ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

DISEÑO E IMPLEMENTACIÓN DE UN FIREWALL PARA DISPOSITIVOS IOT UTILIZANDO RASPBERRY, APLICANDO TECNOLOGÍAS IPS PARA DETECTAR TRÁFICO MALICIOSO

Trabajo de Titulación

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

AUTOR:

ANDRÉS MATEO ÁLVAREZ RAMÍREZ

Riobamba – Ecuador

2022

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

DISEÑO E IMPLEMENTACIÓN DE UN FIREWALL PARA DISPOSITIVOS IOT UTILIZANDO RASPBERRY, APLICANDO TECNOLOGÍAS IPS PARA DETECTAR TRÁFICO MALICIOSO

Trabajo de Titulación

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

AUTOR: ANDRÉS MATEO ÁLVAREZ RAMÍREZ

DIRECTOR: Ing. MARCO VINICIO RAMOS VALENCIA MSc.

Riobamba – Ecuador

2022

Yo, ANDRÉS MATEO ÁLVAREZ RAMÍREZ, declaro que el presente Trabajo de Titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación; El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 20 de julio del 2022

Andrés Mateo Álvarez Ramírez

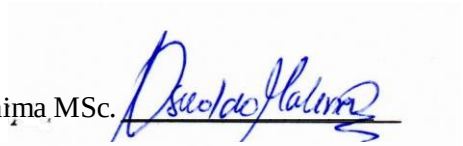
180447523-2

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

El tribunal del Trabajo de Titulación certifica que: El Trabajo de Titulación; Tipo: Proyecto Técnico, **DISEÑO E IMPLEMENTACIÓN DE UN FIREWALL PARA DISPOSITIVOS IOT UTILIZANDO RASPBERRY, APLICANDO TECNOLOGÍAS IPS PARA DETECTAR TRÁFICO MALICIOSO**, de responsabilidad del señor **ANDRÉS MATEO ÁLVAREZ RAMÍREZ**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal autoriza su presentación.

	FIRMA	**FECHA**
Ing. Oswaldo Geovanny Martínez Guashima MSc. **PRESIDENTE DEL TRIBUNAL**		2022-07-20
Ing. Marco Vinicio Ramos Valencia MSc. **DIRECTOR DEL TRABAJO DE TITULACIÓN**		2022-07-20
Ing. Alberto Leopoldo Arellano Aucancela MSc. **MIEMBRO DE TRIBUNAL**		2022-07-20

DEDICATORIA

Dedico el presente proyecto con especial cariño a mis padres Jorge y Rosa, que siempre han estado para apoyarme en todas las etapas de mi vida, así como también a mis hermanos David y Naty por ser el ejemplo de nunca dejar de luchar por los objetivos y también a mis cuñados Isaac, Nani, y mi sobrinita Manu, que con su compañía han llenado de alegría mi vida.

Andrés

AGRADECIMIENTO

A la Escuela Superior Politécnica de Chimborazo, por permitirme culminar mis estudios y ser el impulso para continuar progresando como persona de bien. A todos los docentes y personal administrativo de la carrera de Ingeniería Electrónica en Telecomunicaciones y Redes que de forma directa o indirecta contribuyeron en mi formación académica, incentivando la curiosidad y la investigación. Igualmente quiero agradecer a mis amigos que también fueron un gran apoyo en los momentos difíciles y aunque en muchos de los casos nuestras vidas tomaron rumbos diferentes siempre estaré agradecido con todos.

Andrés

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE GRÁFICOS

INDICE DE ANEXOS

INDICE DE ABREVIATURAS

ARP	Protocolo de resolución de direcciones
AMQP	Protocolo avanzado de colas de mensajes
CAGR	Mide la tasa de crecimiento anual recolectando datos por 5 años.
DARPA	Agencia de Proyectos de Investigación Avanzada del Departamento de Defensa
DCCP	Protocolo de Control de Congestión de Datagramas
DDoS	Ataque de Denegación de Servicio Distribuido
DNS	Sistema de Dominio de Nombres
DHCP	Protocolo Dinámico de Configuración de Host
FTP	Protocolo de Transferencia de archivos
GSM	Sistema Global para las Comunicaciones Móviles
HTTP	Protocolo de Transferencia de Hipertexto
HTTPS	Protocolo Seguro de Transferencia de Hipertexto
IDS	Sistema de Detección de Intrusos
IGMP	Método de escucha del tráfico Multidifusión
IoT	Internet de las Cosas
IP	Protocolo de Internet
IPS	Sistema de Prevención de Intrusos
IPSEC	Protocolo de Seguridad de Internet
JSON	Notación de Objetos de JavaScript
LTE	Evolución a Largo Plazo
L2TP	Protocolo de Túnel de Capa 2
MAC	Control de Acceso al Medio
MARS	Sistema responsivo de análisis y monitoreo
MQTT	Protocolo de transporte de telemetría de cola de mensaje
OSI	Interconexión de Sistemas Abiertos
PC	Computadora Personal
RFID	Identificador por Radiofrecuencia

RESUMEN

El objetivo del presente proyecto fue diseñar e implementar un Firewall de nueva generación utilizando las tecnologías de detección y prevención de intrusos (IDS-IPS), para una red domestica de internet de las cosas (IoT), teniendo como finalidad proteger la red de ataques informáticos. Se analizó las diferentes metodologías existentes en el área, permitiendo que se desarrollara el proyecto mediante una metodología híbrida y finalmente se verificó que la metodología propuesta cumpliera con los objetivos propuestos. En primer lugar, se realizó la recolección de datos para conocer las vulnerabilidades de dispositivos de internet de las cosas, una vez conocidas las vulnerabilidad se estudió las posibles soluciones mediante es estado del arte de la seguridad informática y la Taxonomía, posteriormente se implementó el escenario en el que se realizaron las pruebas necesarias para determinar la solución mas eficiente y finalmente se aplicó una solución que se ajustaba a las características de la red sin afectar al rendimiento de la red y las características limitadas de los dispositivos. Finalmente, en las conclusiones se detalló los resultados obtenidos al ejecutar ataques informáticos y como respondió el sistema de protección, y de igual manera en las recomendaciones se describió todos los ajustes que fueron necesarios realizar para que el sistema tenga un comportamiento adecuado, además de proponer varias soluciones a diferentes problemas de software que aparecieron mientras se desarrollaba el proyecto.

Palabras clave: <FIREWALL> <SISTEMA DE PREVENCIÓN DE INTRUSOS> <INTERNET DE LAS COSAS (IOT)> <RASPBERRY PI> <SEGURIDAD INFORMATICA>.

SUMMARY

This research aimed to design and implement a new generation Firewall using an intrusion detection and prevention system (IDS-IPS) for an Internet of Things (IoT) home network to protect the network from cyber-attacks. The different existing methodologies in the area were analyzed, allowing the project to be developed through a hydrid method, and finally, it was verified that the proposed methodology met the proposed objectives. In the first place, the data collection was carried out to know the vulnerabilities of internet of things devices; once the vulnerability was known, the possible solutions were studied through the state of the art of computer security and Taxonomy, and later the scenario was implemented. The tests were conducted to determine the most efficient solution, and finally, a solution that adjusted to the network's characteristics without affecting the network's performance and the devices' limited characteristics was applied. Finally, the conclusions were detailed, the results obtained when executing cyber-attacks and how the protection system responded, and in the same way in the recommendations, all the adjustments that were necessary to make so that the system has an acceptable behavior were described in addition to proposing several solutions to different software problems that appeared while the project was developing

Keywords: <FIREWALL> <INSTRUSION PREVENTION SYSTEM> <INTERNET OF THINGS (IOT)> <RASPBERRY PI> <COMPUTER SECURITY>.

0602546103

INTRODUCCIÓN

Actualmente los dispositivos con los que las personas se conectan a internet se han multiplicado considerablemente, La diversidad de estos equipos han estructurado lo que ahora conocemos como Internet de las Cosas (IoT por sus siglas en ingles), esta visión permite que sensores y otros dispositivos puedan intercambiar información a través de internet.

El incremento de ataques informáticos mediante diversas técnicas ha provocado que cada vez se requiera un análisis más profundo en los sistemas y equipos informáticos, de forma que la red minimice al máximo los daños que un ataque pueda producir.

Por esta razón el desarrollo del presente proyecto está enfocado en el diseño e implementación de un firewall que sea capaz no solo de filtrar accesos a la red, sino que también pueda analizar el tráfico con el fin de prevenir ataques informáticos aprovechando las fortalezas que brindan los sistemas IPS (Sistema de Prevención de intrusos), dentro de un entorno IoT doméstico.

Para conseguir los objetivos planteados en el proyecto será necesario conocer cada uno de los sistemas y equipos; con este fin, en el primer capítulo se plantean los conceptos necesarios sobre seguridad informática, técnicas de seguridad informática, tráfico malicioso, firewalls, IDS, IPS; así como también el software con el que se desarrolló el proyecto y hardware utilizado. El estudio teórico de todos estos elementos permite que el desarrollo del proyecto considere los parámetros recomendados por las instituciones especializadas en el área de la seguridad informática.

Al desarrollar el marco metodológico en el capítulo dos será necesario implementar de forma física una red IoT administrada por placas Raspberry Pi, ya que estos dispositivos cumplirán la función de Gateway atreves de los servicios que nos ofrecen las plataformas Webthings Gateway y raspAP.

La función más importante que debe cumplir una de las placas Raspberry Pi es la de firewall IPS siendo este el objetivo principal del proyecto, ya que mediante este sistema debemos ser capaces de identificar las anormalidades que surgen en el tráfico cuando la red se encuentra bajo un ataque informático, de manera que tome acciones en los momentos iniciales antes de que la red sufra una degradación en sus servicios.

Al someter nuestro sistema de seguridad ha ataques informáticos será posible conocer el grado de seguridad que se puede brindar a una red doméstica mediante la implementación de un sistema de acción preventiva, también determinar si existe alguna alteración en la comunicación de los

dispositivos IoT. Después de analizar el rendimiento de todo el sistema, en el capítulo tres se aportarán conclusiones y recomendaciones basadas en los resultados obtenidos para contribuir a las investigaciones y realización de proyectos futuros.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

1.1. Antecedentes

El mundo digital y de las telecomunicaciones se encuentra en un punto de transformación irreversible; ya que, al dar el paso hacia una conexión global de dispositivos electrónicos, nuestro estilo de vida cambiará drásticamente. El internet de las cosas conocido como IoT (Internet of Things) lo que busca es la digitalización de todos los dispositivos que interactúan con nuestro entorno, de forma que sean capaces de darle sentido a la información que transmiten o reciben en tiempo real, debido a que esta sería la manera en que podemos optimizar nuestras actividades y el consumo de los recursos. Todo este proceso actualmente se conoce como la industria 4.0, la cual abarca diversas activades, desde algo tan simple como encender las luces de nuestra casa de forma remota hasta monitorear los signos vitales de una persona (Incibe 2020, pp. 4-7)

Al encontrarse la tecnología IoT en un proceso temprano de desarrollo la principal característica con la que cuenta esta tecnología es su capacidad de transferir datos a cualquier parte del mundo mediante internet, también se ha convertido en el punto más crítico de todo el sistema IoT, ya que brinda a los ciberdelincuentes acceso a todos nuestros datos personales si la red por la que viajan estos datos no se encuentran protegidos correctamente (Incibe 2020, pp. 8-14).

Investigaciones sobre ataques informáticos como los realizados por Kaspersky concluyen que uno de los vectores con los que cuentan los atacantes para ejecutar sus operaciones es la poca importancia que los usuarios le dan a la protección de sus dispositivos, sin considerar que su información personal está siendo transferida por los dispositivos y que para un atacante capturar estos datos le puede traer ganancias de diferentes maneras.

También se ha demostrado mediante el uso de honeypots que gran parte de los ataques infectan a los dispositivos con malware permitiendo crear redes remotas desde las cuales dirigir ataques DDoS o convertirlos en proxy para aumentar el anonimato de los ciberdelincuentes. Una característica que resalta de los ataques IoT es que, al estudiar su anatomía, el ataque no comprende procedimientos muy complejos, pero en contrapartida son muy sigilosos, siendo prácticamente imposibles de detectar para un usuario promedio; uno de los malware más utilizados por ofrecer estas características es Mirai, el cual está implicado en el 39% de todos los ataques a dispositivos IoT. Otro de los métodos más utilizados es la fuerza bruta la cual intenta descifrar claves y contraseñas de usuarios, siendo muy efectivo en los casos donde los dispositivos

mantienen sus credenciales de fábrica, actualmente el software que ha tenido mayor participación en este tipo de ataques es el malware Nyadrop, siendo encontrado en el 38,57% de los ataques. A nivel Global el índice de más infecciones posiciona a China como el país que sufre más ataques informáticos para dispositivos IoT alcanzando el 30% de todos los ataques a nivel global. Dentro de la región de Latinoamérica el país con mayor número de ataques es Brazil con el 19% de todos los ataques registrados (latam.kaspersky, 2021, p.5).

Desgraciadamente en Ecuador no existen estadísticas claras sobre ataques hacia dispositivos IoT pero considerando que el uso de softwares piratas es común, adquisición a gran escala de equipos chinos y la poca importancia que se da al área de la seguridad informática convierten al país en un punto atractivo para ejecutar ataques, esto se puede comprobar analizando otro tipo de ataques, por ejemplo Ecuador se encuentra en el noveno puesto a nivel global de víctimas de phishing y cuadragésimo primero en infecciones por malware en dispositivos móviles (latam.kaspersky, 2019, p.21).

Todos estos factores proyectan un incremento en la necesidad de contratar profesionales en el área de seguridad informática y sistemas más eficientes de seguridad, de forma que la industria pueda dar el paso hacia una conectividad IoT mucho más segura. Los puntos en los que se intenta cimentar el futuro tecnológico que nos espera son el compromiso social de forma que seamos conscientes y responsables de la tecnología que utilizamos, también profesionales que adapten sus conocimientos a las situaciones que se requieran y sistemas mucho más "inteligentes" capaces de auto depurarse intentando, de ser posible, predecir situaciones antes de que sean una amenaza real (Vialynk, 2021, p.10).

1.2. Formulación del problema

¿Es posible crear un entorno seguro para proteger a los dispositivos IoT mediante la implementación de un firewall con la Rasberry Pi aplicando tecnologías IPS para detectar tráfico malicioso?

1.3. Sistematización del problema

¿Cuáles es el nivel de protección actual de los dispositivos IoT ante un ataque informático y cómo afecta a una red doméstica?
¿Los parámetros que se analizan en un sistema de seguridad de prevención de intrusiones (IPS) se encuentran presentes en una red IoT?
¿Cuál es el comportamiento del tráfico de los dispositivos IoT ante un ataque informático?

¿La aplicación de un firewall con tecnología de seguridad de prevención de intrusiones (IPS) ofrece un nivel de protección mayor que al no utilizar este sistema en una red IoT?

1.4. Justificación teórica

El hecho de que nos encontremos en un punto donde todas nuestras actividades se digitalicen mediante sensores y actuadores ha permitido a los sistemas computacionales concentrar una cantidad de datos nunca antes esperada y es fundamental convertirla en información útil para nuestro propio beneficio, y es por este motivo que los sistemas IoT aparecen para aprovechar toda esta información, creando múltiples vías de recolección de datos y solucionando tareas cada vez más complejas.

Actualmente no es extraño que mediante un smartphone podamos tener un control total de nuestros electrodomésticos, pero el riesgo de ser víctimas de ataques informáticos también ha incrementado, convirtiéndose en un punto débil de toda red que implemente este tipo de tecnologías, y hasta hace unos pocos años lo habitual era atacar a computadores personales o servidores para degradar sus servicios o sustraer sus datos, pero en el caso de los dispositivos IoT, las intenciones de un atacante son diferentes y las consecuencias igualmente diferentes.

Por ejemplo, creando redes de botnets que permiten ataques DDoS hacia una empresa se puede dañar su prestigio, confianza y proyección, o en el caso de una persona puede ser afectado su perfil digital. Por este motivo es evidente que el comportamiento, usos y tipos de datos que manejan no son similares entre computadores y dispositivos IoT, estas circunstancias requieren un estudio específico basado en las características de los dispositivos IoT.

Además, se requiere adaptar las técnicas de protección para que se enfoquen el comportamiento de los dispositivos antes de sufrir un ataque, de forma que se configure el sistema de protección a medida de los requerimientos de la red, esto significa que al implementar un sistema de seguridad debemos tomar en cuenta características de los dispositivos por ejemplo el tipo de arquitectura con el que son construidos debido a que los dispositivos IoT no realicen tareas de alto procesamiento pero requieren bajas latencias y servicios de red específicos.

Tampoco es posible que los mismos dispositivos IoT apliquen mecanismos de protección avanzados debido a que sus recursos de hardware y software son limitados, por lo que es necesario transferir esta carga computacional a equipos externos capaces de protegerlos. Investigaciones ya proponen aspectos que deberían ser tomados en cuenta al diseñar un sistema de seguridad para este tipo de dispositivos; entre los puntos más importantes se encuentran analizar escenarios

habituales en los que puedan presentarse vulnerabilidades de seguridad, esto incluye modificación de contraseñas por defecto, configuración y limitación de los servicios requeridos por la red, actualización de dispositivos y de firewall, reconocimiento de comportamientos anormales por parte de los equipos.

Bajo estos parámetros el objetivo en el que se centra el presente trabajo es el de contribuir un sistema de bajo costo capaz de garantizar la seguridad dentro de una red IoT, con la capacidad de detectar y actuar ante el tráfico malicioso generado por ataques informáticos, teniendo un enfoque preventivo, para los casos en que el ataque se encuentre en una fase temprana y no ha sido capaz de afectar a los dispositivos IoT presentes en la red.

La investigación aporta al desarrollo de sistemas de seguridad informática capaces de adaptarse a los ambientes domésticos de los dispositivos IoT y así brindar una capa de protección adicional a las ya existentes, evitando la instalación de equipos costosos, que ofrecen servicios sobredimensionados en comparación a lo que realmente requiere una red IoT doméstica.

1.5. Justificación aplicativa

Este proyecto se centra en la aplicación de sistemas informáticos de seguridad para prevenir y proteger una red doméstica de dispositivos IoT, tales como cámaras IP, monitores IP, asistentes por voz, smart TV, Smartphone, entre otros; de forma que la información que manejen estos equipos se encuentre protegida de la forma más robusta y eficiente posible.

Los métodos con que cuentan actualmente y más utilizadas por los dispositivos IoT son las autentificaciones con usuario y contraseña, las cuales son fáciles de descubrir si no son robustas, certificados que validan la identidad de un servicio, pero requieren una constante validación y se vuelven inseguros y fáciles de clonar, también la utilización de VPN privadas y servicios propios de las empresas lo que impide convergencia entre tecnologías además de que el sistema se vuelve dependiente de la empresa proveedora del servicio VPN.

Todos estos métodos impiden que las empresas dedicadas a la administración y análisis de redes puedan dar una solución eficiente. Por este motivo es importante realizar investigaciones con el fin de desarrollar métodos que se adapten a las características de los sistemas IoT, los cuales requieren muy baja latencia, estabilidad en la comunicación y anchos de banda relativamente bajos, por lo que implementar sistemas de seguridad que no se ajusten a estas características producirían desperdiciar recursos, además, de causar molestias al usuario promedio que no tiene conocimientos avanzados para su correcta configuración.

Conseguir un sistema domestico que garantice la seguridad de nuestros datos requiere que se apliquen las recomendaciones por instituciones especializadas y utilizar tecnologías que vayan acorde con las necesidades actuales, por lo que proyectos de estas características pueden aportar ideas en benéfico de una conectividad segura y eficiente para la sociedad.

1.6. Objetivos

1.6.1. Objetivo general

Diseñar e implementar firewall utilizando una Raspberry Pi para dispositivos IoT aplicando técnicas de seguridad informática y utilizando la tecnología IPS para detectar tráfico malicioso.

1.6.2. Objetivos específicos

- Estudiar el nivel de protección que se presentan actualmente en los dispositivos IoT.
- Definir las características y el comportamiento del tráfico en condiciones normales y bajo ataques informáticos.
- Diseñar una red para dispositivos IoT mediante la Raspberry Pi y aplicar las configuraciones necesarias.
- Analizar el tráfico de la red implementada con los dispositivos IoT identificando las anomalías producidas por un ataque informático.
- Verificar el nivel de protección que se obtiene ante los ataques informáticos al aplicar el firewall con sistema de prevención de intrusión dentro de la red IoT.

CAPÍTULO II

2. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS

2.1. Seguridad Informática

Para entender lo que significa la Seguridad Informática primeramente se debe entender que la palabra seguridad define un estado de bienestar debido a la ausencia de riesgo presente en un ser o un objeto. Dentro de la informática es considerada una ciencia interdisciplinaria enfocada en evaluar y gestionar los riesgos de forma que se puedan planear métodos para evitar o prevenir situaciones desfavorables. Su definición engloba 4 acciones indispensables que se deben cumplir (Castro et al., 2018, pp. 13-28):

- Prevención del Riesgo
- Transferir el riesgo
- Mitigar el riesgo
- Aceptar el riesgo

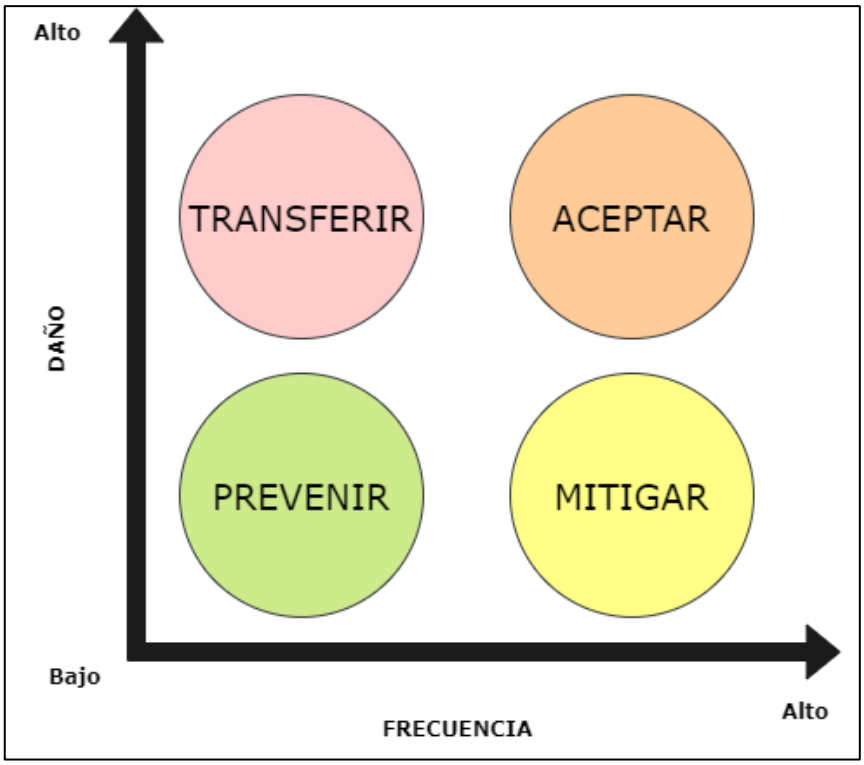


Figura 1-2: Acciones según las características de los ataques informáticos.

Fuente: (Castro et al., 2018).

Por lo tanto, la Seguridad Informática está definida como la ciencia enfocada en la seguridad del medio informático; cabe señalar que la informática abarca los procesos, técnicas y métodos enfocados en el procesamiento, almacenaje y transmisión de la información (Castro et al., 2018).

La seguridad informática como una disciplina encargada de plantear y diseñar las normas, procedimientos, métodos y técnicas con el fin de que un sistema de información sea seguro, confiable y sobre todo que tenga disponibilidad. De forma práctica mediante la Seguridad Informática se intenta minimizar los riesgos a los que está expuesta la información, debido al grado de complejidad de los sistemas que recopilan datos, se generan vectores de riesgo para el mismo sistema, y nunca se podrá garantizar que un sistema está cien por ciento seguro. La Seguridad Informática debe considerar tres partes fundamentales en las que se centran los análisis(Castro et al., 2018):

- Usuarios, los cuales pertenecen al eslabón más débil de la cadena ya que siempre existe la posibilidad de un error voluntario o involuntario y son mucho más difíciles de auditar.
- La información, es el recurso más importante de la seguridad informática ya que todo el esfuerzo se centra en salvaguardar su contenido
- La infraestructura, de todas las partes es la que mayor control se puede tener y se debe en gran parte a la manipulación que se realiza en el sistema, por el cual se puede alcanzar las otras dos partes de la seguridad informática. Es importante siempre considerar que pueden existir problemas complejos como un acceso de personal no autorizado, pero también daños comunes como un incendio.

2.1.1. Definiciones de Vulnerabilidad, Amenaza, Riesgo

- **Vulnerabilidad:** hace referencia a los puntos débiles conocidos que se presentan en el sistema, pueden ser causados por defectos propios del sistema y que bajo este conocimiento permiten realizar un ataque (Tamayo Ottati, 2020).
- **Amenaza:** se relaciona con la probabilidad de que, al desarrollar un ataque informático, este sea capaz de afectar a nuestro sistema, siendo conocidas las vulnerabilidades o no. En el caso de que nunca antes se haya detectado dicha amenaza se denomina ataque de día cero (Tamayo Ottati, 2020).
- **Riesgo:** mediante el riesgo se intenta definir el nivel de daño que podría sufrir el sistema a ser víctima de un ataque, pudiendo existir perdida de información, o daño irreparable de algún componente del sistema(Tamayo Ottati, 2020).

Figura 2-2: Parámetros que intervienen en la protección de sistemas informáticos.

Fuente: (Rivas 2020).

2.1.2. Definición de taxonomía

La taxonomía de forma general es definida como el método de estructurar la información y que puede estudiarse separándola en categorías y subcategorías, de forma que se extraiga las entidades que puedan ser relacionadas y de esta forma llegar a ser comprendidas (Codina, 2019).

En el campo de la seguridad informática es fundamental disponer de una taxonomía común ya que de esta forma se podrá extraer la información necesaria para comprender el ataque. Hay que ser conscientes de que no todos los ataques tienen las mismas características y al aplicar la taxonomía seremos capaces de clasificar y relacionar los ataques con las medidas que se deban tomar para su prevención, contención y erradicación de ataques informáticos (Incibe, 2019).

En el mundo de la ciberseguridad existen muchos casos en los que se aplica la taxonomía para resolver un ataque informático, un ejemplo claro es como los sistemas de correo electrónico descartan los mensajes maliciosos ya que en estos sistemas es necesario estudiar la amenaza desde varias perspectivas para que el sistema descarte de forma correcta los mensajes no deseados. Basados en la taxonomía se puede plantear en primer lugar como se clasifican las amenazas por ejemplo dividiéndolos en mensajes maliciosos, spam o correo gris; por otra parte, se puede tomar en cuenta la autenticidad del remitente, siendo el caso de que la dirección de origen sea una cuenta temporal, suplantada o falsa. Con estos parámetros muchos de los sistemas de correo electrónico son capaces de identificar correos maliciosos pero otros sistemas pueden ir mucho más profundo buscando datos específicos del remitente, intensiones o motivación (Hassold, 2019).

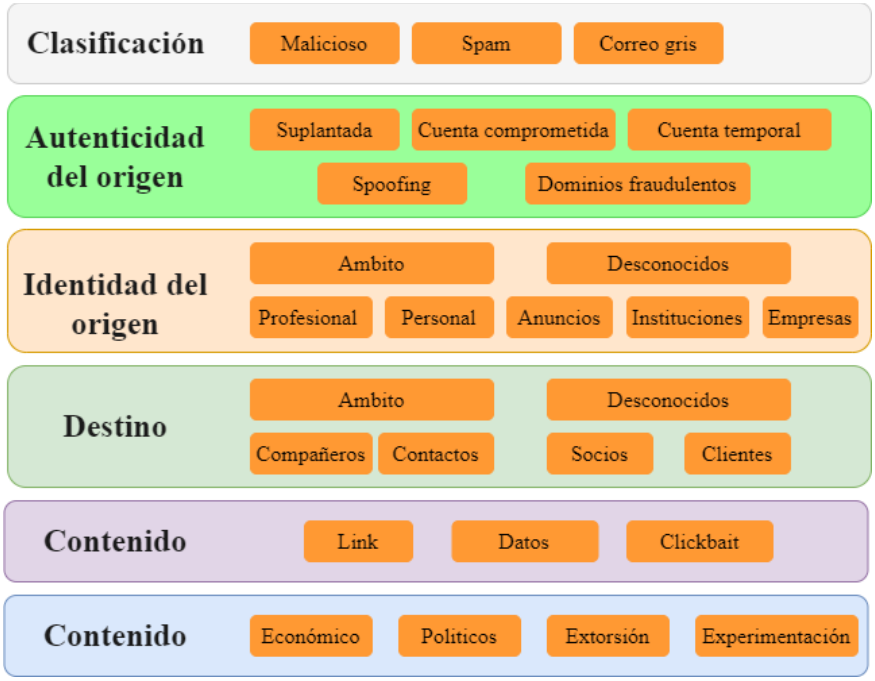


Figura 3-2: Aplicación de Taxonomía para detectar correos electrónicos maliciosos.

Fuente: (Rivas 2020).

Otras de las aplicaciones que se dan a la Taxonomía es el conocido como Kill Chain o en otras palabras cadena de exterminio, la cual fue definida en los años 90 para operaciones militares y que años más tarde fue implementada en la industria de la ciberseguridad, mediante este método se busca definir las fases por las que el atacante atravesó en búsqueda de cumplir su objetivo (Rivas 2020).

En los orígenes de esta metodología el objetivo de aplicar la Taxonomía en los ejercicios bélicos era identificar, atacar y destruir a los intrusos, pero años más tarde el concepto fue utilizado en investigaciones académicas y empresas relacionadas al área de las redes, para finalmente definirse como un método para modelar las intrusiones en una red informática. Gracias a este método no es necesario sufrir un ataque para poder estudiar nuestra red ya que el investigador es el encargado de analizar su propia red, para poder ejecutar este análisis es necesario realizar las siguientes etapas (Rivas 2020):

- Reconocimiento: Es el punto de partida en el cual debemos centrar nuestros objetivos ya que un estudio demasiado extenso crearía demasiadas variables para analizar y podría entorpecer el estudio, normalmente un atacante utiliza este tiempo para fijar vulnerabilidades y considerar los métodos de defensa que pueden estar implementados. También es el momento de recopilar información sobre la organización

- El siguiente paso es el conocido como militarización (weaponization) en el cual seleccionamos las herramientas que permiten explotar las vulnerabilidades anteriormente detectadas y que permitan llegar a los objetivos planteados.
- El tercer punto a tratar es el de entrega (delivery) mediante el cual se ejecuta la distribución de paquetes infectados, entre los más comunes se debe analizar puertos USB, correos electrónicos y sitios vulnerables.
- La etapa más crítica es la de explotación ya que dependiendo de la efectividad del ataque será posible cumplir esta acción, si los pasos previos fueron realizados correctamente el sistema o aplicación quedaran al descubierto debido a que el código malicioso fue ejecutado.
- Al ser posible tener contacto con el sistema victima el atacante necesariamente debe crear puertas traseras que le permitan acceder y mantener la comunicación mientras busca la información deseada. Dentro de la Taxonomía es definido como instalación.
- Una vez que el sistema fue completamente vulnerado se realiza la etapa de mando y control en el cual el atacante fortalece su estrategia y crea canales de comunicación mucho más estables.
- Finalmente se ejecutan las acciones sobre el objetivo pudiendo ser el filtrado de información, destrucción o el secuestro, entre lo más comunes.

Actualmente es el método más común aplicado por organizaciones para detectar sus propias vulnerabilidades y estar preparados en el caso de que se efectué un ataque real. Pero una de las falencias que se podría encontrar al aplicar la Taxonomía es que solo se puede realizar el estudio dentro del perímetro de nuestra red, dejando a un lado puntos vulnerables que dependen de otras organizaciones, por este motivo es necesario integrar áreas de análisis avanzados, proporcionados por empresas especializadas y que generan estadísticas recopiladas del internet (Rivas 2020).

2.2. Internet de las Cosas (IoT)

El internet de las Cosas (IoT) es la forma mediante la cual es posible conectar dispositivos cotidianos a internet, de forma que los elementos físicos puedan enviar sus datos a otros dispositivos con diversos objetivos. Los campos con los que se puede interactuar son innumerables ya que es posible conectar desde dispositivos de salud permitiendo llevar un control de signos vitales en tiempo real, pasando por utensilios de cocina lo que nos permitiría optimizar tiempo, y hasta conectando prendas de vestir o accesorios como por ejemplo zapatillas de deportistas con los cuales se puede generar estadísticas de rendimiento (redhat, 2019).

Para que un dispositivo se pueda considerar IoT es necesario que sea capaz de recibir y transferir datos a través de una red informática y con la intervención humana mínima. Esto ha sido posible en primer lugar gracias a la miniaturización de sistemas de cómputo, ya que de esta forma pueden

ser implementados en diversos dispositivos “cotidianos” sin alterar su forma y funcionamiento, y por otra parte debido a las increíbles velocidades con las que se cuenta actualmente para transferir y recibir información. Una de las principales aplicaciones y de las más comunes que hacen uso de dispositivos IoT son los hogares inteligentes los cuales funcionan enviando, recibiendo y analizando datos de forma permanente en un ciclo de retroalimentación (redhat, 2019).

2.2.1. Arquitectura de red con dispositivos IoT

Para que una red doméstica se considere IoT es necesario definir las partes que deben integrar esta. Los primeros componentes con los que se debe contar son los sensores y actuadores que son los encargados de supervisar o controlar, según sea el caso, alguna cosa o proceso físico, estos dispositivos son los primeros en captar datos de su entorno por ejemplo temperatura, humedad, composición química, etc.; así como tomar una acción rápida mediante un actuador. Por este motivo la capacidad de enviar y recibir datos es fundamental, la principal diferencia de un sensor normal es que los dispositivos IoT cuentan con un módulo de preprocesamiento, pudiendo ser propio del dispositivo o en el borde de la comunicación entre el dispositivo y el internet (Jahnke, 2020, p.1).

También es necesario contar con un sistema de adquisición de datos (DAS) que es el encargado de recoger todos los datos de los sensores y la estructura de forma que sean compatibles con los protocolos de internet ya que estos deben ser enviados mediante Wi-Fi o Ethernet hacia los dispositivos donde se realizara el procesamiento de los datos recopilados, además los DAS deben ser capaces de comprimir o filtrar la información de forma que adquiere un tamaño optimo y poder ser transmitido (Jahnke, 2020, p.15).

El dispositivo de borde en algunos casos es capaz de realizar el análisis y procesamiento de datos de forma que si existe un fallo con la comunicación de internet no se vea afectada la red y se pueda seguir trabajando. En los sistemas más avanzados se implementa métodos de aprendizaje automatizado permitiendo que la propia red se auto regule sin tener que esperar las ordenes desde un servidor en la nube. Otro de los aspectos que caracteriza a los dispositivos IoT es que todo el trabajo de procesamiento y almacenaje es transferido por medio de internet a sistemas mucho más potentes de forma que el usuario final puede optimizar sus recursos (Jahnke, 2020).

La arquitectura de red enfocada a dispositivos IoT no cuenta con una única definición y cada administrador configurara su red dependiendo de sus necesidades, sin embargo, se puede definir la estructura por capas para tener una mejor comprensión. Una de las primeras arquitecturas que se establecieron fue la arquitectura de 3 capas, la cual está dividida en (García, 2020):

- **Capa de percepción:** en la que se encuentran los sensores mediante los cuales el sistema identifica su alrededor recopilando información y convirtiéndola en pulsos eléctricos los cuales son más fáciles de transmitir, por ejemplo, los GPS, termómetros, tarjetas RFID, etc.
- **Capa de red:** se considera la capa central de la red IoT ya que comprende todos los métodos de transmisión para que los dispositivos puedan conectarse y comunicarse hacia la nube o hacia otros dispositivos.
- **Capa de aplicación:** es la encargada de presentar de forma gráfica la información dándole un sentido a la información para que el usuario pueda tener la percepción de que interactúa con un sistema inteligente.

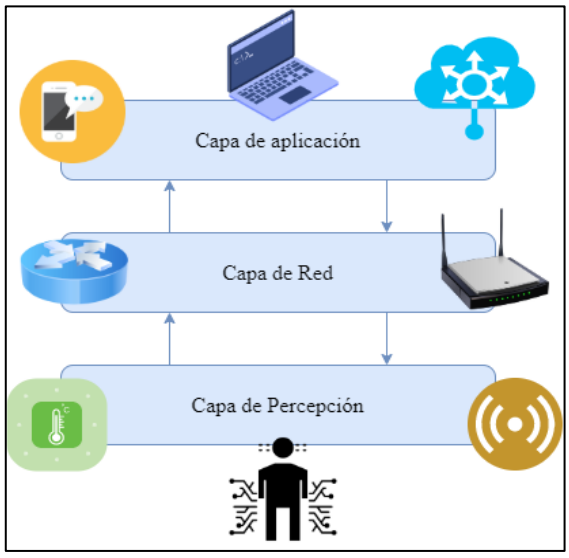


Figura 4-2: Arquitectura básica de una red IoT.

