Universidad CENFOTEC
Maestría Profesional en Ciberseguridad
Escuela de Ciberseguridad

Tema:

"Propuesta de un sistema de monitoreo y detección de eventos de seguridad para tráfico de red en dispositivos IoT residenciales utilizando una arquitectura cloud en AWS"

Elaborado por:
Oscar Arias Quesada

Abril 2026

Declaratoria de derechos de autor

Yo, Oscar Eduardo Arias Quesada, autor de este trabajo presentado para optar por el grado de Maestría en Ciberseguridad de la Universidad Cenfotec, declaro que soy el creador de este documento.

Autorizo la consulta y el uso de este documento para fines académicos y de investigación. Asimismo, queda prohibido cualquier uso o distribución de este material con fines de lucro o comerciales sin mi autorización expresa.

Agradecimientos

A mi hermana, por ser la voz que no me permitió rendirme. Gracias por tu insistencia constante, porque desde pequeños no te rendiste en tu empeño de que continuara mis estudios y por recordarme mi capacidad incluso en los momentos de mayor cansancio.

A mi esposa y a mi hijo, por ser mi refugio y mi mayor sacrificio. Gracias por su paciencia infinita durante las largas jornadas de trabajo en esta tesis y por comprender las ausencias que las clases nocturnas me impusieron. Ustedes son, sin duda, el motor que me impulsa a seguir adelante y la razón principal detrás de cada esfuerzo por superarme profesionalmente.

TRIBUNAL EXAMINADOR

Este proyecto fue aprobado por el Tribunal Examinador de la carrera: Maestría Profesional en Ciberseguridad y Seguridad de la Información, requisito para optar por el título de grado de **Maestría,** el estudiante Oscar Arias Quesada.

REBECA ESQUIVEL FLORES (FIRMA)
Firmado digitalmente por REBECA ESQUIVEL FLORES (FIRMA)
Fecha: 2026.06.23 19:15:49 -06'00'

M.Sc. Rebeca Esquivel Flores
Tutor

EDGAR RICARDO ZAMORA GATGENS (FIRMA)
Digitally signed by EDGAR RICARDO ZAMORA GATGENS (FIRMA)
Date: 2026.06.23 19:11:29 -06'00'

M.Sc. Edgar Zamora Gatgens
Lector 1

JASON ULLOA HERNANDEZ (FIRMA)
Firmado digitalmente por JASON ULLOA HERNANDEZ (FIRMA)
Fecha: 2026.06.29 14:41:57 -06'00'

M.Sc. Jason Ulloa Hernández
Lector 2

San José, Costa Rica, 23 de junio de 2026

Índice de Contenido

Índice de Tablas

Índice de Figuras

Abstract

En el contexto residencial actual, el crecimiento exponencial de dispositivos IoT ha multiplicado los vectores de ataque sobre las redes domésticas, exponiendo datos sensibles y recursos críticos a amenazas persistentes. Frente a esta realidad, la presente tesis propone un sistema integral de monitoreo y detección de eventos de seguridad centrado en el análisis de tráfico de red generado por dispositivos IoT residenciales. Utilizando servicios de AWS para ingestión de datos, se diseña una canalización escalable que captura, analiza y verifica flujos de red, permitiendo la identificación temprana de patrones anómalos mediante una serie de reglas predefinidas.

La propuesta contempla un módulo de gestión centralizada que analiza alertas, además de un sistema de notificaciones automatizadas ante incidentes críticos. El enfoque combina criterios de desempeño, costo y cumplimiento de buenas prácticas de seguridad en la nube, garantizando alta disponibilidad y tolerancia a fallos. Se valida la eficacia del sistema mediante pruebas de intrusión simuladas y escenarios de ataque reales, demostrando mejoras sustanciales en tiempo de detección y reducción de falsos positivos, lo cual refuerza la postura de defensa proactiva en entornos domésticos inteligentes.

Palabras Clave: IoT, monitoreo, tráfico de red, cloud, AWS.

