOOD

Fig. 25. Ataque TCP detectado en ELK

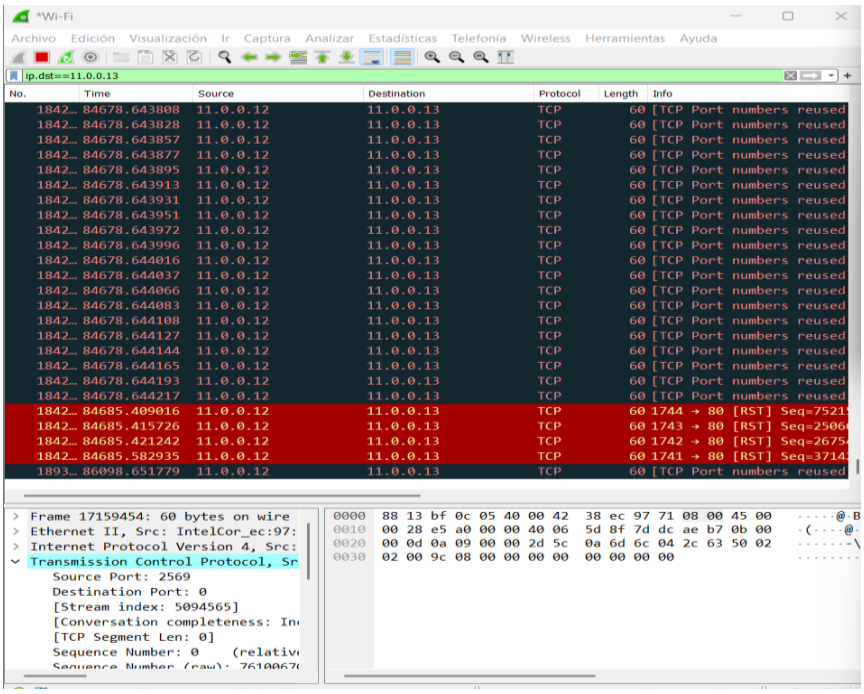


Fig. 26. Saturación de la Red

En la Figura 27 se observa el ataque UDP enviado, mientras que en la Figura 28 se aprecia su detección inicial.

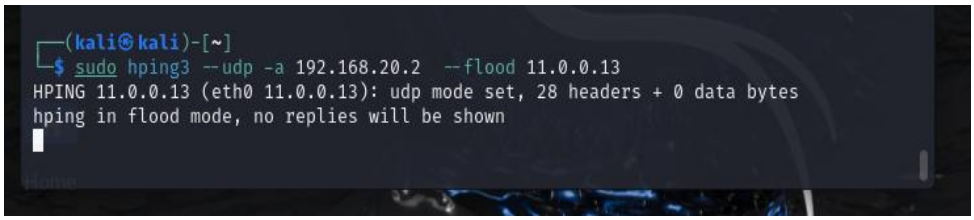


Fig. 27. Ataque UDP enviado

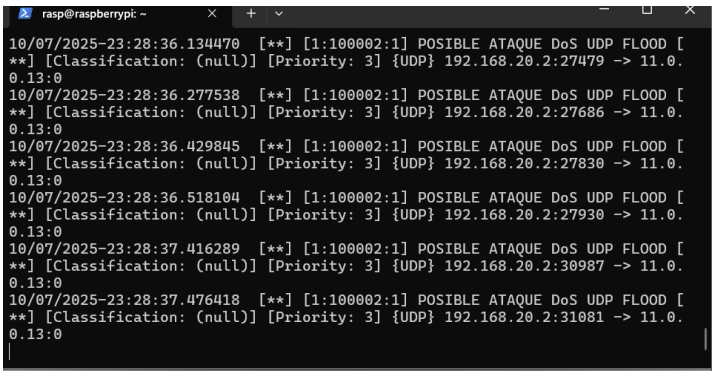
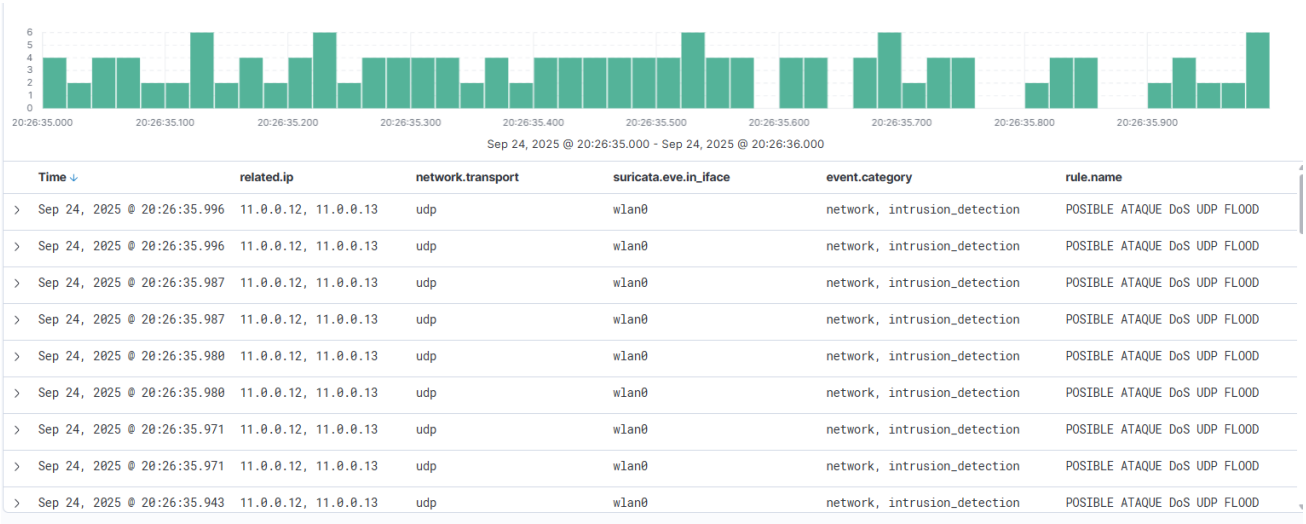


Fig. 28. Detección el Ataque UDP

La figura 29 confirma, a través de la pila ELK, que el ataque UDP fue registrado correctamente, y en la Figura la Figura 30 puede notarse la saturación generada en la red durante la prueba.



Time	related.ip	network.transport	suricata.eve.in_iface	event.category	rule.name
Sep 24, 2025 @ 20:26:35.996	11.0.0.12, 11.0.0.13	udp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS UDP FLOOD
Sep 24, 2025 @ 20:26:35.996	11.0.0.12, 11.0.0.13	udp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS UDP FLOOD
Sep 24, 2025 @ 20:26:35.987	11.0.0.12, 11.0.0.13	udp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS UDP FLOOD
Sep 24, 2025 @ 20:26:35.987	11.0.0.12, 11.0.0.13	udp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS UDP FLOOD
Sep 24, 2025 @ 20:26:35.980	11.0.0.12, 11.0.0.13	udp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS UDP FLOOD
Sep 24, 2025 @ 20:26:35.980	11.0.0.12, 11.0.0.13	udp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS UDP FLOOD
Sep 24, 2025 @ 20:26:35.971	11.0.0.12, 11.0.0.13	udp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS UDP FLOOD
Sep 24, 2025 @ 20:26:35.971	11.0.0.12, 11.0.0.13	udp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS UDP FLOOD
Sep 24, 2025 @ 20:26:35.943	11.0.0.12, 11.0.0.13	udp	wlan0	network, intrusion_detection	POSIBLE ATAQUE DoS UDP FLOOD

Fig. 29. Ataque UDP detectado en ELK

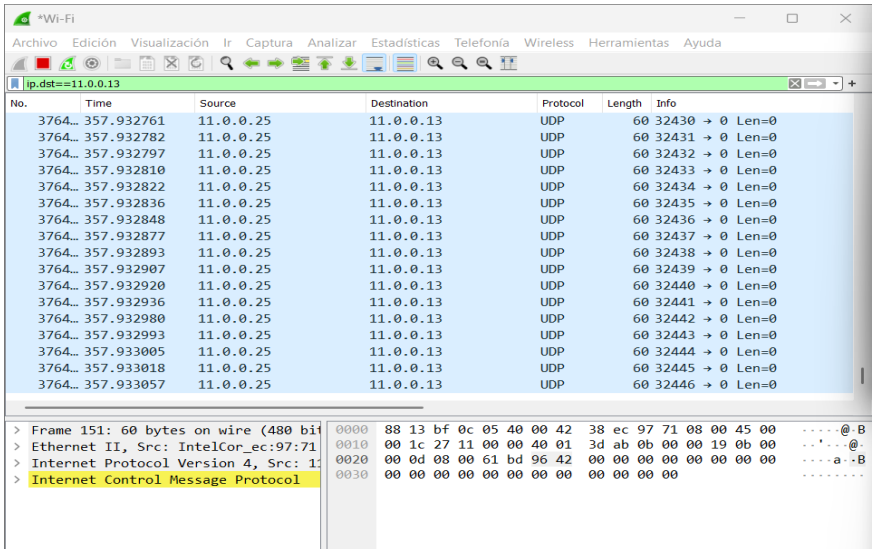


Fig. 30. Saturación de la Red

En la Figura 31 se observa el ataque ICMP enviado, mientras que en la Figura 32 se aprecia su detección inicial.