Fuente: (García, 2020).

Otra arquitectura, aunque complementaria de la anterior incluyendo las capas de transporte la cual integra los diferentes tipos de redes que pueden ser utilizados para transmitir y recibir la información por ejemplo 5G, Wif-Fi, Zigbee, NFC, etc. También se define una capa de procesamiento en la cual se encuentran los dispositivos de frontera encargados del procesamiento y análisis de la información que proviene de la capa de transporte. Por último se considera la capa de aplicación como una capa de negociación ya que agrupa no solo las aplicaciones usadas por el usuario sino también la plataforma de administración y desarrollo del sistema IoT (Garcia, 2020, p.15).

Por otra parte, la arquitectura planteada por Cisco denominada niebla, se basa en que la capa de procesamiento y monitoreo está mucho más cerca de la capa de percepción vista anteriormente, de esta forma se puede reducir la latencia, la seguridad y la eficiencia de la red. Para conseguir estas características hacen uso de nodos (nodos de niebla) los cuales son capaces de operar de

forma autónoma ya que las decisiones se toman de forma local, este planteamiento permite generar redes más homogéneas logrando llegar a múltiples ambientes, consiguiendo una configuración de capas jerárquicas para que dependiendo las necesidades puedan ser representadas como un todo, de esta forma es posible administrar y detectar fallos de forma más rápida, además de automatizar su mantenimiento y al estar constituidas por dispositivos con capacidades de procesamiento se puedan programar ciertas acciones.

De forma interna cada nodo debe tener la capacidad de manejar características como seguridad, almacenamiento, preprocesamiento y monitoreo, de esta forma se evita redundancia y desperdiciar recursos de la red (García, 2020, p.30) .

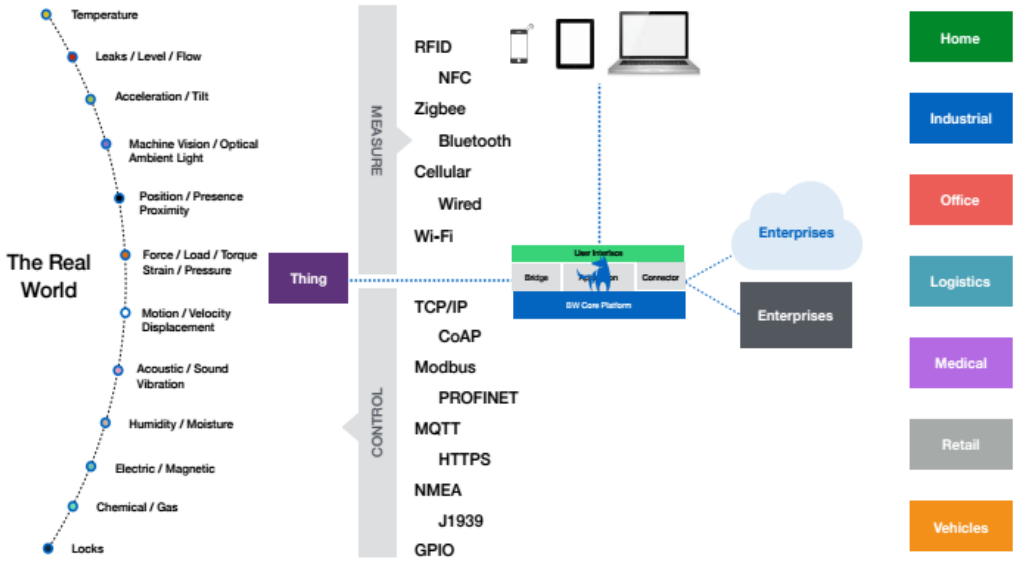


Figura 5-2: Protocolos presentes en la arquitectura de una red IoT

Fuente: (jecrespom; 2018).

2.3. Tráfico

2.3.1. Definiciones

En primer lugar es necesario dar a conocer la definición de trafico de red, también conocido como tráfico de datos o simplemente como tráfico, se define como los datos que se desplazan por una red en un momento determinado, los datos de la red se componen de paquetes, que son la unidad fundamental de los datos que se transmiten, al momento de ser transmitidos los datos, estos son divididos en paquetes por lo que al llegar al receptor deben ser rearmados en el orden correcto para que puedan ser legibles, por este motivo los paquetes tienen dos partes muy bien definidas,

la carga que son datos sin procesar y los encabezados o metadatos que contienen las direcciones IP de origen y destino. Existen 4 tipos de tráficos y son (Solarwinds, 2021):

- **Tráfico congestionado:** el cual se caracteriza por el alto consumo de ancho de banda
- Trafico promedio, hace referencia al ancho de banda consumido por el tráfico en un determinado tiempo de trabajo.
- **Tráfico interactivo:** ya que los datos compiten por consumir todo el ancho de banda disponible y si no está regido por prioridades puede generar tiempos de respuesta más lentos.
- **Tráfico sensible a la latencia:** ya que depende de las características del medio por el que se transmite y si no es monitoreado puede generar deficiencias por el alto consumo del ancho de banda.

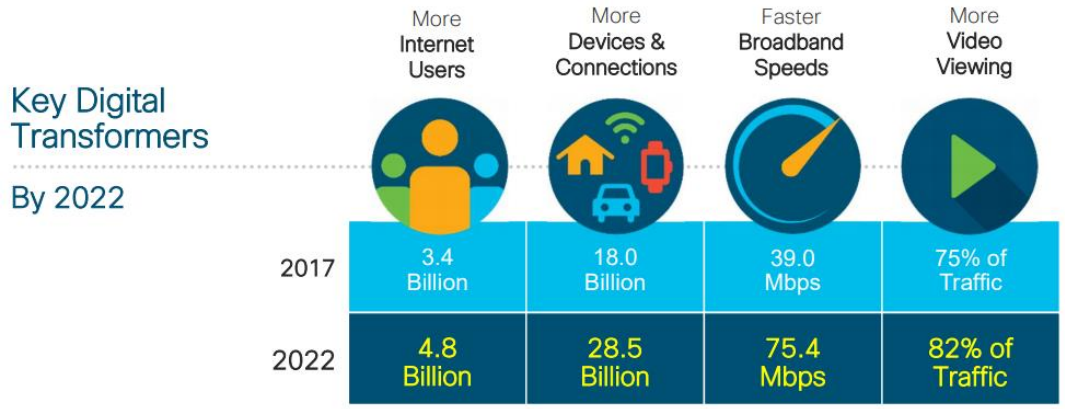


Figura 6-2: Tendencia de crecimiento de tráfico del internet desde 2017 hasta el 2022.

Fuente: (Cisco citado en honim.typepad, 2019)

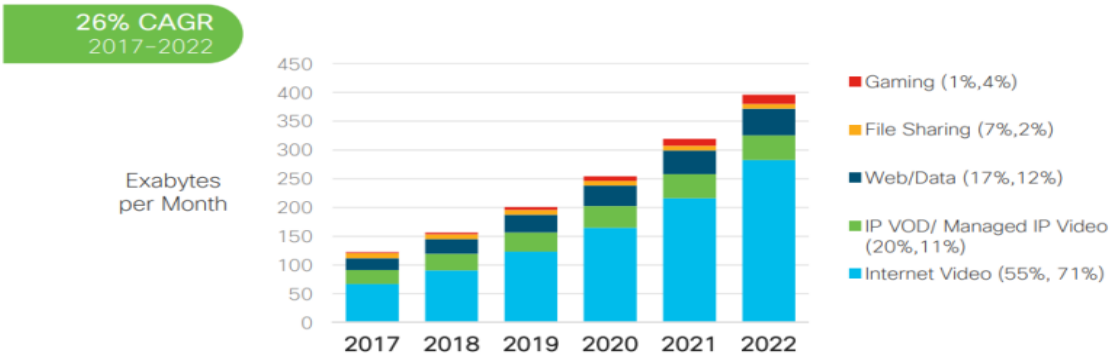


Gráfico 1-2: Tráfico global según el tipo de servicio.

Fuente: (Cisco citado en honim.typepad; 2019).

2.3.2. Tráfico malicioso

El tráfico malicioso puede ser reconocido mediante diferentes características al realizar un análisis, mediante las herramientas que se encuentran disponibles para esta actividad. Entre los

aspectos a tener en cuenta en un análisis de tráfico se tiene por ejemplo a la fuente desde donde se origina el tráfico, esto debido a que puede ser que se haya capturado el servicio y se encuentre utilizando todos los recursos para su beneficio, también puede existir una inundación de puertos en búsqueda de respuestas, cabe señalar que este comportamiento puede deberse a Broadcast o Multicast y es aquí cuando los patrones de tráfico son necesarios y se requiere encontrar anomalías, Además se puede dar el caso de flujos de información excesivos ya sea que contenga información legitima o solo paquetes jumbo los cuales se encargan de desperdiciar anchos de banda, creando latencia en la red (Tamayo Ottati 2020).

- **Anomalía: l**a definición de anomalía bajo un punto de vista informático es la presencia de patrones irregulares dentro de un historial homogéneo, los puntos que presenten características irregulares en comparación al resto permiten extraer información con la cual determinar los motivos de estas anomalías (ibiblio, 2019).
- **Intrusión:** se considerado al incidente en el cual la seguridad se encuentra vulnerada o en riesgo por parte de un atacante que intenta evitar los protocolos de autorización (Tamayo Ottati 2020, p.10).

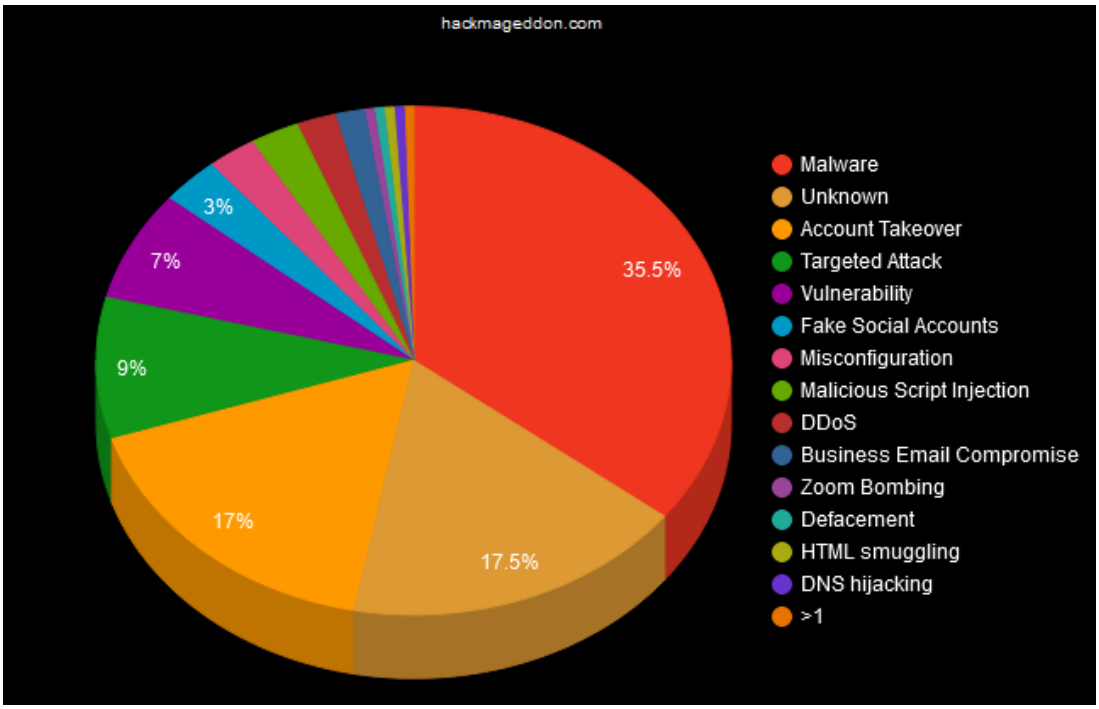


Gráfico 2-2: Ciberataques detectados en el tráfico de internet.

Fuente: (hackmageddon; 2021).

2.4. Firewall

Un Firewall se define como un dispositivo que se implementa en una red con el fin monitorear y filtrar el tráfico que circula dentro de la red, para este fin utiliza políticas también conocidas como

reglas de seguridad, que son definidas bajo un estudio y necesidades de la red, frecuentemente separa la red interna con las conexiones públicas o internet, de forma que mantenga alejadas las amenazas y el tráfico peligroso (checkpoint, 2021,p.2).

Desde sus inicios los firewalls han sido utilizados para el filtrado de paquetes en la capa 4 del modelo OSI (capa de Transporte), desgraciadamente esto deja la puerta abierta para que ataques camuflen software malicioso en las capas superiores por ejemplo malware. Por este motivo se fue necesario desarrollar Firewall a nivel de aplicación o también conocidos como Proxy, con los cuales se puede hacer análisis más profundos pero en contrapartida si no se configuran correctamente generan cuellos de botella (Hillstone, 2021, p.4).

Para que los Firewalls consigan una protección mucho más intuitiva actualmente el desarrollo se enfoca en que los sistemas de próxima generación, Next-Generation Firewall (NGFW), los cuales además de cumplir con las acciones básicas de un Firewall se especializan en la búsqueda de paquetes malformados, protocolos anormales o cualquier comportamiento extraño presente en el tráfico de la red, así como también balanceo de carga, detección de Botnet, Spam, Antivirus, entre otros servicios; todo esto gracias a la integración de inteligencia artificial y procesamiento en la nube y análisis de Big Data (Hillstone, 2021, p.1).

2.4.1. Clasificación de Firewall

La clasifican los Firewalls dependen del criterio del autor, por lo que no existe una sola clasificación, pero frecuentemente se dividen según la aplicación que tengan; existiendo los siguientes grupos (Alonso, 2020, p.10):

- **Firewall de filtrado de paquetes:** es el Firewall más convencional y más antiguo ya que determina el paso o descarte de los paquetes dependiendo de su configuración y de los parámetros de control que se encuentren dentro del sistema. Este tipo de sistemas comprueban la información existente como la IP de destino, la IP de origen, el tipo de paquete, número de puerto, etc., pero sin abrir el paquete y por lo tanto sin inspeccionar el contenido. Después de esta operación los paquetes que no cumplan con los criterios de red serán descartados del enrutador. La principal ventaja de estos sistemas es su bajo consumo de recursos y su baja complejidad.

- **Firewall de pasarela de nivel de circuito:** utiliza un sistema simple de análisis de las puertas de enlace de los circuitos de comunicación, en otras palabras, verifica los protocolos TCP con los que se realiza la transmisión, de forma que se garantice la legitimidad de la sesión. Por este motivo

el punto débil de esta clase de Firewalls se encuentra en que permiten el paso de software malicioso ya que solo comprueba que se utilice el protocolo correcto para la comunicación.

- **Firewall de inspección con estado:** se define como la combinación de los casos anteriores en donde se analiza la estructura de los paquetes y los protocolos de comunicaciones, de esta forma se puede elevar el nivel de seguridad en comparación con los anteriores, pero esto implica un mayor consumo de recursos además de generar latencia dentro de la red.

- **Firewall Proxy:** este tipo de Firewall trabajan en la capa de aplicación filtrando el tráfico que atraviesa la red desde su propio nodo lo que implica que el firewall debe ser direccionado al Firewall antes de que los paquetes se distribuyan en la red, muchos de estos sistemas vienen integrados en aplicaciones y son capaces de utilizar servicios de la nube o trabajar a distancia desde un servidor dedicado para este fin. En estos Firewall podemos encontrar capacidades de análisis profundos siendo capaces de identificar malware. Una característica resaltable es que al utilizar un servidor independiente para analizar el tráfico brinda anonimato al cliente final ya que el destino de todo el tráfico que se dirija al cliente debe en primer lugar ser dirigido hacia la aplicación o servidor proxy, en contrapartida puede causar una disminución en la velocidad de descarga y subida de datos dentro de la red

- **Firewall de Próxima Generación (NGFW):** no existe un punto de partida definido para considerar el inicio de los Firewalls de próxima generación, pero en teoría lo que se intenta conseguir con las nuevas propuestas es que los Firewalls sean mucho más intuitivos y que su capacidad de análisis sea mucho más profunda sin perder las características de sus predecesores. También se busca integrar otros servicios de seguridad de forma que el alcance cubra muchas más capas de red, lo que incrementara la seguridad dentro de la red. Actualmente todos los esfuerzos se encuentran centrados en desarrollar sistemas que tomen encuentra el comportamiento de la red para que se pueda prevenir los ataque antes de que ocurran

Otra clasificación que se puede mencionar separa a los Firewalls en tres grupos, y a pesar de que el criterio para agruparlos es superficial facilita definir qué tipo de sistema se va a implementar en una solución de seguridad informática. Estos grupos son (Alonso, 2020, p.14):

- **Firewall de Software:** los Firewalls de software son herramientas que podemos instalarlo dentro de un dispositivo y brinda protección a este independientemente del estado del resto de la red. En contrapartida el Firewall consumirá los recursos existentes en el dispositivo, además de que el software debe ser compatible con su arquitectura.

- **Firewall de Hardware: s**on los equipos dedicados, generalmente con características de enrutador ya que todo el tráfico de una red deberá atravesar por el dispositivo antes de que llegue a cualquier cliente, en su mayoría son utilizados para brindar seguridad perimetral, esto significa que separa a la red privada de una organización, del internet o cualquier dispositivo que no pertenezca a la red y que intente establecer una conexión hacia los recursos internos de la organización. Evidentemente su mayor falencia se encuentra en su compatibilidad con otros sistemas de seguridad, viéndose limitado por las características de su arquitectura.

- **Firewall en la nube:** cada vez es más frecuente hacer uso del procesamiento que brinda la nube y en el caso de firewall también se aprovecha este servicio mediante el uso de servidores Firewall as a Service (FaaS), el funcionamiento es muy similar al de un proxy ya que el tráfico previamente analizado por el Firewall es el que se envía hacia los clientes. Evidentemente su principal beneficio es el de poder escalar sistemas y economizar en hardware.

2.4.2. Características de un Firewall

Los Firewalls se utilizan para separar redes privadas de las públicas impidiendo que la información y características de la red sean visibles fuera de la zona de protección, además pueden configurarse varios Firewalls en diferentes niveles de la red y cada un impedir el tráfico que no cumple cierto criterio de protección (Briceño V., 2020, p.12).

Para que un Firewall pueda realizar su trabajo debe contar con reglas o políticas que son los criterios mediante los cuales el sistema determina si permite o niega el paso al contenido del tráfico que analiza. De forma general existen dos tipos de reglas como mínimo en un Firewall, las restrictivas, que impiden el paso a todo lo que no esté explícitamente definido y por otra parte las reglas permisivas, las cuales no restringirán el paso al tráfico que no se encuentre explícitamente impedido, por este motivo es fundamental evitar ambigüedades ya que precisamente serán estos puntos débiles los que harán uso los atacantes para vulnerar un sistema de protección con firewall. Bajo esta consideración es mucho más recomendable aplicar reglas restrictivas ya que definen exactamente todo lo que no está permitido y que es conocido, mientras que aplicar reglas permisibles puede permitir el paso a paquetes que no se hayan contemplado (infotecs, 2019, p.6).

De forma general y más conocidos son los firewalls que vienen por defecto en los sistemas Windows y Linux, los cuales trabajan de forma diferente ya que administran las conexiones de aplicaciones y dispositivos de forma diferente. Por una parte, esta Windows el cual forma parte del sistema como un programa más del usuario, lo que permite que otros programas se conecten directamente con él y comuniquen las necesidades de conexión para que se encuentren disponibles

cuando las necesiten. Además, evidentemente son más amigables para el usuario, y pueden ser administrados sin tener conocimientos avanzados. Por otra parte, los sistemas GNU/Linux se consideran más avanzados ya que con los conocimientos suficientes se puede modificar aspectos como los ciclos de envío de los paquetes ICMP de forma que los atacantes no puedan reconocer la red (GuidesMania, 2021, p.37).

Cabe señalar que los dos sistemas hacen uso de las tablas de enrutamiento para ejecutar el filtrado, pero mientras Windows lo administra como un programa, en GNU/Linux está inherente en el kernel del sistema. Específicamente en GNU/Linux se hace uso del servicio IPTABLES el cual es una aplicación de línea de comandos que puede ser configurado, a pesar de que cada sistema trae sus propias reglas de forma predeterminada, mediante IPTABLES se crean tablas compuestas las cuales se encargan de comparar los paquetes con las condiciones internas de cada regla, al analizar un paquete, este es etiquetado (TARGET) y se busca una coincidencia que conecta con las acciones disponibles por el Firewall como son ACCEPT, DROP, RETURN. IPTABLES puede analizar el tráfico de tres formas diferentes las cuales son el tráfico entrante (INPUT), el tráfico que atraviesa el dispositivo (FORWARD), y el tráfico saliente (OUTPUT) (Diana C., 2021, p.12).

2.4.3. Ventajas y desventajas

Las principales ventajas que nos brinda un Firewall es que podemos blindar nuestra red antes de que nos veamos afectados por un ataque informático y cuando seamos víctima de uno tener la capacidad de reaccionar reduciendo los efectos negativos que nos pueda causar, por otra parte permite separar redes y que cada una tenga sus propias reglas de administración, al nivel de que pueden ser invisibles unas con otras, otra ventaja resaltable es su adaptabilidad, pudiendo ser configurados solo para un dispositivo o para cubrir todo un grupo de dispositivos.

Entre sus desventajas hay que señalar independiente del tipo de Firewall que se utilice es la degradación en el ancho de banda o latencia que puede presentar la red cuando se implementa un Firewall. Por otra parte, se ven seriamente afectados en los casos de ataques de día cero ya que como se describió previamente los Firewalls necesariamente deben tener definidos los parámetros de análisis ya sea registrados en bases de datos locales o en la nube, además no se puede delegar toda la carga de protección al Firewall debido a que la sobrecarga de tareas puede llevar a la generación de falsos positivos.

2.4.4. Características de dispositivos disponibles en el mercado

Actualmente se ofrecen soluciones de protección por parte de distintas empresas expertas a nivel mundial, de las cuales a continuación se resaltará las enfocadas a soluciones de redes domésticas y empresas pequeñas, sin necesariamente estar enfocadas a redes IoT.

Tabla 1-2: Características de dispositivos Firewall.

Características / Modelos	Área de implementación	Tecnologías	Servicios preconfigurados	Servicios adicionales	Inconvenientes conocidos	Precios
Ubiquiti Unifi Security Gateway (USG)		3 puertos Gb -Velocidad de 3Gbps para paquetes 512bytes	-Inspección profunda de paquetes. -IDS. IPS.	-Enrutador WiFi. -Gestión vía CLI. -Monitoreo de dispositivos. -Configuración de VLANs. QoS	-Requiere conocimientos avanzados	$129,00
Firewalla	-Domestico	-WiFi 2.4Ghz. -Velocidad entre 100Mbps y 500Mbps.	-IDS. -IPS. -Antivirus. -Antimalware. Antiphishing. -OpenVPN.	-Control parental. -Administración via aplicación móvil.	-El servicio IPS solo permite velocidad máxima de 100Mbps. -Puede no ser compatible con todos los routers.	$108,00
Bitdefender Box 2	-Domestico	-WiFi 2.4Ghz y 5Ghz. -2 Puerto Gb -Velocidad de 1Gbps	-Antitracker. -Navegacion Segura Safe Online Banking. -Antivirus	-Enrutador WiFi. -Total Security unlimited para IoT. -Control parental. -Monitoreo de dispositivos IoT basado en la nube.	-Suscripción para servicios avanzados -Licencia del antivirus por suscripción	$171,00
Cortafuegos Inteligente CUJO	-Domestico	-WiFi 2.4Ghz -Velocidad de 1Gbps	-Protección contra acceso remote. -Antimalware. -Antiphishing. -Antivirus. -Antibotnet	-Enrutador WiFi. -Uso de algoritmos IA. -Control parental. -Configuración asistida.	-Solo puede ser administrado por aplicación movil	$94,95

Cortafuegos VPN de próxima generación Zyxel	-Domestico -Pequeña empresa	-1 puerto Gb para WAN. -4puertos Gb para LAN. -1 puerto SFP Gb para enlace de fibra óptica. -Velocidad de Firewall entre 200Mps y 350Mbps.	-VPN. -Sistema unificado de amenazas (UTM).	-Soporte IPv6. -Failover multi-WAN. -Inspección de contenido web. -Control de aplicaciones. -Antivirus. IPS -Servicios OneSecurity	-Licencia de suscripción anual	$179,97
Cortafuegos SinicWall TZ400	Pequeña empresa.	Velocidad entre 900Mbps y 1.3Gbps. -Soporte de 100 puertos adicionales.	-Sistema unificado de amenazas (UTM) -Antivirus -Monitoreo de red -VPN.	-Inspección profunda de paquetes de tráfico de internet. -Conexión móvil SSL	-Suscripción de licencia para los servicios de protección.	$845,71
FortiGate 30E	Pequeña empresa.	-4 puertos de 1Gbps. -Velocidad de Firewall entre 150Mbps y 950Mbps.	-Sistema unificado de amenazas (UTM). -IPS. -Filtrado web. -VPN.	-Segmentación de la red mediante VDOM (Firewalls independientes).	-Suscripción continua.	$274,54
Cisco Meraki MX64W	Pequeña empresa.	-WiFi 2.4Ghz. -Velocidad de 1.2Gbps. -Velocidad de Firewall 250Mbps. -Máximo 50 usuarios.	-Filtrado avanzado de contenido -Cisco Threat Grid -Protección avanzada contra malware	-Enrutador WiFi -Capacidad de gestión desde la nube -Segmentación de red WiFi (Máximo 4 SSID)	-Licencia de 1 año.	$743,87

Protectli Firewall 4 puertos Gigabit	-Doméstico. -Pequeña empresa.	-Memoria RAM de 4GB. -Disco SSD de 32GB Velocidad de 1Gbps.	-servicios básicos de Firewall.	-Enrutador LAN/WAN	-Requiere conocimientos avanzados. -Requiere instalación de los servicios de protección.	$309,00
WatchGuard Firebox T15	-Pequeña empresa.	-WiFi 2.4Ghz y 5Ghz. -Velocidad entre 90Mbps y 400Mbps.	-VPN. -IDS. -Antivirus. -Antiransonware -Prevención de perdida de datos.	-Enrutador WiFi. -Access Point.	-Soporte de un año. -Máximo 5 usuarios.	$331,98 Soporte ilimitado por 90 días.

Fuente: (tpempresas, 2019)

Realizado por: Álvarez**,** Andrés, 2022.

Como se puede apreciar en la tabla, existe una gran variedad de dispositivos para proteger una red y se pueden incluir diversos sistemas de protección para reducir los ataques que pueda sufrir la red, pero los precios no son bajos y en algunos casos requieren una constante renovación del servicio. En contrapartida con el sistema propuesto en este proyecto los únicos costos a cubrir son los del hardware como son las tarjetas Arduino, Rasbperry Pi, dispositivos IoT, cables de conexión, entre otros, mientras que no existe un costo por parte del software como son el sistema operativo, herramienta IPS, software de monitoreo y de pruebas, ya que son Open Source. Cabe señalar que se puede pagar suscripciones para acceder a repositorios con reglas preconfiguradas. El principal inconveniente en este caso es el de configurar todo el sistema y adecuarlo a las necesidades de la red.

2.5. Sistema de detección de intrusos (IDS)

Una intrusión para un sistema no solo se considera cuando existe un acceso no autorizado sino a todo tipo de actividad que comprometa el comportamiento del mismo, así como su integridad, confidencialidad o disponibilidad. Al ser este tipo de ejercicios un problema evidente para la seguridad informática se requiere de dispositivos y sistemas especializados en esta área tales como los Sistemas de Detección de Intrusos (IDS). Este grupo teóricamente integra todo hardware o software que vigila dentro de su red actividades intrusivas, pero de forma partica esta denominación solo se asigna a sistemas que reaccionan de forma automática ante un evento intrusivo. Estos sistemas tuvieron sus primeras apariciones en los años 80s y desde entonces se han ido perfeccionando al punto de que hoy en día todo sistema de seguridad intenta detectar ataques antes de que se lleven a cabo (Gutiérrez y Guerrero 2019, p. 18).

2.5.1. Clasificación de IDS

Los IDS se encuentran clasificados en dos grandes grupos, los sistemas que enfocan su análisis a todo un dominio de red y los que se centran en un solo host, de forma que permite ajustar los esfuerzos de acuerdo al sistema que se vigila y como debe realizarlo el sistema (Gutiérrez y Guerrero 2019, p. 18):

- **IDS basados en red:** los sistemas IDS basados en red (NIDS) trabajan con el tráfico generado por los paquetes ya sean entrantes o salientes, los cuales dependiendo de sus características alertan al sistema para que a su vez sean monitorizados. El sistema debe ser instalado en el equipo que tenga acceso a toda la red que se desea analizar, generalmente HUBS, puntos de acceso o equipos con capacidades de enrutamiento, de forma que el monitoreo abarque todos los dispositivos existentes en la red. Los IDS pueden ser ajustados para que realicen un

monitoreo a toda la red o parte de ella en el caso de existir VLANS, siempre y cuando las interfaces del equipo en donde se encuentra el IDS estén configuradas en modo promiscuo en relación a la red de estudio (Gutiérrez y Guerrero, 2019, p. 19)

Al realizar esta configuración el sistema se encargará de analizar todas las tramas mediante la fragmentación de los campos en búsqueda de bits modificados o incoherentes, poniendo mayor énfasis en las direcciones IP de origen y destino, puertos de origen y destino, FLAGS TCP y campos de datos (Gutiérrez y Guerrero, 2019, p. 19).

- **IDS basados en host:** conocidos como HIDS tiene su principal diferencia con los anteriores en que se enfocan en proteger un único sistema, tal como lo haría un firewall o antivirus instalado en un host, esto implica que el sistema IDS se ejecutara dentro del mismo equipo en segundo plano (background) analizando el comportamiento del equipo frente a la red (Gutiérrez y Guerrero, 2019, p. 19).

- **IDS basados en conocimiento:** hacen uso de bases de datos clasificados en perfiles de vulnerabilidades, que han sido recopilados mediante el estudio de ataques conocidos, por este motivo requieren de una constante actualización y en la mayoría de casos trabajan bajo suscripciones ya que requieren de algún método que provee las bases de datos y de aquí deriva su principal falencia al momento de implementarlos (infotecs, 2019, p.1).

- **IDS basados en comportamiento:** trabajan siguiendo una línea de rendimiento establecida previamente, lo que se cataloga como un comportamiento normal de la red y cuando surgen actividades que salen de los patrones establecidos el sistema los identificara como intrusiones, activando todos los avisos que estén programados. En este caso es necesario realizar un análisis previo de la red para definir de la forma más precisa lo que se entiende dentro de la red como un comportamiento normal poque de otra forma se generaran falsos positivos, por este motivo al implementar este método y la mayoría de casos es necesario investigar los patrones que activaron las alarmas del sistema (infotecs, 2019, p.1)

Cabe señalar que dentro de estos grupos algunos autores como es el caso de infotecs incluyen los grupos en donde los IDS al detectar una actividad ilícita activan más acciones como bloqueo de tráfico o desactivación de puertos, pero cabe señalar que este tipo de acciones no pertenecen de forma nativa a un IDS ya que para este tipo de acciones existen los IPS, sin embargo este tipo de sistemas son determinados IDS activos y en el caso de que solo generen alertas hacia el administrador de red se llaman IDS pasivos.

2.5.2. Características del IDS

Dentro de los IDS basados en host se debe tener muy en cuenta que es necesario cumplir ciertos parámetros como los SIV (verificador de integridad del sistema) los cuales se encargan de comprobar la integridad de los archivos existentes dentro del host en busca de modificaciones no autorizadas por ejemplo los backdoors que se crean al modificar los scripts de red permitiendo el acceso de usuarios y comandos no autorizados. Por este motivo la responsabilidad del usuario incrementa, ya que podría ser el causante de un ataque no registrado por el sistema IDS, debido a esto, sistemas como Solaris y ASET (Automated Security Enhacement Tools) traen servicios pre configurados y requieren de conocimientos previos para ser modificados (Gutiérrez y Guerrero, 2019, p. 19).

Otro de los sistemas con que cuentan los IDS basados en host son los LFM (Monitor de Registro) y Sistemas de Decepción. Los primeros son pequeños programas internos o demonios que se encargan de analizar los logs generados por el sistema al presentarse una actividad anormal, por ejemplo, cuando un host desconocido envía peticiones mediante el protocolo ICMP y esta actividad es registrada, en este caso los IDS buscan definir los patrones que se están empleando para intentar vulnerar el sistema y generar las alertas.

Por otra parte, los Sistemas de Decepción que también son llamados HONEYPOTS se encargan de simular servicios atractivos para un atacante a manera de anzuelo para que los ataques sean dirigidos específicamente a estos servicios, esto ayuda al sistema a registrar sus actividades y tener un mejor conocimiento del tipo de ataque que se intenta desplegar; estos mecanismos generan una protección extra ya que el atacante podría emplear mucho tiempo al intentar vulnerar el sistema (Gutiérrez y Guerrero, 2019, p. 23).

2.5.3. Ventajas y desventajas

El motivo de que sea necesario implementar un sistema IDS es que permite un monitoreo en tiempo real de toda la red, permitiéndonos recopilar datos, determinar alteraciones en la red, detectar de forma preventiva acciones ilícitas y generar estadísticas para tener una mayor percepción del comportamiento de la red (incibe, 2020, p.5).

Por otra parte, y como se pudo determinar anteriormente, los sistemas IDS presentan muchas falencias al generar falsos positivos ya que al ser solo sistemas de detección es necesario analizar cada anomalía y tomar los correctivos necesario para que el sistema trabaje de la forma más óptima posible (incibe, 2020, p.5).

2.6. Sistema de Prevención de intrusos (IPS)

Los Sistemas de Prevención de Intrusos IPS son sistemas que agrupan un conjunto de acciones predefinidas capaces de proteger el sistema de forma proactiva y eficaz, ya sea identificando acciones conocidas o desconocidas de los atacantes (Gutiérrez y Guerrero 2019, p. 25).

El mecanismo fundamental de un IPS es la comparación entre firmas conocidas y firmas sospechosas. Una firma para un IPS es el script que contiene las reglas que definen el patrón de una actividad intrusiva, mientras más específicas sean las reglas mejor se podrá determinar si una actividad es intrusiva y por lo tanto mejor actuará el sistema, cada software tiene su propio método para adquirir las firmas, pero en la mayoría de los casos estos pueden ser modificadas y adecuadas para el sistema.(e-manuales, 2021).

2.6.1. Clasificación de IPS

La principal forma de clasificar a los sistemas IPS es mediante el método de detección, dentro de este grupo se puede definir los siguientes (incibe, 2020, p.5):

- **IPS basados en firmas o reglas:** se define de esta forma debido al uso de base de datos del sistema, en el cual se definen los patrones de los ataques, esta información es constantemente actualizada de forma que el sistema realice una constante búsqueda de coincidencias con las acciones que se realizan en la red.

- **IPS basados en anomalías:** el método utilizado para este caso es el de perfiles, el cual se centra en buscar actividades que alteren los parámetros que previamente se han definido como un comportamiento normal dentro del sistema, pudiendo ser causados por un dispositivo o por el tráfico de la red. Este tipo de análisis requiere más consumo de recursos ya que requiere un procesamiento estadístico interno.

- **IPS basados en políticas:** en este caso declarar de forma explícita las políticas que se desea emplear en el sistema de seguridad, de forma que será necesario introducir en la regla todos los parámetros necesarios para que el sistema funcione, muy similar a lo que se hace en un firewall.

- **IPS basados en detección por Honeypot:** para implementar este modelo es necesario configurar previamente un equipo de señuelo para que atraiga a los atacantes y desplieguen sus mecanismos de forma que se cree un registro de todas las acciones realizadas, para posteriormente crear políticas o que el sistema busque directamente patrones de vulnerabilidad.

Por otra parte, tenemos la clasificación según la tecnología implementada; se definen los siguientes sistemas (infotecs, 2019, p.19):

- **IPS basados en host: s**imilar al caso de los IDS, la función principal es monitorear un dispositivo específico en busca de vulneraciones del sistema. Entre los puntos de interés se encuentra el tráfico de red entrante y saliente, registros del sistema, privilegios de usuarios, ejecución de procesos y modificación de archivos. La aplicación más frecuente que tienen esta clase de sistemas es en el campo de servidores y diversos equipos que requieren mantener su servicio siempre activo (infotecs, 2019, p.19).

- **IPS basados en red:** el enfoque que se da para estos sistemas es el análisis de tráfico fundamentalmente, de forma que analiza todos los componentes del tráfico por ejemplo los protocolos, cabeceras, FLAGS, etc. La búsqueda de actividades sospechosas se debe realizar en tiempo real lo que implica que se debe ejecutar en un dispositivo robusto ya que en la mayoría de casos trabaja en paralelo con herramientas que llevan valores estadísticos de la red o también con otros métodos de seguridad como por ejemplo un firewall (incibe, 2020, p.5).