Capítulo 1. Introducción

1.1 Generalidades

Este trabajo de investigación surge ante la acelerada integración de dispositivos del Internet de las Cosas (IoT) en los hogares costarricenses, fenómeno que ha derivado en una expansión crítica de los vectores de ataque disponibles para la ciberdelincuencia. Según la Information Systems Audit and Control Association (ISACA), la proliferación de estos nodos representa una amenaza latente debido a que gran parte de estos dispositivos son diseñados con controles de seguridad deficientes, lo que facilita el acceso no autorizado a las redes privadas y compromete la integridad del ecosistema digital residencial. (Oluleke Akinpelu, 2024)

1.2 Antecedentes del Problema

El aumento de dispositivos IoT en entornos residenciales ha transformado la vida cotidiana mediante la automatización, la eficiencia energética y la conectividad remota. Sin embargo, esta expansión ha traído consigo una serie de vulnerabilidades que comprometen la privacidad y seguridad de los usuarios. Aniketh Girish y Tianrui Hu en su investigación sobre dispositivos IoT, concluyeron que históricamente, los dispositivos IoT han sido vendidos con una configuración de seguridad por defecto inadecuada y capacidades innecesarias que pueden ser utilizadas indebidamente para permitir ataques a otros sistemas. (Girish, 2024)

Flórez Gutiérrez, Gordillo y Roa realizaron un análisis de literatura reciente en el que identificaron vulnerabilidades comunes en dispositivos IoT domésticos, tales como autenticación débil, falta de cifrado, fugas de información sensible y ausencia de controles de acceso basados en geolocalización. El estudio concluye que estas debilidades pueden ser explotadas por actores maliciosos para obtener datos privados o incluso alterar el funcionamiento de los dispositivos, generando riesgos físicos en el hogar. (Flórez Gutiérrez, 2022)

Por su parte, Gálvez Cisneros llevó a cabo una revisión sistemática de literatura sobre la ciberseguridad en dispositivos IoT de uso doméstico, utilizando el método PRISMA. Su estudio identificó como principales vulnerabilidades la falta de cifrado, contraseñas débiles y ausencia de actualizaciones regulares. Además, propone la adopción de técnicas de anonimización y pseudonimización para proteger

los datos personales, así como la estandarización de protocolos para mejorar la interoperabilidad y seguridad del ecosistema IoT residencial. (Gálvez Cisneros, 2025)

Estos estudios coinciden en que, aunque existen esfuerzos por parte de fabricantes y usuarios para mejorar la seguridad de los dispositivos IoT, los resultados aún son limitados o inconsistentes. En muchos casos, no se ha contratado personal especializado ni se han adoptado soluciones externas de forma sistemática, lo que deja un amplio margen para mejorar la protección del tráfico de red y los eventos de seguridad en entornos domésticos conectados a la nube.

1.3 Definición y Descripción del Problema

La acelerada adopción de dispositivos del Internet de las Cosas (IoT) en los hogares costarricenses ha superado drásticamente la capacidad de defensa de las arquitecturas de red domésticas convencionales (Oluleke Akinpelu, 2024), una situación que se agrava proporcionalmente al incremento en la exposición y penetración del Internet de banda ancha en las viviendas, la cual ya alcanza el 85.4% a nivel nacional (INEC, 2025). Esta vulnerabilidad técnica surge porque los enrutadores residenciales estándar carecen de la potencia de cómputo necesaria para ejecutar tareas críticas de seguridad; En su rol como infraestructura crítica de conectividad, los enrutadores presentan una superficie de ataque significativa que es frecuentemente explotada debido a una combinación de obsolescencia en el firmware, debilidad en los protocolos de cifrado y autenticación, y configuraciones deficientes en la gestión remota. Esta carencia sistemática de mecanismos de seguridad nativos en el dispositivo (on-device security) facilita su compromiso, permitiendo a los actores maliciosos reclutarlos en redes de bots (botnets) y utilizarlos como vector para el movimiento lateral y el control de los dispositivos IoT conectados a la red interna (Bitdefender, 2025)

La selección de Amazon Web Services (AWS) como plataforma de monitoreo de seguridad se justifica por su capacidad para superar las limitaciones de procesamiento de los entornos residenciales mediante el uso de AWS IoT Core. A diferencia de los dispositivos de red locales (on-premise) que dependen de firmas estáticas y carecen de memoria para el análisis histórico, AWS implementa modelos que permiten la detección de anomalías comportamentales en tiempo real,

identificando desviaciones en los patrones de tráfico que pasarían desapercibidas para un firewall convencional (Amazon Web Services, 2021)