Fig. 31. Ataque ICMP enviado

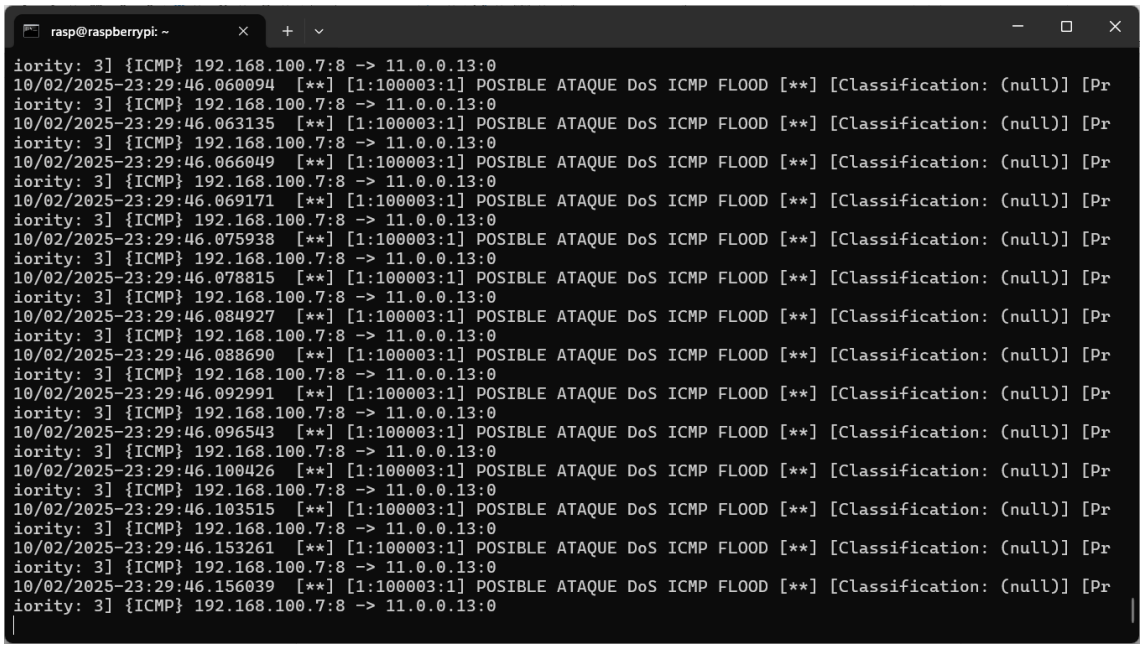


Fig. 32. Detección el Ataque UDP

La Figura 33 confirma, a través de la pila ELK, que el ataque ICMP fue registrado correctamente, y en la Figura 34 puede notarse la saturación generada en la red durante la prueba.

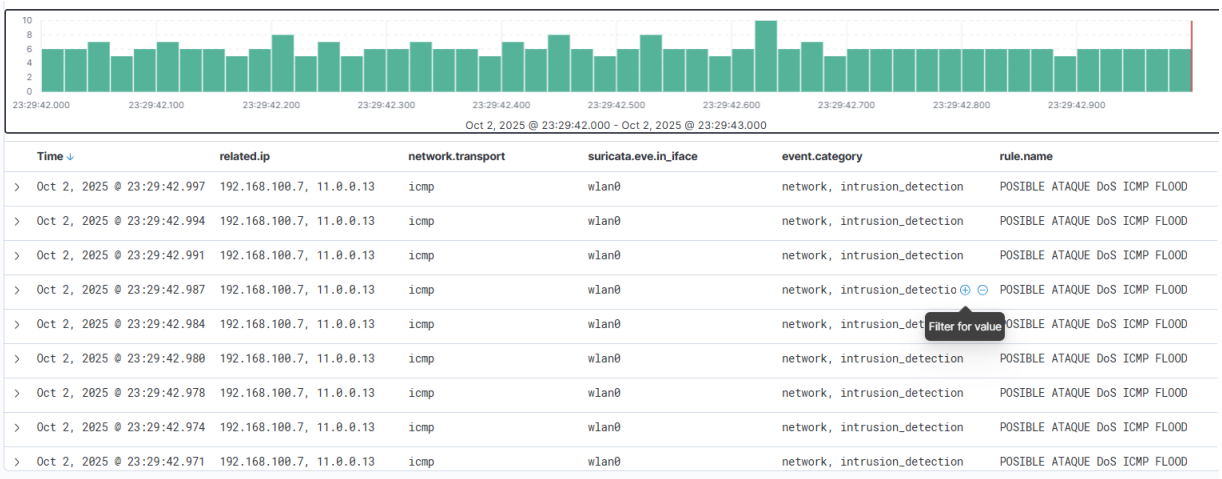


Fig. 33. Ataque ICMP detectado en ELK

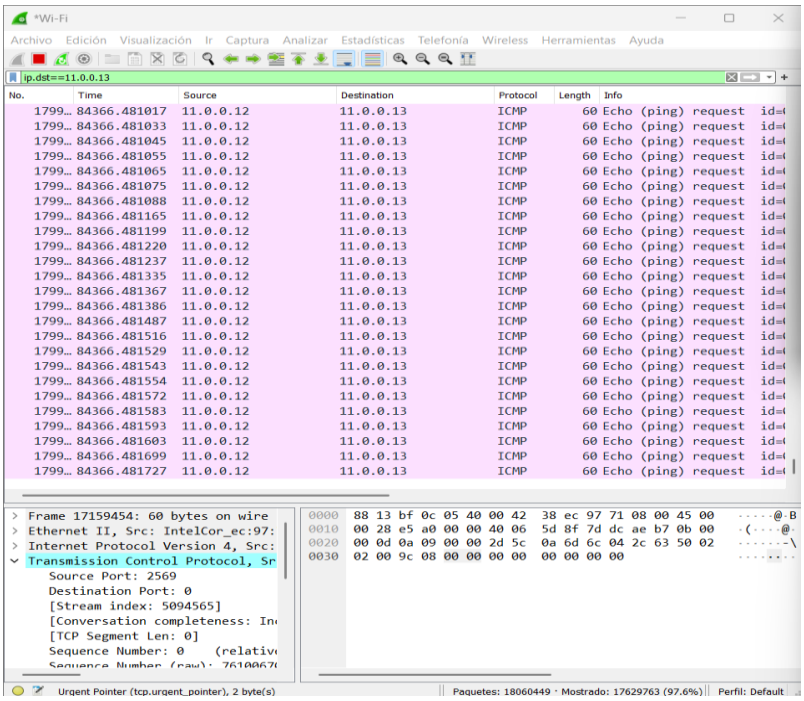


Fig. 34. Saturación de la Red

La figura 35 muestra los porcentajes de detección de los tipos de ataques posibles para el sistema para condiciones donde no existe saturación y la herramienta de Suricata funciona sin inhibirse.

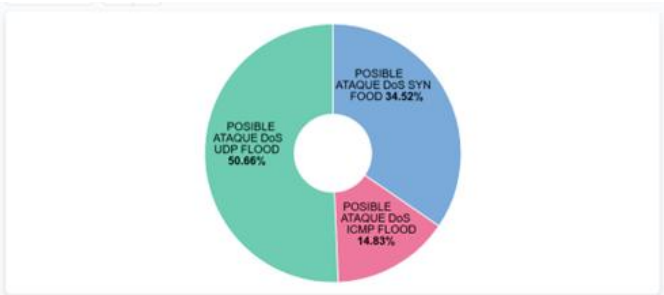


Fig. 35. Porcentajes de detección de ataques DoS

No obstante, en escenarios más cercanos a condiciones reales, cuando se enviaron ataques combinados, el sistema no logró detectarlos adecuadamente. En la Figura 36 se muestra un ejemplo en el que, tras ejecutar una inundación ICMP seguida de un ataque UDP, este último no fue identificado por el sistema.