2.6.2. Características del IPS

Los sistemas IPS tienen una gran capacidad de análisis pudiendo manejar desde paquetes de información completos hasta secciones de datos muy mínimas como un bit, que de otra forma no serían detectados por los sistemas de seguridad. Cabe señalar que la precisión para identificar ataques dependerá de cómo se encuentren estructuradas las instrucciones y a su vez si estas generan congestión dentro de la red, ya que el sistema debe clasificar, inspeccionar y filtrar todo el tráfico antes de que este lo permita salir o entrar (infotecs, 2019, p.12).

Para que el trabajo sea más proactivo sin afectar el comportamiento de la red, los sistemas IPS asignan un filtro que contiene un conjunto de reglas definidas las cuales se dedicaran a controlar el tráfico, segmentando y extrayendo la porción de datos que se desea analizar, mediante estas técnicas se consigue que el tráfico sea clasificado y asignado a los filtros correspondientes consiguiendo unos resultados más consistentes. El uso de filtros individuales trabajando de forma simultánea se denomina procesamiento masivo de paquetes en paralelo, generalmente utilizado en IPS más avanzados, esta clase de procesamiento es más frecuente en hardware dedicado ya que de otra forma se verá afectado el rendimiento (infotecs, 2019).

Las nuevas tendencias de IPS se encuentran evolucionando a sistemas mucho más robustos que sean capases de mantenerse siempre en línea sin que degraden de alguna forma la red, además de que se requiere por parte del sistema una “conciencia de aplicaciones” con el fin de que puedan

identificar las aplicaciones y las políticas propias de cada una de ellas. También se requiere de una "conciencia de contexto" de forma que las decisiones que se tome en el sistema se fundamenten en la búsqueda del mejor beneficio para la red y que actúen dependiendo de las circunstancias que rodean al ataque, de ser posible que reaccionen con mayor rapidez a los ataques más inminentes y perjudiciales (infotecs, 2019, p.12).

2.6.3. Ventajas y desventajas

Entre las principales ventajas de implementar un IPS como método de protección es la forma de administrar los dispositivos conectados pudiendo crear subprocesos dedicados. También este tipo de sistemas permiten automatizar las acciones ante movimientos anómalos en la red mediante las reglas que se hayan definido; lo que implica una mayor facilidad de configuración y administración, siendo la única limitante, la capacidad del software que se esté utilizando. Los ataques con los que tiene una mayor eficiencia son los ataques de fuerza bruta, infecciones por malware, intrusiones, modificación de archivos, por este motivo es que se recomienda ser implementado en sistemas de monitoreo de red (incibe, 2020).

Entre la principal desventaja que impide un despliegue completo de los sistemas IPS es que cuando no se encuentra correctamente configurado y se generan una gran cantidad de falsos positivos el rendimiento disminuye y casos reales de instrucción se puede ignorar (incibe 2020a).

2.7. Software

2.7.1. Herramienta Snort

La herramienta Snort engloba todo un sistema IPS (Sistema de Prevención de Intrusos), el cual es de código abierto y a esto se debe su importancia a nivel mundial. Snort trabaja mediante reglas, las cuales brindan toda la información necesaria para que el sistema detecte actividades maliciosas; mediante estas reglas se crean patrones de vigilancia de forma que el sistema analice los paquetes que atraviesan la red y genera alertas o tome acción con el fin de proteger la red. (Snort, 2021).

Según la página oficial de Snort la herramienta está enfocada para tres usos principales; el primero como rastreador de paquetes tal como lo haría un Sniffer por ejemplo Tcpdump, el segundo uso posible es como registrador de paquetes con el cual es posible depurar el tráfico que circula en la red por ejemplo almacenando logs para un análisis, y el tercer uso mucho más avanzado es el modo IDS/IPS (Sistemas de Detección o Prevención de Intrusos), que como se ha mencionado

anteriormente intenta predecir acciones maliciosas antes de que ocurran y de esta forma generar alertas, tomar acciones y reducir el impacto de un ataque informático.

2.7.1.1. Características

La principal característica con la que se cuenta al momento de contrarrestar un ataque informático es la capacidad de crear nuestras propias reglas frente a comportamientos anormales, de forma que se pueda adaptar a las vulnerabilidades que descubramos en puntos críticos. También es posible programar reglas que nos permitan guardar determinados Logs dependiendo de las condiciones que se requiera o directamente descartar paquetes que no sean deseados. tal como lo haría un Firewall. Cabe señalar que Snort provee un conjunto de reglas predefinidas dentro del directorio Rules, las cuales podemos usar de ejemplo para crear reglas, modificarlas y experimentar. (Ortega, 2017, p.42).

Las principales características de Snort son las siguientes (Marina, 2021, p.12):

- Monitor de tráfico en tiempo real.
- Registro de paquetes.
- Análisis de protocolos.
- Coincidencias de contenido.
- Huellas digitales del sistema operativo.
- Facilidad de instalación en cualquier entorno de red.
- Crear registros.
- Código abierto.
- Baja complejidad para la implementación de reglas.

2.7.1.2. Arquitectura

En cada uno de los modos de trabajo es necesario que Snort cuente con las etapas y componentes necesarios para poder trabajar, por este motivo Snort está diseñado de forma modular y cada sección debe encargarse de una función de forma independiente, lo cual permite un mayor dinamismo al entrar en operación. La arquitectura de Snort no tiene un modelo definido y dependiendo del texto que se utilice como guía podría presentar una estructura diferente ya que las etapas se pueden combinar y ser analizadas como un único modulo; por ejemplo, dividirlo en Capturador de Paquetes, Decodificador de Paquetes, Preprocesador (todos los preprocesadores con los que cuente el sistema), Detector (reglas), Salida. (Ghafir, Prenosil & Hammoudeh citado en Zambrano y Guailacela 2019, p. 6). Pero independientemente del número de módulos en que

se divida la arquitectura siempre deberá contar con las siguientes etapas (Llopis Polvoreda 2017, p. 20-21):

- Interfaz de red.
- Método de captura.
- Filtro BPF.
- Preprocesadoras.
- Motor de reglas.
- Filtrado de eventos.
- Salida.

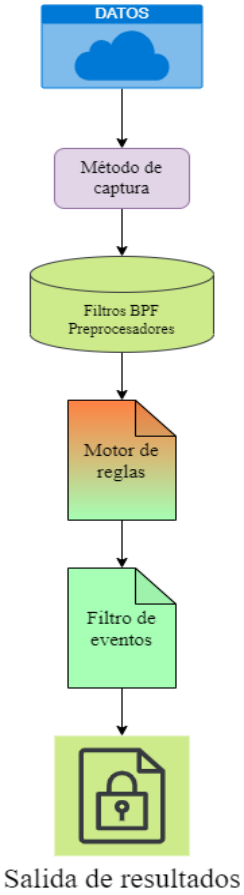


Figura 7-2: Etapas presentes en el análisis de paquetes la herramienta Snort.

Realizado por: Álvarez; Andrés, 2022.

Todas las etapas se caracterizan por cumplir una función específica, en el caso de la interfaz de red establece que puertos deberán ser analizadas, que métodos se utilizaran para el análisis (DAQ, AF-PACKET, PF_RING, NFQ, otros), realizados estos pasos se utilizará un filtro BPF (Berkeley Packet Filter) que aplica un filtrado dependiendo el tipo de interfaz (Llopis Polvoreda 2017, p. 21)

Dependiendo del filtro que se haya seleccionado se aplicará un preprocesador, el cual es el encargado de arreglar, rearmar o modificar los datos para que en la etapa posterior denominada motor de reglas determine las amenazas dentro del tráfico, el análisis se realiza con todas las reglas que se encuentren configuradas, y cada coincidencia ejecutara su acción respectiva. Los preprocesadores más comunes que se utilizan en Snort son los siguientes (Llopis Polvoreda, 2017, p. 21):

- **Frag3:** se encarga de organizar los fragmentos de los paquetes con el fin de que puedan ser analizados correctamente.
- **Stream5:** de igual forma es capaz de ensamblar paquetes, pero también incluye capacidades de gestión de sesiones TCP, UDP e ICMP en búsqueda de anomalías en los paquetes.

- **http_inspect:** analiza la estructura de del protocolo HTTP pudiendo extraer fragmentos tales como cabeceras (header), URI, host, etc.

Mediante el filtro de eventos determinamos el número de veces que una regla ha generado una alerta, de forma que reduzcamos los falsos positivos que se puedan presentar. Los logs donde se guarda la información descriptiva de los eventos se crean como texto o binario, en función de lo que el motor de regla haya detectado. De esta forma los resultados serán presentados según el método configurado como salida, ya sea como un correo electrónico, alerta remota o directamente al administrador por consola (Llopis, 2017, p. 21).

2.7.1.3. Reglas

Las reglas o también denominadas firmas son el elemento fundamental para Snort ya que dependerá como se encuentre estructurada la regla para que pueda detectar anomalías en los paquetes de datos. Como se mencionó anteriormente la profundidad con la que se desee realizar el análisis dependerá del protocolo o capa (OSI o TCP/IP) que se desea revisar, mediante las reglas podríamos analizar los encabezados de las capas tales como de red y transporte (IP, TCP, UDP, ICMP), capas de aplicación (FTP, HTTP, etc) o analizar datos específicos en búsqueda de palabras claves dentro de los paquetes de información (Marina, 2021, p.13).

Para escribir una regla es necesario definir dos secciones; la primera denominada como encabezado de regla (Rule Header), en la cual se debe establecer una acción para los casos de coincidencia, el tipo de paquete para analizar (TCP, UDP, etc) además de la dirección IP de origen, la dirección de destino y el número de puerto a ser escaneado. Por otra parte, es necesario especificar las opciones de regla (Rule Options), esto quiere decir que es necesario especificar que caracteres o datos activarán las alertas de coincidencia, además un valor SID (Snort ID) el

cual debe ser único ya que de otra forma entraría en conflicto con otra regla, este valor puede ir desde el 0 hasta 1.000.000 (Marina, 2021, p.13).

Las opciones de reglas brindan mayor flexibilidad y control para el administrador permitiendo distinguir bits, datos o paquetes específicos; en el caso de que se incluyan expresiones regulares deben ser ingresados en el formato de Pearl mediante la opción PCRE. En el caso de que el texto contenga comillas, es necesario "escaparlas" mediante el uso del carácter Backslash (\ ). Backslash solo se puede usar desde la versión 1.8 de Snort en adelante. Como ejemplo se describirá una regla con la que se desea detectar la palabra "ATAQUE" dentro de la capa de Red (modelo OSI).

En la sección de cabecera establecemos una *alert* como acción para los casos de coincidencia, IP como protocolo a analizar y para la dirección de origen y destino *any any -> any any*, de esta forma analizaremos cualquier dirección IP y cualquier puerto que se encuentre disponible. En la sección de opciones definimos un sid: 1000001 que identifica la regla y el mensaje que deseamos detectar será *msg:" palabra ATAQUE detectada"; content:" ATAQUE".* Entonces la regla quedara de la siguiente forma (Marina, 2021, p.13):

alert ip any any -> any any (sid:1000001;msg"palabra ATAQUE detectada";content:"ATAQUE")

Al instalar la herramienta Snort no se incorporan reglas y por lo tanto éstas deben ser adquiridas desde repositorios oficiales o de terceros, la misma organización de Snort ofrece suscripciones para adquirir reglas basados en la recopilación de vulnerabilidades descubiertas por investigadores, pero también existen comunidades o grupos de entusiastas que crean sus propias bibliotecas y que se encuentran disponibles para cualquier persona que las desee utilizar, en cualquiera de estos casos la principal ventaja es que nos brindan soluciones para los casos más conocidos de ataques, ahorrándonos tiempo al no tener que reescribir las reglas, pero en contraparte la desventaja es que podrían no ajustarse a las características particulares de nuestra red (Ramiro, 2020, p.37).

Las fuentes de donde podemos adquirir reglas para Snort ya sea para implementarlas en una red o con fines investigativos se clasifican en 4 grupos (Ramiro, 2020, p.37):

- **Equipos de Investigación de Vulnerabilidades (VRT):** Reglas de Snort oficiales.
- **Emerging Threates (ET):** Reglas de amenazas emergentes, se actualizan varias veces al día.
- **Reglas de Comunidad:** Reglas de comunidades y entusiastas de la investigación.

- **Reglas caseras y otras:** Reglas creadas y mantenidas localmente.

2.7.2. Herramienta Suricata

Otra de las herramientas con las que se puede contar para la creación de un sistema de defensa informático es Suricata, la misma organización que desarrolla Suricata define la herramienta como un motor de monitoreo de seguridad de red que puede trabajar como IDS/IPS de alto rendimiento. Ésta herramienta es de código abierto, pero su desarrollo se realiza mediante la fundación Open Information Security Fundation (OISF) la cual es administrada sin fines de lucro (suricata.readthedocs.io, 2019).

Suricata frecuentemente es utilizado de tres formas, la primera como IDS en tiempo real gracias a las prestaciones que tiene su motor, también puede ser configurado como IPS generando una prevención sustentable en la red, de igual manera puede ser implementado en un supervisor de seguridad de red NSM y permite realizar procesamiento offline de paquetes PCAP. El método de funcionamiento de Suricata se basa en la inspección de tráfico de red mediante su potente y extensa estructura de firmas ya que cuentan con soporte de lenguaje de comandos *Lua* el cual es un lenguaje de programación enfocado a la creación de script y permite describir datos que luego son analizados por el motor de Suricata, de esta forma es posible detectar amenazas complejas. Para integrarse en los sistemas de supervisión de red utiliza los formatos YAML y JSON pudiendo manejar datos de entrada o salida lo que permite que trabaje de forma dinámica con el resto de sistemas en la red, entre las principales plataformas que administran bases de datos compatibles con Suricata tenemos Splunk, Logstash/Elasticsearch y Kibana (itconnect.lat 2021).

2.7.2.1. Características

Suricata al ser una herramienta de uso avanzado presenta características escalables lo que nos permite aplicar un monitoreo más global de la red, además presenta funciones multi-hilo permitiendo un balanceo de carga entre todos los procesadores disponibles. Lo cual se refleja directamente en la capacidad de procesar un ancho de banda de hasta 10 Gbps(blog.elhacker.net, 2017).

Entre sus principales características tenemos (Alfon, 2021, p.11):

- **Multi-Threaded Processing:** ejecución de procesos/subprocesos de forma simultánea.
- **Automatic Protocol Detection:** permite escribir reglas independientemente del puerto que utilice el protocolo ya que son automáticamente detectados.

- **Performance Statistics:** creación de archivos de estadísticas y análisis de rendimiento.
- **HTTP Log Module:** registro de peticiones almacenadas en formato Log Apache.

2.7.2.2. Arquitectura

El Principal atributo que resalta en Suricata es su multi-core y toda su estructura ésta enfocada en realizar múltiples procesamientos mediante sus métodos de multi-threading, lo que da como resultado analizar múltiples reglas y protocolos de forma simultánea optimizando el rendimiento del sistema y aprovechando el alcance computacional del hardware. Los resultados de igual forma pueden ser generados en múltiples formatos estadísticos tales como pcap, json, unified2, entre otros, todos ellos compatibles con procesadores de ficheros.

Con Suricata mediante los procesos multi-hilos evitamos los problemas de eliminar paquetes al formar colas, algo que si ocurre con Snort; pero esto implica utilizar más recursos del sistema, por este motivo es más recomendado utilizar sistemas GNU/Linux que sistemas Windows ya que en GNU/Linux las instrucciones se ejecutan de forma separada. La estructura de procesamiento de la herramienta Suricata se basa en las siguientes secciones (Mattila, 2020, p. 21):

- Captura de paquetes.
- Decodificación y distribución según la capa de red.
- Detección.
- Salida.

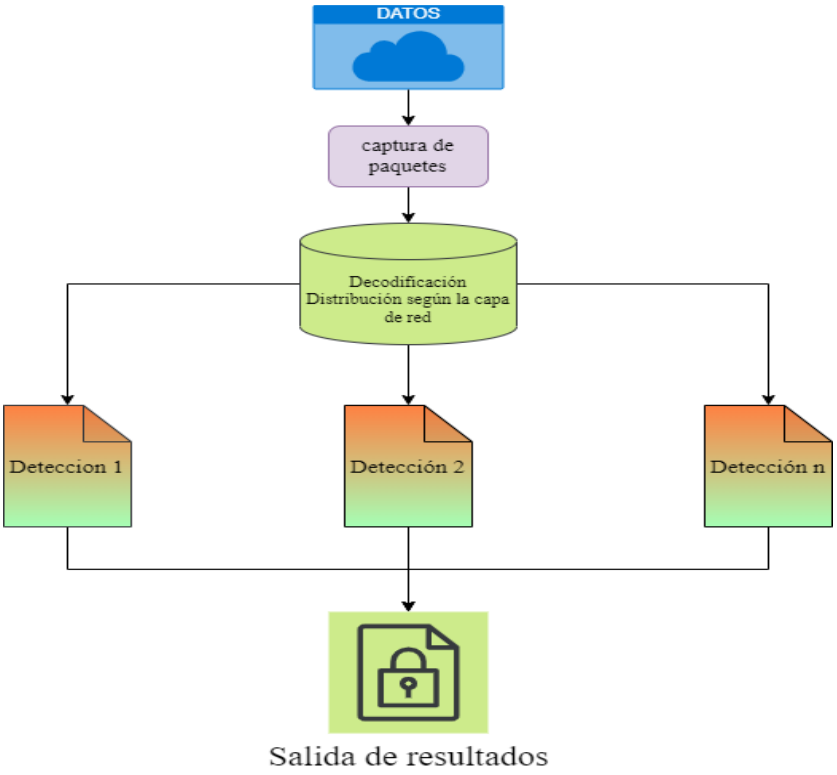


Figura 8-2: Etapas presentes en el análisis de paquetes la herramienta Suricata.

Realizado por: Álvarez; Andrés, 2022.

Al iniciar un análisis es necesario determinar el modo de ejecución ya que de éstos dependen los controladores de captura, pudiendo ser entre los más habituales la opción *–i* para PCAP, *-r* cuando se requiere PCAPFILE o *–q* en el caso de NFQUEUE, entre otros más. Al determinar el modo de ejecución internamente el sistema selecciona varios hilos los cuales a su vez determinaran que módulos, colas y patrones se utilizaran para realizar las tareas solicitadas. Para el tratamiento de paquetes se requiere del uso de tres módulos, los cuales son el capturador de paquetes, decodificación que identifica el protocolo según corresponda la capa (IP, ICMP, TCP, UDP, etc) y finalmente el módulo de procesamiento donde se identifican los datos y se busca patrones. Vale la pena señalar que dentro de los módulos existen subprocesos ejecutándose, de los que se puede destacar los siguientes (Hui Tsai jian, 2019, p.10):

- **Módulo de recepción:** captura los datos de la red.
- **Módulo de decodificación:** decodifica las capas de Enlace, de Red, Transporte; para la capa de aplicación utiliza un módulo dedicado (modelo TCP/IP).
- **Módulo FlowWorker:** distribuye los paquetes, gestiona las sesiones TCP, analiza y procesa los datos de la capa de aplicación.
- **Módulo de veredicto:** dependiendo del resultado del análisis los paquetes son etiquetados para descarte o algún otro tipo de análisis.

- **Módulo RespondReject:** se encarga de enviar un paquete de reinicio al sistema para que puedan conocer el estado del análisis del paquete.
- **Módulo de registro:** registra los resultados después del procesamiento de paquetes.

Los subprocesos se encuentran concentrados en la estructura ThreadVars (Per Thread variable structure) el cual determina las colas de datos de entrada INQ (in-queue) y salida OUTQ (out-queue), éste sistema también permite que las colas de salida vuelvan al sistema para un nuevo análisis en el caso de que sea necesario (Hui Tsai jian, 2019, p.10).

2.7.2.3. Reglas

La estructura de la regla es similar a la herramienta Snort, y desde un punto de vista general se puede dividir en tres secciones, acción, cabecera y opciones; pero al usar una herramienta como Suricata existen muchas más instrucciones que se pueden agregar y permiten que las reglas trabajen de una forma más intuitiva. Mediante éste enfoque se pueden definir las siguientes partes que componen una regla (Mattila, 2020, p. 21):

- **Acción:** ejecuta una orden cuando la regla encuentra coincidencias ALERT, PASS, DROP, REJECT, entre los principales.
- **Protocolo:** define que paquete será escaneado mediante el análisis del protocolo al que pertenece, TCP, UDP, IP, DNS, HTTP, FTP, SSH, etc.
- **Dirección y puerto de origen:** define la dirección IP y el puerto desde donde parten los paquetes a ser analizados, pueden ser direcciones individuales o grupos de direcciones.
- **Sentido en el que viaja el tráfico:** define la dirección en la que el tráfico circulará entre los extremos, solo viajará en un sentido *(*->) o en cualquier sentido (< >).
- **Dirección y puerto de destino:** define la dirección IP y el puerto hacia donde llegan los paquetes que son analizados, igualmente pueden ser direcciones individuales o grupos de direcciones.
- **Opciones:** encierra los parámetros de la regla y cada opción es separada el símbolo (;), las opciones pueden ser simplemente palabras dentro de un texto como “Facebook” o identificadores como “http_header”, además requiere de un identificador único SID (suricata id).

Un claro ejemplo es definir una regla encarga de eliminar todos los paquetes que provenga desde la dirección de red local ($HOME_NET), que utilice cualquier puerto (*any*) y que tengan como destino la dirección 1.2.3.4 que utilice cualquier puerto. Cuando existan coincidencias los Log imprimirán el mensaje de *“Acceso bloqueado”* y el identificador será SID *1000000.* Se debe

tomar en cuenta que los paquetes que viajen en el sentido opuesto a la dirección definida (->) no generarán ninguna alerta (Mattila, 2020, p. 23):

drop ip $HOME_NET any -> 1.2.3.4 any (msg: "Acceso bloqueado"; sid: 1000000;)

2.7.2.4. Elección de la Herramienta IPS

Una vez comprendida la estructura de las herramientas IPS es necesario definir el software con el que se va a trabajar y a pesar de que la protección que brindan es muy similar existen puntos que destacan a una herramienta sobre la otra. Pruebas de escenarios en entornos GNU/Linux (Rasilla Villegas 2020, p. 13-14) demuestran que Suricata es mucho más estable al analizar el tráfico de una red ya que su capacidad de multi-hilo le permite el procesamiento de más paquetes reduciendo las colas y por lo tanto disminuyendo la latencia, pero esto afecta directamente al consumo de recursos del equipo informático, mientras que Snort modera el consumo de recursos y permite que actividades complementarias del sistema no sean afectadas.

Entre los aspectos técnicos que se ha analizado se encuentra una mayor escalabilidad por parte de Suricata, así como más opciones al presentar la información de salida permitiendo formatos tales como Unified2, JSON o YAML mientras que Snort solo soporta Unified (Rasilla Villegas 2020, p. 13)

También se tomó en cuenta que Suricata realiza un analisis más profundo de los protocolos HTTP, FTP, SMB Y DNS siendo capaz de recopilar información adicional en sus Logs como por ejemplo si se requiere almacenar los certificados TLS/SSL (Mendoza et al, 2018, p. 7-8).

Por otra parte, Snort cuenta con una comunicad mucho más amplia la cual facilita el trabajo con esta herramienta y controla de mejor forma la solución a problemas, mientras que en Suricata la información es un poco más escasa además de que las guías existentes no contemplan varios problemas que se puedan presentar al momento de su implementación (Rasilla Villegas 2020, p. 13).

Después de considerar todos estos aspectos que se pueden presentar dentro del proyecto se ha tomado la decisión de utilizar la herramienta Suricata ya que se adapta mejor al tipo de análisis que se desea realizar, todos sus módulos son compatibles para el sistema ARM de Raspberry Pi. Además, evitamos desperdiciar los recursos de procesamiento disponibles por parte de la Raspberry Pi, y adicionalmente al existir información limitada sobre el tema a tratar, se presenta la oportunidad de generar nueva información que puede servir a futuras investigaciones. Adicionalmente se puede resaltar la similitud entre los dos sistemas, lo que permite que una

solución existente en un sistema pueda ser aplicado en el otro tomando en cuenta las configuraciones necesarias.

2.7.3. Wireshark

Wireshark es un software que según su propia página web (Wireshark, 2021) se define como un analizador de protocolos de red más importante y más utilizado en el mundo. El software permite analizar de forma muy detallada el tráfico de una red, por este motivo prácticamente se ha convertido en un estándar al realizar escaneos de red por instituciones gubernamentales, comerciales o independientes, lo que permite tener una retroalimentación con su comunidad mundial de desarrolladores, de esta forma se puede corregir errores y agregar nuevas funciones.

2.7.3.1. Características

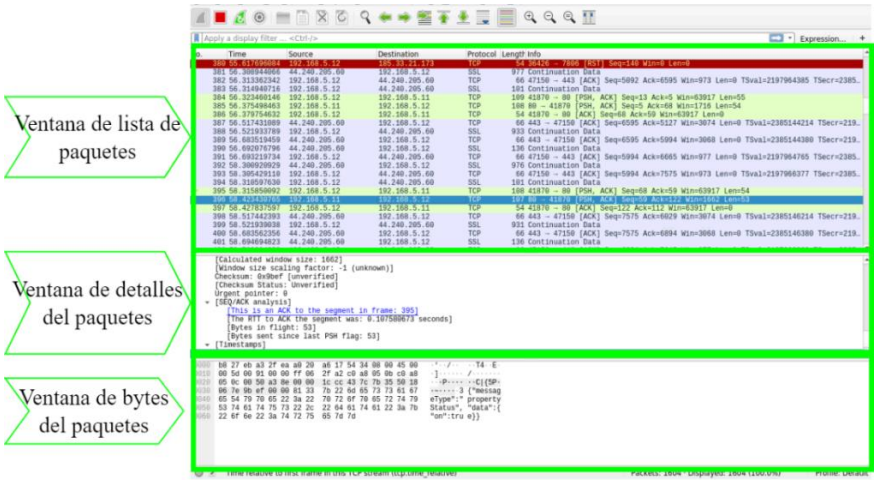
Wireshark se ha convertido en una herramienta imprescindible para el análisis de paquetes en redes, debido al sistema completo de filtros con los que cuenta, pero a su vez esto le agrega cierto grado de complejidad al momento de utilizarlo. La herramienta internamente cuenta con un sistema de filtros con los que se pueden realizar tareas básicas para el análisis de los paquetes que interesan, pero también se puede realizar un estudio más avanzado mediante el uso de expresiones con las que cuenta y de esta forma adecuar los filtros a las características de los paquetes que se busca analizar (López, 2019, p.1).

Esta herramienta se encuentra disponible para los sistemas Windows, MacOS y Linux de forma gratuita, por lo tanto, puede detectar características de una red como son la idoneidad, estabilidad, problemas o flaquezas de seguridad entre otras. El análisis puede realizarse de forma automatizada o de forma manual dependiendo de los parámetros que se configuren en la sección de filtros y expresiones, los parámetros que se pueden ingresar son números de paquetes capturados, numero de archivos, tamaño de los paquetes en KB, MB, GB, así como también el periodo de tiempo de captura en segundos, minutos u horas. La información recopilada puede ser almacenada en diferentes formatos, muchos de ellos siendo estándar para otras herramientas de análisis; entre los formatos más importantes están PCAPNG, PCAP (López, 2019, p.1).

Otra de las características importantes de la herramienta es que su interfaz gráfica facilita la búsqueda de información dentro de los paquetes de red, de forma que en la ventana principal es posible marcar los paquetes mediante un código de colores, el número para distinguir el orden de captura, momento de la captura. Dentro del paquete se puede visualizar la dirección IP de origen y destino, el protocolo empleado (TCP, TLS, ICMP, ARP, etc), además del tamaño del paquete

en bytes. La interfaz gráfica cuenta con tres ventanas, cada una especializada en detallar cierto tipo de información las cuales son(altitudetvm, 2020):

- **Ventana de lista de paquetes:** se muestran los paquetes según el orden en el que son capturados y se desplazan de forma tabular de forma que se puedan visualizar los nuevos paquetes que van ingresando, la información de cada paquete se muestra en fila empezando desde la izquierda a derecha con el número secuencial de captura, tiempo en el que fue capturando el paquete desde que inicio la captura, origen o fuente del paquete, destino, protocolo utilizado, longitud del paquete en bytes y la información que se pueda extraer del paquete, si este se encuentra encriptado no será posible visualizar la información.
- **Ventana de detalles del paquete:** es la ventana intermedia del Wireshark y presenta la información contenida en el protocolo siendo estas características más detalladas, tales como puertos utilizados, la versión de la tecnología utilizada, nombre y características de la interface de red, entre otros.
- **Ventana de bytes del paquete:** en la última ventana inferior se encuentra la información del paquete que se seleccione, es presentada sin ningún procesamiento, el cual está compuesto de 16 bytes hexadecimales y 16 bytes ASCII.



Fuente 9-2: Ventanas presentes en la herramienta Wireshark.

Realizado por: Álvarez, Andrés, 2022.

Existe una sección más en la interfaz del software en la parte superior de las ventanas, denominado filtro de captura y su función es la de permitir al usuario realizar consultas o en otras palabras ingresar los filtros necesarios para que el sistema capture un tipo de paquete específico (altitudetvm, 2020).

De forma detallada la comunidad que desarrolla Wireshark a través de su página oficial destaca las siguientes características de la herramienta (Wireshark, 2021):

- Análisis profundo de cientos de protocolos y continuamente se agregan más.
- Captura en tiempo real y análisis en modo offline.
- Estandarización de paneles para la búsqueda de paquetes.
- Multiplataforma: Windows, Linux, macOS, Solaris, FreeBSD, NetBSD, entre otros.
- Se puede navegar entre los datos capturados de la red a través de la interfaz gráfica (GUI) o por terminal TTY-mode mediante TShark.
- Utiliza los filtros más importantes dentro de la industria informática.
- Agrega análisis de VoIP.
- Lee y escribe muchos formatos de archivos de captura tales como: Tcpdump (Libpcap), Pcap NG, Catapult DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, Network General Sniffer® (comprimido y sin comprimir), Sniffer® Pro y NetXray®, Network Instruments Observer, NetScreen snoop, Novell LANalyzer, RADCOM WAM/LAN Analyzer, Shomiti/Finisar Surveyor, TEktronix K12xx, Visual Networks Visual UpTime, WildPackets EtherPeek/TokenPeek/AiroPeek.

- Capacidad de descomprimir en tiempo real archivos de tipo Gzip.
- Los datos pueden ser leídos en tiempo real desde interfaces de tipo: Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, y varios más si la plataforma lo permite.
- Soporte de desencriptación de protocolos tales como: IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP y WPA/WPA2.
- Se puede aplicar un código de colores a la lista de paquetes para agilizar el análisis y hacerlo más intuitivo.
- Los valores de salida pueden ser exportados en formato XML, PostScript®, o texto plano.

2.7.3.2. Filtros

Wireshark cuenta con filtros que permite realizar búsquedas más precisas dentro del tráfico capturado. Los filtros dentro de Wireshark cumplen dos funciones fundamentales, por una parte, permiten visualizar paquetes específicos (Display Filters) los cuales pueden ser utilizados en cualquier momento y básicamente su función es ocultar los paquetes que no se desea tener presentes en lista y por otra parte adaptar la captura de paquetes (capture Filters) mediante la delimitación del tamaño de una captura bruta, dicha configuración debe realizarse previa a iniciar el ataque y no puede ser modificada cuando la herramienta Wireshark ha iniciado (Baz, 2020, p.21).

- Filtros de captura: como se mencionó anteriormente limitan los paquetes que pueden ser capturados, para su implementación es necesario utilizar el lenguaje de filtros Libcap el cual está compuesto de expresiones primitivas las cuales son identificadores y calificadores; de forma práctica existen 3 argumentos necesarios para definir una expresión. Por una parte, tenemos los identificadores Tipo, los cuales hacen referencia al nombre o un número de identificación, por defecto se establece la expresión host, el cual recopila el tráfico del equipo anfitrión; entre los más usados se encuentra; NET, PORT, PORTRANGE. Expresiones como Gateway host permite escanear el tráfico de la puerta de enlace de la red sin necesidad de conocer su dirección IP (Baz, 2020, p.21).

A este grupo también pertenece la expresión Dirección (DIR) que permite definir una dirección de transferencia o recepción, dentro de este grupo se encuentran parámetros como SRC/DST (source/destination), RA, TA, ADDR1, ADDR2, etc.; hay que considerar que alguna expresión puede ser utilizado para conexiones inalámbricas. En el caso de que se agregue la máscara de red se escaneara al grupo de computadoras que se encuentren dentro del rango de direcciones IP (Baz, 2020, p.21).

Otro grupo importante es el Protocolo (proto) ya que se encargan de reducir el análisis a un protocolo específico, entre los más importantes se encuentran ETHER, FDDI, TR, WLAN, IP, IP6, ARP, TCP, UDP, entre otros; en el caso de que no se especifique un protocolo por defecto se capturara todos los protocolos que puedan ser utilizados por el servicio, por ejemplo UDP y TCP para un servidor DNS (Baz, 2020, p.21).

- Filtros de visualización: Mediante estas instrucciones controlamos lo que deseamos ver en la ventana de lista de paquetes, su estructura primordial requiere que se realicen comparaciones mediante operadores lógicos y en el caso de utilizar expresiones más complejas es necesario el uso de paréntesis. Ejemplos básicos de las expresiones utilizadas son TCP.PORT el cual limita la información visualizada a todos los paquetes que contengan el puerto referido; otro caso es la expresión IP.ADDR el cual presenta el tráfico de la dirección IP especificada o también al utilizar ETH.ADDR mediante el cual se muestra visualizamos el tráfico que atraviesa dicha dirección MAC. Cabe señalar que los ejemplos anteriores presentan el tráfico de origen y de destino (Baz, 2020, p.21).

2.7.4. Tcpdump

Tcpdump es una herramienta similar a Wireshark ya que también permite el análisis de paquetes, pero mediante la introducción de comandos, al no necesitar interfaz gráfica se reduce

considerablemente el tamaño utilizado en disco, puede ejecutarse en cualquier sistema que permita el ingreso de línea de comandos principalmente en sistema UNIX, por este motivo es el método más rápido para supervisar redes en búsqueda de fallos de seguridad o problemas de conexión, entre otros (Sandoval, 2020, p.33).

2.7.4.1. Características

A pesar de que no cuenta con una interfaz gráfica es posible analizar las cabeceras de los paquetes además varias características más; entre las más importantes tenemos (Khurana, 2020):

- Marca de tiempo de captura del paquete.
- Tipo de Protocolo.
- Nombre o IP del host de origen.
- Nombre o IP del host de destino.
- Bandera encargas de indicar el estado de la conexión pudiendo tener posibles valores como S-SYN para el primer paso al establecer conexión, ACK cuando es un paquete de acuse de recibido correctamente, puede venir acompañado de un número, P-EMPUJE señalándole al receptor que los paquetes en su búfer, R-RST para los casos en que se detenga la comunicación, SEQ-1,2,3,etc para diferenciar el número de secuencia, WIN 453 comunica el número de bytes disponibles en el búfer de recepción generalmente acompañado del protocolo, LENGTH el cual presenta la longitud de la carga útil de datos. Cabe señalar que todas las operaciones que se realizan con esta herramienta deben ser teniendo privilegios de súper usuario ya que debe tener acceso a todas las interfaces existentes en el host (Khurana, 2020, p.2).

Otra de las ventajas es que nos permite guardar y abrir archivos de tipo. PCAP que es compatible con otras herramientas como Wireshark, por lo que podríamos guardar LOG de captura para luego ser analizados. Por otra parte es posible leer el contenido de los paquetes mediante la opción *–A* y de ser necesario ingresar la dirección web para un análisis más específico, esto evita recurrir a otras herramientas para desempaquetar el contenido del tráfico (Sandoval, 2020, p.18).

2.7.4.2. Filtros

Los filtros que se manejan en esta herramienta son similares a Wireshark y pueden ser definidos en tres grupos (Sandoval, 2020, p.6):

- **Filtros de protocolo:** encargado de capturar paquetes del protocolo que se defina
- **Filtros de puerto:** permitiéndonos filtrar el tráfico de un puerto especifico

- **Filtros de host:** con el cual podemos definir un host especifico además de distinguir si deseamos un host de origen o destino mediante SRC y DEST.

Todos los filtros pueden usarse por separado o combinados mediante operadores lógicos *y, o* (Sandoval 2020).

2.7.5. Webthings Gateway (Mozilla WebThings)

Webthings Gateway definida por sus propios desarrolladores (Francis et al., 2021) es un software libre capaz de actuar como Gateway de hogares inteligentes, permitiendo a los usuarios monitorear y controlar directamente sus propio dispositivos inteligentes desde la web evitando intermediarios. Al implementar Webthings Gateway en una red doméstica evitamos la necesidad de adquirir y configurar una plataforma IoT para cada marca de dispositivo IoT ya que esta aplicación se basa en los estándares de W3C (World Wide Web Consortium), de forma que no es necesario conocer la estructura interna que maneje cada marca de dispositivo IoT (WebThings, 2021).