1.4 Justificación

La propuesta concreta de un sistema de monitoreo y detección de eventos de seguridad para el tráfico de red de dispositivos IoT residenciales, implementado sobre una arquitectura cloud en AWS, se justifica por la persistente falta de visibilidad y detección temprana en redes domésticas conectadas, donde las comunicaciones locales y hacia la nube facilitan vectores de ataque y filtraciones de datos que actualmente pasan desapercibidos. Al incluir este monitoreo se podrá adoptar un sistema que detecte eventos de seguridad definidos (por ejemplo, escaneos ARP masivos, exfiltración vía SSDP/mDNS, servicios HTTP sin cifrado y comportamiento anómalo asociado a SDKs de terceros), reduzca el tiempo de detección y habilite respuestas técnicas reproducibles y auditables.

Aniketh Girish y Tianrui Hu nos muestran que la persistente falta de visibilidad y detección temprana en redes domésticas conectadas, donde las comunicaciones locales y hacia la nube facilitan vectores de ataque y filtraciones de datos que actualmente pasan desapercibidos, hace que una propuesta concreta de un sistema de monitoreo y detección de eventos de seguridad para el tráfico de red de dispositivos IoT residenciales (Girish, 2024)

1.5 Viabilidad

A pesar de que no existen muchas referencias específicas al respecto, existen estudios de diversos autores, con temas relacionados al IoT residencial (Gálvez Cisneros, 2025), (TrinitySec, 2024), (Girish, 2024). A continuación, se detallarán los puntos de vista técnicos, operativos y económicos por los que se consideran que el proyecto a desarrollarse es viable:

1.5.1 Punto de vista técnico

Este proyecto es viable debido a que se enfoca en herramientas ya existentes, como lo son AWS, IoT y redes, las cuales han sido objeto de estudio hasta cierto nivel durante el curso de la maestría. Sumado a lo anterior, el investigador cuenta con más de 10 años de experiencia en el área de TI, trabajando con estas y otras tecnologías, lo cual le brinda las habilidades técnicas necesarias para un adecuado desarrollo del proyecto.

1.5.2 Punto de vista operativo

Al ser este un proyecto de investigación abarcando únicamente el área residencial, el investigador cuenta con los elementos necesarios para desarrollar pruebas acordes y realizar implementaciones según sea el caso; este acceso se cuenta desde su casa hasta la casa de familiares, ya que estos a su vez cuentan con elementos IoT con los que se pueden realizar las pruebas requeridas.

1.5.3 Punto de vista económico

Económicamente, la propuesta es altamente eficiente. Al utilizar un modelo de arquitectura serverless, los costos se limitan estrictamente al volumen de datos procesados. La implementación de una estrategia de agregación de datos (flujo 3-a-1) en el gateway local reduce significativamente los mensajes enviados a la nube, optimizando el uso de la capa gratuita de AWS o manteniendo los cargos mensuales en niveles mínimos para un hogar promedio. Además, el uso de hardware accesible (como una Raspberry Pi o una computadora reutilizada) evita inversiones iniciales elevadas.

1.6 Objetivos

Para la definición de los objetivos, se utiliza la taxonomía de Bloom, creada en 1956 y la cual a través de los años ha demostrado ser la más completa, tanto por su documentación disponible como por su estructura que facilita la definición de los objetivos.

1.6.1 Objetivo general

- Proponer un sistema de monitoreo y detección de eventos de seguridad para tráfico de red en dispositivos IoT residenciales utilizando una arquitectura cloud en AWS.

1.6.2 Objetivos específicos

- Implementar un módulo de captura y preprocesamiento ligero, capaz de recolectar y anonimizar el tráfico de red residencial de los dispositivos IoT, asegurando una transmisión segura y eficiente hacia la infraestructura cloud en AWS.
- Analizar el comportamiento intrínseco de comunicación (fingerprinting) de al menos tres categorías comunes de dispositivos IoT residenciales (ej., cámaras de vigilancia, smart plugs y bombillos inteligentes) para clasificar las

actividades de red en patrones de operación legítimos y desviaciones indicativas de una amenaza.

- Diseñar una arquitectura de Detección de Intrusiones (IDS) basada en AWS que integre los servicios nativos de la nube (ej., AWS IoT Core, Lambda, Cloudwatch) para la monitorización y la correlación de eventos, proponiendo un modelo de datos escalable para el almacenamiento del tráfico de red residencial.