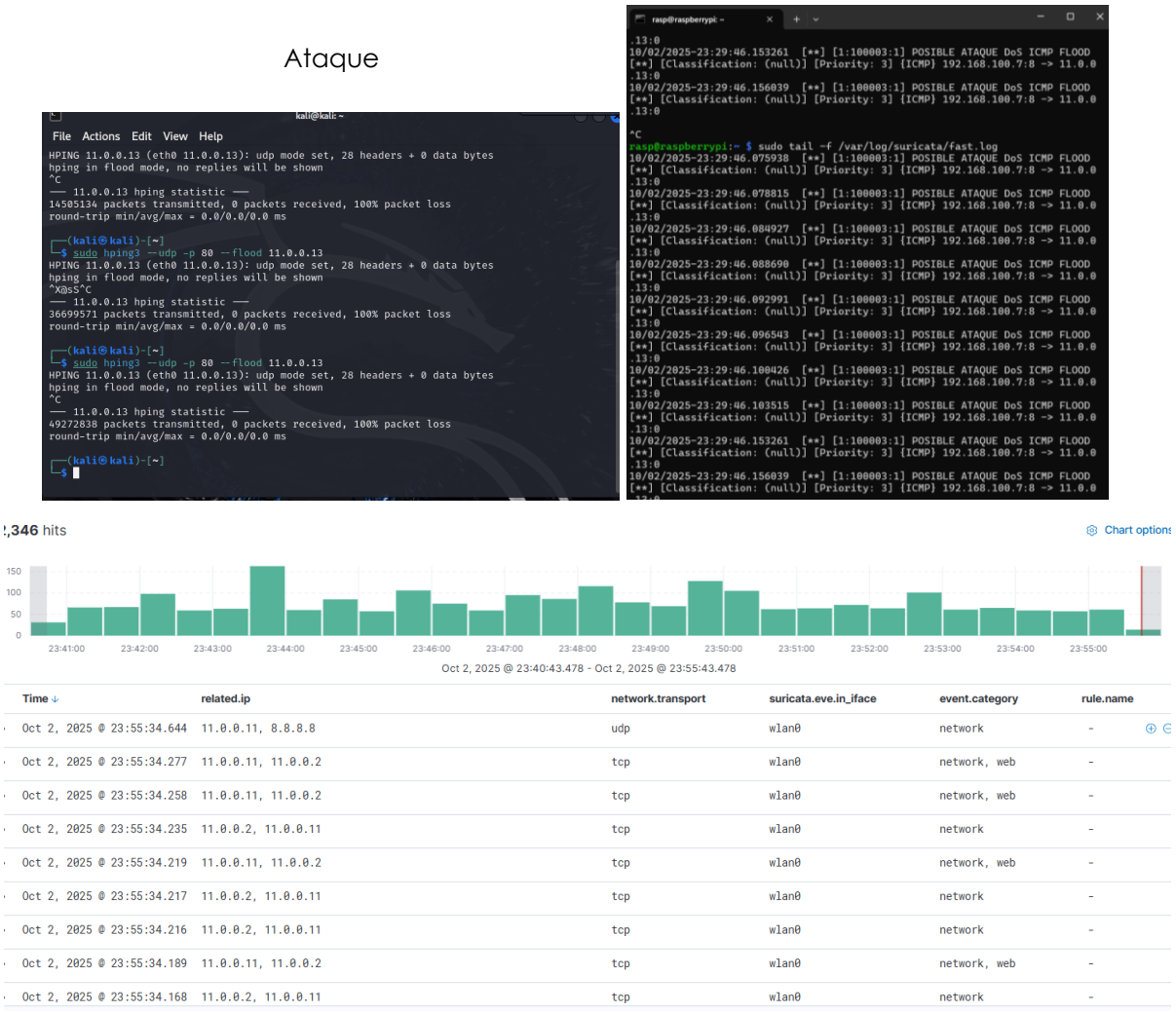


Fig. 36. Ataques UDP sin detectar

Una situación similar ocurrió cuando el IDS se inhibió, como se observa en la Figura 37, donde no se detectó un ataque TCP aun después de reiniciar Suricata.

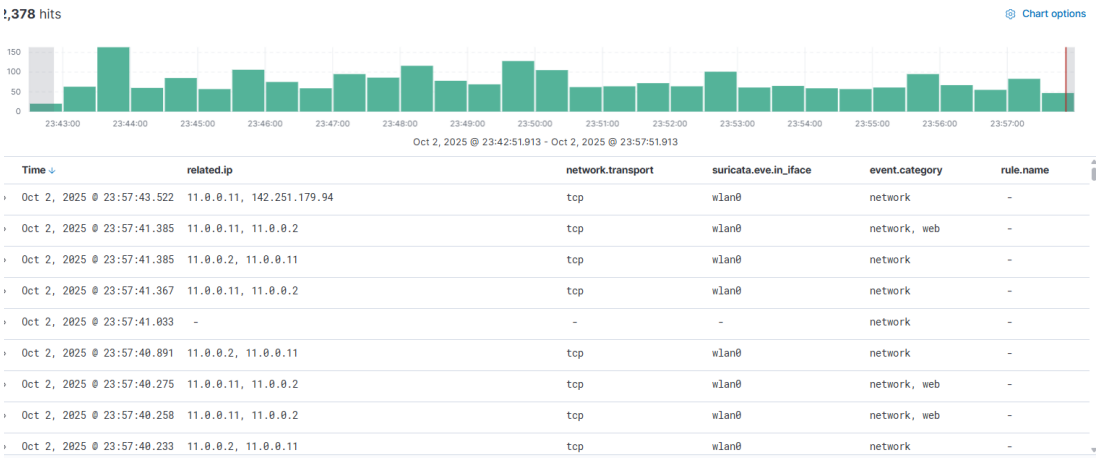


Fig. 37. Ataques TCP sin detectar

La Figura 38 ilustra claramente esta condición: al enviar simultáneamente un ataque UDP y uno TCP, Suricata solo detectó el primero; sin embargo, al mantener únicamente el ataque UDP, este fue detectado de manera normal.

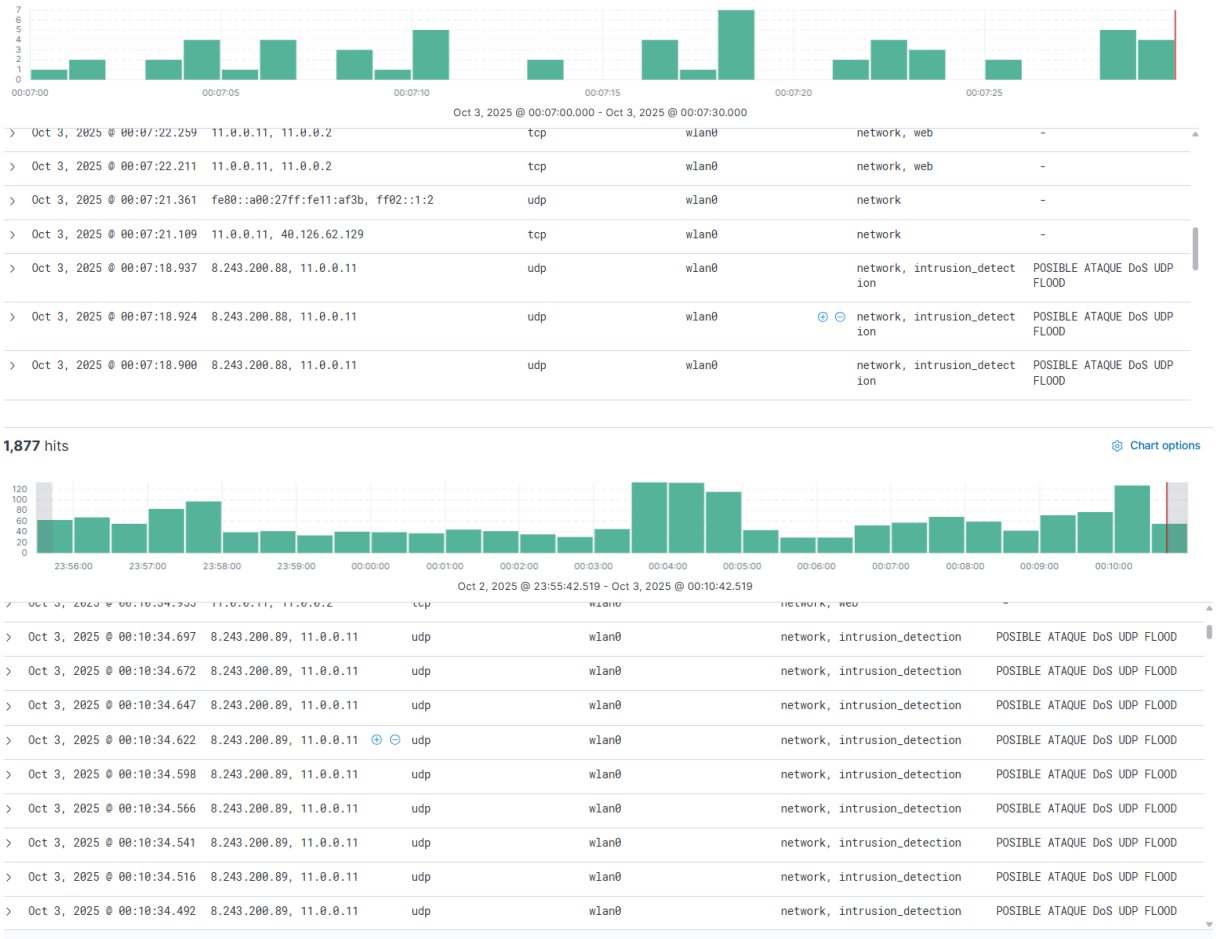


Fig. 38. Ataques UDP y TCP simultáneos.