El objetivo primordial de Webthings, es la de permitir la conexión de dispositivos físicos mediante la red global. Su propuesta se basa en la creación de un internet de las cosas descentralizado conectando dispositivos mediante direcciones URLs, de esta forma será posible vincular, descubrir y definir un modelo y protocolo de datos estándar para que sean interoperables (WebThings, 2021).

Inicialmente el proyecto se lanzó mediante la propuesta del proyecto Project Things realizado por un equipo de Mozilla en junio del 2017 seis meses después de haber presentado un documento técnico en el que se estudiaba la posibilidad de que Mozilla contribuyera en un ecosistema emergente IoT, Ben Francis señala que "queríamos aplicar las lecciones aprendidas de la World Wide Web al internet de las cosas para crear un IoT que ponga a las personas en primer lugar, donde las personas puedan moldear su propia experiencia y estén empoderados, seguras e independientes"(Francis, 2020, p.14).

Como se mencionó anteriormente el proyecto inicio con el apoyo de Mozilla y por este motivo inicialmente el proyecto fue llamado Mozilla Webthings, utilizando el dominio https://iot.mozilla.org pero a través de los años en que ha ido avanzando el proyecto ha tomado mayor identidad y ha creado una comunidad consolidada. Actualmente el proyecto ha cambiado de nombre pasando a llamarse desde ahora en adelante WebThings Gateway, usando el dominio https://webthings.io. Este cambio se debe a que Mozilla redujo inversión directa al proyecto transfiriendo el control y la responsabilidad a la comunidad existente alrededor del proyecto. Se

espera que este cambio permita una contribución continua y fácil para nuevas propuestas de código abierto, de forma que WebThings Gateway tenga un alcance mundial (Bryant, 2020, p.1).

La transición de dominio se realizó en el año 2020 teniendo como fecha límite el 31 de diciembre por lo que todos los proyectos que utilizaban el dominio de Mozilla debieron hacer el traspaso a WebThings mediante una actualización presentada como banner de la misma plataforma (Francis, 2020). La última versión desarrollada con el anterior dominio fue la 0.12 pero Actualmente se encuentra en la versión 1.0 la cual cuenta con un gran soporte de dispositivos y ha incrementado la biblioteca de Addons. Según la hoja de ruta del proyecto y los convenidos con los patrocinadores se espera comenzar con el desarrollo de una versión 2.0 (Francis, 2020, p.8).

2.7.5.1. Características

WebThings está pensada para que trabaje en la capa de aplicación del modelo OSI combinando varios protocolos de IoT y acoplándolo a tecnologías ya existentes subyacentes de la web. Las principales tecnologías compatibles con el software son Zigbee, ZWave, Bluetooth, HomeKit, Weave, Wi-Fi. Actualmente cuenta con más de cien addons desarrollados para WebThings, lo que ha permitido unir una amplia gama de diferentes protocolos y dispositivos a la web lo que también beneficia a la interfaz de usuario de WebThings.(webthings, 2021).

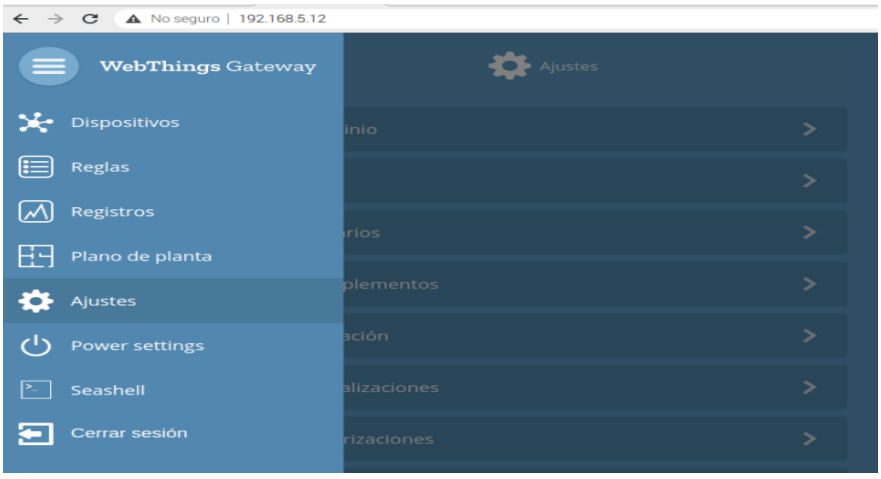


Figura 10-2: Página principal de Webthings.

Realizado por: Álvarez, Andrés, 2022.

WebThings está compuesta de dos partes principales el Framework y el Gateway, la combinación de estas secciones permiten que el entorno de WebThings pueda trabajar como un servidor para los distintos dispositivos IoT, al levantar los servicios por primera vez el sistema nos pedirá crear un subdominio el cual puede ser por ejemplo *"reddomestica.webthins.io"*. Al acceder a la interfaz

del administrador podemos realizar cualquier tipo de modificación, desde agregar nuevos dispositivos hasta crear subrutinas o eliminar los dispositivos (Francis, 2020, p.2).

El servidor tiene tres formas de ser montado, puede ser mediante la instalación del software en un sistema Linux, utilizando servicios en la nube como Docker y mediante el uso de una placa Raspberry el cual puede trabajar como Access Point y servidor de los dispositivos IoT (webthings, 2021),

Este último caso será utilizado para la realización del presente proyecto. Otra de las características fundamentales a mencionar es como el sistema obtiene las características de los dispositivos y determina que acciones puede realizar con estos. El método propuesto por el proyecto WebThings y la parte más importante de su sistema es el trabajo que se realiza mediante APIs (Application Programming Interface) ya que son secciones de código encargadas de comunicar servicios o aplicaciones entre sí, en el caso específico de WebThings las APIs se encargan de compartir las especificaciones del dispositivo además de los protocolos empleados de forma que esta información se pueda estandarizar y sea útil para otros dispositivos dejando de lado aspectos irrelevantes para la web como el tipo de hardware empleado (Prader y Engineer, 2018). La función principal de las APIs son las siguientes:

- Asignar a cada dispositivo una dirección URL.
- Utilizar un modelo de datos estándar para describir los dispositivos, sus propiedades y sus funciones. Para crear esta información se utiliza JSON objects.
- Las APIs también se encargan de supervisar y controlar los dispositivos ya sea por http o WebSockets.
- Las APIs están disponibles de forma que todos dispositivos sean capaces de manejar dichos estándares, serán capaces de solicitar información y de esta forma monitorear y controlar a otros dispositivos.

Por otra parte, la interfaz del administrador presenta 3 secciones principales mediante las cuales podemos interactuar con los dispositivos. En primer lugar encontramos la sección de descubrimiento de dispositivos mediante el cual el sistema busca la existencia de un nuevo dispositivo que pueda ser agregado, para que los dispositivos puedan ser descubiertos el servidor hace uso de su parte de Gateway de forma que los dispositivos IoT deben conectarse a la red perteneciente de WebThings, una vez conectado el dispositivo aparecerá en la página de WebThings y podrá ser agregado para interactuar con él (Prader y Engineer, 2018, p.1).

La segunda Sección es la encargada de presentar todos los dispositivos que se encuentran conectados y disponibles para WebThings, de esta forma podemos conocer datos importantes como el tipo de dispositivo conectado, nombre del dispositivo, si se encuentra encendido o apagado; también dentro de esta pestaña es posible interactuar con los dispositivos ya sea encendiéndolos o apagándolos o cambian el icono con el que se representa al dispositivo. La última sección y la más atractiva es la sección de reglas en donde se puede cambiar las propiedades de un dispositivo sin la necesidad de una interacción humana, de forma que el entorno creado por la red de WebThings se pueda considerar "inteligente" siendo capaz de reaccionar a eventos predefinidos, ya sea por la activación de un sensor o un horario establecido (Prader y Engineer, 2018, p.3).

También se puede señalar que permite cargar diagramas del domicilio de forma que se pueda conocer la distribución de los dispositivos. Por otra parte mediante Settings se puede modificar el nombre del subdominio, configurar una clave de acceso para el administrador, agregar o elimiar Addons, buscar actualizaciones, activar la comunicación ssh, entre otros (Prader y Engineer, 2018). Los componentes más importantes de WebThing son 3 (webthings, 2021):

- **WebThings Gateway:** está compuesto del back-end para el servidor diseñado en Node.js y del front-end para el cliente, es la primera parte del software en cargarse y se encarga de descubrir los dispositivos IoT asi como facilitar la conexión hacia la red, internamente almacena la información mediante SQLlite aprovechando la simplicidad que esta herramienta presenta (Stegeman, 2020)
- **WebThings Framework:** es una recolección de componentes de software reutilizables enfocados en la creación de complementos web para WebThings y se comunican directamente con las APIs, de forma que sean descubiertos por el Gateway y presentados al cliente. Mediante las líneas de código se debe describir las capacidades del dispositivo IoT y los métodos en que se debe administrar y monitorear estos. Los componentes pueden ser creados mediante Node.js, Python, Java, Rust o Arduino (webthings, 2021).
- **Addons:** son un complemento esencial para los dispositivos, se encarga de adaptar los protocolos que cada uno de los dispositivos utiliza y que el resto del software necesita para poder entenderlos, para este objetivo se requiere definir las propiedades del dispositivo, acciones posibles y eventos. Los Addons deben ser capaces de manejar los dispositivos de hardware conectados directamente por ejemplo los puertos GPIO de la Raspberry, así como también servicios en la nube por ejemplo datos meteorológicos.

2.7.6. Arduino-IDE

Para poder programar las placas y módulos de Arduino es necesario hacer uso del software adecuado para este fin, por consecuencia es indispensable el uso del IDE (Integrated Development Environment) de Arduino ya que facilitara muchas de las tareas que se pueden realizar con las placas. Mediante este software es posible escribir, depurar, editar y grabar programas (sketches) para que realicen las acciones que se desea (Arduino, 2021, p.2).

Una de las principales ventajas de usar este software es que puede ser instalado en sistemas Windows, Mac OS y Linux debido a que las librerías y driver con los que se controlan las placas y módulos están disponibles para todos los sistemas nombrados de forma que nos garantiza la misma respuesta siempre por parte de las placas, independientemente del sistema que utilicemos. La ventana de Arduino ofrece todos los recursos necesarios para programar y también para realizar depuraciones de código ya que contamos con botones de verificación de código, subida mediante el programa se graba en la placa así como también la opción de un monitor serial mediante el cual se puede visualizar el comportamiento de la placa (Arduino 2021, p.20).

2.8. Hardware

2.8.1. Raspberry Pi

El dispositivo Raspberry Pi (RPi) es un pequeño computador de bajo costo el cual dispone de todas las características que se necesita en un ordenador personal, todo concentrado en una sola placa. Su potencial permite navegar por internet, crear, modificar y eliminar archivos, trabajar en proyectos de programación, electrónicos o interactuar con otros dispositivos (Gareth, 2020, p. 8).

Antes de existir la placa Raspberry Pi tal como la conocemos ahora se creó la Fundación Raspberry Pi sin ánimos de lucro en el 2021, la cual desde sus inicios a enfocado sus esfuerzos en fomentar la educación informática para todo el mundo, en especial a los sectores de bajos recursos. Por este motivo desde el primer modelo de Raspberry Pi se ha intentado reducir costos sin afectar al usuario, por este motivo se han creado varios modelos, los cuales pueden adaptarse a las necesidades que requiera un proyecto, un ejemplo claro es el modelo Raspberry Pi Zero, el cual omite algunas características tales como puertos USB y el puerto de red con cable, pero a pesar de estas carencias es capaz de ejecutar software escrito para un modelo diferente (Gareth, 2020, p. 8).

2.8.2. Raspberry PI 3B

El modelo con el cual se trabaja en el presente proyecto es el RPi v3B ya que sus características son muy favorables para un sistema IoT. Este modelo tiene incorporado un CPU Cortex-A53 de cuatro núcleos a 64bits con 1,2GHz de procesamiento por lo que es capaz de responder un 50% más rápido que el modelo anterior (RPi v2B) y diez veces más rápido que el primer modelo. Evidentemente mantiene la compatibilidad con todos los periféricos diseñados para los modelos anteriores, esto incluye los pines GPIO y la conexión con sensores (Raspberry Pi 3B - Raspberry Pi, 2019).

El cambio más importante que se produjo en este modelo fue la incorporación de conexión Wi-Fi de 802.11n y Bluetooth 4.1 de esta forma se dejó de depender de puerto Ethernet para acceder a internet, además de que gracias a las velocidades que presentan estas tecnologías se puede establecer comunicación inalámbrica con sensores y periféricos. Evidentemente es capaz de ejecutar todos los sistemas operativos diseñados para Raspberry Pi incluyendo el sistema operativo Windows 10 IoT. Sus principales características son las siguientes (Raspberry Pi 3B - Raspberry Pi, 2019):

- Chipset Broadcom BCM2837 a 1,2GHz.
- ARM Cortex-A53 de 64bits con 4 núcleos.
- LAN inalámbrica 802.11 b/g/n.
- Bluetooth 4.1 de bajo consumo.
- Coprocesador multimedia de doble núcleo Videocore IV®.
- Memoria LPDDR2 de 1GB (compatible con todas las distribuciones ARM GNU/Linux y Windows 10 IoT).
- 1 puerto Ethernet 10/100 Mbps.
- 1 conector de video/audio HDMI.
- 1 conector 3.5mm audio/video compuesto.
- 4 puertos USB 2.0.
- 40 pines GPIO.
- Antena de chip (Wi-Fi y Bluetooth integrados).
- Conector de pantalla DSI.
- Ranura de tarjeta microSD (Sistema Operativo).
- Dimensiones de 85mm x 56mm x 17mm.

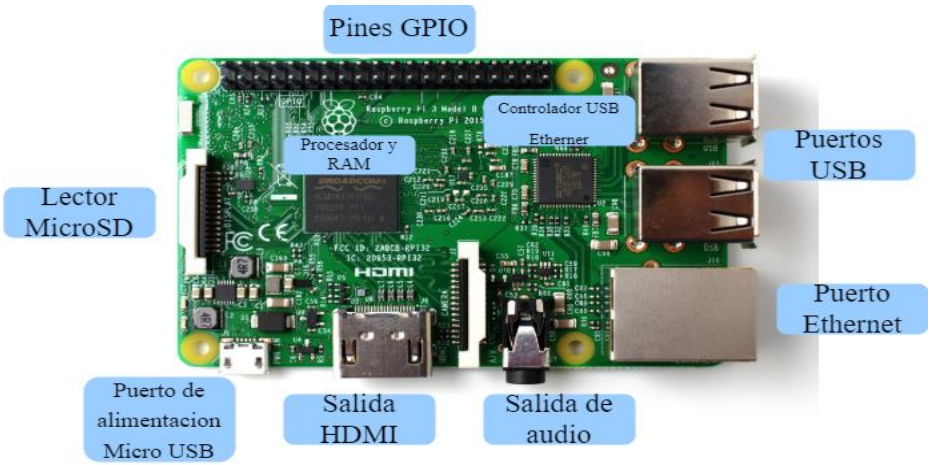


Figura 11-2: Componentes presentes en la placa Raspberry Pi 3B.

Realizado por: Álvarez, Andrés, 2022.

2.8.3. Raspberry Pi 4B

Otro de los modelos presentes en el desarrollo de este proyecto es el RPi v4B lanzado en junio del 2019, el cual cuenta con un procesador RISC (Reduced Instruction Set Computer) de 4 núcleos simétricos, ARM A72 a 64bits a 1,5GHz de velocidad. A partir de éste modelo se cuenta con el chip Broadcom BCM2711 que tiene una unidad dedicada específicamente al procesamiento grafico "VideoCore" VI de 32bits a 500MHz, además de mejorar aún más las comunicaciones inalámbricas al incorporar Ggabit Ethernet, Wi-Fi 802.11 b/g/n/ac y Bluetooth 5.0 (Gonzalez Alonso, 2021, p. 25).

Entre sus principales características tenemos las siguientes:(Cruz Garcés, 2021, p. 10)

- Procesador Broadcom BCM2711 de 64bits a 1,5GHz.
- Memoria RAM de 4GB LPDDR2 SDRAM.
- Interface de video Full-size HDMI.
- Entrada de audio 4-pole stereo, puerto I/O compuesto de video.
- Puerto Gigabit Ethernet.
- Antena Wi-Fi 2.4GHz y 5GHz IEEE 802.11 b/g/n/ac.
- Bluetooth 4.2 BLE.
- Puertos USB 2x2.0 y 2x3.0.
- Puerto para cámara CSI.
- Puerto DSI para pantallas táctiles.

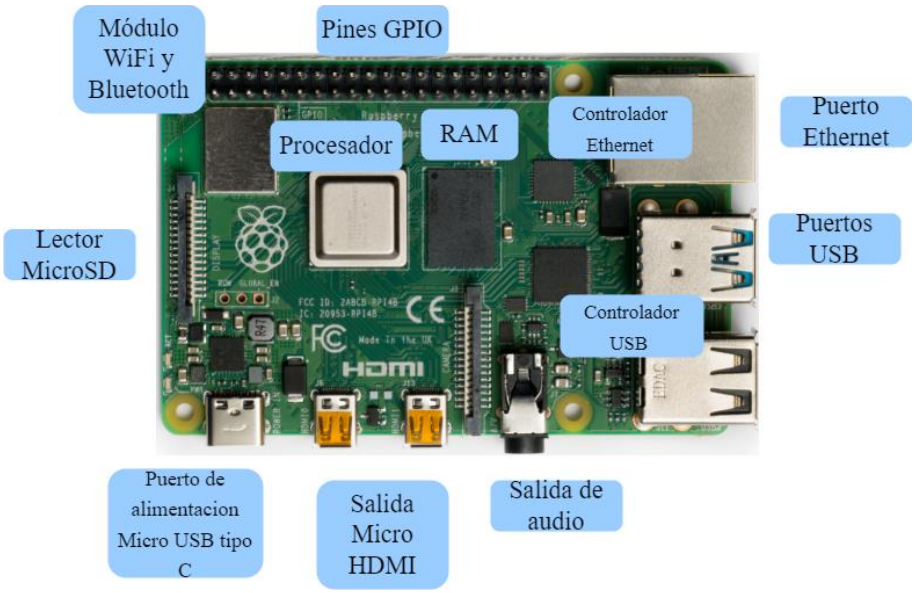


Figura 12-2: Componentes presentes en la placa Raspberry Pi 4B.

Realizado por: Álvarez, Andrés, 2022.

2.8.4. Módulo ESP8266 MOD

El módulo ESP8266 toma el nombre del chip Wi-Fi de bajo costo que trae incorporado, capaz de procesar protocolos TCP/IP provenientes de sus múltiples pines que pueden trabajar como entrada o salida y modificar los estados con los dispositivos que se comunica. Presenta características superiores con respecto a dispositivos como microchip o Atmel ya que cuenta con un microcontrolador Tensilica Xtensa LX106, el cual tiene una arquitectura RISC a 80MHz y 512KB de memoria flash. Soporta temperaturas de -40°C a 125°C por lo que es ideal para el control de actuadores en proyectos IoT, además cuenta con su propio convertidor analógico/digital (Analog to Digital Converter, ADC) de 10bits, el cual requiere un voltaje de 2.5V a 3.6V para un correcto funcionamiento. Sus principales características son las siguientes (Játiva et al, 2018, p. 3):

- CPU RISC de 32bits-Tensilica Xtensa LX106, oscilador 80MHz.
- Memoria RAM de instrucciones 64KB, RAM de datos 96KB.
- Memoria externa flash QSPI-512KB a 4MB (máximo 16MB).
- Wi-Fi IEEE 802.11 b/g/n (TR switch, balun, LNA, amplificador de potencia de RF y red de adaptación de impedancias, soporte de autentificación WEP y WPA/WPA2).
- 16 pines GPIO (I/O propósito general).
- SPI, I^2C.
- Interfaz I2S con DMA (comparte pines con GPIO).
- Pines dedicados a UART, 1xUART para transmisión a través del pin GPIO2.
- Convertidor ADC de 10bits

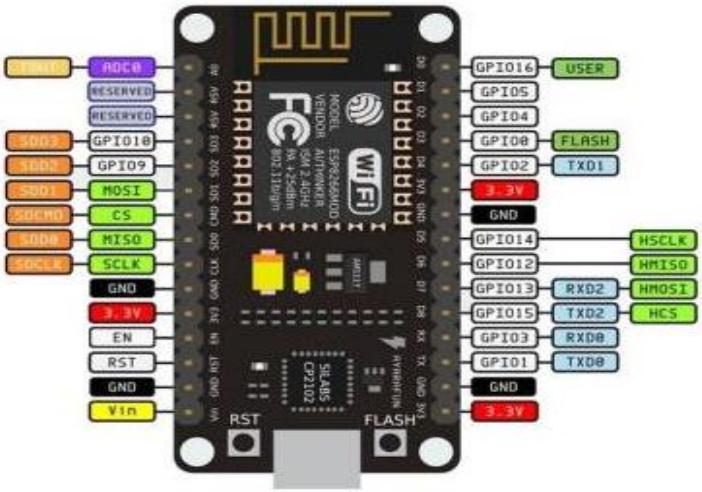


Figura 13-2: Distribución de pines del módulo ESP8266 MOD.
Fuente: (mrwatt, 2022).

2.8.5. Módulo WROOM 32

El módulo WROOM 32 diseñado por Espressif Systems presenta características superiores al modelo ESP8266 ya que está dirigido a servir de puente de comunicación entre dispositivos que no cuentan con esta característica, de forma que cualquier sensor pueda formar parte de una red IoT. Esto es posible debido a que el dispositivo es que es capaz de ejecutar aplicaciones en tiempo real lo que reduce los tiempos de espera y envío de datos dentro de la red (Benito Herranz, 2019, p. 16).

El chip que incorpora (del cual se toma el nombre ESP32) forma parte del módulo que contiene un oscilador de 40MHz, memoria flash y una antena que varía según el modelo, además de un PCB que permite conexiones serie/USB, botones de BOOT y RESET, y pines de usos múltiples. Entre sus principales características tenemos (Benito Herranz, 2019, p. 16).

- Voltaje de alimentación 3.3V (2.7V-3.6V) DC.
- Corriente de 80mA (fuente superior a 500mA).
- Voltaje lógico I/O 3.3V.
- SoC ESP32 (ESP32-D0WDQ6).
- CPU dual core Tensilica LX6 de 32bits.
- Oscilador de 240MHz.
- SRAM 520KB.
- Memoria flash externa 4MB.

- 34 pines digitales GPIO.
- 2 UART.
- 3 SPI.
- 2 I^2C.
- Interfaz SD.
- 3 Timers 16bits.
- PWM de led: 16 canales independientes (16bits).
- 2 ADC de 12bits.
- 2 DAC de 8bits.
- Wi-Fi con protocolo 802.11 b/g/n/e/i (802.11 a 150Mbps).
- Wi-Fi con certificación RF: FCC/CE/IC/TELEC/KCC/SRRC/NCC.
- Wi-Fi con rango de Frecuencia de 2.4GHz-2.5GHz.
- Wi-Fi seguridad WPA/WPA2/WPA2-Enterprise/WPS.
- Protocolo de red IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT.
- Protocolo bluetooth V4.2 BR/EDR y BLE.
- Bluetooth radios: NZIF receptor con sensibilidad -97dBm, class-1, class-2, class-3 transmitter, AFH.
- Bluetooth audio: CVSD y SBC.
- Stack de protocol TCP/IP integrado.

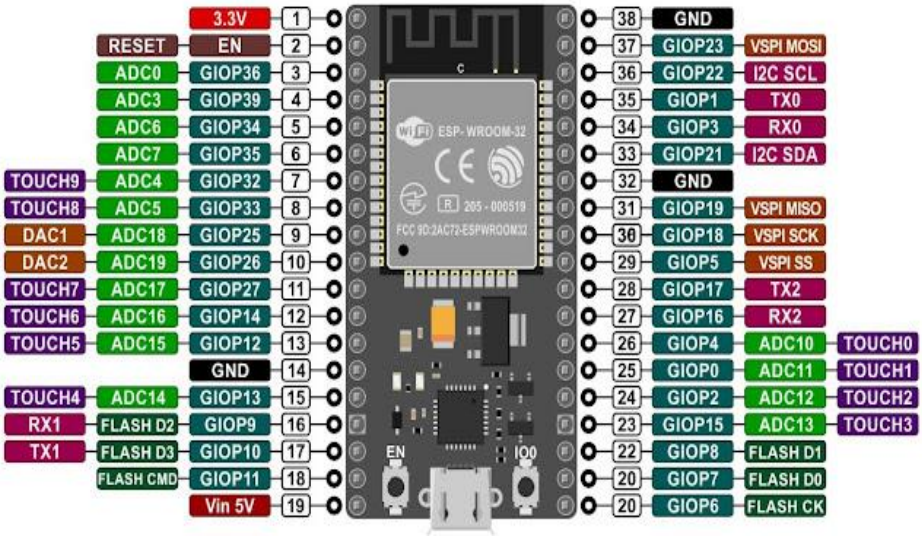


Figura 14-2:. Distribución de pines del módulo WROOM 32.

Fuente: (Asanza, 2022)

2.8.6. Módulo HL 525

El módulo HL 525 fabricado por la empresa Songle está compuesto de 2 Relays capaces de soportar 250V/10A, los Relays son de alta calidad debido a que tienen aislamiento eléctrico en canales separados, sus características los convierten en componentes ideales para trabajar con proyectos basados en Arduino, Raspberry Pi, ESP8266 (NodeMCU y Wemos), Teensy y Pic (naylampmechatronics, 2021).

El módulo entra en funcionamiento cuando recibe un "0" lógico (0V) y cambia de un estado normalmente abierto (NO) a un estado normalmente cerrado (NC); en el caso de que se aplique un "1" lógico (5V) regresará a su estado inicial. Los optoacopladores encargados de controlar estos valores lógicos encienden y apagan sus leds para que sea más fácil detectar el estado en el que se encuentran los Relays. Sus características principales son (naylampmechatronics, 2021):

- Voltaje de operación de 5V DC.
- Señal de control TTL (3.3V o 5V).
- 2 Relay con canales independientes (2 CH).
- Modelo de Relay SRD-05VDC-SL-C.
- Capacidad máxima de 10ª/250VAC, 10ª/30VDC.
- Corriente máxima de actuadores 10A(NO), 5A(NC).
- Tiempo de acción de 10ms/5ms.
- Activación con 0V (0 lógico).
- Entradas octoacopladas.
- Indicadores Led de activación.

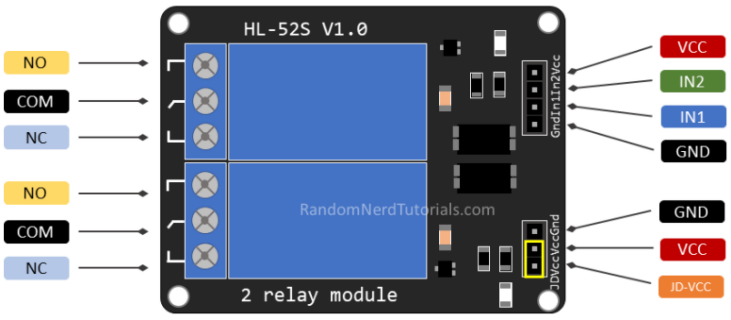


Figura 15-2: Distribución de pines del módulo HL 525

Fuente: (Vijay, 2022)

CAPITULO III

3. MARCO METODOLÓGICO

3.1. Metodología

En el presente capitulo se describe el procedimiento sobre el diseño e implementación de un firewall para dispositivos IoT utilizando Raspberry Pi, aplicando tecnologías IPS para detectar tráfico malicioso, que tiene como objetivo proporcionar una visión global en términos de seguridad para dispositivos basados en el internet de las cosas.

Para lograr este fin se detalla la metodología científica y tecnológica a seguir, la cual se realizará mediante el método experimental y que inicialmente define todas las características necesarias para posteriormente realizar las pruebas, recolección de resultados y su posterior análisis para el desarrollo de la investigación. La metodología científica que se aplica para el cumplimiento de cada una de las etapas en el diseño e implementación de un firewall para dispositivos IoT utilizando Raspberry Pi, aplicando tecnologías IPS para detectar tráfico malicioso se detalla a continuación.

El desarrollo del proyecto se basa en la metodología propuesta por el Instituto Nacional de Ciberseguridad de España (INCIBE); se ha escogido esta metodología ya que es el resultado de las experimentaciones realizadas por dicha institución. Además, esta metodología tiene un enfoque preventivo que es lo que se busca al implementar un sistema IPS de forma que los parámetros que se descubran en la red antes de sufrir un ataque sean el mejor aliado al desplegar el sistema de protección informático.

De forma específica las etapas inician con un estudio descriptivo para conocer el escenario y obtener las características más importantes; posteriormente se analiza a través de la investigación analítica y explicativa la definición del escenario, esta clasificación es necesaria para establecer las bases desde donde parte él trabaja. Bajo estos parámetros la metodología propuesta atraviesa 4 pasos fundamentales los cuales son:

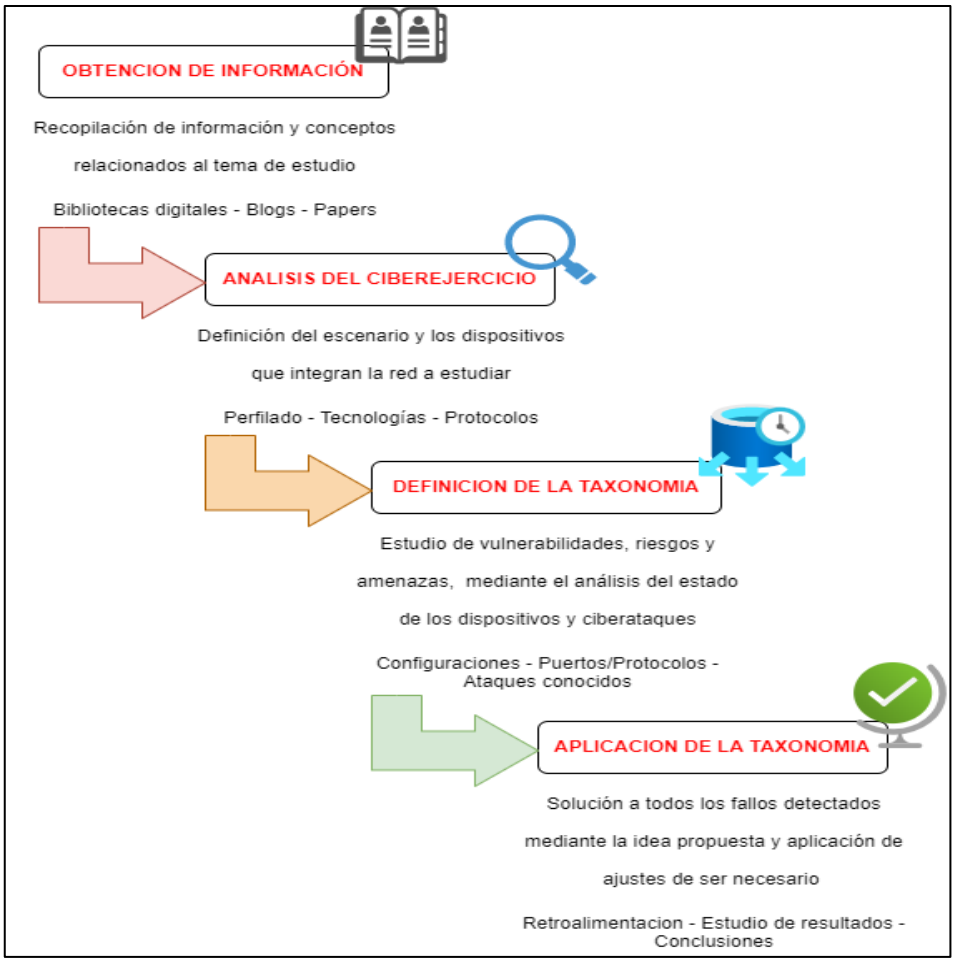


Figura 1-3: Diagrama de las etapas que intervienen dentro de la metodología.

Fuente: (incibe, 2019).

3.1.1. Obtención de información

En la primera etapa es necesario la adquisición de información relacionada con el tema a tratar ya sea de páginas web o libros, de forma que se entienda la estructura y comportamiento de los diferentes elementos que componen el proyecto, esto se puede evidenciar en el capítulo 1, en donde se describen los conceptos, características y demás detalles de cada uno de los elementos que intervienen en el desarrollo del proyecto.

3.1.1.1. Estado del arte de la protección de dispositivos IoT.

Desde la primera vez que se utilizó el termino IoT (Internet of Things) en 1999 en el instituto de Massachusetts la intensión de estas nuevas tecnologías fue la de dotar nuevas capacidades a los dispositivos electrónicos, permitiéndolos interactuar con su entorno y uniendo mucho más el mundo digital con el físico, esta unión ha generado nuevos retos para el mundo tecnológico y específicamente a las ciencias informáticas, ya que se abren nuevas áreas de investigación. Un

área que siempre ha estado presente es la relacionada con la seguridad, ya que diversos estudios académicos demuestran que las tecnologías IoT aun no alcanzan la madurez esperada. (Mauricio 2019, p. 26).

Desde este punto de vista, continuamente se van realizando diversos informes que intentan recopilar los puntos más importantes que surgen dentro del área de los dispositivos inteligentes, entre ellos se encuentra el informe presentado en el 2005 por la Unión Internacional de Telecomunicaciones (UIT) (UIT citado en Mauricio 2019, p. 27) en el cual se hace referencia a nuevas capacidades de los dispositivos IoT como por ejemplo asumir la identidad de una persona dentro del mundo virtual, esto permite que la comunicaciones entre cosas y personas se más fácil pero en contra partida nuestros datos personales quedan expuesto dentro del mundo virtual si no existen los sistemas de seguridad lo suficientemente blindados para protegernos.

Otros informes relacionados con el mundo de la seguridad dentro del campo del internet (Zabalo Arteche 2019, p. 8) hablan sobre una de las áreas más perjudicadas por las redes inseguras, específicamente se refiere a la industria, ya que estudios revelan que la ciberseguridad es uno de los puntos menos desarrollados y por este motivo se intenta mediante la Industria 4.0 corregir estas falencias dentro del área. Estos mismos estudios demuestran que las técnicas más utilizadas en ataques de redes industriales es la ingeniería social ya que es el método más rápido y fácil de obtener información, métodos más elaborados involucran el uso de malware en el que se integran a los virus, troyanos y diversas secciones de código alterado que son inyectados a los dispositivos; también se menciona directamente al malware ya que sus efectos pueden ser muy perjudiciales y requieren su propio estudio (Zabalo Arteche, 2019, p. 8).

De igual forma un fallo que se presenta muy frecuentemente y no siempre se puede corregir en los dispositivos IoT son las vulnerabilidades de Software, los cuales son explotados por ciberdelincuentes al descubrir fallos en secciones de código de los dispositivos, el tratamiento no es fácil ya que por una parte la industria no es capaz de actualizarse al mismo ritmo de la tecnología lo que deja gran parte de su software anticuado y por otro lado si los fallos no son reportados a tiempo pueden ser propagados en poco tiempo a centenares de dispositivos, permitiendo que los atacantes obtengan el conocimiento necesario para aprovecharlo. Otro de los ataques que han venido incrementado al pasar de los años es el conocido como DoS (Denia of Service) así como su versión más avanzada DDoS (Distributed Denia of Service) el cual ha provocado que las empresas dediquen un área específica para el monitoreo de esta clase de movimientos anormales dentro de la red (Zabalo Arteche, 2019, p. 8).

Como resultado de diversas investigaciones y principalmente existiendo la necesidad de definir métodos y recomendaciones para el estudio de vulnerabilidades en las redes de comunicación, se

crearon normas ISO las cuales están enfocadas en gestionar, organizar, así como identificar los riesgos dentro de una red. De forma específica la norma centrada en la seguridad de dispositivos conectados a la red, es la ISO/IEC 27033 Network Security, la cual se divide en 6 secciones (PECB, 2022):

- ISO/IEC 27033-1: Realizar un mapeo de la red, generar una visión sobre los conceptos y orientación de la gestión de seguridad de la red, así como los requisitos de seguridad.
- ISO/IEC 27033-2: Directrices para el diseño, la implementación y documentación desde un punto de vista científico, considerando la arquitectura de seguridad.
- ISO/IEC 27033-3: Demostración del escenario y sus amenazas, integra técnicas de diseño y problemas de control.
- ISO/IEC 27033-4: Enfocado a las directrices del riesgo, enfocado a los controles de puertas de enlace, de forma que se pueda proteger los flujos de información entre redes.
- ISO/IEC 27033-5: Presenta las guías para implementar y supervisar controles de redes privadas permitiendo comunicaciones remotas.
- ISO/IEC 27033-6: Se enfoca en los riesgo, diseño y control de redes inalámbricas IP, integrando las redes de área personal WPAN, área local WLAN y metropolitanas (WMAN).