1.7 Alcances y Limitaciones

1.7.1 Alcances

Desplegar un sistema de monitoreo y detección de eventos, utilizando herramientas existentes en AWS, por lo que se centra en la integración de arquitectura en la nube y orquestación de servicios. Además, se estarán utilizando para esta investigación, las amenazas más comunes que se pueden encontrar a nivel residencial, por lo que se excluyen ataques avanzados, de zero-day o ataques de ingeniería social.

1.7.2 Limitaciones

Se excluyen explícitamente los ataques de día cero y las técnicas de ingeniería social, dado que el enfoque se centra exclusivamente en el comportamiento del tráfico de red y no en el factor humano. En este contexto, el sistema se restringe al análisis de metadatos y patrones de flujo (flow-based analysis), omitiendo la inspección de contenido en paquetes cifrados (SSL/TLS) cuando no se dispone de la clave privada. Asimismo, a nivel de infraestructura, no se contempla el diseño de nuevos dispositivos físicos, basándose el proyecto estrictamente en la utilización de hardware comercial existente de dispositivos IoT.

1.8 Marco de Referencia Organizacional y Socioeconómico

En los últimos años, el concepto de casa inteligente ha dejado de ser un servicio exclusivo de los sectores con más poder adquisitivo para convertirse en una realidad accesible a todos. Esto se debe a una disminución de los costos de los diferentes dispositivos IoT, así como a las diferentes formas que se ofrecen en el mercado de servicios de conectividad siendo uno de ellos plataformas en la nube.

1.8.1 Historia

Dada la evolución tecnológica que se ha vivido en los últimos años con el auge del concepto del Internet de las cosas, se ha transformado el panorama de la vida cotidiana en la mayoría de los hogares costarricenses convirtiendo viviendas tradicionales en casas inteligentes. Esto se logra gracias a que las redes domésticas permiten que múltiples dispositivos como computadoras personales, teléfonos inteligentes, asistentes virtuales, cámaras de seguridad, electrodomésticos inteligentes y sistemas de entretenimiento etc. se interconectan de forma fácil y rápida. De forma paralela los servicios en la nube han evolucionado para ofrecer plataformas escalables, flexibles y accesibles que permiten implementar soluciones de seguridad y análisis de datos de manera ágil.

Este escenario evidencia la necesidad de brindar a los hogares modernos mecanismos de protección confiables y fáciles de usar, que los mantengan protegidos de posibles ataques, identificando las principales vulnerabilidades de las diferentes redes que se utilizan en el hogar.

1.8.2 Tipo de Negocio y Mercado Meta

Este proyecto abarca un ámbito de ciberseguridad residencial pensado como un sector emergente dado el crecimiento en digitalización de los hogares, por lo que se busca ofrecer una solución que permita proteger los dispositivos IoT conectados a las redes WIFI de cualquier casa que posea dichos aparatos.

Por consiguiente, la solución ha sido diseñada para satisfacer los requerimientos de usuarios domésticos de diversos rangos de edades. Dado que se busca poder proteger a todos los usuarios de la vivienda se piensa en una solución que sea accesible e intuitiva para comenzar a crear conciencia de las buenas prácticas en ciberseguridad para los miembros de toda la familia.

1.8.3 Misión, Visión y Valores

A pesar de que el proyecto está definido a un entorno residencial, se busca tener un enfoque de que un hogar sea visto como una unidad operativa tecnológica ya que se gestionan recursos digitales por medio de los dispositivos conectados.

Siendo así se tiene:

Misión

Promover la seguridad digital en el entorno residencial mediante la implementación de buenas prácticas y tecnologías accesibles que fortalezcan la protección de los datos, dispositivos y usuarios del hogar.

Visión

Ser un modelo de referencia en la aplicación de ciberseguridad doméstica, contribuyendo a la creación de hogares digitalmente seguros y conscientes de los riesgos tecnológicos emergentes.

Valores

- **Responsabilidad digital**: Uso ético y seguro de la tecnología en el hogar.
- **Accesibilidad**: Soluciones aplicables sin necesidad de conocimientos técnicos avanzados.
- **Privacidad**: Protección de la información personal de todos los miembros del hogar.