4.3 Discusión

La presente sección analizó de forma integral los resultados obtenidos durante la ejecución de ataques DoS en el entorno IoT implementado, considerando los registros de Suricata, la información procesada en la pila ELK y el análisis estadístico realizado a partir de los datos recopilados. El objetivo fue evaluar la efectividad del sistema de detección implementado, la coherencia de los datos obtenidos y el comportamiento del modelo estadístico aplicado.

Para desarrollar este apartado se aplicó un análisis estadístico destinado a evaluar la efectividad del sistema de detección implementado. Se llevó a cabo un análisis descriptivo basado en los datos obtenidos desde la interfaz ELK, específicamente en relación con los paquetes enviados y las alertas detectadas por el IDS como se indica en la Tabla 5.

TABLA V
DATOS OBTENIDOS DE PAQUETES ENVIADOS

paquetes_enviados	
Mínimo	8800
Primer Cuartil	9300
Mediana	9600
Media	9625
Tercer Cuartil	9975
Máximo	10800
No Registrados/Fallos	11

En primera instancia, el análisis descriptivo evidenció patrones relevantes en las dos variables de estudio: paquetes_enviados y alertas_detectadas. En la Tabla 5, correspondiente a los estadísticos de paquetes_enviados, se observó que los valores oscilaron entre 8800 y 10800. La media (9625) y la mediana (9600) mostraron una distribución estable con ligera asimetría positiva, mientras que el rango intercuartílico confirmó la concentración de la

mayoría de los datos en un intervalo estrecho. De manera complementaria, la Tabla 6, referente a alertas_detectadas, mostró valores comprendidos entre 5386 y 6966, con una media (6248) y mediana (6290) muy próximas entre sí, lo que evidenció una distribución casi simétrica. En ambas tablas se identificaron 11 valores faltantes coincidentes, lo que indica que las interrupciones en la captura afectaron simultáneamente a ambas variables.

TABLA VI
DATOS OBTENIDOS DE ALERTAS DETECTADAS

alertas_detectadas	
Mínimo	5386
Primer Cuartil	5889
Mediana	6290
Media	6248
Tercer Cuartil	6622
Máximo	6966
No Registrados/Fallos	11

La Figura 39 presentó el histograma de las alertas detectadas, donde se pudo observar una concentración importante entre 5800 y 6900 alertas, con picos en los rangos altos. Esta tendencia reflejó que el sistema respondió con mayor intensidad durante los momentos de incremento significativo en el tráfico malicioso.

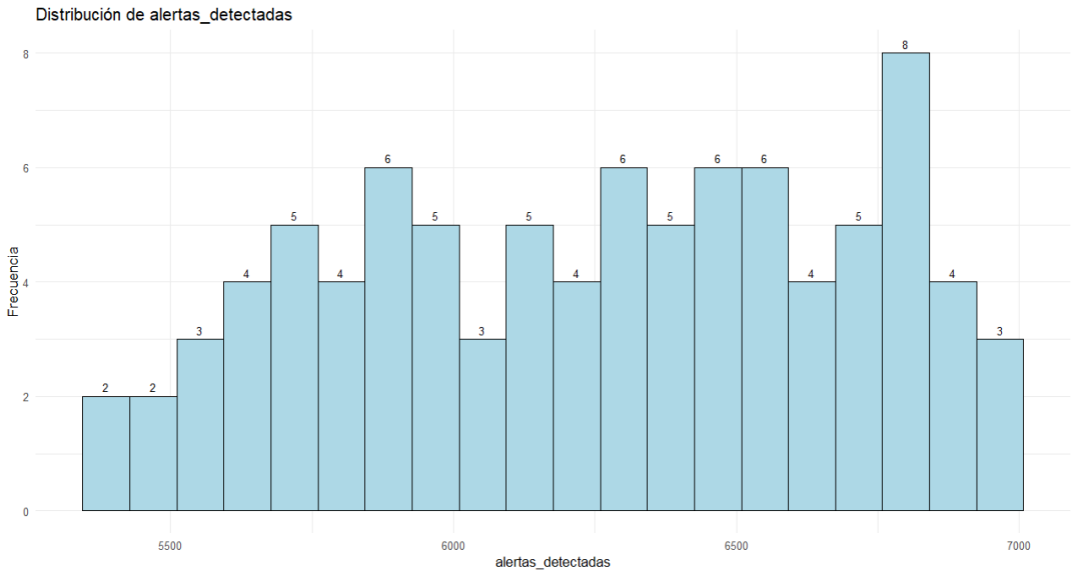


Fig. 39. Histograma alertas detectadas

Por otro lado, la Figura 40 mostró la matriz de dispersión entre paquetes_enviados y alertas_detectadas. En esta figura se evidenció una tendencia ascendente definida, lo cual reflejó la existencia de una relación lineal positiva entre las dos variables, consistente con la operación esperada de un IDS basado en firmas.

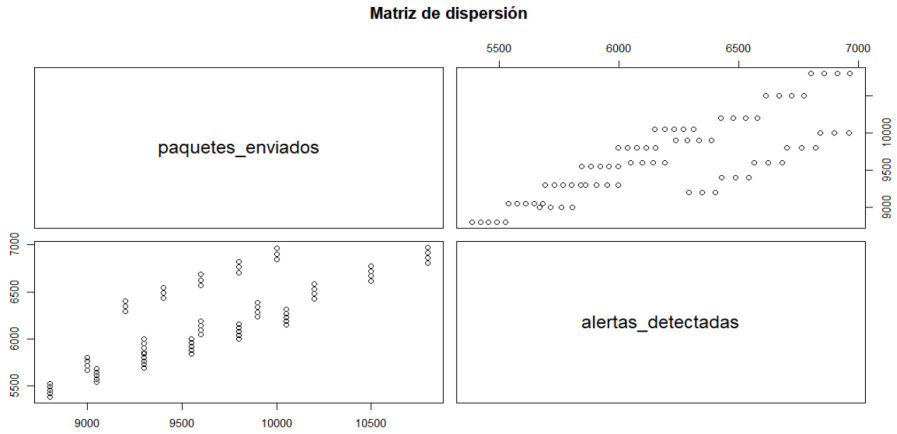


Fig. 40. Matriz de Dispersión

La relación entre las variables también se confirmó mediante la matriz de correlación presentada en la Tabla 9 y representada gráficamente en la Figura 41. En esta matriz se observó un coeficiente de correlación moderado–alto (aproximadamente 0.75), lo que indica que existió una relación directa entre el incremento de paquetes enviados y el aumento de alertas detectadas. La diagonal con valores de 1 representó la autocorrelación natural de cada

variable consigo misma, mientras que la ausencia de coeficientes negativos evidenció que no existieron relaciones inversas entre las métricas analizadas.

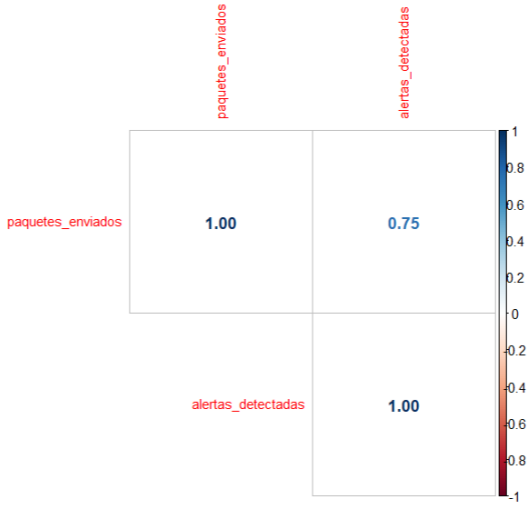


Fig. 41. Matriz de Correlación

En contextos experimentales con ataques simultáneos o consecutivos, el Sistema de Detección de Ataques (IDS) evidenció restricciones significativas. Según las Figuras 36, 37 y 38 de la sección previa, Suricata no logró identificar todos los ataques cuando se ejecutaron múltiples vectores en intervalos breves o bajo condiciones de saturación extrema. En determinados escenarios, se documentó únicamente el primer ataque y se desestimaron los subsiguientes.