Un ejemplo claro de la aplicación de estos parámetros relacionados al estudio que se ha venido realizando en el área de las redes IoT, es el desarrollado por la Industria IoT 4.0 en donde se considera la separación en campos todo el entorno que involucra la comunicación del dispositivo, de forma que en el campo de control se investigue el sistema encargado de gestionar los actuadores, también llamados entidades abstractas. Por otra parte, se analiza el campo de operación el cual integra las plantas y fuentes de poder que alimentan a todos los sistemas y que requieren un control que los monitorice. También se encuentra el campo de información el cual integra los datos manejados por los distintos componentes del sistema dentro de la red, en el campo de aplicación se agrupan los datos generados por los campos anteriores y se les da un sentido permitiendo que el sistema haga uso de los datos recogidos anteriormente. Finalmente, el campo de negociación hace uso de los datos generados durante el proceso de fabricación y se los convierte en información útil para el área de gestión de la empresa, aquí ya se puede extender a sectores como clientes, servicios de venta y planificación. (Zabalo Arteche, 2019, p. 30).

Por otra pare Institutos de investigación como el INCIBE enfocan el estudio de la seguridad en redes IoT mediante los vectores de ataques, de donde se definen puntos importantes como por ejemplo los errores en la implementación de dispositivos IoT principalmente en el área empresarial ya que no se segmenta de forma adecuada la red, lo que permite accesos a otros

recursos que no deberían estar visibles. También se considera los casos en los que el atacante se encuentra dentro de la red y es capaz de analizar el tráfico que circula en la red (incibe, 2021).

De igual forma INCIBE recomienda el análisis de vulnerabilidades presentes en las configuraciones deficientes, en muchas ocasionados mantener los parámetros del fabricante permite que el atacante tenga acceso a las plataformas que administran los dispositivos. Y otro de los puntos que pocas veces se considera es la capacidad de acceso físico a los dispositivos, además del desconocimiento de los usuarios, lo que facilita en gran medida el acceso de un atacante a la red y a los dispositivos (incibe, 2021).

Una vez recopilada la información se procede a generar una tabla en la que se puede valorar el nivel de información disponible que existe para el caso de aplicación. En el caso específico del presente proyecto al integrar 3 tecnologías dentro de la misma solución, los niveles que se obtuvieron fueron los siguientes:

Tabla 1-3: Nivel de disponibilidad de información en textos y publicaciones virtual.

Información disponible	Nivel de disponibilidad
Información sobre redes IoT	Alta
Información sobre tráfico malicioso	Bajo
Información sobre firewall, IDS, IPS	Media

Realizado por: Álvarez, Andrés, 2022.

3.1.2. Análisis del ciberejercicio

3.1.2.1. Planteamiento y Diseño del escenario

El escenario que se implementara para el desarrollo del presente proyecto tiene la siguiente distribución:

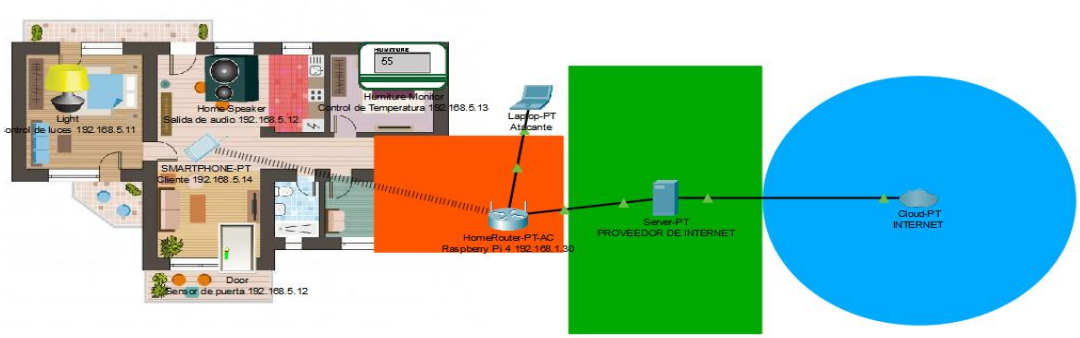


Figura 2-3: Escenario planteado para la red IoT doméstica.

Realizado por: Álvarez, Andrés, 2022.

El presente escenario está diseñado como una subred que contiene a los dispositivos IoT, se deriba de la red principal que distribuye el router domestico y que es la puerta de conexión con internet. Al agrupar a todos los dispositivos IoT en una subred es más eficiente monitorear su comportamiento además que las pruebas se pueden realizar sin el riesgo de afectar a otras areas de la red. En la presente red existen los siguientes dispositivos:

Tabla 2-2: Distribución de dispositivos según su direccion IP y servicio.

Dirección IP	Dispositivo	Servicios
192.168.1.1	Router del Proveedor	Gateway
192.168.1.30	Raspberry Pi 4B	Gateway de subred-eth0
192.168.5.1		Gateway de subred-wlan1
192.168.5.10	Raspberry Pi 3B	Monitor de red
192.168.5.12		Servidor WebThings
192.168.5.11	ESP8266	Control de lampara
192.168.5.13	WROOM 32	Sensor de temperatura
192.168.5.14	Smartphone	Cliente-ventana de administración

Realizado por: Álvarez, Andrés, 2022.

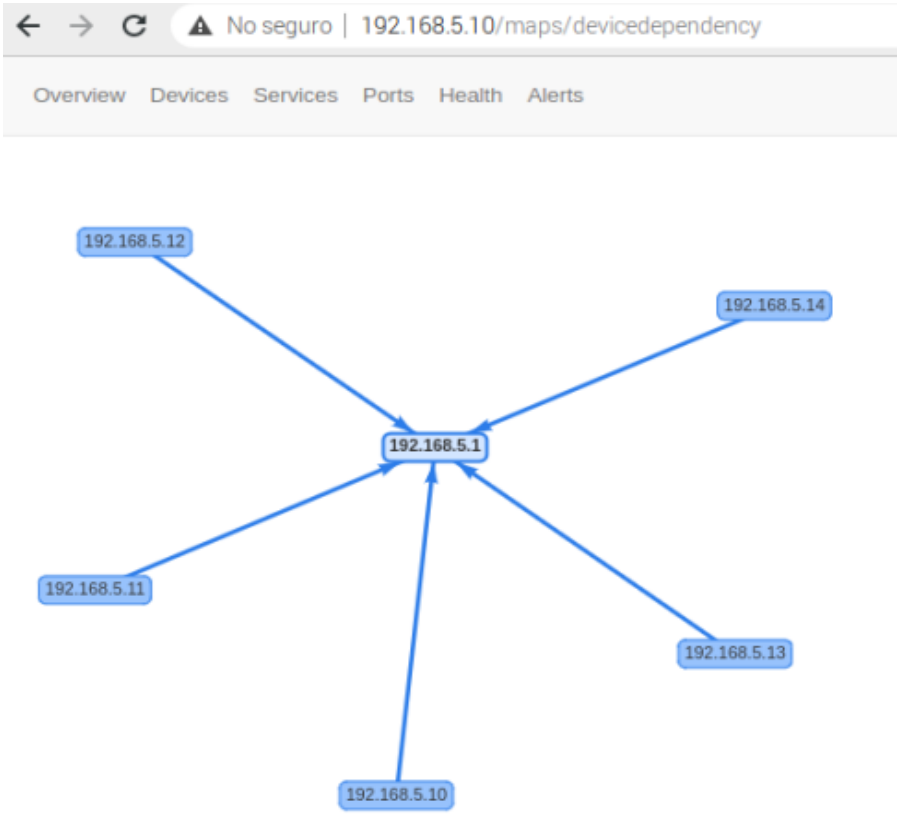


Figura 3-3: Vista de la red desde el servicio de monitoreo.

Realizado por: Álvarez, Andrés, 2022.

En el diagrama se puede observar que se trata de una red doméstica IoT, la cual consta de un enrutador ONT el cual es proporciona por el proveedor de servicio de internet y es quien nos permite tener salida hacia internet, desde este dispositivo se ha creado una red interna específica que utilizará la dirección IP 192.168.1.30 en un extremo y en el otro la dirección 192.168.5.1 como puerta de enlace y será por donde los dispositivos IoT tienen vista hacia internet. Para conseguir esta subred se hará uso de un dispositivo Raspberry Pi el cual hará las funciones de enrutador con salida a internet y punto de acceso para que los diferentes dispositivos IoT puedan conectarse de forma inalámbrica, y también se instalará la herramienta IPS dentro de este dispositivo. De esta forma se consigue trabajar dentro de una red controlada, en la que se puede monitorear y analizar todo el tráfico que circula sin interferir con las comunicaciones del resto de la red principal.

Los dispositivos IoT serán controlados mediante la plataforma Webthings Gateway la cual se encuentra instalada dentro de un segundo dispositivo Raspberry Pi, así como también una plataforma de monitoreo de red mediante el cual se podrá obtener diferentes parámetros para conocer el comportamiento de la red.

3.1.2.2. Configuraciones

- **Raspberry Pi**

Antes de realizar cualquier acción en la tarjeta Raspberry Pi 4B es necesario cargar un sistema operativo y definir algunas configuraciones para la cuenta de usuario.
Para empezar, será necesario cargar el sistema operativo correspondiente para la tarjeta Raspberry Pi, como se analizó en el capítulo anterior el procesador ARM Cortex-A72 del modelo 4B permite ejecutar sistemas de 64bits como es el caso de alguna versión de Ubuntu y de Debian para sistemas ARM, pero las guías de desarrollo con las que cuentan señalan que no son versiones estables y pueden presentarse bugs o incompatibilidad de software, por este motivo se utilizara la versión estable de Raspberry Pi OS de 32bits.

Anteriormente era necesario descargarse el sistema operativo en formato *.img* y cargarlo en la memoria microSD, pero desde el año 2020 oficialmente Raspberry cuenta con su propio software llamado *Raspberry Pi Imager* que se encuentra actualmente en su versión v1.3, permite formatear la tarjeta microSD y cargar el sistema operativo sin necesidad de descargarlo. En el caso que necesitemos cargar un sistema operativo diferente y que no aparece en las opciones podemos seleccionarlo desde nuestro propio disco.

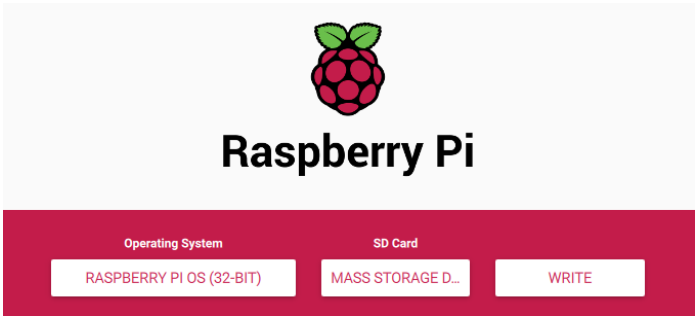


Figura 4-3: Ventana principal del software Raspberry Pi Imager.

Realizado por: Álvarez, Andrés, 2022.

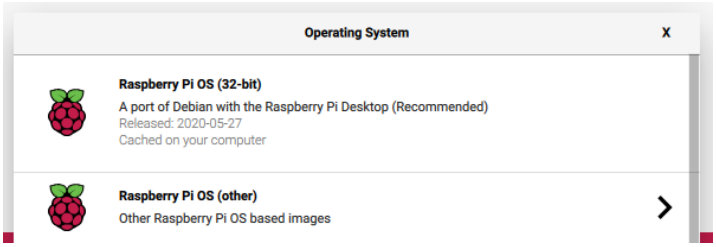


Figura 5-3: Sistemas Operativos disponibles para instalar.

Realizado por: Álvarez, Andrés, 2022.

Una vez insertada la tarjeta microSD y encendida la placa Raspberry Pi será necesario realizar varios ajustes dentro del sistema. Entre los principales ajustes que se debe realizar, el primero es el colocar una contraseña para el super usuario (root), además se recomienda al usuario por defecto llamado *pi* asignarle una clave y activar el modo restrictivo para todos los casos ya que por defecto el sistema no solicita clave al ingresar comandos que modifican el sistema.

Figura 6-3: Modificación del password del usuario root.

Realizado por: Álvarez, Andrés, 2022.

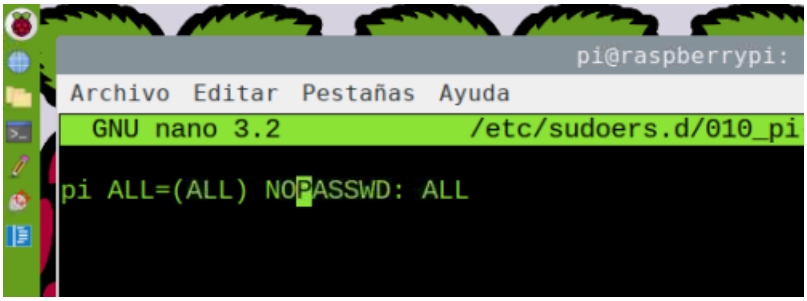


Figura 7-3: Configuración por defecto del usuario Pi.

Realizado por: Álvarez, Andrés, 2022.

Una vez realizados los pasos anteriores estamos en la posibilidad de conectarnos a una red wifi y actualizar el sistema mediante *apt-get update && apt-get upgrade –y*.

También será necesario instalar el software necesario para que la tarjeta Raspberry Pi trabaje como Gateway de los dispositivos IoT, para el proyecto en particular se agregará una antena Wi-Fi USB que servirá más adelante para que los dispositivos IoT la usen con punto de acceso.

Por defecto el sistema operativo no incorpora los divers necesarios para la antena D-link 131 a pesar de que es compatible con el sistema, por este motivo será necesario cargar el driver correspondiente. Es necesario descargar los archivos binarios alojados en GitHub, correspondientes a la versión de la antena, una vez obtenidos los binarios se debe modificar el archivo "Makefile" ya que por defecto está configurado para que ejecuten los archivos correspondientes en computadoras de 32 bits, una vez realizado el cambio se ejecuta el archivo.

```
120  ######### Notify SDIO Host Keep Power During Syspend
121  CONFIG_RTW_SDIO_PM_KEEP_POWER = y
122  ####################### MP HW TX MODE FOR VHT ########
123  CONFIG_MP_VHT_HW_TX_MODE = n
124  ####################### Platform Related #############
125  CONFIG_PLATFORM_ARM_RPI = n
126  CONFIG_PLATFORM_ARM_AARCH64 = n
127  CONFIG_PLATFORM_I386_PC = y
128  CONFIG_PLATFORM_ANDROID_X86 = n
129  CONFIG_PLATFORM_ANDROID_INTEL_X86 = n
130  CONFIG_PLATFORM_JB_X86 = n
131  CONFIG_PLATFORM_ARM_S3C2K4 = n
132  CONFIG_PLATFORM_ARM_PXA2XX = n
133  CONFIG_PLATFORM_ARM_S3C6K4 = n
134  CONFIG_PLATFORM_MIPS_RMI = n
135  CONFIG_PLATFORM_RTD2880B = n
136  CONFIG_PLATFORM_MIPS_AR9132 = n
137  CONFIG_PLATFORM_RTK_DMP = n
138  CONFIG_PLATFORM_MIPS_PLM = n
139  CONFIG_PLATFORM_MSTAR389 = n
140  CONFIG_PLATFORM_MT53XX = n
```

Figura 8-3: Arquitectura de la antena WiFi D-Link 131.

Realizado por: Álvarez, Andrés, 2022.

Si ejecutamos el comando *ifconfig* podremos apreciar que ahora se cuentan dos tarjetas WiFi, wlan0 corresponde a la tarjeta interna de Raspberry y wlan1 es la antena usb D-link 131.

```
wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
       ether 54:2a:a2:5c:1e:1d  txqueuelen 1000  (Ethernet)
       RX packets 0  bytes 0 (0.0 B)
       RX errors 0  dropped 0  overruns 0  frame 0
       TX packets 0  bytes 0 (0.0 B)
       TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
       inet 192.168.1.15  netmask 255.255.255.224  broadcast 192.168.1.31
       inet6 fe80::5313:b4c8:a21f:97e2  prefixlen 64  scopeid 0x20<link>
       inet6 2800:370:d9:5040:fce6:5d79:17ec:2c0e  prefixlen 64  scopeid 0x0<
```

Figura 9-3: Verificación del estado de la antena WiFi D-Link 131

Realizado por: Álvarez, Andrés, 2022.

3.1.2.3. Instalación de Webthings Gateway

Por otra parte, es necesario agregar el paquete de WebThing Gateway al sistema para que pueda cumplir con la función de Gateway. Existen tres opciones disponibles en la página de WebThings.io, entre ellas se encuentra la opción destinada a placas Raspberry Pi, mediante la cual todo el sistema se convierte en un servidor remoto y el cual se manipula mediante un navegador web, la siguiente opción destinada a sistemas GNU/Linux en la que encontramos archivos diseñados para las arquitecturas más utilizadas, así como los recursos necesarios para ser compilados. La última opción está destinada a la utilización de contenedores de Docker destinada a servicios de cloud computing.

Figura 10-3: Opciones disponibles para instalar WebThings Gateway.

Realizado por: Álvarez, Andrés, 2022.

Es necesario instalar todas las dependencias necesarias para que el sistema pueda ejecutarse correctamente.

```
 GNU nano 4.8                          New Buffer                          Modified
1 sudo apt install curl
2 sudo apt install libboost-python-dev libboost-thread-dev libbluetooth-dev libglib2.0-dev
3 sudo apt install libusb-1.0-0-dev libudev-dev
4 sudo apt install libffi-dev
5
6
```

Figura 11-3: Comandos para instalar las dependencias de WebThings Gateway.

Realizado por: Álvarez, Andrés, 2022.

Después de agregar todas las dependencias se procede a clonar el repositorio para seguidamente ejecutarlo.

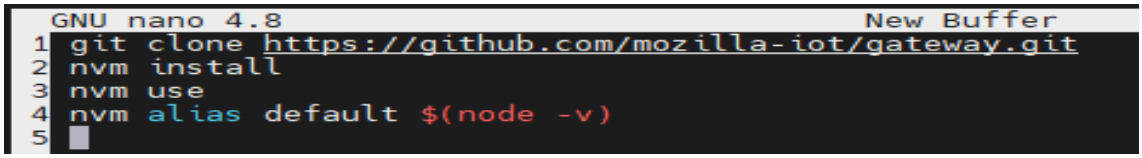


Figura 12-3: Comandos para clonar el repositorio y compilación.

Realizado por: Álvarez, Andrés, 2022.

Se puede verificar la versión además de que al ejecutar el sistema podemos acceder a su ventana de inicio mediante un navegador como se puede observar en el anexo C.

3.1.2.4. Instalación del sistema de monitoreo

El sistema que se instalará para realizar el monitoreo del tráfico de la red será LibreNMS ya que a diferencia de otros sistemas este es compatible con la arquitectura de Raspberry Pi por su soporte a chips ARM, además presenta las características que se requieren para el proyecto como son presentar el tráfico de red en tiempo real, ver las características y rendimiento de los dispositivos además de crear gráficas históricas del tráfico.

En primer lugar, es necesario instalar las librerías y dependencias necesarias para que el software pueda funcionar ya que la plataforma trabaja con un servidor Apache el cual está conectado a una base de datos que funciona mediante MariaDB y además requiere las librerías de PHP v7.3 o más actuales, por este motivo es necesario ejecutar los comandos para instalar todas estas dependencias.

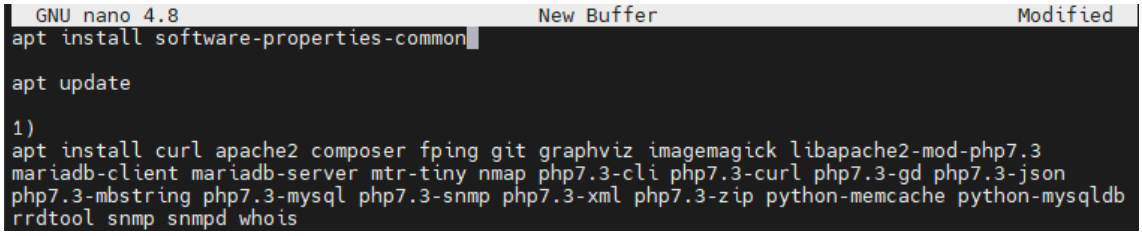


Figura 13-3: Comando de instalación de dependencias y software necesario.

Realizado por: Álvarez, Andrés, 2022.

A continuación, se procede a crear los directorios y usuarios, así como sus grupos necesarios ya que de esta forma podremos administrar los permisos y recursos del sistema que utilizara el software.

```
2)
/usr/sbin/useradd librenms -d /opt/librenms -M -r

3)
usermod -a -G librenms www-data
```

Figura 14-3: Configuración del directorio del usuario para librenms.

Realizado por: Álvarez, Andrés, 2022.

En este punto ya podemos copiar los recursos existentes en Github mediante los cuales se procederá a la instalación, antes de ejecutar los archivos correspondientes para la instalación se debe configurar los permisos adecuados.

```
4)
cd /opt/
git clone https://github.com/librenms/librenms.git

5)
chown -R librenms:librenms /opt/librenms

6)
chmod 770 /opt/librenms

7)
apt-get install acl

8)
setfacl -d -m g::rwx /opt/librenms/rrd /opt/librenms/logs /opt/librenms/bootstrap/cache/
 /opt/librenms/storage/

9)
setfacl -R -m g::rwx /opt/librenms/rrd /opt/librenms/logs /opt/librenms/bootstrap/cache
 /opt/librenms/storage/
```

Figura 15-3: Obtención y configuración de los recursos del software librenms.

Realizado por: Álvarez, Andrés, 2022.

Una vez instalados todos los recursos y verificada la instalación se realiza la configuración de la base de datos, en la que se creará una nueva base de datos asignándole un usuario y una contraseña además de definir los permisos adecuados.

```
##chown -R librenms:librenms /opt/librenms
##chmod 770 /opt/librenms

10)
su - librenms

11)
./scripts/composer_wrapper.php install --no-dev
exit

12)
systemctl restart mysql

13)
mysql -uroot -p      ##password
>CREATE DATABASE librenms CHARACTER SET utf8 COLLATE utf8_unicode_ci;
>CREATE USER 'librenms'@'localhost' IDENTIFIED BY '##password'; ##password
>GRANT ALL PRIVILEGES ON librenms.* TO 'librenms'@'localhost';
>FLUSH PRIVILEGES;
>exit
```

Figura 16-3: Configuración de la base de datos.

Realizado por: Álvarez, Andrés, 2022.

La siguiente configuración necesaria que se debe hacer involucra al servidor apache, para que este pueda conectarse con la base de datos se deben definir los parámetros necesarios como la zona horaria, puerto a utilizarse, etc.

```
14)
nano /etc/mysql/mariadb.conf.d/50-server.cnf
*****************************
[mysqld]
innodb_file_per_table=1
lower_case_table_names=0
*****************************
(guardar configuracion)

15)
systemctl restart mysql


16)
nano /etc/php/7.3/apache2/php.ini
*****************************
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = America/Guayaquil # <==depende la ubicacion
*****************************
(guardar configuracion)

17)
nano /etc/php/7.3/cli/php.ini
*****************************
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = America/Guayaquil
*****************************
```

Figura 17-3: Ajuste de los archivos de configuración.

Realizado por: Álvarez, Andrés, 2022.

Para terminar la configuración ajustamos los archivos *.conf* de los servicios para que se ajuste a nuestras necesidades, lo que garantiza la comunicación con el resto de servicios, tales como la base de datos y el contenido web.

```
18)
a2enmod php7.3
a2dismod mpm_event
a2enmod mpm_prefork

19)
nano /etc/apache2/sites-available/librenms.conf
*****************************
<VirtualHost *:80>
DocumentRoot /opt/librenms/html/
ServerName debian-librenms.FIE-Tesis.com

AllowEncodedSlashes NoDecode
<Directory "/opt/librenms/html/">
Require all granted
AllowOverride All
Options FollowSymLinks MultiViews
</Directory>
</VirtualHost>
*****************************
(guardar configuracion)

20)
a2dissite 000-default
a2ensite librenms.conf
a2enmod rewrite
systemctl restart apache2

21)
cp /opt/librenms/snmpd.conf.example /etc/snmp/snmpd.conf

22)
nano /etc/snmp/snmpd.conf
```

Figura 18-3: Ajuste de los archivos de configuración.

Realizado por: Álvarez, Andrés, 2022.

.

```
nano /etc/snmp/snmpd.conf
*****************************
#Change RANDOMSTRINGGOESHERE to your preferred SNMP community string
com2sec readonly  default         public
*****************************
(guardar configuracion)

23)
curl -o /usr/bin/distro https://raw.githubusercontent.com/librenms/librenms-agent/master/snmp>

24)
chmod +x /usr/bin/distro

25)
systemctl restart snmpd

26)
cp /opt/librenms/librenms.nonroot.cron /etc/cron.d/librenms
```

Figura 19-3: Ajuste de los archivos de configuración.

Realizado por: Álvarez, Andrés, 2022.

Una vez terminada la configuración deberemos dirigirnos hacia un navegador web para poder ingresar a la interfaz del sistema de monitoreo. Si la instalación se ha realizado correctamente deberemos visualizar en la pestaña del navegador la verificación correspondiente

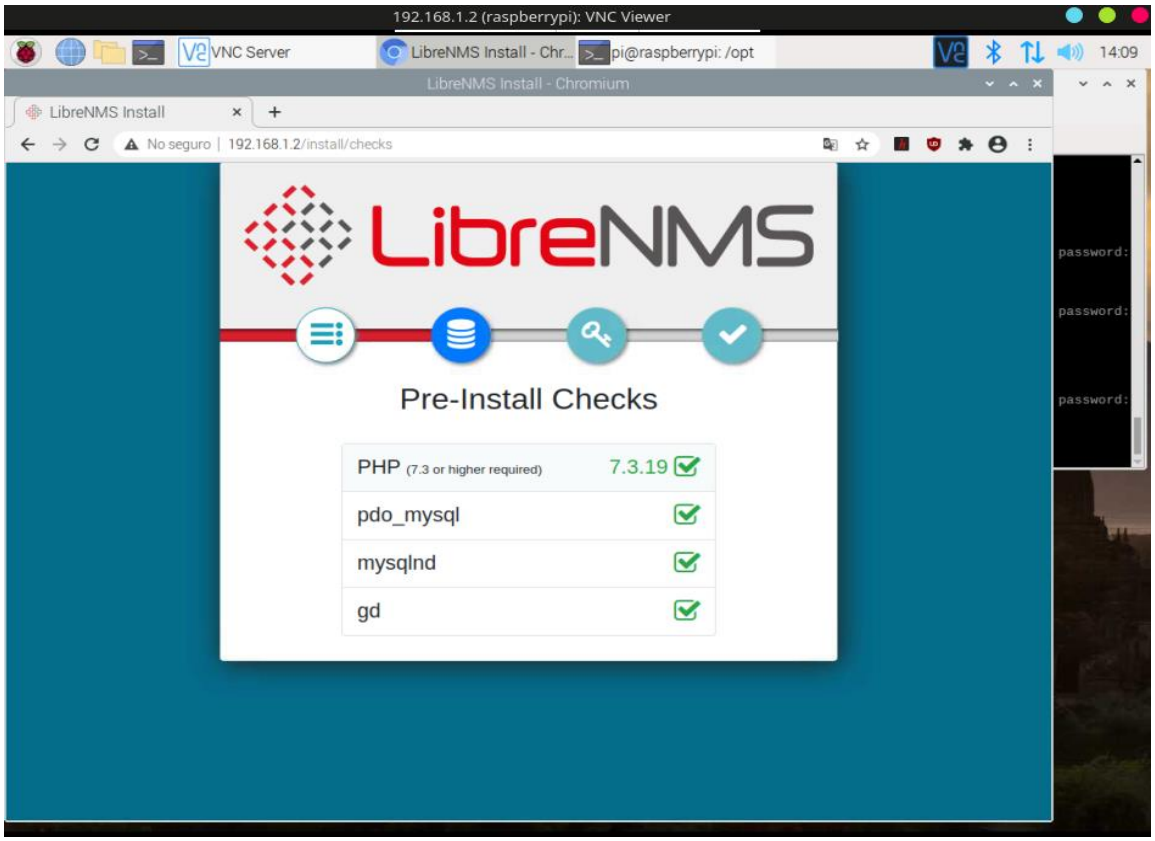


Figura 20-3: Página de verificación de todos los servicios instalados de Librenms.

Realizado por: Álvarez, Andrés, 2022.

3.1.2.5. Demostración de la instalación de la herramienta Snort

A modo de demostración y comprobación de que la herramienta Snort es compatible con la arquitectura ARM de Raspberry Pi se procedió a la instalación siguiendo los siguientes pasos.
En primer lugar, es necesario instalar los recursos necesarios de la herramienta. Dependiendo del sistema en el que se va a instalar habrá que ejecutar el comando correspondiente para instalar la herramienta y las dependencias que se necesiten.

```
1--sudo apt-get install snort
2--sudo dpkg-reconfigure snort
service snort restart


##dependencias necesarias
tar xvf tcpdump-4.9.3.tar.gz
3--(sudo apt-get install tcpdump)
./configure
4--sudo apt-get install libpcap-dev
make
sudo make install
5--sudo apt-get install libpcre3 libpcre3-dev

6--sudo apt-get install -y dnet-common
7--dpkg-reconfigure dnet-common
```

Figura 21-3: Instalación de dependencias necesarias.

Realizado por: Álvarez, Andrés, 2022.

Para verificar que la herramienta se ha instalado de forma adecuada se puede revisar los archivos de configuración y comprobar que no existe ningún problema.

```
##configuracion
/etc/snort/snort.conf

##ejecutar todas las acciones desde /etc/snort/

##Validad configuracion
$sudo snort -T -c /etc/snort/snort.conf

##directorio de reglas
/etc/snort/rules/sites.rules
```

Figura 22-3: Instalación del software Snort.

Realizado por: Álvarez, Andrés, 2022.

3.1.2.6. Instalación de la herramienta Suricata

La instalación de Suricata presenta varias alternativas, la primera es descargando el código fuente directamente de su página oficial y compilarlo nosotros mismos, la segunda alternativa es instalarlo desde los repositorios de Debian, el inconveniente de esta opción es que tiene una actualización más lenta, pero nos garantiza contar con la versión estable del software. Para este

proyecto se hará uso del repositorio de Debian y se ejecutará todas las operaciones que se necesitan para instalar Suricata de una forma correcta.

```
root@raspAP:/home/raspAP# apt search suricata
Ordenando... Hecho
Buscar en todo el texto... Hecho
fever/oldstable 1.0.5-2 armhf
  fast, extensible, versatile event router for Suricata's EVE-JSON format

golang-github-jasonish-go-idsrules-dev/oldstable 0.0~git20170503.0.c646b91-2 all
  Go IDS rule parser

libhtp-dev/oldstable 1:0.5.30-1 armhf
  HTTP normalizer and parser library (devel)

libhtp2/oldstable 1:0.5.30-1 armhf
  HTTP normalizer and parser library

oslinkwatch/oldstable 1.0-2 armhf
  automatic maintenance of Suricata monitoring interfaces

suricata/oldstable 1:4.1.2-2+deb10u1 armhf
  Next Generation Intrusion Detection and Prevention Tool

suricata-oinkmaster/oldstable 1:4.1.2-2+deb10u1 all
  Integration package between suricata and oinkmaster

suricata-update/oldstable 1.0.3-2 armhf
  tool for updating Suricata rules
```

Figura 23-3: Verificación del software Suricata en los repositorios Debian.

Realizado por: Álvarez, Andrés, 2022.

Una vez instalado el software se crearán los archivos correspondientes de configuracion y una vez ejecutada la herramienta se puede combrobar que todos los servicios internos requeridos se encuentran activos, principalmente para nuestro caso los servicios de IDS/IPS. Es importante verificar que se encuentren disponibles los servicios de IDS/IPS ya que si solo instalamos la versión mínima de la herramienta no estarán presentes y no se podrá hacer uso de estos.

```
raspAP@ras... ✖ | raspAP@ras... ✖ | raspAP@ras... ✖ | raspAP@ras... ✖ | * raspAP@r... ✖

● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2021-12-22 00:31:48 -05; 8s ago
     Docs: man:suricata(8)
           man:suricatasc(8)
           https://suricata-ids.org/docs/
  Process: 27405 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid
 Main PID: 27406 (Suricata-Main)
    Tasks: 10 (limit: 4915)
   CGroup: /system.slice/suricata.service
           └─27406 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid

dic 22 00:31:47 raspAP systemd[1]: Starting Suricata IDS/IDP daemon...
dic 22 00:31:48 raspAP suricata[27405]: 22/12/2021 -- 00:31:48 - <Info> - Configuration node 'rule-files' redefined.
dic 22 00:31:48 raspAP suricata[27405]: 22/12/2021 -- 00:31:48 - <Notice> - This is Suricata version 4.1.2 RELEASE
dic 22 00:31:48 raspAP systemd[1]: Started Suricata IDS/IDP daemon.
```

Figura 24-3: Verificación del estado de la instalación del software suricata.

Realizado por: Álvarez, Andrés, 2022.

3.1.3. Definición de la taxonomía

Como se definió en el capítulo anterior al aplicar la Taxonomía es necesario seccionar los ataques para identificar sus características y la forma en que estos actúan dentro de la red. Si combinamos estos conocimientos con lo estudiado dentro de la seguridad informática se obtiene como resultado los puntos que deberán ser analizados en búsqueda de fallos de seguridad. Por lo tanto, para mejorar nuestra capacidad de prevención y acción frente a los ataques informáticos, el estudio se divide en los tres aspectos importantes de la seguridad informática, vulnerabilidades, amenazas, y riesgos.

3.1.3.1. Identificación de vulnerabilidades

Las vulnerabilidades de una red pueden ser descubiertas mediante sus métricas, aunque aparenten ser acciones inofensivas, ya que permiten reconocer la red, y recopilar información de sus dispositivos, siendo esta una técnica muy simple de ejecutar. Comandos como *Tracert* (en Windows) o *Traceroute* (en GNU/Linux), p*ing*, y herramientas como *NMAP* permiten al atacante conocer la distribución de los dispositivos dentro de la red mediante sus direcciones IP y de esta forma crear un mapa de la red, lo que facilita la búsqueda de puntos vulnerables en la red.

En la Raspberry Pi que actúa como Gateway al ejecutarse estos comandos obtenemos el valor TTL (time to live) el cual será igual a 64 por tratarse de un sistema GNU/Linux, en otros casos este valor puede variar dependiendo de los saltos que existan en la red y aunque no es un valor completamente confiable y nos permite tener un primer contacto con las características de la red. La opción *-c* nos permite realizar un conteo de paquetes y enviar solo una cantidad definida.

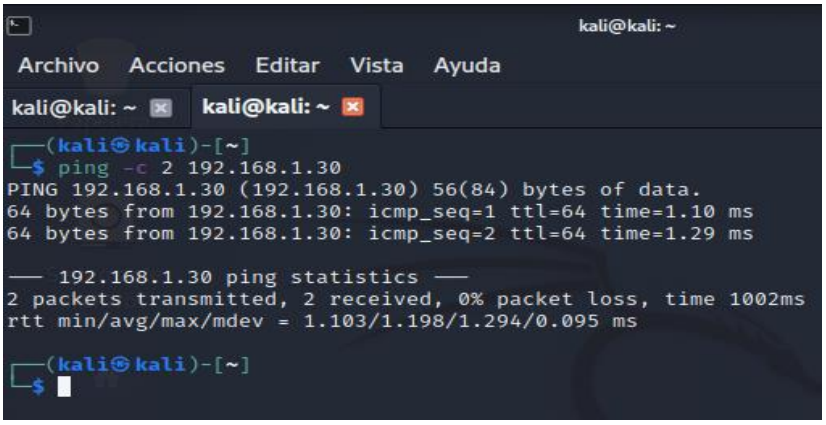


Figura 25-3: Reconocimiento inicial de la red.

Realizado por: Álvarez, Andrés, 2022.

Como se puede observar, en este caso particular el valor de TTL es de 64 ya que los sistemas operativos con los que se trabaja Raspberry Pi están diseñados bajo el sistema GNU/Linux y no existen dispositivos intermedios que disminuyan el valor de TTL. Una vez definido el sistema operativo con el que se está trabajando procedemos a examinar puertos abiertos mediante la ejecución de NMAP y haciendo uso de la opción *-Pn*

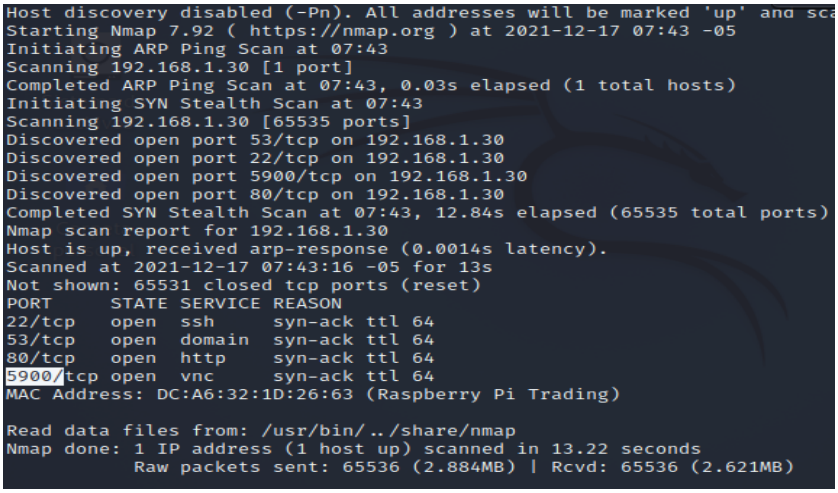


Figura 26-3: Reconocimiento avanzado de la red mediante NMAP.