1.9 Estado de la cuestión

La ciberseguridad residencial ha ganado relevancia mundial ante el crecimiento de dispositivos conectados en un hogar, lo que ha generado nuevas vulnerabilidades. Diversos estudios han abordado la protección de redes domésticas mediante técnicas como minería de datos y aprendizaje automático, pero aún existe una brecha en la sistematización de estos enfoques. Por ello, este proyecto plantea una revisión sistemática que identifica investigaciones relevantes, define criterios de inclusión y exclusión, y selecciona fuentes académicas confiables. Los resultados permiten establecer patrones comunes y proponer estrategias efectivas para fortalecer la seguridad digital en entornos residenciales.

1.9.1 Planificación de la revisión

Se formula en esta etapa una pregunta clara y bien definida del tema de investigación. Se hace una búsqueda de documentación existente con el objetivo de conocer el desarrollo académico que existe, posibles áreas débiles que se puedan ampliar y verificar que no se esté duplicando los estudios realizados en otras investigaciones.

1.9.1.1 Formulación de la pregunta

La formulación de la pregunta de investigación tiene como propósito delimitar el alcance de la búsqueda bibliográfica, permitiendo la revisión sistemática de estudios que aporten al campo de la ciberseguridad residencial. Al definir de forma precisa la pregunta, se facilita la identificación de trabajos pertinentes, se optimiza el proceso de selección de fuentes y se garantiza que los resultados obtenidos tengan un impacto significativo en el marco conceptual y metodológico de la tesis.

1.9.1.1.1 Foco de la pregunta

El alcance de la pregunta debe centrarse en identificar las técnicas que se han aplicado para proteger los hogares digitales y casas inteligentes. Al delimitar la búsqueda de hacia estudios que traten soluciones prácticas y teóricas en redes domésticas se logra un enfoque más integral.

1.9.1.1.2 Amplitud y calidad de la pregunta

En esta sección se crea la pregunta de investigación, que responderá en forma objetiva cuáles son los desarrollos actuales respecto a sistemas de monitoreo y detección de eventos de seguridad en sistemas IoT. Para ello se elabora un listado de términos relevantes que son considerados componentes clave y que facilitan la búsqueda de información.

1. Problema

La falta de monitoreo y detección de eventos de seguridad en el tráfico de red generado por dispositivos IoT residenciales constituye una brecha crítica en la ciberseguridad doméstica. Esto expone a los usuarios a riesgos como la interceptación de datos, secuestro de dispositivos, etc. Aunque los hogares pueden ahora incluir rápidamente dispositivos IoT estos suelen carecer de mecanismos para auditar o alertar ante la presencia de conexiones fuera del seno familiar.

2. Pregunta

Una vez definido el problema, se formula la pregunta de investigación:
¿Cuáles son los desarrollos actuales respecto a sistemas de monitoreo y detección de eventos de seguridad en sistemas IoT?

3. Palabras clave y sinónimos

Con el fin de facilitar la búsqueda de documentos se elabora la siguiente lista de palabras claves que se van a utilizar para la búsqueda de identificación de documentos y trabajos relacionados con la investigación; algunas de estas palabras

están en el idioma inglés, ya que hay unas varias publicaciones en ese idioma, las mismas se muestran en la tabla 1.

A continuación, un listado de estas palabras:

Tabla 1. Listado de palabras

Palabra en español	**Equivalente en inglés**	**Sinónimos / Términos Relacionados**
Internet de las cosas	Internet of Things (IoT)	Dispositivos inteligentes, tecnología conectada
Dispositivos inteligentes	Smart Devices	Tecnología IoT, aparatos conectados
Monitoreo	Monitoring	Supervisión, vigilancia, observación continua
Tráfico de red	Network Traffic	Flujo de datos, comunicación de red, paquetes de red
Seguridad	Security	Protección, defensa digital, ciberseguridad
Ciberseguridad	Cybersecurity	Seguridad informática, protección de datos
Arquitectura en la nube	Cloud Architecture	Infraestructura cloud, diseño distribuido
Amazon Web Services	Amazon Web Services (AWS)	Servicios cloud, infraestructura escalable
Eventos de seguridad	Security Events	Incidentes, alertas, anomalías, amenazas
Intrusión	Intrusión	Ataque, acceso no autorizado, penetración
Sistema de detección	Detection System	IDS, sistema de monitoreo, análisis de tráfico