Este comportamiento constituye una vulnerabilidad significativa en contextos reales de Internet de las Cosas (IoT), donde los atacantes frecuentemente combinan estrategias para eludir los sistemas de seguridad. Desde un enfoque estadístico, se implementó un modelo de regresión múltiple para evaluar la incidencia de los paquetes enviados en las alertas detectadas. La Tabla 7 ofrece una síntesis del modelo, en la que se evidencian los coeficientes de las variables independientes y su correspondiente significancia. A pesar de que el modelo consiguió detallar una porción considerable de la variabilidad observada, el análisis puso de manifiesto que la suposición de normalidad de los residuos no fue completamente alcanzada.

TABLA VII
PARAMETROS OBTENIDOS DE LA REGRESIÓN LINEAL MÚLTIPLE

1. Modelo:	$lm(formula = alertas\_detectadas \sim paquetes\_enviados + tipo\_ataque, data = datos)$				
2. Residuos:	Mínimo	1 Cuartil	Mediana	Tercer Cuartil	Máximo
	-94.225	-49.048	-1.423	43.363	84225
3. Coeficientes	Estimado	Error Estándar		Valor t	Significancia
Interceptados	453.49682	118.90085		3.814	0.000257 ***
paquetes_enviados	0.64276	0.01234		52.075	< 2e-16 ***
tipo-ataqueTCP	-508.32352	14.24587		-35.682	< 2e-16 ***
tipo_ataqueUDP	-668.01687	14.07349		-47.463	< 2e-16 ***
Códigos de significancia	‘ *** ’ 0.001 ‘ ** ’ 0.01 ‘ * ’ 0.05 ‘ . ’ 0.1 ‘ ’ 1				
4. Error Residual Estándar	53.86 en 86 grados de libertad (Se eliminaron 11 observaciones por datos faltantes)				
5. R cuadrado múltiple	0.9856				
6. R cuadrado ajustado	0.9851				
7. Estadístico F	1958 con 3 y 86 grados de libertad				
8. Valor P	< 2.2e-16				

Donde la línea 1 se puede resumir de la siguiente manera: Linear Model lm() es la función de R que ajusta un modelo lineal, mismo que sirve para hacer regresiones lineales simples o múltiples, es decir, modelos donde una variable depende de otras de forma lineal. La parte de la fórmula: alertas_detectadas ~ paquetes_enviados + tipo_ataque, R, el símbolo ~ se lee como "en función de" o "depende de", así en este caso, alertas_detectadas depende de “paquetes_enviados” y tipo_ataque.

Por lo tanto, le efectividad se va a predecir a través de las alertas_detectadas, conformando la variable dependiente o respuesta, mientras que paquetes_enviados y tipo_ataque se convierten en variables independientes numérica (explicativa) y categórica (TCP, UDP, ICMP), que el programa convierte en variables dummy automáticamente. El argumento data = datos es el data frame llamado datos, donde buscará las columnas: alertas_detectadas, paquetes_enviados, tipo_ataque. El operador <- guarda el modelo ajustado dentro del objeto modelo.

Algo relevante a considerar es que todos los coeficientes son altamente significativos P menor a 2e-16, lo que indica que las variables explicativas (paquetes_enviados, tipo-ataqueTCP ,tipo_ataqueUDP) aportan información relevante al modelo. Así mismo, la Calidad del ajuste $R^2 = 0.9856$ indica que aproximadamente el 98.56% de la variabilidad de las alertas detectadas, resultando excelente R^2 ajustado de 0.9851 dado por el número de predictores; sigue siendo muy alto, lo que sugiere que el modelo no está sobre ajustado. El error estándar residual de 53.86 indica que las predicciones tienen, en promedio, un error de ±53.86 alertas, por su parte el estadístico F de 1958 y el valor p menor a 2.2e-16 sugieren que el conjunto del modelo es estadísticamente significativo.

A continuación, en la figura 42 se presentan los gráficos de diagnóstico empleados:

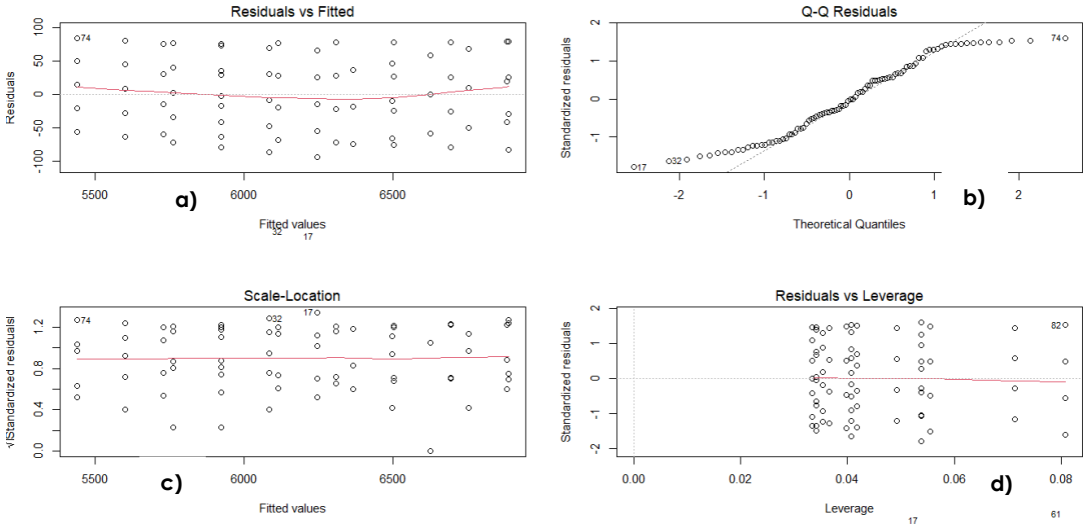


Fig. 42.Gráficos del Modelo de Regresión Múltiple

En primer lugar se tiene el gráfico Residuals vs Fitted (Valores Residuales vs Valores Ajustados) cuyo fin es verificar si los residuos tienen media cero y varianza constante (homocedasticidad), en la figura 42a los puntos están dispersos de forma aleatoria, sin patrón claro, y la línea roja es casi horizontal, lo que indica el supuesto de linealidad se está cumpliendo; sin embargo, hay algunos valores atípicos (outliers), como el 74, que podrían influir en el modelo ya que podrían representar flujos de tráfico anómalos o picos de ataques DoS que el modelo debe manejar con cuidado, en este caso, estos valores atípicos pueden darse debido a flujos de tráfico anómalos o picos de actividad sospechosa, característicos de un ataque; por tanto, su identificación resulta relevante para el posterior refinamiento del sistema de detección.

El siguiente gráfico denominado Q-Q de Residuos tiene el objetivo de verificar si los residuos siguen una distribución normal, la Figura 42b muestra que la mayor parte de los puntos siguen la línea diagonal, pero en los extremos (cola superior e inferior) se desvían, lo que supone la presencia ligeramente colas pesadas o valores extremos, lo cual se traduce en que existen anomalías o tráfico irregular que no se ajustan a la normalidad. En el escenario de IoT, esto puede significar que algunos ataques DoS tienen patrones fuera del rango habitual del comportamiento normal, lo que podría representar intentos de ataque DoS, lo cual indica que un modelo lineal puede ser útil para tendencias generales, pero limitado para capturar anomalías complejas.

El tercer gráfico presente en la Figura 42c de nombre Scale-Location buscó comprobar que la homogeneidad de la varianza de los residuos sea constante a lo largo de los valores ajustados (homocedasticidad), es así como, se puede observar que los puntos están dispersos de manera bastante uniforme, y la línea roja es casi horizontal, sin embargo, algunos puntos extremos pueden estar afectando un poco la uniformidad y confirma nuevamente la existencia de observaciones atípicas, en este caso práctico, demuestra que el

sistema tiene una buena estabilidad al predecir comportamiento normal, pero los eventos anómalos aún podrían requerir un tratamiento especial (clustering o modelos no lineales como Random Forest e incluso redes neuronales).

El último gráfico es el de Residuals vs Leverage(Apalancamientos) cuyo fin es el de detectar observaciones influyentes que podrían afectar fuertemente los resultados del modelo, así, en la Figura 42d se observan que algunos puntos como el 820 tienen mayor leverage, lo que indica que tienen una fuerte influencia en el modelo, lo cual puede representar un tráfico inusual o eventos anómalos (ataques) que distorsiona la regresión, de nuevamente en este caso, dichos puntos de alta influencia pueden representar flujos de ataque DoS o comportamientos fuera del rango normal de la red, lo cual justifica su tratamiento especial en fases posteriores de análisis o mediante el uso de modelos más robustos frente a valores extremos.

Como último paso en el análisis analítivo se realizó la comprobación de los supuestos, mediante la aplicación de varios modelos estadísticos:

a) Normalidad de los residuos

TABLA VIII
TEST DE NORMALIDAD DE SHAPIRO-WILK

Data	Residuals(modelo)
W	0.94143
Valor P	0.000514

El test de Shapiro-Wilk se utilizó para evaluar si los residuos del modelo siguen una distribución normal, con el fin de cumplir uno de los supuestos fundamentales en la regresión lineal clásica; es así como, se manejan dos hipótesis:

Hipótesis nula (Ho): Los residuos se distribuyen normalmente.