Realizado por: Álvarez, Andrés, 2022.

Como se puede observar existen 4 puertos abiertos de los cuales se puede determinar bajo que servicios se encuentran ejecutando y de esta forma estructurar los ataques que serían más efectivos para esta red.

Si deseamos tener la información más detallada de los servicios que se encuentran utilizando los puertos abiertos mediante el comando de *nmap sCV -p* se puede obtener la siguiente información:

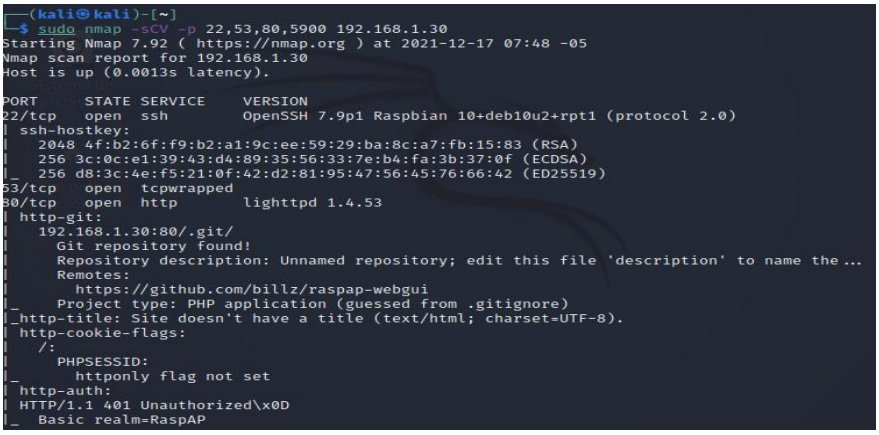


Figura 27-3: Reconocimiento avanzado de la red mediante NMAP.

Realizado por: Álvarez, Andrés, 2022.

Las puertas traseras son el resultado de descuidos surgidos al dejar un puerto abierto y permiten que se ejecuten acciones remotas para comprometer al dispositivo, en este caso puede estar abiertos debido a los servicios que se están ejecutan para mantener la comunicación activa.

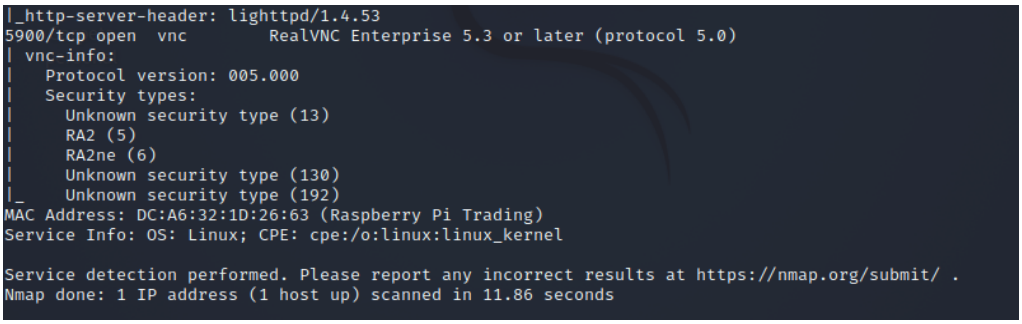


Figura 28-3: Descubrimiento de puertos activos en la red.

Realizado por: Álvarez, Andrés, 2022.

Otro de los problemas que se puede detectar en la Raspberry Pi Gateway es que la página web del administrador permite la inyección de scripts, así como también inyección de código malicioso contra las bases de datos. Esto se determina haciendo uso de la herramienta Nikto la cual necesita definir la dirección de la página mediante la opción *-h* y para guardar el informe se aplica la opción *-o*.

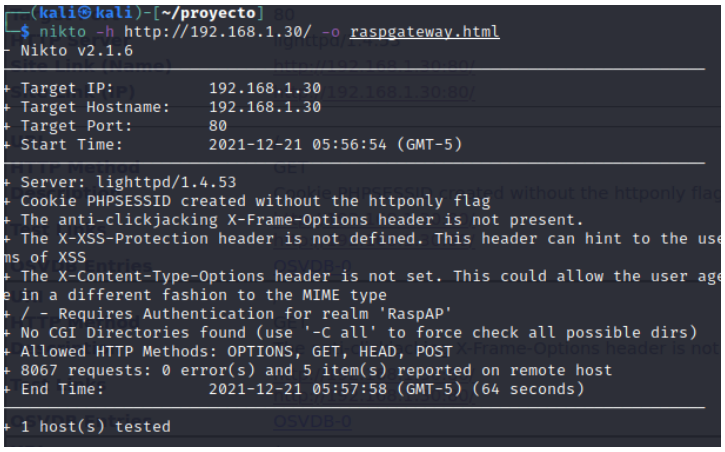


Figura 29-3: Análisis de la red mediante la herramienta Nikto.

Realizado por: Álvarez, Andrés, 2022.

Realizando un escaneo en el servidor de Webthings bajo las mismas condiciones expuestas anteriormente se puede detectar que de igual forma nos encontramos delante de un sistema operativo basado en Linux, el cual no presenta ninguna restricción al escanear su estado.

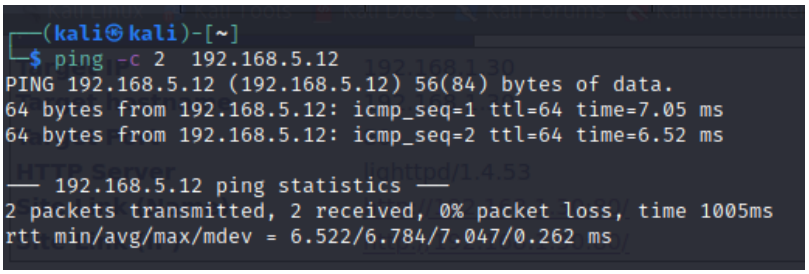


Figura 30-3. Reconocimiento inicial de la subred.

Realizado por: Álvarez, Andrés, 2022.

Como se puede observar en la imagen Webthings es uno de los puntos más vulnerables dentro de la red ya que utiliza muchos más puertos y en gran parte los servicios que se están ejecutando carecen de sistemas de protección.

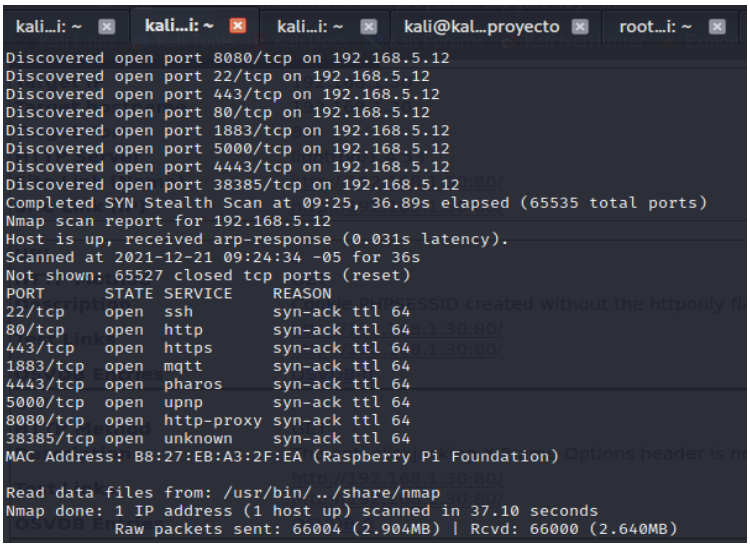


Figura 31-3: Reconocimiento de puertos activos en la subred.

Realizado por: Álvarez, Andrés, 2022.

Además, se puede observar que todos los protocolos utilizados por los dispositivos IoT con Webthings se encuentran expuestos y son fáciles de identificar sin realizar mucho esfuerzo lo que igualmente debilita la red.

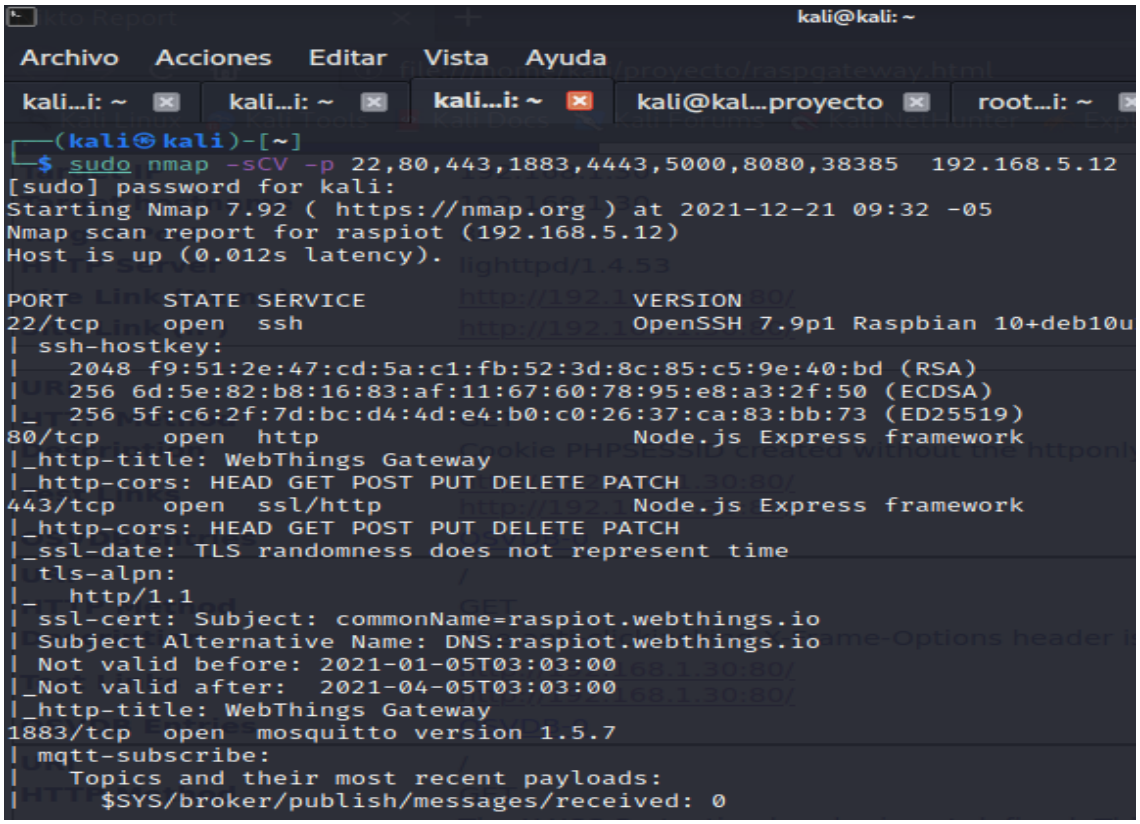


Figura 32-3: Reconocimiento avanzado de los puertos activos en la red.

Realizado por: Andrés Álvarez; 2022.

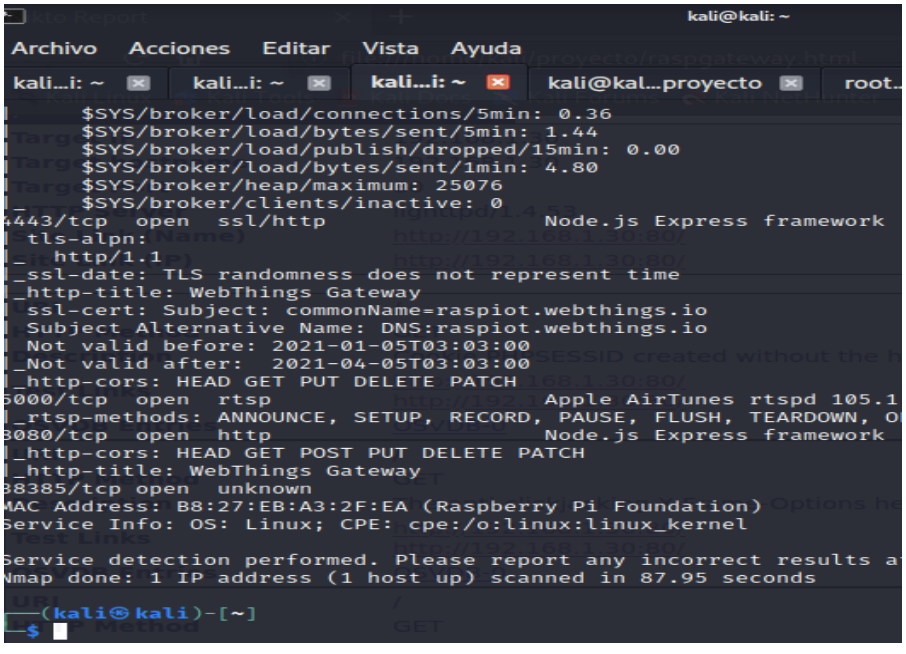


Figura 33-3: Reconocimiento avanzado de los puertos activos en la red.

Realizado por: Álvarez, Andrés, 2022.

Cuando se utiliza la herramienta Nikto se puede observar que el dispositivo que ejecuta Webthings tiene vulnerabilidades similares a las del Gateway y un aspecto a destacar es que al realizar los ataques hacia los servidores el tráfico es mucho más elevado con respecto al tráfico promedio generados por los dispositivos IoT tal como se puede apreciar en el anexo F. Las contraseñas son un grave problema de vulnerabilidad debido a que si no son modificadas presentan un punto extremadamente sensible para la red permitiendo que el atacante acceda rápidamente a un dispositivo y lo manipule.

Un claro ejemplo de esto es el usuario por defecto que viene en el sistema operativo de Raspberry Pi, el cual es "Raspberry" y con el suficiente número de intentos es posible acceder al sistema y aún peor tener el control total del dispositivo de forma remota, por ejemplo, si se realiza un ataque hacia SSH.

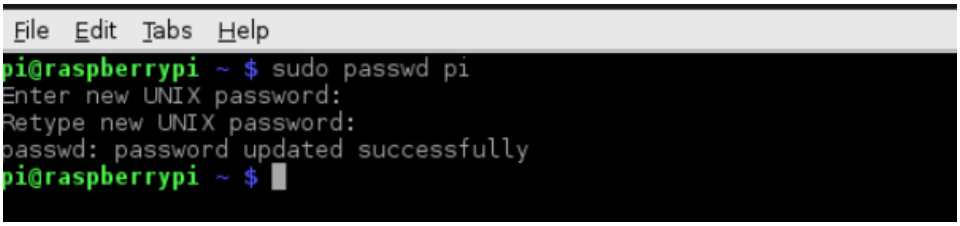


Figura 34-3: Verificación de la seguridad del password de Raspberry Pi.

Realizado por: Álvarez, Andrés, 2022.

También se puede observar que al momento de levantar el servicio en el Gateway por defecto la clave es genérica a pesar de que la encriptación es WPA2, la limitación del dispositivo impide ejecutar una encriptación WPA3 y como alternativa nos ofrece WPA+WPA2 lo que sigue siendo un punto débil si un ataque de fuerza bruta o con diccionarios se ejecuta hacia la clave del Wi-Fi.

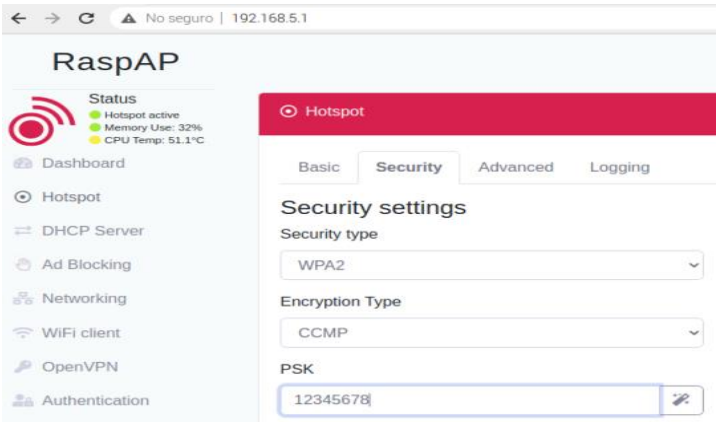


Figura 35-3: Verificación de la seguridad del password del Access Point.

Realizado por: Álvarez, Andrés, 2022.

Dentro de los dispositivos IoT los métodos de encriptación son mínimos y el acceso a la información de la red con la que entabla una comunicación puede quedar expuesta, este aspecto

suele ser unos de lo más comunes dentro de los dispositivos IoT , ya que se intenta reducir latencias en la comunicación transmitiendo información en texto plano y evidentemente reduce la seguridad; no todos los dispositivos ofrecen la capacidad de comunicaciones blindadas debido a que es necesario procesadores más potentes capaces de realizar grandes operaciones en pocos segundos, todos estos factores se resumen en la capacidad que ofrecen a los atacantes de leer los archivos de configuración.

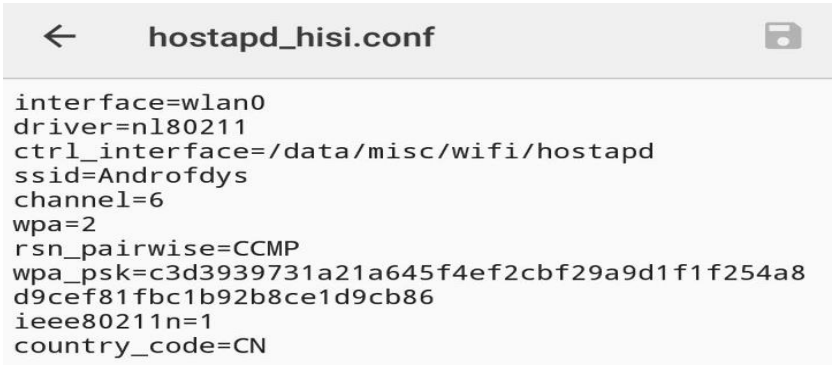


Figura 36-3: Archivo de configuración utilizada por los dispositivos IoT.

Realizado por: Álvarez, Andrés, 2022.

En la figura 37-2 es posible definir la interface por la cual se está conectando el dispositivo, el canal y el tipo de encriptación que cuenta la clave, en este caso se cuenta con un método de encriptar muy frecuente en redes Wi-Fi, pero no es uno de los métodos más robustos y si la clave no cuenta con las características adecuadas puede ser vulnerada. En otros casos directamente todo el código requerido por el dispositivo debe ser cargado en texto plano para que este pueda funcionar y en consecuencia no se encripta, dejando en evidencia no solo la contraseña sino también otra información del dispositivo.

Figura 37-3: Código internet de los dispositivos IoT.

Realizado por: Álvarez, Andrés, 2022.

Mediante este análisis se consigue definir los puntos débiles con los que cuenta la red IoT, y estos conocimientos adquiridos permite seleccionar las amenazas más efectivas hacia la red.

3.1.3.2. Identificación de amenazas

Como se mencionó anteriormente las amenazas son todas las acciones existentes que realiza un atacante para perjudicarnos, por este motivo se ha seleccionado los ataques SQL injection, inyección de scripts denominado XSS y ataque de denegación de servicio DoS. Es fundamental dar mayor atención a estos ataques debido a que afectan directamente a una de las propiedades más importantes de una red IoT que es la disponibilidad.

El objetivo de los dos primeros ataques es similar, lo que se intenta es inyectar secciones de código que permitan hacerse con el control de la información, esto es posible ya que los atacantes pueden inyectar instrucciones en el caso de XSS o consultas si se trata de una base de datos y de esta forma escalar en privilegios dentro de la red hasta el punto de confundir al resto de dispositivos conectados

Por otra un ataque de DoS o técnicas más avanzadas DDoS intentan congestionar las vías de comunicación de la red, esto es posible ya que toda red tiene un límite de transferencia o recepción, de forma que si aumentamos carga de transferencia al tráfico ya existente, la red difícilmente será capaz de gestionar toda la información y llegado a un punto simplemente dejará de responder. Para una red IoT sufrir ataques como estos perjudica directamente a los servicios, impidiendo que se realicen las acciones esenciales de los dispositivos IoT.

3.1.3.3. Identificación de riesgos

Entre los ataques expuestos anteriormente, los más perjudiciales son los inyectan códigos maliciosos se corre el riesgo de perder la configuración de los dispositivos IoT, lo que produciría tener que volver a configurar los dispositivos y ajustar los cronogramas de trabajo de los dispositivos, un claro ejemplo son las reglas que se pueden crear en Webthings, permitiendo que los dispositivos reaccionen a los eventos producidos por otros dispositivos. Uno de los casos son las reglas para que cuando el pulsador de una ventana sea presionado se active una alarma.

Figura 38-3: Regla definida para el control de una alarma y un pulsador.

Realizado por: Álvarez, Andrés, 2022.

Además, la gran parte de dispositivos contiene certificados firmados donde se almacena información del dispositivo, en otras palabras, cuando un dispositivo inicia comunicación con otro intercambia sus certificados para que de esta forma se conozcan sus características y la transferencia de datos se adecue dependiendo del dispositivo, así como las API utilizadas ampliamente por la plataforma WebThings, y si no son actualizados de forma continua, un atacante puede falsificar los certificados y engañar a los dispositivos IoT con conexiones fraudulentas.

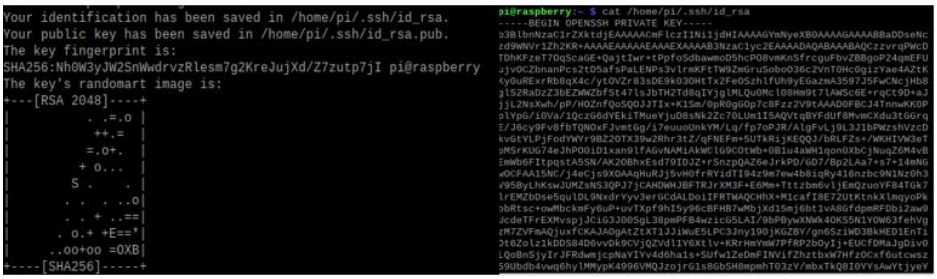


Figura 39-3: Certificado y clave para una conexión SSH.

Realizado por: Álvarez, Andrés, 2022.

En la imagen vemos el estado actual del certificado de la comunicación SSH de Raspberry Pi, un atacante podría clonar el certificado y asignarlo a un dispositivo diferente perjudicando la comunicación en la red, de forma que los dispositivos se comunicarán con un clon sin detectar la diferencia entre el dispositivo original y el falso. También cabe mencionar que cualquier información a la que pueda tener acceso un atacante puede ser utilizado para violar nuestra privacidad y servir de señuelo para extraer más información personal.

3.1.4. Aplicación de la Taxonomía

Una vez realizado el estudio de las vulnerabilidades, amenazas y riesgos existentes en la red el IPS que se plantea como solución debe cumplir las siguientes características:

- El IPS será configurado a nivel de red como dispositivo de frontera de forma que controlará el tráfico que entre y salga del dispositivo Gateway.
- Deberá generar alertas cuando se detecte alguna anomalía en el tráfico red.
- Todas las acciones que sean identificadas como anomalías deberán ser almacenado mediante logs para tener un historial del comportamiento de la red.
- Se implementará las reglas necesarias para mitigar todos los problemas detectados anteriormente, y se ajustará a las características de la red para tener un mejor rendimiento.

3.1.4.1. Configuración de la Herramienta IPS

Antes de implementar cualquier solución es fundamental comprobar el estado de la herramienta IPS ya que de esta forma descartaremos fallos producidos por una incorrecta configuración. Para verificar que el sistema se encuentre ejecutándose en primer lugar se debe ejecutar el comando *systemctl start suricata.services* en modo de super usuario, de forma que todos los servicios entren en funcionamiento.

Todos los archivos de configuración los encontramos en los directorios donde se encuentra instalado Suricata, esto podemos realizarlo mediante el comando *whereis suricata*; entre los archivos de configuración que se deben revisar se encuentra el suricata.yaml y es fundamental que no tenga ningún tipo de error además de que reconozca la ruta donde se encuentran configuradas las reglas.

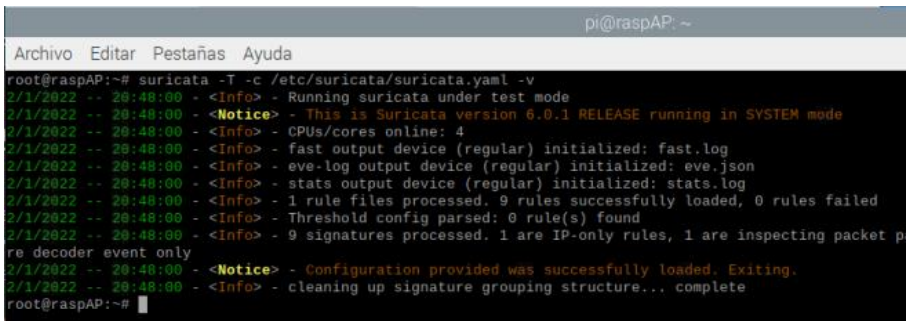


Figura 40-3: Verificación de los archivos de configuración de Suricata.

Realizado por: Álvarez, Andrés, 2022.

Al ejecutar el comando *iptables -I FORWARD -j NFQUEUE* ordenamos que todo el tráfico que atraviesa la red se dirija hacia la herramienta IPS a través del módulo NFQ, podemos verificar que el tráfico de la red se dirige hacia el IPS de forma que lo podemos analizar con la herramienta Suricata. Existen 2 opciones adicionales que se pueden configurar, la primera es cuando se transfiere todo el tráfico que ingresa al puerto del servidor y la segunda opción es que se redireccione todo el tráfico que sale a través del puerto del servidor.

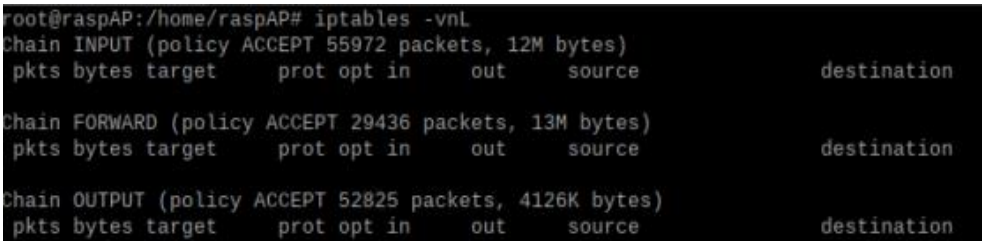


Figura 41-3: Verificación del estado de IPTABLES.

Realizado por: Álvarez, Andrés, 2022.

3.1.4.2. Tratamiento de los ataques mediante la implementación de reglas en el IPS

Antes de proceder con la creación de nuestras reglas se debe crear el archivo en donde serán almacenadas todas las reglas, para este fin el archivo debe tener la extensión *.rules* y debe ser agregada su ruta en el archivo de configuración.

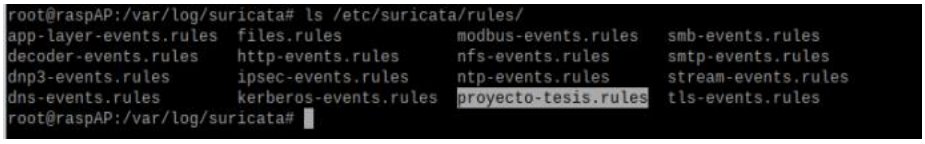


Figura 42-3: Creación del archivo para las reglas de Suricata.

Realizado por: Álvarez, Andrés, 2022.

- **Ataques de tipo virus, malware, troyanos.**

Las pruebas y antecedentes determinan que el riesgo que existe ante un ataque de software malicioso es bajo debido a que el atacante debería en primer lugar acceder a la red y buscar un dispositivo que ejecute el software malicioso. En el caso de una red IoT lo que se intenta es tomar el control de los dispositivos IoT, ya que de esta forma se puede camuflar las actividades del atacante y falsificar su identidad.

Con este objetivo el software más utilizado es el llamado Mirai el cual está clasificado como malware y su función principal es infectar dispositivos IoT para ejecutar un ataque DDoS, la

forma en que este software es implantado puede ocupar diferentes vías ya sea encapsulado en forma de troyano o encapsulado en otro tipo de archivos por ejemplo un certificado. Para el caso de estudio se analizará el tráfico que atraviesa el dispositivo Gateway ya que como se dijo anteriormente lo que el ataque intentara es infectar todos los dispositivos disponibles en la red. Esto se realiza de la siguiente manera:

Figura 43-3: Traspaso del tráfico de red desde IPTABLES hacia NFQ de Suricata.
Realizado por: Álvarez, Andrés, 2022.

Como se puede observar se ha configurado el tráfico para que se dirija hacia el módulo NFQ el cual es el encargado de tomar el tráfico de la red y enviarlo hacia el sistema IPS para que sea analizado. Las opciones de entrada y salida de tráfico se mantienen con la configuración por defecto. Las reglas que han sido definidas hacen uso de la capacidad de la herramienta Suricata para conectarse con repositorios los cuales guardan características del malware Mirai de forma que el sistema identifique las diferentes variantes que pueden camuflar un malware de este tipo. Las reglas cubren 3 aspectos fundamentales.

La primera característica es intentar identificar las palabras contenidas dentro del software malicio mediante la opción *Depth* este argumento lo que nos permite es analizar los bytes que componen un payload y buscar patrones de código hexadecimal que se ajusten a la estructura del malware, además de que se analizan las extensiones de los archivos, haciendo que coincidan los patrones conocidos con las secciones de información analizada mediante *endswith*, este argumento ajusta el texto de forma que las palabras que contienen una extensión coincidan con los patrones definidos dentro de la regla, además de que en algunos casos se trata de evadir los sistemas utilizando mayúsculas o minúsculas, para ignorar esta características utilizamos *nocase*, así como también si existe el rastro de una conexión P2P con otro equipo que es una característica presente en un ataque Botnet sea analizado el protocolo HTTP en búsqueda de coincidencias con las direcciones normalmente utilizadas por los atacantes.

```
#Ataques de virus-Troyanos-Malware
alert dns $HOME_NET any -> any any (msg:"Ataque de tipo Virus-Troyano-Malware"; dns_query;
content:"tr096.pw"; depth:8; nocase; endswith; fast_pattern; sid:0000001;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"Ataque de tipo Virus-Troyano-Malware";
flow:established,to_server; content:"/gate.php?cmd="; http_uri; content:"&botnet="; http_uri;
content:"&userid="; content:"&os="; http_uri; metadata:former_category MALWARE; sid:0000002;)
alert dns $HOME_NET any -> any any (msg:"Ataque de tipo Virus-Troyano-Malware"; dns_query;
content:"xpknpxmywqsr.support"; depth:20; nocase; endswith; fast_pattern; sid:0000003;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"Ataque de tipo Virus-Troyano-Malware";
flow:established,to_server; content:"GET"; http_method; content:"/install/"; http_uri; nocase;
classtype:trojan-activity; sid:0000004;)
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"Ataque de tipo Virus-Troyano-Malware";
flow:established,from_server; content:"|16|"; content:"|0b|"; within:8; content:"|55 04 03|";
distance:0; content:"|0b 2a|.tr553.com"; distance:1; within:12; threshold: type limit, track
by_src, count 2, seconds 60; classtype:trojan-activity; sid:0000005;)
```

Figura 44-3: Reglas para los casos de tipo virus, troyanos, malware.

Realizado por: Álvarez, Andrés, 2022.

- **Ataques de tipo Sniffer**

Los ataques de tipo Sniffer están centrado en el análisis del tráfico que circula por la red para mediante esta información generar paquetes o protocolos maliciosos en contra de la red. Como se pudo observar en las pruebas de reconocimiento de red, un atacante se podría centrar en la alteración de los protocolos HTTP, UDP, TCP MQTT, SSL, TLS, entre los principales, ya que estos son los que utilizan los dispositivos IoT para transmitir su información, un claro ejemplo se produciría si mediante el protocolo TCP se inunda la red con segmentos SYN (SYN flooding) que contengan direcciones IP falsas, lo que causa por una parte incrementar el volumen de tráfico afectando al rendimiento pero también se puede aprovechar para que dispositivos desprotegidos entablen comunicación con destinos perjudiciales.

Otras acciones sospechosas que se pueden encontrar son las realizadas al escanear la red como por ejemplo al utilizar el protocolo ICMP mediante la generación de *ping*, o con las herramientas NMAP y NIKTO y el mayor inconveniente que se presenta es que no siempre se puede distinguir entre acciones legítimas o peligrosas. Para estos casos las reglas configuradas toman las siguientes condiciones.

Las reglas en primer lugar especifican el protocolo en el que estan enfocados por ejemplo ICMP o TCP, además como se trata de acciones preventivas consideramos el echo de que el ataque será dirigido desde agentes externos hacia la red por cualquier puerto disponible, por lo tanto colocamos *$EXTERNAL_NET any -> $HOME_NET any*, por otra parte y fundamentalmente agregamos la bandera que identifica a SYN que es *s* y finalmente restringimos el número de paquetes enviados mediante *threshold* con el cual establecemos el limite de alertas antes de activas la regla y un tiempo de espera para reducir la congestión.

```
#Ataques de tipo Sniffer
alert icmp any any -> any any (msg:"Ataque de tipo Sniffer"; sid:0000006;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Ataque de tipo Sniffer"; flow:to_server;
flags: S,12; threshold: type both, track by_dst, count 100, seconds 5; classtype:misc-activity;
sid:0000007;)
alert tcp any any -> any any (msg:"Ataque de tipo Sniffer"; flags:F; sid:0000009;)
alert tcp any any -> any any (msg:"Ataque de tipo sniffer"; flags:0; sid:0000010;)
alert tcp any any -> any any (msg:"Ataque de tipo Sniffer"; flags:FPU; sid:0000011;)
alert icmp any any -> any any (msg:"Ataque de tipo Sniffer"; dsize:0; sid:0000012;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"Ataque de tipo Sniffer";
flow:established,to_server; content:"User-Agent|3a| AutoGetColumn"; http_header;
classtype:attempted-recon; sid:0000013;)
#alert mqtt any any -> any any (msg:"Ataque de tipo Sniffer";
mqtt_method:con_req;threshold:type limit, track by_src, count 50, seconds 60; sid:1000009;)
#alert mqtt any any -> (msg:"Ataque de tipo Sniffer"; mqtt_method:pub; threshold: type limit,
track by_src, count 100, seconds 60; classtype:bad-unknown; sid:1000010;)
```

Figura 45-3: Regla para los casos de tipo sniffer.

Realizado por: Álvarez, Andrés, 2022.

- **Ataques de tipo XSS**

Los ataques XSS se encargan de enviar secciones de código a las páginas web intentando que se ejecute algún comando y de esta forma obtener el control de recursos de las sesiones web, una vez conseguido este objetivo puede alterar el contenido web o crear sesiones fraudulentas, normalmente se ejecutan en formato Javascript debido a que este lenguaje permite a las páginas web interactuar con otros servicios, así como cookies y APIs, por lo tanto es fundamental identificar y descartar secciones de datos que contengan este formato, un atacante intentará camuflar acciones "fuera de contexto" a diferencia de lo que se realiza en un código normal.

Uno de los casos que integran las reglas está enfocada en la actividad del puerto 80 que es utilizado por el protocolo http y buscar entre los paquetes secciones de instrucciones encerradas entre *</script>*, mediante *nocase* incluimos los caracteres en mayúsculas y minúsculas.

```
#Ataque de tipo XSS
alert http any any -> $HOME_NET 80 (msg:"Ataque de tipo XSS"; flow:to_server, established;
uricontent:"</script>"; nocase; sid:0000014;)
alert http $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Ataque de tipo XSS";
flow:to_server,established; content:"GET "; depth:4; uricontent:"/etc/passwd?format=";
uricontent:"><script>alert('xss')"; uricontent:"traversal="; classtype:attempted-recon;
sid:0000015;)
alert http $EXTERNAL_NET any -> $HTTP_SERVERS any (msg:"Ataque de tipo XSS";
flow:to_server,established; content:"Springenwerk"; http_user_agent; threshold: type limit,
count 1, seconds 60, track by_src; classtype:attempted-recon; sid:0000016;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Ataque de tipo XSS"; flow:to_server,
established; content:"Alert(document.cookie)"; classtype:attempted-admin; sid:0000017;)
```

Figura 46-3: Regla para los casos de tipo XSS.

Realizado por: Álvarez, Andrés, 2022.

- **Ataque de tipo SQL injection**

Para bloquear este tipo de ataques es fundamental identificar las consultas dirigidas a las bases de datos, con el fin de que la base de datos no responda a estas consultas y exponga la información almacenada, o que ingrese una instrucción que dañe la información.