Fingerprinting	Fingerprinting	Huella digital de comportamiento, perfil de tráfico
Anonimización	Anonymization	Ocultación de identidad, anonimato de datos
Pseudonimización	Pseudonymization	Sustitución de identidad, protección de datos personales
Vulnerabilidad	Vulnerability	Debilidad, riesgo, exposición
Cifrado	Encryption	Encriptación, codificación de datos
Autenticación	Authentication	Verificación de identidad, control de acceso
Panel de control	Dashboard	Interfaz gráfica, consola de monitoreo
Alta disponibilidad	High Availability	Tolerancia a fallos, redundancia
Pruebas de intrusión	Penetration Testing	Simulación de ataques, evaluación de seguridad
Red de bots	Botnet	Red de dispositivos comprometidos, ataques distribuidos
Comunicación local	Local Communication	Tráfico interno, red doméstica
Comunicación con la nube	Cloud Communication	Interacción cloud, tráfico hacia servicios cloud
Notificaciones automáticas	Automated Notifications	Alertas automáticas, avisos de seguridad
SDK	Software Development Kit	Kit de desarrollo, herramientas de programación

Protocolo de respuesta	Incident Response Protocol (IRP)	IRP, plan de contingencia
Escaneo ARP	ARP Scanning	Detección de dispositivos, descubrimiento de red
SSDP	Simple Service Discovery Protocol	Descubrimiento de servicios, protocolo SSDP
mDNS	Multicast DNS	Resolución de nombres local, protocolo mDNS
Clasificación de tráfico	Traffic Classification	Segmentación de tráfico, análisis de red

Fuente: Elaboración propia

4. Intervención

Extraer los artículos y documentos de mayor relevancia para la investigación y analizar los resultados obtenidos para determinar cómo se han desarrollado formas o sistemas de monitoreo ante eventos de seguridad en redes domésticas que tienen dispositivos IoT.

5. Control

Durante el inicio de la investigación se cuenta con un conocimiento básico adquirido por el autor de la tesis durante sus años como profesional a cargo de sistemas de redes empresariales y también como un usuario de dispositivos IoT en su casa de habitación. Sin embargo, muchos términos y conceptos tuvieron que ser buscados desde cero**.**

6. Efectos

Se espera contar con la documentación relevante a partir de las búsquedas realizadas, que permita comprender las metodologías más efectivas en el análisis de tráfico de red y detección de eventos de seguridad en entornos domésticos con dispositivos IoT. Esto incluye identificar técnicas de minería de datos aplicables a la clasificación de comportamientos anómalos, así como conocer los esfuerzos previos realizados en Costa Rica en este campo.

7. Medida de salida

Para la documentación encontrada se realiza una revisión de la calidad y aportes de esta en sitios web especializados para tal fin.

8. Población

La población de estudio está constituida por usuarios de dispositivos IoT en entornos residenciales de Costa Rica, que dispongan de infraestructura de red Wi-Fi doméstica.

9. Aplicación

Los resultados de la investigación pueden ser útiles para las personas interesadas en fortalecer la seguridad digital en su hogar, de los distintos dispositivos conectados a la red de la casa que actualmente no posean ningún sistema de monitoreo ni detección de eventos.

10. Diseño Experimental

Durante el diseño experimental se realiza un análisis exhaustivo y clasificación de los estudios obtenidos, priorizando la calidad del contenido y su relevancia para los objetivos planteados. Esto permite conservar únicamente la documentación más confiable y útil para el desarrollo de la investigación. Así, se garantiza una base sólida y pertinente para la propuesta del sistema de monitoreo en entornos IoT residenciales.

1.9.1.2 Selección de fuentes

Esta sección detalla las fuentes utilizadas para identificar estudios primarios relevantes que respaldan el desarrollo de la investigación.

1.9.1.2.1 Definición del criterio de selección de fuentes

Para seleccionar las fuentes se consideraron las fuentes reconocidas en el ámbito de la ciberseguridad, redes, IoT y computación en la nube. Se toman primeramente las que cuentan con respaldo académico y técnico que ofrezcan además documentación actualizada.

1.9.1.2.2 Lenguaje de estudio

Las búsquedas bibliográficas se realizan tanto en español como en inglés, con el fin de ampliar los resultados y acceder a investigaciones internacionales que complementen el análisis local.