Hipótesis alternativa (H1): los residuos no se distribuyen normalmente.

El programa entrega un valor de W = 0.94143 mismo que se aleja de 1 y un valor de p = 0.000514, que es menor a 0.05, lo que implica que se rechaza la hipótesis nula (H_0); por consiguiente, los residuos del modelo no siguen una distribución normal, es decir, existen desviaciones significativas respecto a la normalidad, en este caso práctico, significa que un modelo lineal tradicional puede no resultar suficiente para detectar eficazmente eventos anómalos en especial tráfico IoT.

b) Homocedasticidad (Breusch-Pagan)

La prueba de Breusch–Pagan, se utilizó para verificar si existe heterocedasticidad (es decir, si la varianza de los errores no es constante) en un modelo de regresión, bajo las siguientes hipótesis:

H_0: Los residuos son homocedásticos.

H_1: Los residuos son heterocedásticos.

Obteniendo los siguientes resultados:

TABLA IX
TEST DE BREUSCH-PAGAN

Data	Modelo
BP	2.3566
GL (Grados Libertad)	3
Valor P	0.5018

Como se muestra en la Tabla 9, el valor P al ser 0.5018 (mayor a 0.05), no se rechaza H_0, por ende, no hay evidencia estadísticamente significativa de heterocedasticidad en el modelo.

Si el modelo se utiliza para clasificar o predecir ataques DoS (por ejemplo, analizando tráfico de red, número de paquetes, tiempo de respuesta, etc.), la homocedasticidad refuerza que el comportamiento del modelo es consistente bajo distintas

condiciones de tráfico IoT, lo cual ayuda a asegurar que la detección de ataques no se ve afectada por variaciones anómalas en los datos o por ruido en ciertas condiciones.

c) Multicolinealidad

El comando VIF en R sirvió para evaluar la multicolinealidad entre las variables independientes de un modelo de regresión; en este caso, las variables son: paquetes_enviados y tipo_ataque. Es así como, valores bajos de VIF (< 5 o < 10 según criterio) indican que no hay multicolinealidad preocupante entre las variables, el programa entrega como resultado valores cercanos a 1.1 que son muy bajos, lo cual sugiere que las variables son independientes entre sí. En el sistema IDS desarrollado significa que la variable paquetes_enviados (#de paquetes transmitidos) no depende fuertemente del tipo_ataque (por ejemplo, UDP flood, ICMP flood, etc.); por su parte, ambas aportan información distinta y complementaria para identificar patrones de ataque.

TABLA X
MULTICOLINEALIDAD

	GVIF	DF	$GVIF^{\frac{1}{2*DF}}$
paquetes_enviados	1.143748	1	1.069462
tipo_ataque	1.143748	2	1.034148

d) Autocorrelación (independencia de residuos)

El test de Durbin-Watson (DW) se usó para detectar autocorrelación en los residuos de un modelo de regresión, cuyo objetivo es comprobar si los errores del modelo están correlacionados, lo que podría indicar un patrón no aleatorio (por ejemplo, una dependencia temporal). Esto se maneja mediante dos hipótesis:

H_0: No existe autocorrelación de primer orden en los residuos.

H_1: Existe autocorrelación (positiva o negativa).

La interpretación del modelo viene dada por:

Un valor DW ≈ 2 sugiere ausencia de autocorrelación .

Un valor < 2 indica autocorrelación positiva.

Un valor > 2 indica autocorrelación negativa.

Aquí se obtuvo lo que se muestra en la tabla 11:

TABLA XI
TEST DURBIN-WATSON

Autocorrelación	Estadístico D-W	Valor P
0.07781389	1.8044988	0.52

Un DW muy cercano a 2, y el p-valor = 0.52 (> 0.05), lo que sugiere que no se rechaza la hipótesis nula de independencia, es decir no hay autocorrelación significativa en los residuos, además que el sistema es estable y no presenta comportamiento predecible que un atacante podría explotar. Si hubiera existido autocorrelación, se interpretaría como que el modelo no está captando correctamente la dinámica temporal de las conexiones IoT o del tráfico de red.

e) Linealidad

La figura 43 se usó para comprobar dos supuestos estadísticos importantes en modelos de regresión que son: Linealidad, es decir si la relación entre las variables independientes y la dependiente es lineal; y la Homocedasticidad (homogeneidad de varianza). Los residuos al presentar una dispersión constante a lo largo de todos los valores ajustados y con varianza constante, implica que:

- El modelo captura correctamente la relación entre las variables de tráfico y la ocurrencia de ataques;
- No hay sesgo sistemático, el modelo no sobreestima ni subestima los valores;
- El IDS mantiene estable rendimiento frente a distintos niveles de tráfico IoT.

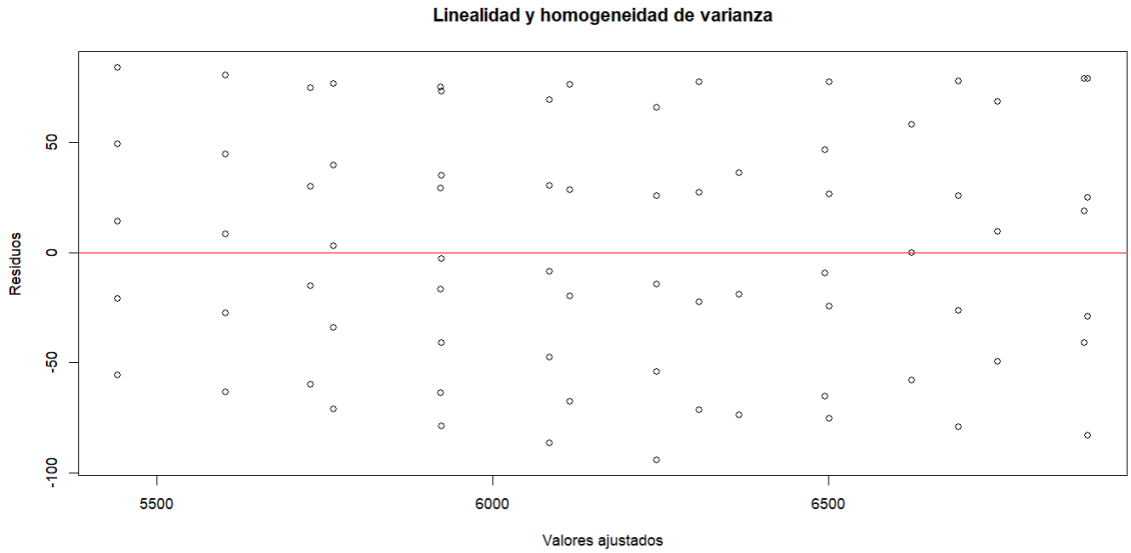


Fig. 43.Gráficos del Modelo de Regresión Múltiple

Este comportamiento se verificó mediante los diagnósticos presentados en la Tabla 7, los cuales mostraron indicadores asociados a heterocedasticidad, distribución de residuos y puntos de influencia. De igual manera, la Figura 42 complementó esta evaluación al mostrar los gráficos Residuals vs Fitted, Q–Q Plot y Scale–Location. En el gráfico Q–Q se evidenciaron ligeros desvíos respecto a la normalidad; sin embargo, el análisis confirmó que los supuestos de independencia de residuos y multicolinealidad se mantuvieron dentro de parámetros aceptables. Finalmente, la Figura 43 permitió identificar valores atípicos o residuales con influencia moderada, reafirmando la estabilidad general del modelo a pesar de las limitaciones detectadas.

En conjunto, los resultados permitieron concluir que el sistema de detección operó adecuadamente bajo condiciones controladas y fue capaz de identificar ataques DoS simples con alta efectividad. Sin embargo, los escenarios de tráfico combinado y saturación extrema revelaron fallos en la detección.

CAPITULO V CONCLUSIONES Y RECOMENDACIONES

Conclusiones

El presente estudio permitió demostrar que los dispositivos IoT son altamente vulnerables a ataques de denegación de servicio, especialmente a través de técnicas como SYN Flood, UDP Flood e ICMP Flood, que lograron saturar el ancho de banda y comprometer los recursos de los equipos conectados. La construcción del sistema de detección basado en Raspberry Pi, Suricata y la pila ELK evidenció que es posible implementar una solución de monitoreo accesible, funcional y de bajo costo, capaz de capturar, analizar y visualizar tráfico malicioso en tiempo real dentro de un entorno doméstico.

Los resultados reflejaron que Suricata detectó correctamente los ataques básicos y mantuvo un comportamiento constante en las alertas que generó. La pila ELK permitió observar el tráfico y comprender la dimensión y duración de los ataques. Sin embargo, cuando hubo ataques al mismo tiempo o uno tras otro, el sistema no funcionó: muchos ataques no fueron identificados (especialmente en momentos de gran carga), lo que mostró que el IDS pierde eficacia ante eventos que se repiten o tienden a repetirse.