Por lo tanto, las reglas se centrarán en reconocer las comunicaciones con el servidor de base de datos, esto se logra mediante *Flow: established, to_server*. De esta forma centramos el análisis en la comunicación que haya sido capaz de conectarse con el servidor ya que los paquetes serán dirigidos solo al servidor.

La efectividad de las reglas se encuentra en la sección de *content* ya que se intenta realizar un *match* entre la sección del tráfico que contenga instrucciones SQL y las frases que se hayan configurado en la regla, de igual forma ignoraremos las mayúsculas y las minúsculas mediante la opción *nocase*. Además, mediante la instrucción *http_uri* analizaremos las terminaciones de las direcciones *url* que se almacenaran en el buffer del sistema IPS, esta opción se complementa con *http_client_body* enfocando el análisis en lo que contengan el cuerpo del paquete enviado por el cliente hacia el servidor.

Por otra parte, también se aumenta la protección si bloqueamos las acciones que se puedan realizar con el script *bsqlbfv1.2-th.pl* mediante *content:"bsqlbf"* este script escrito en lenguaje Perl es capaz de entablar comunicación con diferentes plataformas de bases de datos tales como MySQL, PostgreSQL, Oracle. De forma que las reglas se definen de la siguiente manera:

```
#Ataque de tipo SQL injection
alert http $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"SQL injection";
flow:established,to_server; content:"bsqlbf"; http_user_agent; nocase;
classtype:web-application-activity; sid:0000018;)
alert http any any -> any any (msg:"Ataque de tipo SQL injection"; flow:established, to_server;
content:"'"; nocase; http_uri; sid:0000019;)
alert http any any -> any any (msg:"Ataque de tipo SQL injection"; flow:established, to_server;
content:"union"; nocase; http_uri; sid:0000020;)
alert http any any -> any any (msg:"Ataque de tipo SQL injection"; flow:established, to_server;
content:"select"; nocase; http_uri; sid:0000021;)
alert http any any -> any any (msg:"Ataque de tipo SQL injection"; flow:established, to_server;
content:"insert"; nocase; http_uri; sid:0000022;)
alert http any any -> any any (msg:"Ataque de tipo SQL injection"; flow:established, to_server;
content:"delete"; nocase; http_uri; sid:0000023;)
```

Figura 47-3: Reglas casos de tipo SQL injection.

Realizado por: Álvarez, Andrés, 2022.

- **Ataque de DDoS**

Debido a las características que se presentan en la red, el ataque más perjudicial que se podría sufrir es la denegación de servicios. En primer lugar las técnicas que se han desarrollado para desplegar esta clase de acciones actualmente son muy avanzadas siendo capaz de dirigir el tráfico malicioso a servicios específicos y dejar sin respuesta a las aplicaciones que las utilizan aplicaciones; por otra parte se complican las acciones de prevención o mitigación del ataque ya que se puede camuflar con el tráfico propio de la red por ejemplo en el caso de la red IoT podría aparentar ser un sensor que transmite información en tiempo real tal como lo realiza el sensor de temperatura. Además, Las soluciones tradicionales intentan descubrir la dirección IP desde donde

proviene el ataque para bloquearlo, pero fácilmente podría evadirse esta protección generando direcciones IP aleatorias o utilizando dispositivos infectados.

Tomando en cuenta todas estas características es necesario reducir la capacidad de los paquetes de mantenerse presentes en la red mediante el argumento *ttl:64* además de que impedimos la conexión de equipos lejanos, se desea analizar todas las conexiones que vayan destinadas al servidor mediante *Flow:to_service*, y los campos fundamentales a considerar son las banderas PUSH que obliga al receptor a procesar los paquetes son mantenerlos en el buffer, así como el ACK que se encarga de confirmar la recepción de mensaje pero usado de forma indebida mantiene al servidor reenviando paquetes.

Otra de la forma con las que se cuenta para atraer la atención del servidor es mediante la bandera SYN que constantemente le solicita al servidor establecer una comunicación. Finalmente establecemos un límite de mensajes destinados hacia el servidor mediante *threshold:type threshold, track by_dst; count 500, seconds 60;*

```
#Ataque de tipo DoS y DDoS
alert tcp any any -> any any (msg:"Ataque de tipo DoS y DDoS"; ttl:64; flow:to_server; flags:S;
threshold:type threshold, track by_dst, count 500, seconds 60; classtype:attempted-dos;
sid:0000024;)
alert tcp any any -> any any (msg:"Ataque de tipo DoS y DDoS"; ttl:64; flow:to_server;
flags:PA; threshold:type threshold, track by_dst, count 500, seconds 60;
classtype:attempted-dos; sid:0000025;)
```

Figura 48-3: Regla para casos de tipo DoS y DDoS.

Realizado por: Álvarez, Andrés, 2022.

CAPITULO IV

4. MARCO DE RESULTADOS Y DISCUSIÓN

Para comprender los resultados generados por el caso de estudio se realizará una comparación entre el estado normal de la red, bajo los ataques informáticos propuestos y el comportamiento una vez aplicado el sistema de prevención IPS. Con los valores obtenidos se podrá determinar las ventajas y desventajas que se presentan en el escenario.

4.1. Comportamiento de la red bajo condiciones normales de uso

En primer lugar, se determinó el consumo de recursos promedio de las placas Rasberry Pi para establecer una base, actualmente la placa Rasberry Pi 4 de 4Gb de memoria RAM, con un procesador de 64 bits se encuentra ejecutando el software de enrutamiento RaspAP WiFi, además de procesos menores que también aportan al consumo de procesamiento. Como resultado el promedio de consumo de procesador es de 63,59% como se puede apreciar en el gráfico 1-4.

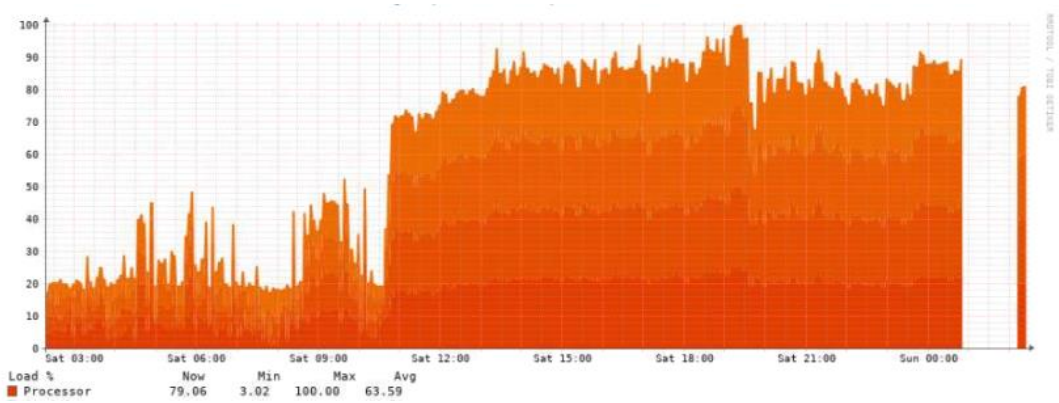


Gráfico 1-4: Consumo de los núcleos del procesador en condiciones normales.

Realizado por: Álvarez, Andrés, 2022.

Por otra parte, el consumo de memoria RAM se sitúa en un 43% de promedio y en el caso de la memoria SWAP un 62% tal como se puede apreciar en la imagen

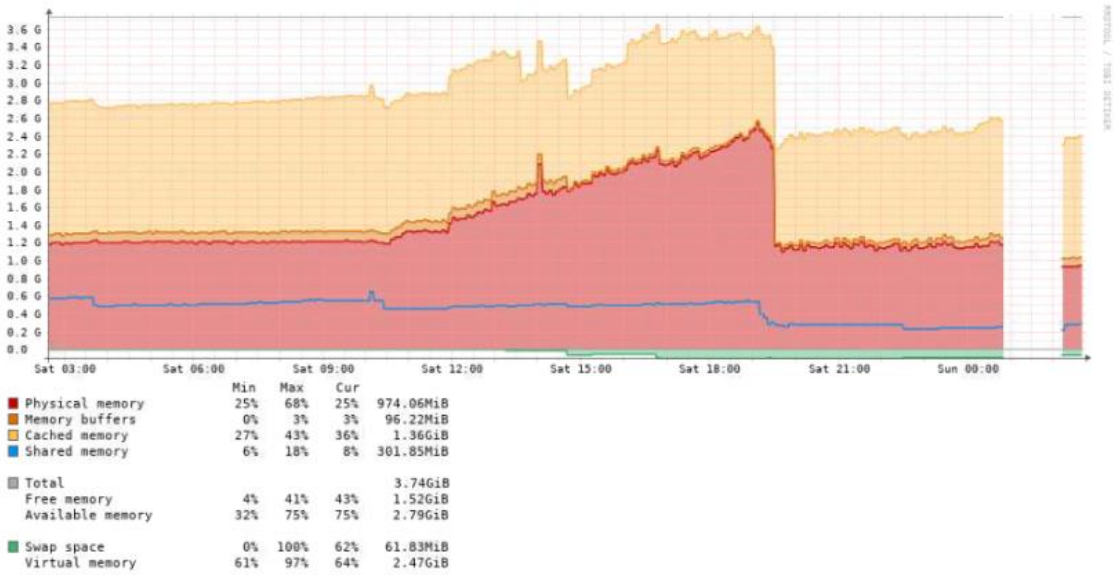


Gráfico 2-4: Consumo de las memorias internas de una placa Rasberry Pi

Realizado por: Álvarez, Andrés, 2022.

Con respecto al tráfico Mediante el software instalado podemos determinar dos aspectos presentes en la red, en primer lugar, la tasa media de transferencia de la red LAN por el puerto wlan1 es de 17.92kbps en bajada y 123.86kbps de subida.

Gráfico 3-4: Ancho de banda consumido por el tráfico presente en la red.

Realizado por: Álvarez, Andrés, 2022.

En segundo lugar, el volumen de datos generados por todos los dispositivos y atraviesa la red es de promedio 7.41MB de bajada y 63.56MB de subida.

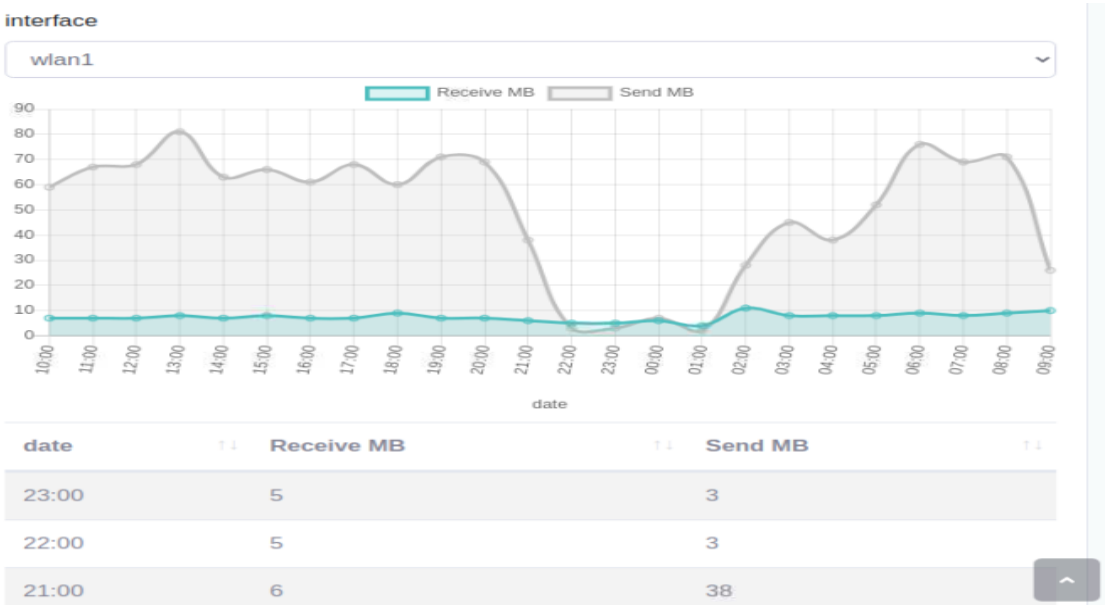


Gráfico 4-4: Volumen promedio de datos transmitidos.

Realizado por: Álvarez, Andrés, 2022.

Finalmente, como es de esperarse al no ejecutar ningún tipo de ataque o escaneo de red los valores de alertas y detecciones son mínimos y solo son activados por mensajes de control propios de la red tal como se puede visualizar en la figura 1-3.

Figura 1-4: Información generada por Suricata mediante el archivo eve.json.

Realizado por: Álvarez, Andrés, 2022.

4.1.1. Comportamiento de la red bajo ataques

En primer lugar, se observará el comportamiento de los dispositivos y la red al realizar un escaneo de puertos y búsqueda de vulnerabilidades. Como se puede observar, las acciones pueden pasar desapercibidas ya que se podría considerar parte del tráfico normal de la red y el procesamiento y la memoria tampoco se ven afectadas, cabe señalar que estas acciones tienen como objetivo permitir al atacante reconocer el entorno, siendo estos considerados como ataques pasivos.

Figura 2-4: Información generada por Suricata dentro de Logs.

Realizado por: Álvarez, Andrés, 2022.

Pero al utilizar herramientas más avanzadas, como es el caso de Hydra que ejecuta pequeñas pruebas de penetración mediante el protocolo SSH, los identificadores de los ataques se activan, demostrando que las reglas reconocen las acciones ejecutadas por parte del atacante, tal como se puede apreciar en la figura 3-3 y figura 4-3.

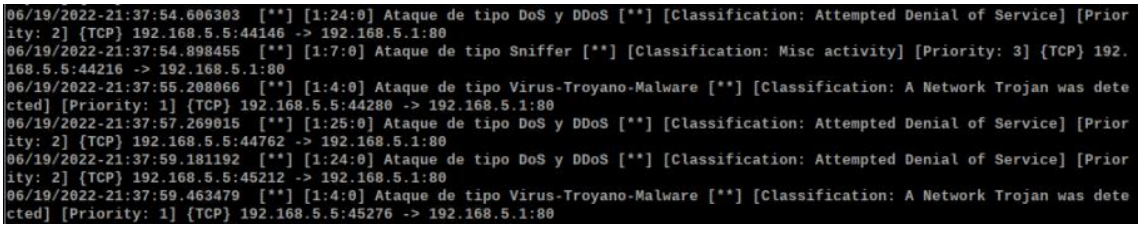


Figura 3-4: Información generada por Suricata dentro de Logs.

Realizado por: Álvarez, Andrés, 2022.

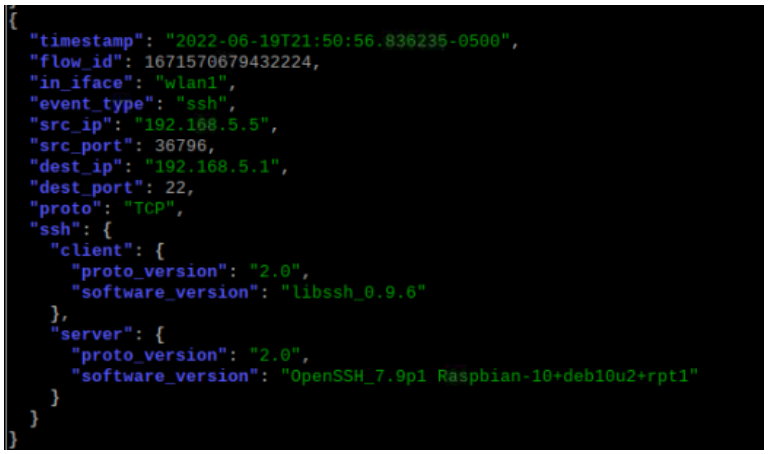


Figura 4-4: Información presente en el log de Suricata.

Realizado por: Álvarez, Andrés, 2022.

Al ejecutar la herramienta hping3 que simulan ataques DoS y DDoS se puede visualizar que efectivamente se altera el comportamiento de la red y que mientras más tiempo se mantenga activo el ataque será mucho más perjudicial. Se puede apreciar las diferentes direcciones IP que se comunican con el servidor a pesar de que no pertenecen a esta sección de red.

Figura 5-4: Información presente en el log de Suricata.

Realizado por: Álvarez, Andrés, 2022.

A pesar de que el despliegue solo duro unos minutos y que estaba limitado por las características de la máquina virtual en la que se ejecuta, es evidente que la red sufrió un deterioro. En este caso es mucho más evidente el consumo de ancho de banda, ya que este se eleva drásticamente lo que genera picos de tráfico anormales dentro de la red.

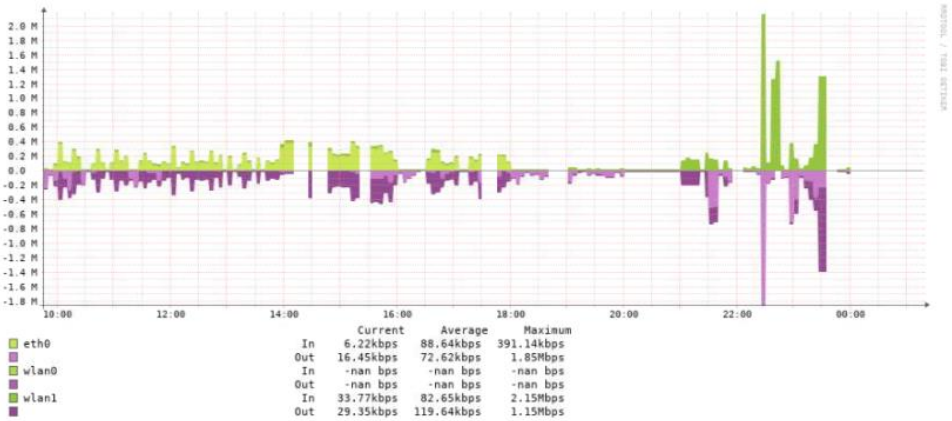


Gráfico 5-4: Consumo de ancho de banda bajo ataque de tipo DDoS.

Realizado por: Álvarez, Andrés, 2022.

Además, el volumen de datos transferido fue de 221MB de bajada y 112MB de subida en sus picos más altos. Si comparamos estos valores con el caso anterior en donde no existía actividad

maliciosa, se ha generado un comportamiento anormal ya que el volumen de bajada es superior al de subida

Gráfico 6-4: Datos presente en la red bajo un ataque de tipo DDoS.

Realizado por: Álvarez, Andrés, 2022.

Visualmente las consecuencias también se presentan en las ventanas de aplicación, como se puede apreciar en la figura, ya que impide a los servicios comunicarse y estos informan los problemas en la comunicación, así como también un incremento considerable en el consumo de recursos.

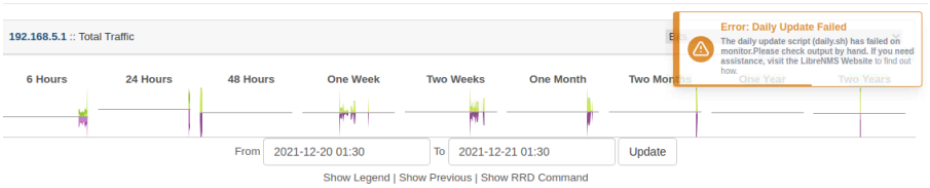


Figura 6-4: Error generado en LibreNMS por desconexión.

Realizado por: Álvarez, Andrés, 2022.

Se evidencia que el procesamiento se elevó hasta llegar a un promedio de 85% y se generaron problemas para manipular el sistema, motivo por el cual fue necesario detener la prueba.

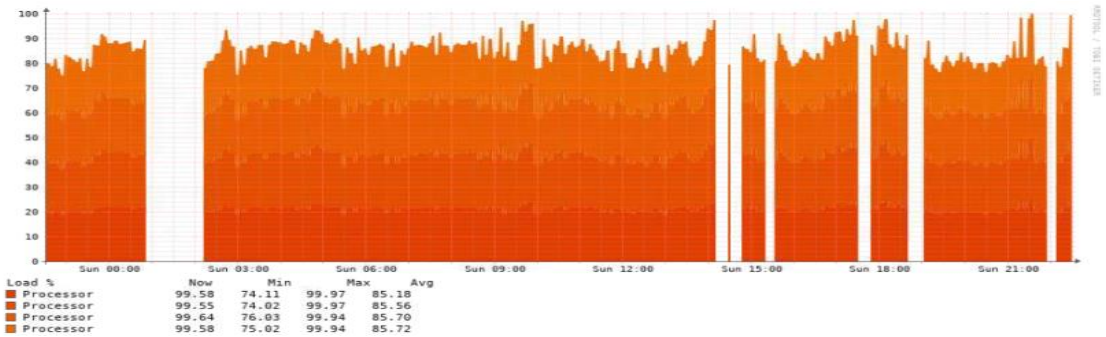


Gráfico 7-4: Consumo de los núcleos del procesador bajo un ataque informático.

Realizado por: Álvarez, Andrés, 2022.

4.1.2. Resultados de utilizar el sistema IPS en la red

Al momento de ejecutar la herramienta Suricata en modo IPS las alertas son parecidas al caso anterior, en el que solo actuaba como IDS, pero la principal diferencia es que se puede evidenciar la acción que realiza el sistema IPS al detectar un paquete peligroso, como se aprecia en la imagen nos señala la IP de origen y a que dirección está destinada, también se puede apreciar el puerto que fue utilizado por el atacante, así como el protocolo; por otro lado y fundamentalmente se puede identificar que el paquete fue bloqueado mediante la regla que contiene el SID 7, y también podemos visualizar el mensaje asignado a la regla.

Figura 7-4: Información presente en el log del archivo eve.json.

Realizado por: Álvarez, Andrés, 2022.

Con respecto al rendimiento del equipo en el que se ejecuta el sistema IPS, existe un ligero incremento en el consumo de recursos debido a que los paquetes deben ser transmitidos desde el filtro de IPTABLES hacia NFQUEUE, también causa un ligero retardo en la comunicación, pero no interfiere en el control de los dispositivos.

En conclusión, se puede definir que todas las acciones implementadas en un sistema IPS se enfocan en la previenen de los ataques informáticos, pero también nos brinda un beneficio mucho más importante; ya que, al convertir a nuestra red en un objetivo difícil de analizar por parte del atacante, no conocerá las debilidades de la red y esto nos permite detectar las fallas existentes y corregirlas. Una vez realizadas las pruebas correspondientes se puede determinar el nivel de peligrosidad de los ataques informáticos, y cuáles serían utilizados para aprovechar las vulnerabilidades, amenazas y riesgos de la red IoT planteada. El análisis se puede conceptualizar en la siguiente tabla:

Tabla 1-4: Resultados de ataques informáticos a los que está expuesta la red.

Nivel de peligrosidad	Descripción	Tipo de ataque
Critico	Afecta directamente al rendimiento de la red siendo capaz de dejarla inutilizable completamente, y aunque existan una planificación preventiva por parte del sistema, mientras no se conozca el origen del ataque, será difícil mitigarlo por completo	Ataque de tipo DDoS
Alto	Los ataques de tipo Sniffer son muy variados y pueden aprovechar diversas secciones de los paquetes para atacar a la red, aun teniendo las configuraciones adecuadas para prevenir este tipo de ataques siempre existe la posibilidad de que el sistema apruebe como legítimo un paquete infectado	Ataque de tipo Sniffer
Medio	La relativa facilidad con que pueden ser ejecutados los ataques de SQL injection se suma al daño que pueden producir a una red, ya que las bases de datos son el componente más importante de una red. La pérdida de datos no solo afecta a la infraestructura sino también a los usuarios	Ataque de tipo SQL injection
Medio	Afecta principalmente a las sesiones de los clientes, puede sustraer credenciales con información delicada y crear perfiles fraudulentos	Ataque de tipo XSS
Bajo	Desde un punto de vista preventivo para los sistemas IPS es mucho más fácil identificar virus, troyanos o malware ya que se cuenta con una biblioteca muy amplia de muestras y constantemente se actualizan con nuevas muestras, por este motivo es posible descartar todos los paquetes infectados y reducir al mínimo las acciones perjudiciales de esta clase de archivos.	Ataques de tipo Virus, Troyanos, Malware

Realizado por: Álvarez, Andrés, 2022.

CONCLUSIONES

- Se ha logrado diseñar e implementar en un ambiente real el sistema de prevención de intrusos IPS propuesto, demostrando que puede ser tomada en cuenta al buscar una solución preventiva ante ataques informáticos.

- Al estudiar los dispositivos IoT se comprobó los mínimos niveles de seguridad que presenta una red IoT, ya que los dispositivos que la integran están diseñados mediante interfaces amigables para el usuario y lo más intuitivas posible, sacrificando la seguridad del dispositivo, de forma que la capacidad de prevenir o resistir un ataque informático recae en los usuarios y en los dispositivos destinados a adminístralos.

- Fue Posible determinar el comportamiento de la red bajo condiciones normales y ante un ataque, demostrando, así como se deteriora la red, principalmente en el consumo de ancho de banda el cual presento alteraciones muy evidentes cuando existe un ataque informativo, y por otra parte en el rendimiento de los dispositivos los cuales se vieron deteriorados. Estas características podrían servir como pista inicial de que en la red está siendo comprometida.

- Se logró diseñar la red IoT funcional a pesar de que eran dispositivos de baja potencia y que sus características están enfocadas hacia aplicaciones de índole general; de forma que en próximos estudios se podría conseguir un mejor rendimiento si se utilizara dispositivos con un mejor enfoque hacia las redes IoT.

- Al analizar el tráfico de la red fue evidente que existen una gran posibilidad de alterar el contenido de los paquetes que viajan en la red y estos ser distribuidos por toda la red, motivo por el cual al implementar reglas para un sistema IPS no deben estar enfocadas en las aplicaciones que ejecutan los ataques sino, en los protocolos y las deficiencias que estos generan ya que siempre estarán presentes en una red y son mucho más complicados de detectar.

- Se verificó el nivel de protección del sistema IPS dentro de diferentes situaciones y se puede afirmar que estos sistemas son una importante opción a tomar en cuenta cuando se desea proteger una red IoT, lo fundamental para entender el nivel de protección de un sistema IPS es que actuará de mejor manera si las reglas integran el mayor número de características de un ataque.

RECOMENDACIONES

- Después de realizar toda la instalación del sistema LibreNMS hay que tener en cuenta que se presentaran errores la primera vez que se ejecuta el sistema ya que necesita las credenciales con que fue creada la base de datos, para que de esta forma pueda ser verificada, mientras no se ingresen los datos correspondientes se presentara un mensaje en la configuración como en el anexo I.

- Se pueden encontrar fallos al intentar guardar los datos de la red y por lo tanto no se podrán generar las gráficas, cuando exista este problema veremos el mensaje presente en el anexo J. Para corregir este fallo es necesario ejecutar el siguiente comando:
su – librenms
git pull
./daily.sh

- Al momento de editar el archivo de configuración de la herramienta Suricata *suricata.yaml,* así como cuando se escribe una firma dentro de la carpeta *rules/* es fundamental escribir desde el margen izquierdo sin dejar espacios ya que el software al momento de ser ejecutado nos imprimirá en pantalla un error a pesar de que nuestro código sea correcto y podría causar confusión y contratiempos.

- Las herramientas Suricata y Snort no cuentan con una interface gráfica para la arquitectura ARM en la cual está diseñado Raspberry Pi, de forma que, si se desea un sistema IPS mucho más amigables para el administrador, se recomienda utilizar equipos con arquitecturas compatibles con las interfaces gráficas.

- Se debe tener en cuenta que al enviar todo el tráfico hacia el módulo NFQ de Suricata, mientas no se definan reglas o acciones a realizar, los paquetes generaran colas y esto pueden afectar a la conectividad de los demás dispositivos, por lo que es recomendable solo transferir el tráfico cuando se encuentren configuradas las reglas y los módulos.

BIBLIOGRAFÍA

ALFON. *Suricata. Entendiendo y configurando Suricata. Parte I. Seguridad y Redes* [en línea]. 2021, [Consulta: 17 agosto 2021]. Disponible en: https://seguridadyredes.wordpress.com/2011/02/22/ids-ips-suricata-entendiendo-y-configurando-suricata-parte-i/.

ALONSO, N. *Los 10 mejores firewall o cortafuegos para Windows. Grupo Atico34* [en línea] 2020 [Consulta: 10 enero 2022]. Disponible en: https://protecciondatos-lopd.com/empresas/mejores-firewall-windows/.

ALTITUDETVM. *Conozca la comprensión de Wireshark y sus funciones y formas, completa* [en línea] 2020 [Consulta: 11 septiembre 2021]. Disponible en: https://altitudetvm.com/es/internet/1096-kenali-pengertian-wireshark-beserta-fungsi-dan-cara-kerjanya-lengkap.html.

ARDUINO. *Software de Arduino | Arduino.cl - Compra tu Arduino en Línea*. [en línea] 2021 [Consulta: 15 octubre 2021]. Disponible en: https://arduino.cl/programacion/.

ARDUINOVE. *Módulo ESP32 ESP-WROOM-32 WiFi +Bluetoot*h. [en línea] 2021 [Consulta: 30 julio 2021]. Disponible en: http://www.arduinove.com/index.php?route=product/product&product_id=544.

ASANZA, V. *Especificaciones del módulo ESP32. ▷ Especificaciones del módulo ESP32* [en línea] 2022 [Consulta: 9 febrero 2022]. Disponible en: https://vasanza.blogspot.com/2021/07/especificaciones-del-modulo-esp32.html.

BAZ, E.*Profesor Cyber: Uso básico de Wireshark (Parte 3: Los filtros). Profesor Cyber* [en línea] 2020 [Consulta: 25 septiembre 2021]. Disponible en: https://profesorcyber.blogspot.com/2020/04/uso-basico-de-wireshark-parte-3-los.html.

BRYANT, D. *An Important Update on Mozilla WebThings. Mozilla Discourse* [en línea] 2020 [Consulta: 10 octubre 2021]. Disponible en: https://discourse.mozilla.org/t/an-important-update-on-mozilla-webthings/67764.

CHECKPOINT. *¿Qué es un firewall? Check Point Software ES* [en línea] 2021 [Consulta: 10 enero 2022]. Disponible en: https://www.checkpoint.com/es/cyber-hub/what-is-firewall/.

CODINA, L. *Qué son las taxonomías y cómo se aplican a sitios web. Lluís Codina* [en línea] 2019 [Consulta: 21 diciembre 2021]. Disponible en: https://www.lluiscodina.com/taxonomia-sitio-web/.

CRUZ, R. *Desarrollo de un sistema para reconocimiento de texto y conversión a audio, utilizando Raspberry Pi para personas no videntes* [en línea] 2021 Quito-Ecuador: Universidad Politecnica Salesiana sede Quito. [Consulta: 14 noviembre 2020]. Disponible en: https://dspace.ups.edu.ec/bitstream/123456789/19924/1/UPS%20-%20TTS295.pdf.

DIANA C. *Tutorial de Iptables - Asegura tu VPS Ubuntu con un Firewall de Linux. Tutoriales Hostinger* [en línea] 2021 [Consulta: 12 enero 2022]. Disponible en: https://www.hostinger.es/tutoriales/iptables-asegurar-ubuntu-vps-linux-firewall/.

FRANCIS, B. *Flying the Nest: WebThings Gateway 1.0 – Mozilla Hacks - the Web developer blog. Mozilla Hacks – the Web developer blog* [en línea] 2020 [Consulta: 10 octubre 2021]. Disponible en: https://hacks.mozilla.org/2020/12/flying-the-nest-webthings-gateway-1-0.

GARCIA, L. *¿Cuáles son las arquitecturas y componentes de una red IoT? Redes Móviles* [en línea] 2020 [Consulta: 15 diciembre 2021]. Disponible en: https://redesmoviles.com/iot/arquitecturas-iot/.

GARETH. *The Official Raspberry Pi Beginner's Guide, 4th Edition and Translations. Gareth Halfacree* [en línea] 2021 [Consulta: 16 julio 2021]. Disponible en: https://freelance.halfacree.co.uk/2020/11/the-official-raspberry-pi-beginners-guide-4th-edition-and-translations/.

GONZALEZ, M. *Modelo de consumo de energía de Raspberry Pi 4 B con escalamiento de frecuencias* [en línea] 2021, ciudad de Mexico-Mexico: Instituto Politécnico Nacional. [Consulta: 14 noviembre 2020]. Disponible en: https://www.escom.ipn.mx/posgrado/tesis/Tesis_065_Miguel_Angel_Gonzalez_Alonso.pdf.

GUIDESMANIA. *What Is The Difference Between Windows Firewall And Linux Firewall - GuidesMania*. [en línea] 2021 [Consulta: 12 enero 2022]. Disponible en: https://guidesmania.com/difference-between-windows-firewall-and-linux-firewall/.

HACKMAGEDDON. *Cyber Attacks Statistics. HACKMAGEDDON* [en línea] 2020 [Consulta: 5 febrero 2022]. Disponible en: https://www.hackmageddon.com/2021/01/13/2020-cyber-attacks-statistics/.

HASSOLD, C. *The Threat Taxonomy: Types of Cybercrime and a Framework. Agari* [en línea] 2020 [Consulta: 4 febrero 2022]. Disponible en: https://agari.com/email-security-blog/threat-taxonomy-framework-cyber-attacks/.

HIGUERA, L. *Diseño de una tarjeta de desarrollo con ESP8266 orientada a wireless y microcontroladores para IoT*. [en línea] 2019, pp. 51. [Consulta: 16 noviembre 2020]. Disponible en: http://201.159.223.180/bitstream/3317/13360/1/T-UCSG-PRE-TEC-ITEL-346.pdf.

HILLSTONE, M. *Los diferentes tipos de Firewall y cuál usar para la seguridad de la red. Hillstone Networks* [en línea] 2020 [Consulta: 10 enero 2022]. Disponible en: https://www.hillstonenet.lat/blog/los-diferentes-tipos-de-firewall-y-cual-usar-para-la-seguridad-de-la-red/.

HONIM.TYPEPAD. *Internet Traffic will increase to 396 exabytes per month by 2022. vzw BiASC asbl* [en línea] 2019 [Consulta: 5 febrero 2022]. Disponible en: https://honim.typepad.com/biasc/2019/12/internet-traffic-will-increase-to-396-exabytes-per-month-by-2022-.html.

IBIBLIO. *Detección de anomalías.* [en línea] 2016 [Consulta: 15 junio 2020]. Disponible en: https://www.ibiblio.org/pub/Linux/docs/LuCaS/Manuales-LuCAS/doc-unixsec/unixsec-html/node286.html.

INCIBE. *Taxonomía*. *INCIBE-CERT* [en línea] 2019 [Consulta: 21 diciembre 2021]. Disponible en: https://www.incibe-cert.es/taxonomia.

INCIBE. *¿Qué son y para qué sirven los SIEM, IDS e IPS? INCIBE* [en línea] 2020 [Consulta: 29 noviembre 2021]. Disponible en: https://www.incibe.es/protege-tu-empresa/blog/son-y-sirven-los-siem-ids-e-ips.

INFOTECS. *Firewall: Cortafuegos*. [en línea] 2019 [Consulta: 12 enero 2022]. Disponible en: https://infotecs.mx/blog/firewall-cortafuegos.html.

INFOTECS. *Sistema de Detección de Intrusos*. [en línea] 2019 [Consulta: 15 diciembre 2021]. Disponible en: https://infotecs.mx/blog/sistema-de-deteccion-de-intrusos.html.

INFOTECS. *IPS: Sistema de Prevención de Intrusos*. [en línea] 2019 [Consulta: 29 noviembre 2021]. Disponible en: https://infotecs.mx/blog/ips-sistema-de-prevencion-de-intrusos.html.

JÁTIVA, J et al. *Encendido y apagado de un Foco con atenuación desde cayenne y un módulo ESP 8266*. [en línea], pp. 5. [Consulta: 16 noviembre 2020]. Disponible en: https://www.researchgate.net/profile/JorgeMorales/publication/325406493_Encendido_y_apagado_de_un_Foco_con_atenuacion_desde_cayenne_y_un_modulo_ESP_8266/links/5b0c9a45aca2725783ec49c9/Encendido-y-apagado-de-un-Foco-con-atenuacion-desde-cayenne-y-un-modulo-ESP-8266.pdf.

JECRESPOM. *Arquitecturas IoT. Aprendiendo Arduino* [en línea] 2018 [Consulta: 5 febrero 2022]. Disponible en: https://aprendiendoarduino.wordpress.com/2018/11/11/arquitecturas-iot/.

KHURANA, V. *¿Cómo capturar y analizar el tráfico de red con tcpdump? Geekflare* [en línea] 2020 [Consulta: 9 octubre 2021]. Disponible en: https://geekflare.com/es/tcpdump-examples/.

LATAM.KASPERSKY. *Kaspersky registra 45 ataques por segundo en América Latina*. [en línea] 2019 [Consulta: 7 enero 2022]. Disponible en: https://latam.kaspersky.com/blog/kaspersky-registra-45-ataques-por-segundo-en-america-latina/15274/.