El análisis estadístico confirmó que los datos recopilados son consistentes. Las distribuciones de paquetes enviados y las alertas detectadas mostraron estabilidad y sincronía. La matriz de correlación mostró una relación positiva fuerte entre las dos variables, lo que indicó que, a más tráfico malicioso, hubo un aumento en las alertas. El modelo de regresión múltiple ayudó an analizar los datos, pero el hecho de que no se cumpliera el supuesto de normalidad mostró que el modelo no pudo explicar completamente la variabilidad de los eventos observados. A pesar de esto, el comportamiento de los residuos fue lo suficientemente estable como para que el modelo se considere útil desde un enfoque evaluativo.

En su totalidad, el sistema implementado demostró ser adecuado para la detección de ataques DoS básicos y para proporcionar un apoyo visual y analítico al tráfico observado en una red de Internet de las Cosas doméstica. Sin embargo, su rendimiento experimentó una reducción en respuesta a ataques simultáneos, y su capacidad analítica se vio restringida por las limitaciones de hardware propias a la Raspberry Pi y por la sensibilidad de Suricata frente a situaciones de elevada saturación. El proyecto facilitó una solución técnica aplicable y demostró que la ciberseguridad doméstica puede robustecerse a través de herramientas de desarrollo abierto, siempre que se tomen en cuenta sus limitaciones y se realicen mejoras continuas.

Recomendaciones

Es aconsejable mejorar Suricata con nuevas reglas, firmas específicas para dispositivos IoT y ajustar los umbrales de alerta. Esto permitiría una mejor detección en casos de múltiples vectores de ataque o saturaciones prolongadas. Asimismo, sería adecuado optimizar el hardware que se está utilizando (tarjetas de red dedicadas o una Raspberry Pi más potente) para disminuir la pérdida de paquetes en los ataques.

Se sugiere realizar más pruebas con ataques distribuidos, tráfico encriptado, cargas altas y dispositivos IoT heterogéneos para evaluar el comportamiento del sistema en condiciones más realistas. Resultaría provechoso integrar métodos de aprendizaje automático para identificar patrones anormales sin limitarse a firmas predefinidas, expandiendo la capacidad del sistema con el fin de detectar nuevos ataques o variantes no registradas.

Integrar mecanismos de notificación automática que permitan alertar al usuario en tiempo real a través de correo electrónico, servicios web o plataformas de mensajería, evitando la necesidad de supervisión constante. Finalmente, se aconseja evolucionar el sistema hacia la creación de un Gateway doméstico de ciberseguridad que combine filtrado, inspección profunda de paquetes y monitoreo avanzado, ofreciendo una protección más completa y escalable para redes IoT domésticas.

BIBLIOGRAFÍA

[1] M. Gelgi, Y. Guan, S. Arunachala, M. Samba Siva Rao, and N. Dragoni, "Systematic Literature Review of IoT Botnet DDOS Attacks and Evaluation of Detection Techniques," *Sensors (Basel)*, vol. 24, no. 11, p. 3571, Jun. 2024, doi: 10.3390/S24113571.

[2] Software Engineering Institute, "1997 Tech Tip: Denial of Service Attacks." Accessed: Sep. 30, 2025. [Online]. Available: https://www.sei.cmu.edu/library/1997-tech-tip-denial-of-service-attacks/

[3] R. Vishwakarma and A. Jain, "A survey of DDoS attacking techniques and defence mechanisms in the IoT network," *Telecommun Syst*, vol. 73, no. 1, pp. 3–25, Jan. 2020, doi: 10.1007/S11235-019-00599-Z/METRICS.

[4] A. Pakmehr, A. Aßmuth, N. Taheri, and A. Ghaffari, "DDoS attack detection techniques in IoT networks: a survey," *Cluster Comput*, vol. 27, no. 10, pp. 14637–14668, Dec. 2024, doi: 10.1007/S10586-024-04662-6/FIGURES/6.

[5] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions," *Electronics 2020, Vol. 9, Page 1177*, vol. 9, no. 7, p. 1177, Jul. 2020, doi: 10.3390/ELECTRONICS9071177.

[6] F. Alsakran, G. Bendiab, S. Shiaeles, and N. Kolokotronis, "Intrusion Detection Systems for Smart Home IoT Devices: Experimental Comparison Study," *Communications in Computer and Information Science*, vol. 1208 CCIS, pp. 87–98, Jan. 2021, doi: 10.1007/978-981-15-4825-3_7.

[7] B. Paten, "IoT Security Challenges: Device Vulnerability & Attack Stats | PatentPC." Accessed: Sep. 30, 2025. [Online]. Available: https://patentpc.com/blog/iot-security-challenges-device-vulnerability-attack-stats?

[8] K. O. Chee, M. Ge, G. Bai, and D. D. Kim, "Unveiling the evolution of IoT threats: Trends, tactics, and simulation analysis," *Comput Secur*, vol. 157, p. 104537, Oct. 2025, doi: 10.1016/J.COSE.2025.104537.

[9] A. A. ; Alahmadi *et al.*, "DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions," *Electronics 2023, Vol. 12, Page 3103*, vol. 12, no. 14, p. 3103, Jul. 2023, doi: 10.3390/ELECTRONICS12143103.

[10] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, “A vision of IoT: Applications, challenges, and opportunities with China Perspective,” *IEEE Internet Things J*, vol. 1, no. 4, pp. 349–359, Aug. 2014, doi: 10.1109/JIOT.2014.2337336.

[11] M. Ansari, S. A. Ali, and M. Alam, “Internet of things (IoT) fusion with cloud computing: current research and future direction,” *International Journal of Advanced Technology and Engineering Exploration*, vol. 9, no. 97, pp. 1812–1845, 2022, doi: 10.19101/IJATEE.2021.876002.

[12] IOT-ANALYTICS, “Estado del IoT en 2018: El número de dispositivos IoT ahora es de 7 mil millones – El mercado se está acelerando.” Accessed: Sep. 30, 2025. [Online]. Available: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/

[13] J. López, “Detección y Mitigación de Ataques de Denegación de Servicio en Redes IoT Usando Inteligencia Artificial (IA) y Técnicas de Aprendizaje Automático (ML),” Tesis de Grado. Universitat Oberta de Catalunya., 2024.

[14] LevelBlue, “Open-Source Intrusion Detection Tools Overview | LevelBlue.” Accessed: Sep. 30, 2025. [Online]. Available: https://levelblue.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview

[15] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, “Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms,” *Sensors 2024, Vol. 24, Page 713*, vol. 24, no. 2, p. 713, Jan. 2024, doi: 10.3390/S24020713.

[16] K. Ashton, “Esa cosa del ‘Internet de las cosas’ - RFID JOURNAL.” Accessed: Oct. 01, 2025. [Online]. Available: https://www.rfidjournal.com/expert-views/that-internet-of-things-thing/73881/

[17] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013, doi: 10.1016/J.FUTURE.2013.01.010.

[18] ITU, “International Telecommunication Union (ITU). (2012). Overview of the Internet of Things. ITU-T Y.2060. - Buscar con Google.” Accessed: Oct. 01, 2025. [Online]. Available: https://www.google.com/search?q=International+Telecommunication+Union+(ITU).+(2012).+Overview+of+the+Internet+of+Things.+ITU-T+Y.2060.&rlz=1C1CHZN_enEC1095EC1095&oq=International+Telecommunicat

ion+Union+(ITU).+(2012).+Overview+of+the+Internet+of+Things.+ITU-T+Y.2060.&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBBzE5MWowajSoAgCwAgE&sourceid=chrome&ie=UTF-8

[19] R. Singh and S. Gill, "Edge AI: A survey," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 71–92, Jan. 2023, doi: 10.1016/J.IOTCPS.2023.02.004.

[20] I. Ficili, M. Giacobbe, G. Tricomi, and A. Puliafito, "From Sensors to Data Intelligence: Leveraging IoT, Cloud, and Edge Computing with AI," *Sensors 2025, Vol. 25, Page 1763*, vol. 25, no. 6, p. 1763, Mar. 2025, doi: 10.3390/S25061763.

[21] S. Almass and S. K. Chowdhary, "Comprehensive Study on Cyber Security and Cyber Attacks," *Proceedings - 1st International Conference on Electronics, Communication and Signal Processing, ICECSP 2024*, 2024, doi: 10.1109/ICECSP61809.2024.10698540.

[22] Fortinet, "¿Qué es un ataque de hombre en el medio (MITM)? Tipos y ejemplos | Fortinet." Accessed: Oct. 26, 2025. [Online]. Available: https://www.fortinet.com/lat/resources/cyberglossary/man-in-the-middle-attack

[23] M. Stampar, "Data Retrieval over DNS in SQL Injection Attacks," Mar. 2013, Accessed: Oct. 26, 2025. [Online]. Available: http://arxiv.org/abs/1303.3047

[24] Malwarebyts, "¿Qué es el ransomware? | Protección contra ransomware." Accessed: Oct. 26, 2025. [Online]. Available: https://www.malwarebytes.com/es/ransomware

[25] E. Andrés, F. Quilumbango, R. Omar, A. Paredes, M. Vinicio, and Y. Yugsi, "Protección contra ataques de ingeniería social: análisis cualitativo de estrategias, tecnologías y patrones específicos," *Religación*, vol. 10, no. 44, pp. e2501310–e2501310, Oct. 2025, doi: 10.46652/RGN.V10I44.1310.

[26] T. Sasi, A. H. Lashkari, R. Lu, P. Xiong, and S. Iqbal, "A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges," *Journal of Information and Intelligence*, vol. 2, no. 6, pp. 455–513, Nov. 2024, doi: 10.1016/J.JIIXD.2023.12.001.

[27] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, Apr. 2004, doi: 10.1145/997150.997156.

[28] M. Roesch, "Snort-Lightweight Intrusion Detection for Networks," 1999.

[29] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A. R. Sadeghi, "DIoT: A Federated Self-learning Anomaly Detection System for IoT," *Proc*

Int Conf Distrib Comput Syst, vol. 2019-July, pp. 756–767, Apr. 2018, doi: 10.1109/ICDCS.2019.00080.

[30] S. Plans, "Desarrollo de una SIEM mediante ELK," Tesis de Grado. Universitat de Barcelona, 2019.

[31] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, Feb. 2020, doi: 10.1016/J.JISA.2019.102419.

[32] Kaspersky, "Seguridad de la Internet de las cosas." Accessed: Oct. 01, 2025. [Online]. Available: https://latam.kaspersky.com/resource-center/preemptive-safety/best-practices-for-iot-security

[33] EDPS, "Aprendizaje federado | Supervisor Europeo de Protección de Datos." Accessed: Oct. 05, 2025. [Online]. Available: https://www.edps.europa.eu/press-publications/publications/techsonar/federated-learning_en

[34] A. Dessiatnikoff, R. Akrout, E. Alata, M. Kaaniche, and V. Nicomette, "A clustering approach for web vulnerabilities detection," *Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing, PRDC*, pp. 194–203, 2011, doi: 10.1109/PRDC.2011.31.

[35] Kali, "hping3 | Herramientas de Kali Linux." Accessed: Oct. 05, 2025. [Online]. Available: https://www.kali.org/tools/hping3/

[36] Raspberry Pi, "¿Que es Raspberry Pi? - Raspberry Pi." Accessed: Oct. 27, 2025. [Online]. Available: https://raspberrypi.cl/que-es-raspberry/

[37] Incibe, "Glosario de términos de ciberseguridad," 2020.

[38] Lady Cañón, "Ataques Informáticos, Ethical Hacking y Conciencia de Seguridad Informática en niños.," 2015.

[39] L. Fernández, "Ataque TCP SYN: Qué es y cómo mitigar este ataque DoS." Accessed: Oct. 05, 2025. [Online]. Available: https://www.redeszone.net/tutoriales/seguridad/ataque-syn-que-es/

[40] J. Polivio, "Simulación y análisis de mecanismos de defensa ante los ataques de denegación de servicio (DoS) en redes de área local convergentes," Tesis de grado. Escuela Politécnica Nacional, 2011.

[41] Snort, "Snort - Network Intrusion Detection & Prevention System." Accessed: Oct. 06, 2025. [Online]. Available: https://www.snort.org/

[42] J. Astudillo, F. Flores, A. Macías, and A. Aranda, “Adaptación del IDS/IPS Suricata para que se pueda convertir en una solución empresarial,” 2012.

[43] CertSi, “Diseño y Configuración de IPS, IDS y SIEM en Sistemas de Control Industrial,” 2017. [Online]. Available: http://www.incibe.es.

[44] ToBeIt, “¿Qué es Elastic Stack o Stack ELK? - ToBeIT.” Accessed: Oct. 06, 2025. [Online]. Available: https://tobeit.es/que-es-stack-de-elastic-o-stack-elk/

ANEXOS

Anexo 1

Configuración del Access Point

Para la creación del punto de acceso (AP), se parte de generar un archivo en la ruta /etc/hostapd/hostapd.conf. Es necesario completar varias líneas de configuración. La estructura del archivo variará según el hardware específico y el esquema de seguridad que se desee implementar, y podría configurarse de la siguiente forma:

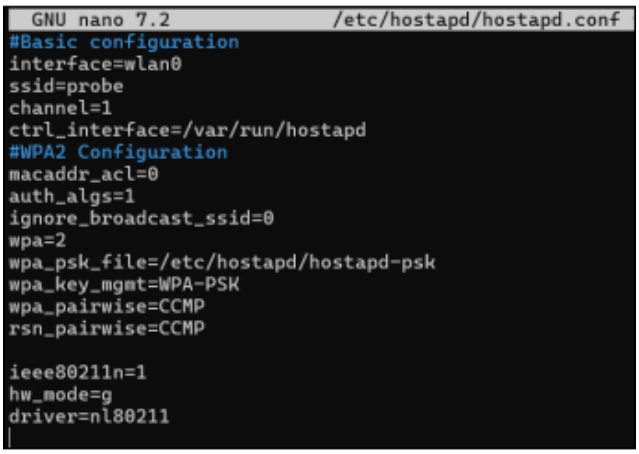


Fig. 44.Archivo hostapd.conf

Para redirigir el tráfico adecuadamente entre una red inalámbrica y la internet, se requiere ajustar un servicio NAT y permitir la redirección IPv4 en las interfaces de red de la red. Inicialmente, se debe ajustar o agregar la línea pertinente en el archivo "/etc/sysctl.conf" con la orden

net.ipv4.ip_forward=1

La línea de configuración anterior servirá para habilitar la redirección IPv4 al inicio del sistema. esto permitirá una navegación óptima entre ambas redes.

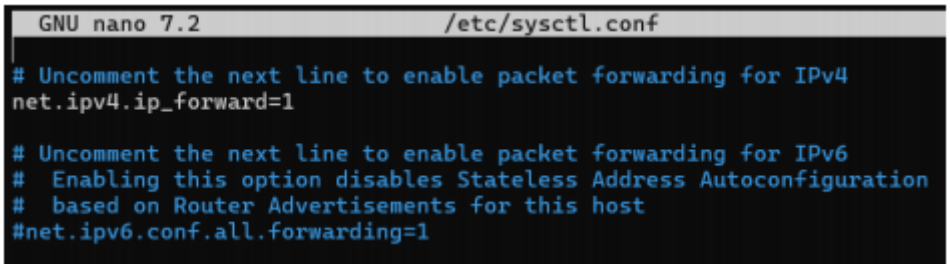


Fig. 45.Archivo sysctl.conf

Anexo 2.

Configuración de las reglas en Suricata

Para crear reglas específicas se va a la ruta ***/etc/suricata/rules*** colocando un nombre al archivo, por ejemplo, (ejemplo.rules). Este paso de establecer reglas entorno a los protocolos TCP, UDP e ICMO para detectar ataques DoS es crucial; Esto incluye alertar sobre conexiones excesivas en TCP, identificar patrones anómalos en UDP y volumen inusual de solicitudes en ICMP. Esta técnica asegura una defensa sólida, al mismo tiempo que adopta un enfoque ético al proteger los recursos y mantener la integridad de la red.

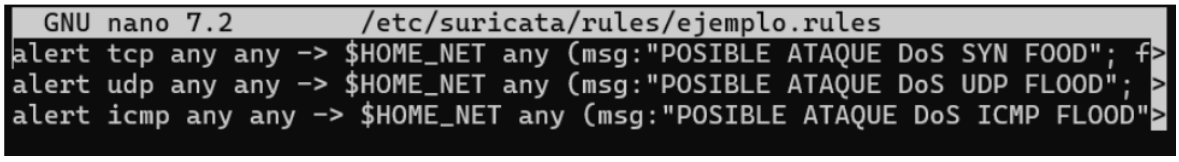


Fig. 46.Reglas de Suricata

Lo anterior se puede interpretar de la siguiente manera:

- Alert: Es la acción que se tomará cuando se cumpla la regla.
- Tcp, Udp, Icmp: Especifican los protocolos.
- any any → any 80: La regla se aplicará a cualquier dirección IP y puerto origen, con destino a cualquier dirección IP en el puerto 80 (HTTP).
- msg: Cuando se active la regla, suricata mostrará este mensaje.
- threshold: Son las condiciones para que se considere un ataque (más de 20 conexiones en 1 segundo desde la misma fuente).

- sid: Identificador único de la regla.
- rev: # de revisión de la regla.