LLOPIS, J. *Sistema de monitorización del IDS Snort* [en línea] 2017, Valencia-España: Universitat Politècnica de València. Disponible en: https://riunet.upv.es/bitstream/handle/10251/88474/LLOPIS%20-%20Sistema%20de%20monitorización%20del%20IDS%20Snort.pdf?sequence=1&isAllowed=y.

LÓPEZ, J. *Analizando paquetes de red con Wireshark Análisis de paquetes de red con Wireshark. Blogthinkbig.com* [en línea] 2019 [Consulta: 11 septiembre 2021]. Disponible en: https://blogthinkbig.com/analizar-paquetes-de-red-wireshark.

MARINA, **A.** *Así es Snort, el sistema de detección de intrusos más popular*. *Grupo Atico34* [en línea] 2021 [Consulta: 12 agosto 2021]. Disponible en: https://protecciondatos-lopd.com/empresas/snort-deteccion-intrusos/.

MATTILA, T. *Integration of arctic node threat intelligence sharing platform with suricata*. [en línea] 2020, pp. 59. Disponible en: https://core.ac.uk/download/pdf/344911373.pdf.

MAURICIO, A. *El estado del arte sobre el internet de las cosas. amenazas y vulnerabilidades de seguridad informática evidenciadas desde la domotica*. En: Accepted: 2019-11-06T21:58:46Z [en línea] 2019, pp. 26. [Consulta: 16 marzo 2022]. Disponible en: http://repository.unad.edu.co/handle/10596/28446.

MENDOZA, A. *Informe comparativo snort - suricata*. *StuDocu* [en línea]. [Consulta: 16 junio 2022]. Disponible en: https://www.studocu.com/co/document/universidad-ecci/informatica-forense/informe-snort-suricata/23687485.

ORTEGA, D. *Qué es Snort: Primeros pasos. OpenWebinars.net* [en línea] 2017 [Consulta: 15 junio 2020]. Disponible en: https://openwebinars.net/blog/que-es-snort/.

RAMIRO, R. *Reglas SNORT , detección de intrusos y uso no autorizado. CIBERSEGURIDAD .blog* [en línea] 2020 [Consulta: 17 agosto 2021]. Disponible en: https://ciberseguridad.blog/reglas-snort-deteccion-de-intrusos-y-uso-no-autorizado/.

RASILLA, V. *Desarrollo de una prueba de concepto de un detector de intrusiones para su aplicación en prácticas de laboratorio* [en línea]. septiembre 2020. S.l.: s.n. Disponible en: http://hdl.handle.net/10902/19231.

TAMAYO, S. *Riesgos informáticos, Amenazas y Vulnerabilidad • OpenUIDE. OpenUIDE* [en línea] 2020 [Consulta: 21 diciembre 2021]. Disponible en: https://globalimf.com.ec/openuide/blog/gestion-de-riesgos-informaticos/.

TPEMPRESAS. *Los 10 mejores firewall de hadware en el 201*9. *Totalplay Empresas* [en línea] 2019 [Consulta: 17 enero 2022]. Disponible en: https://tpempresas.com/los-10-mejores-firewalls-de-hardware-para-redes-domesticas-y-de-pequenas-empresas-2019.

VIALYNK. *¿Cómo es la ciberseguridad en el futuro?* [en línea] 2021 [Consulta: 8 enero 2022]. Disponible en: https://www.vialynk.com/post/c%C3%B3mo-es-la-ciberseguridad-en-el-futuro.

VIJAY, G. *Control AC Appliances (Web Server). Instructables* [en línea] 2022 [Consulta: 9 febrero 2022]. Disponible en: https://www.instructables.com/ESP8266-Relay-Module-Control-AC-Appliances-Web-Ser/.

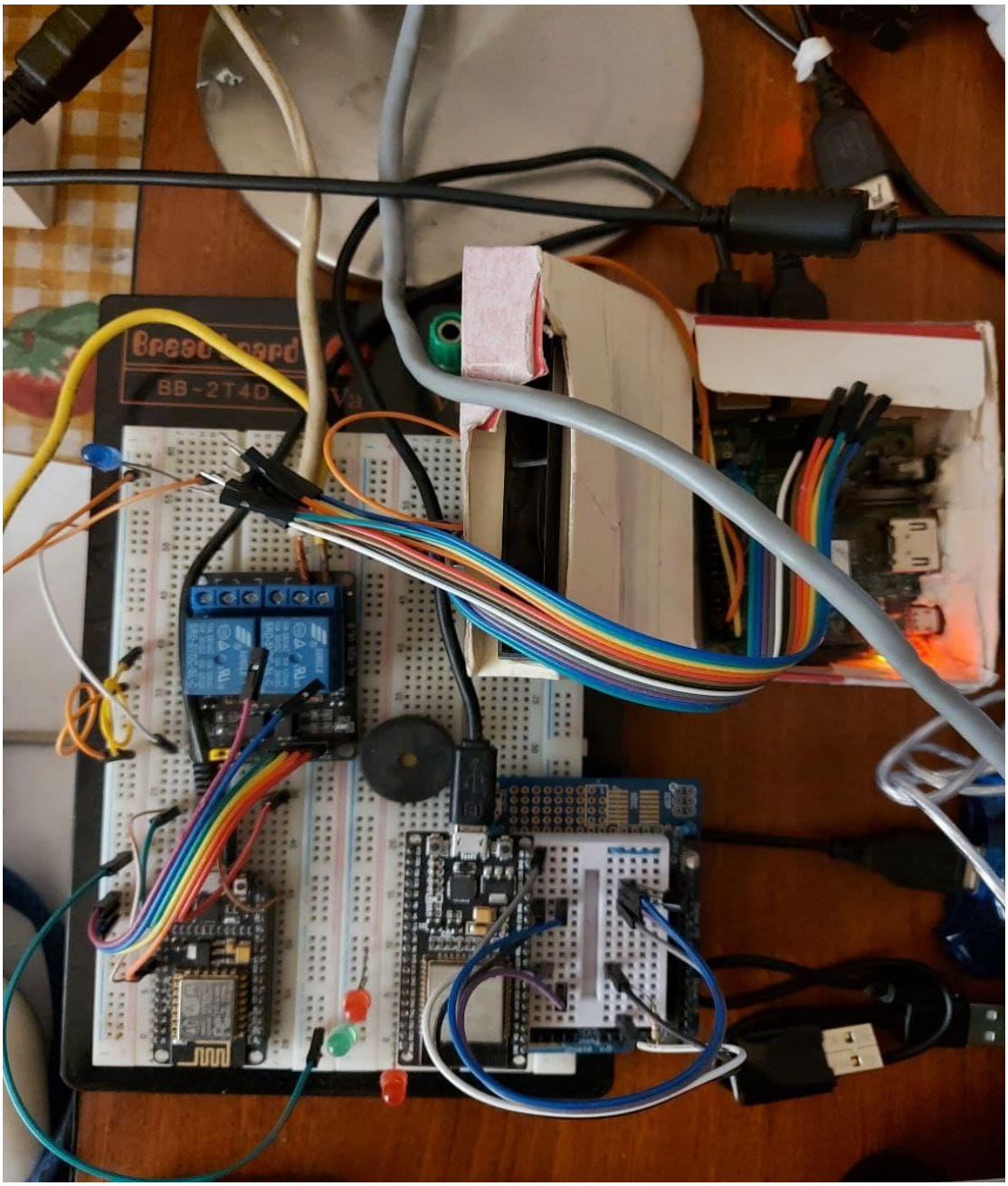
ZABALO, E. *La ciberseguridad como norma. Estudio del estado del arte en estándares y certificación en materia de seguridad cibernética aplicada a industria* [en línea] 2019, pp. 9. [Consulta: 16 marzo 2022]. Disponible en: https://addi.ehu.es/handle/10810/32240.

ZAMBRANO, A. y GUAILACELA, F. *Análisis de la eficiencia de los IDS open source Suricata y Snort en las PYMES* [en línea] 2019. Disponible en: http://repositorio.uees.edu.ec/bitstream/123456789/2926/1/ZAMBRANO%20BARBERAN%20ALFONSO%20%26%20GUAILACELA%20ROMERO%20FRANKLIN.pdf.

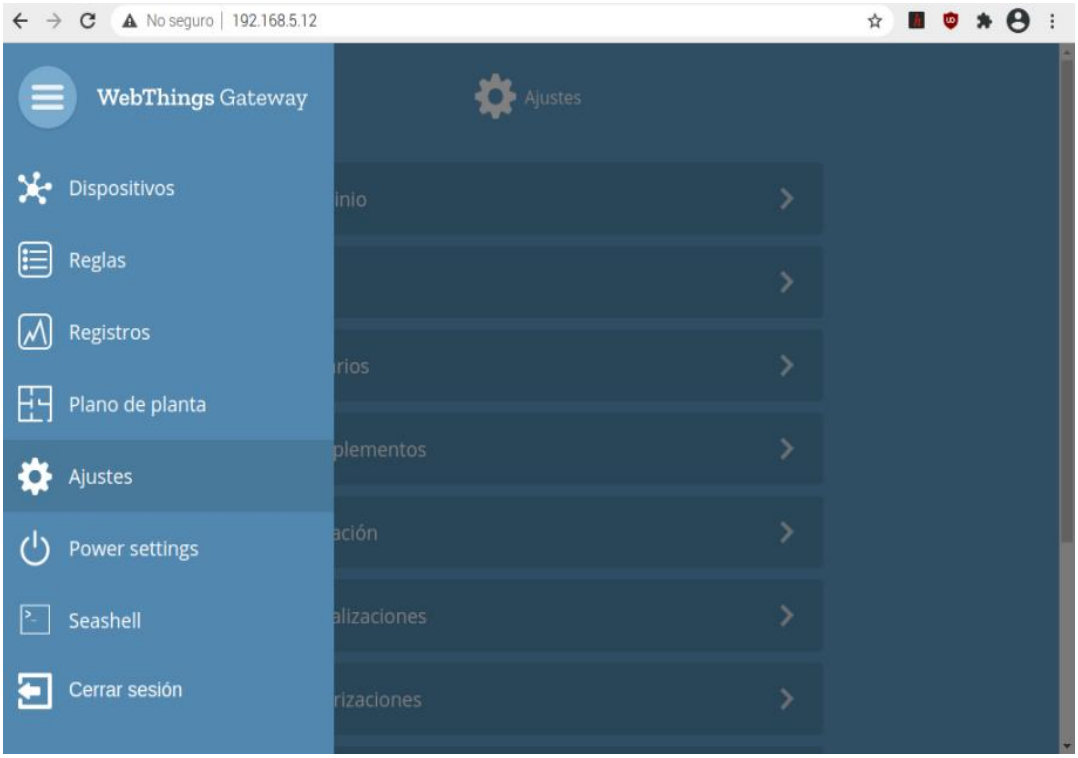
ANEXOS

ANEXO A: ESTADO FÍSICO DE LA TARJETA RASPBERRY PI

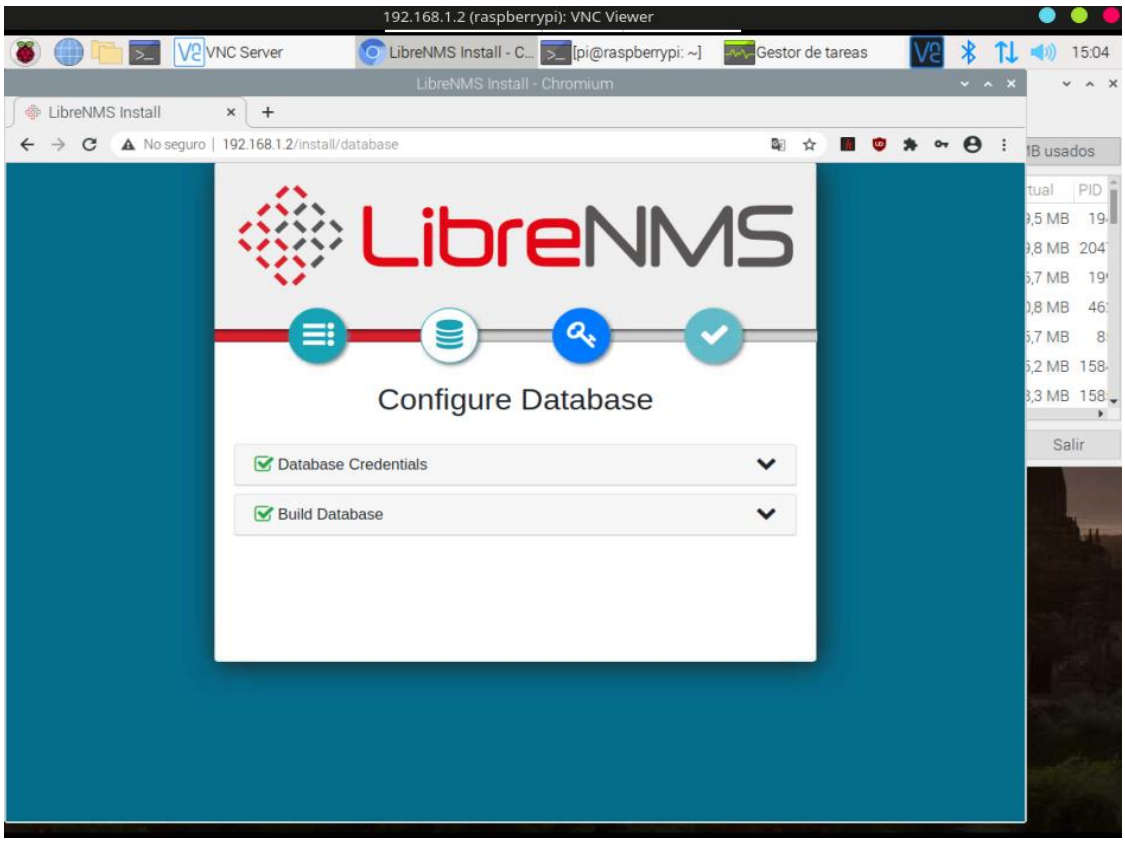
ANEXO B: DISTRIBUCIÓN DE COMPONENTES IOT



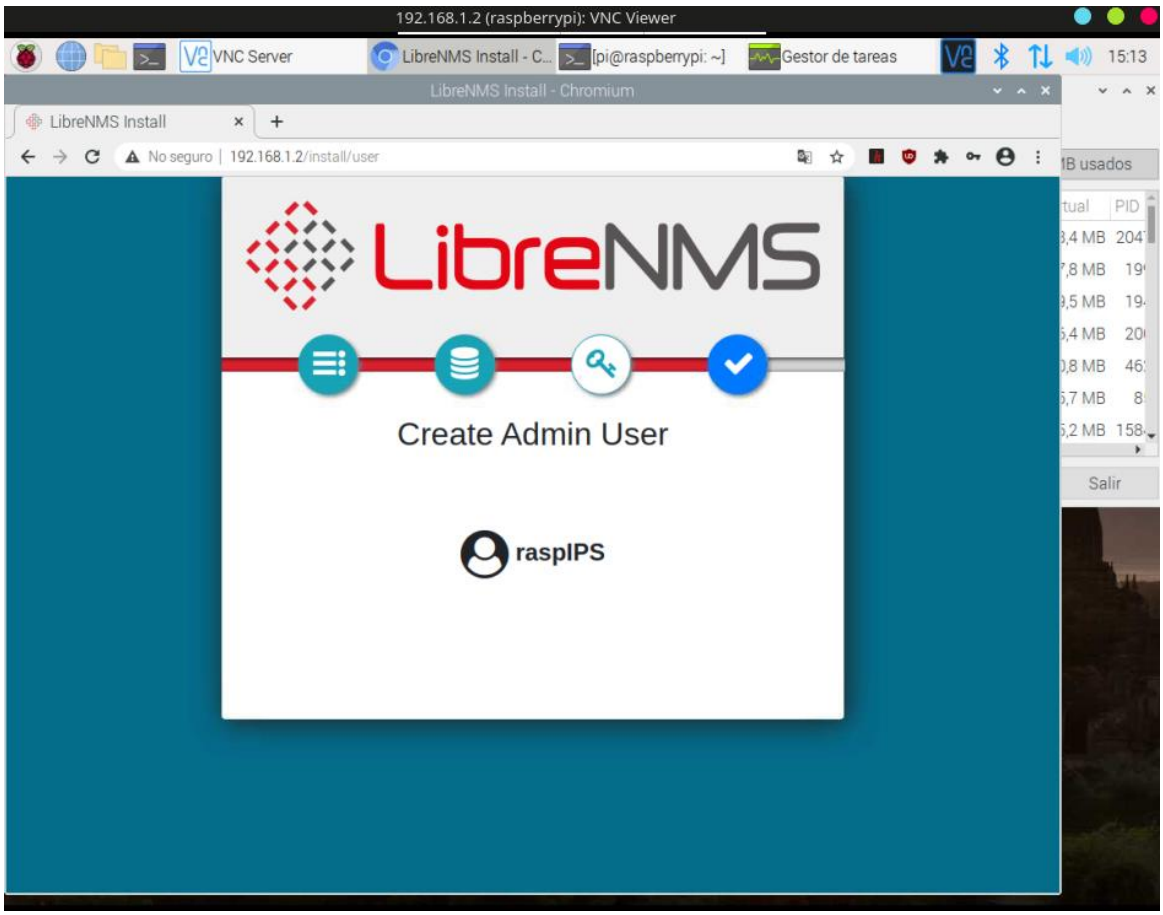
ANEXO C: VENTANA PRINCIPAL DE WEBTHINGS DESDE UN NAVEGADOR



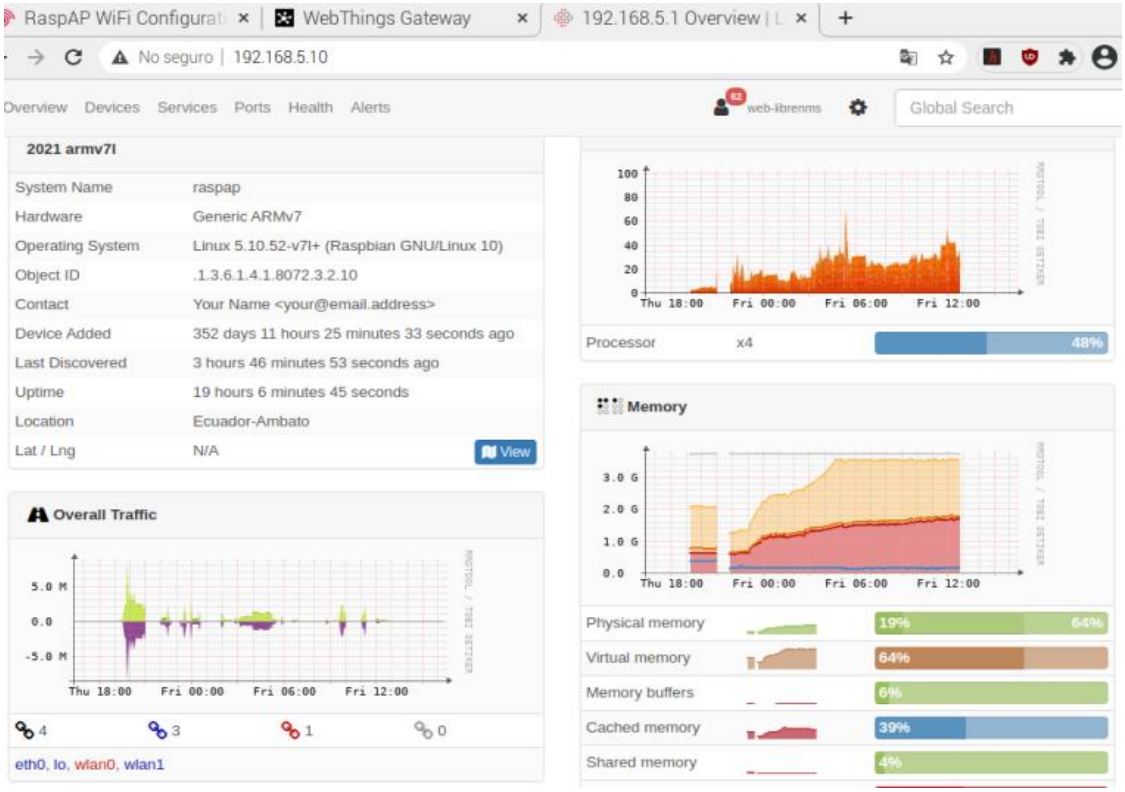
ANEXO D: VENTA DE VERIFICACIÓN DE LA CONFIGURACION DE LIBRENMS



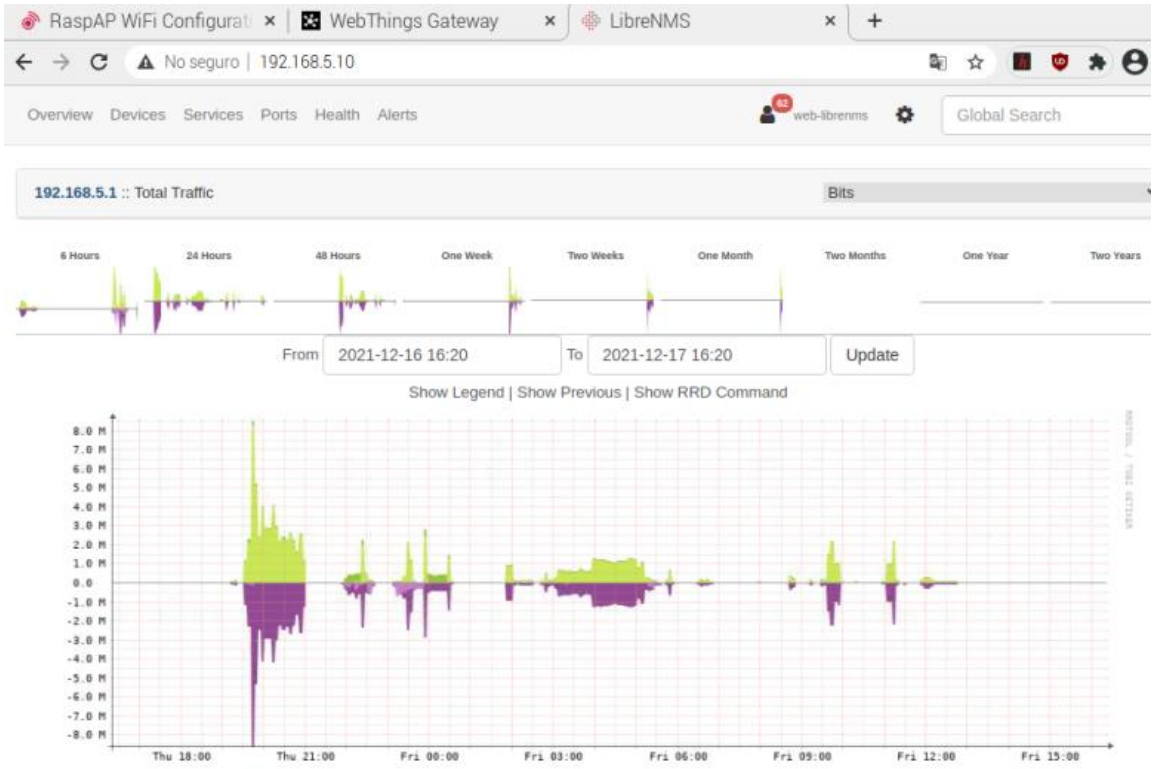
ANEXO E: VERIFICACIÓN DEL USUARIO PARA LIBRENMS



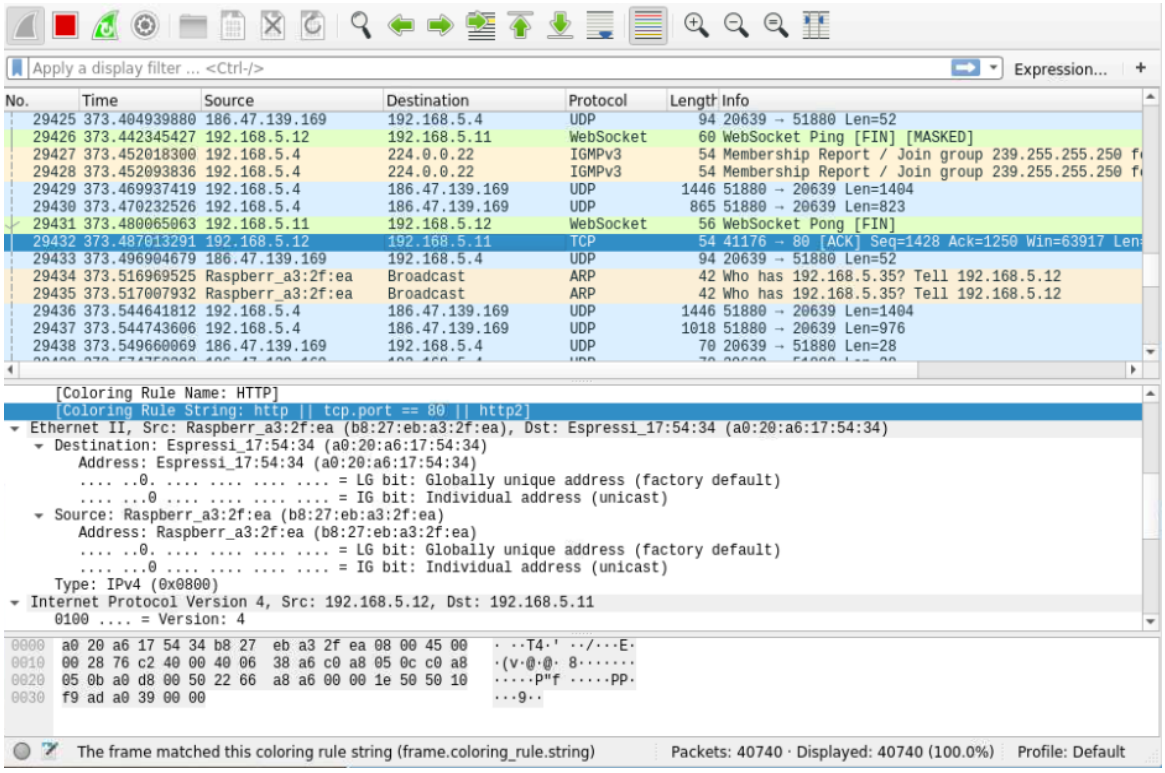
ANEXO F: VENTANA PRINCIPAL DE LA HERRAMIENTA LIBRENMS



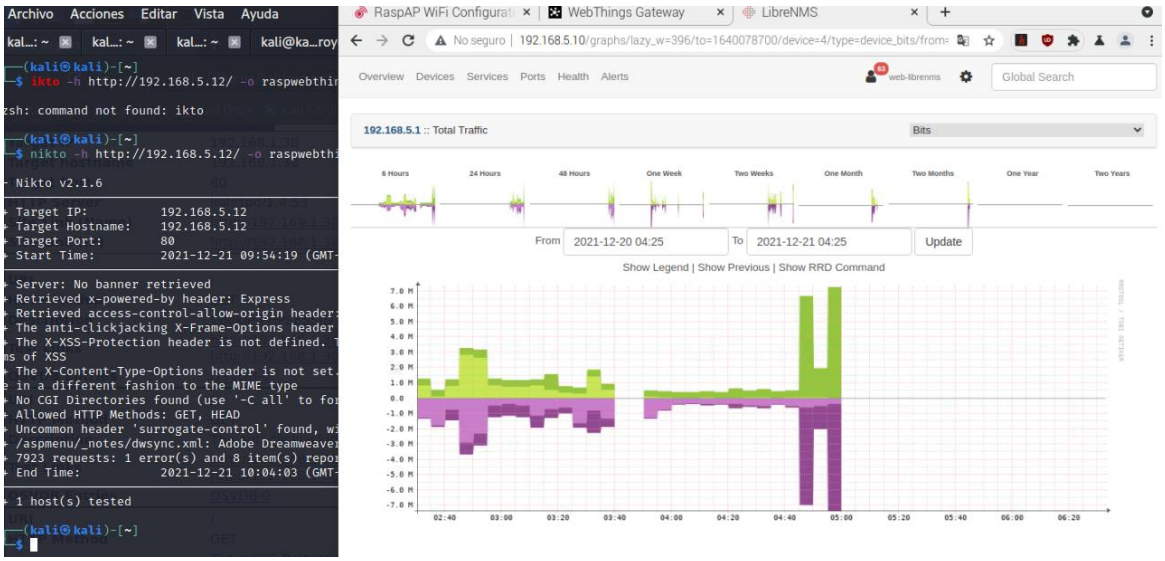
ANEXO G: GRÁFICA DEL COMPORTAMIENTO DE TRÁFICO DENTRO DE LA RED IOT.



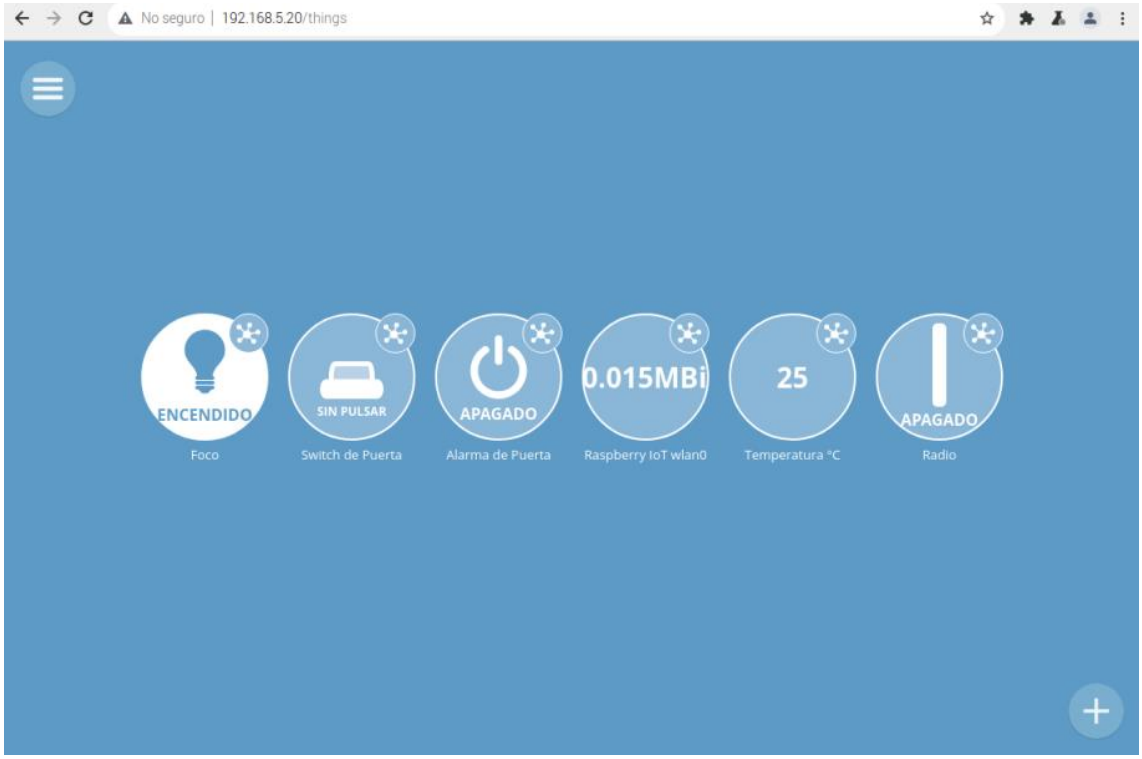
ANEXO H: ANALISIS DE LOS PAQUETES DEL MODULO ESP8266



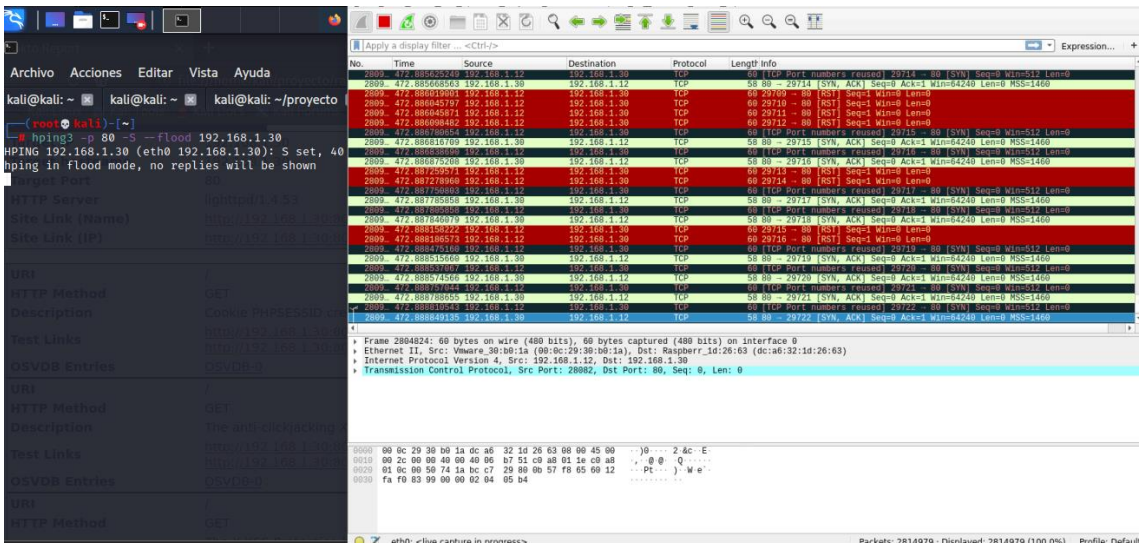
ANEXO F: EJECUCION DE COMANDOS PARA SIMULAR UN ATAQUE



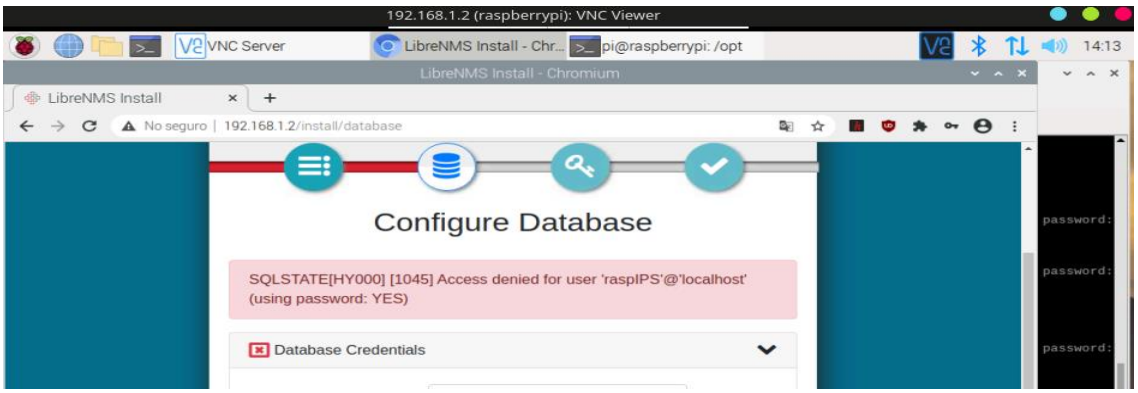
ANEXO G: PANEL DE CONTROL DE IOT DE WEBTHINGS GATEWAY



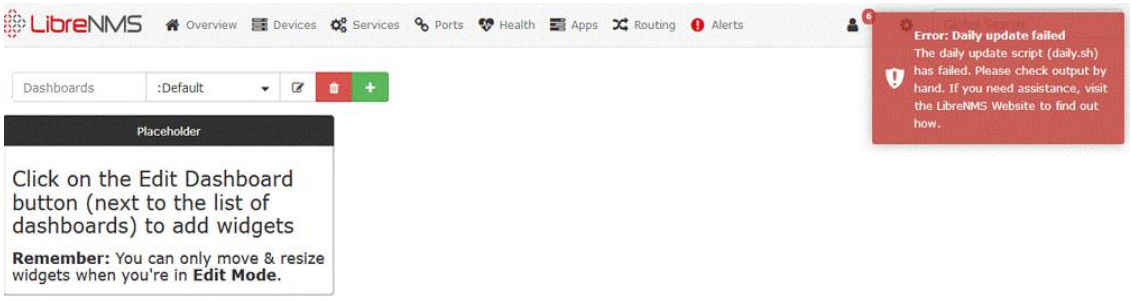
ANEXO H: PAQUETES DE UN ATAQUE DOS MEDIANTE WIRESHARK



ANEXO I: ERROR DEL SERVIDOR LIBRENMS AL CONECTARSE



ANEXO J: ERROR DE ACTUALIZACIÓN DEL SERVIDOR LIBRENMS



espoch | Dirección de Bibliotecas y Recursos del Aprendizaje

UNIDAD DE PROCESOS TÉCNICOS Y ANÁLISIS BIBLIOGRÁFICO Y DOCUMENTAL

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 28 / 09 / 2022

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: ANDRÉS MATEO ÁLVAREZ RAMÍREZ
INFORMACIÓN INSTITUCIONAL
Facultad: INFORMÁTICA Y ELECTRÓNICA
Carrera: TELECOMUNICACIONES
Título a optar: INGENIERA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES
f. Analista de Biblioteca responsable: Lcdo. Holger Ramos, MSc.

1880-DBRA-UPT-2022