1.9.1.2.3 Identificación de fuentes

Aquí se detalla la selección de las fuentes, una descripción de cómo se realizaron las búsquedas y cuáles fueron las fuentes utilizadas.

1. Método de selección de fuentes:

El método de selección se basa en la evaluación del respaldo técnico y académico de cada fuente, su especialización en temas de tecnología y ciberseguridad, y la facilidad para acceder a sus contenidos. Se priorizan artículos, tesis, informes técnicos y publicaciones institucionales que aporten evidencia sólida para la investigación.

2. Cadena de búsqueda:

Las cadenas de búsqueda utilizadas tienen combinación de “OR” y “AND”.

("IoT" OR "Internet de las cosas") AND ("detección de eventos de seguridad" OR "security event detection") AND ("tráfico de red" OR "network traffic") AND ("residencial" OR "residential") AND ("AWS" OR "Amazon Web Services") AND ("implementación" OR "implementation" OR "arquitectura cloud" OR "cloud architecture") AND ("algoritmos" OR "algorithms") AND ("técnicas" OR "techniques")

3. Lista de fuentes

Para el desarrollo de la investigación se utilizan las siguientes fuentes, estas garantizan la calidad de los artículos académicos y publicaciones recientes:

1. IEEE Xplore
2. Scopus
3. SpringerLink
4. Repositorio PUCE (Pontificia Universidad Católica del Ecuador)
5. Ciencia Latina – Revista Científica Multidisciplinar
6. Revista Científica Arbitrada Multidisciplinaria PENTACIENCIAS (PUCE)
7. RIIT (Revista Internacional de Investigación e Innovación Tecnológica)

1.9.1.2.4 Selección de fuentes después de la evaluación

Un aspecto que se toma en cuenta para la selección de fuentes es la facilidad que se tiene para acceder al material, así como la calidad de los documentos encontrados en aras de obtener los que aporten más al desarrollo del proyecto.

1.9.1.2.5 Comprobación de las fuentes

En este momento no se cuenta con criterio experto para la selección de las fuentes; sin embargo, al ser librerías de universidades importantes en sus países e instituciones reconocidas a nivel mundial se garantiza la calidad de estas.

1.9.1.3 Selección de los estudios

Una vez que se ha realizado el proceso de selección de fuentes se definen cuáles de estas van a ser incluidas en el análisis final.

1.9.1.3.1 Definición del criterio de inclusión y exclusión de estudios

Se utilizan los criterios detallados en la tabla 2, para incluir o excluir un documento. Los artículos que cumplan los requisitos son candidatos para incluir.

Tabla 2. Criterios de inclusión y exclusión de documentos

Pregunta de la investigación	**Términos principales para criterio de inclusión**	**Criterios de exclusión**
¿Qué trabajos se han realizado en el área de monitoreo y detección de eventos de seguridad en dispositivos IoT residenciales utilizando arquitectura cloud?	"IoT", "Internet de las cosas", "tráfico de red", "detección de eventos", "seguridad", "AWS", "cloud", "minería de datos", "algoritmos", "técnicas"	- Documentos sobre IoT sin relación con seguridad o monitoreo de tráfico de red. - Estudios sobre seguridad en redes empresariales o industriales. Publicaciones que no incluyan arquitectura cloud o servicios como AWS.

Fuente: Elaboración propia.

1.9.1.3.2 Definición de tipos de estudio

La definición de los tipos de estudios está relacionada con la pregunta de investigación, se crea la siguiente tabla para determinar los requisitos para definir los artículos de interés.

Tabla 3. Tipos de preguntas para definir artículos

Pregunta de investigación	**¿Quién?**	**¿Qué?**	**¿Cómo?**	**¿Dónde?**

¿Qué trabajos se han realizado en el área de monitoreo y detección de eventos de seguridad en dispositivos IoT residenciales utilizando arquitectura cloud?	Usuarios residenciales y fabricantes de dispositivos IoT	Dispositivos IoT, tráfico de red, eventos de seguridad, arquitectura cloud, AWS	Monitoreo, detección, clasificación, correlación de eventos, respuesta automatizada	Redes domésticas, entornos residenciales, plataformas cloud como AWS

Fuente: Elaboración propia.

1.9.1.3.3 Procedimiento para la selección de los estudios

Para la selección de estudios relevantes en el contexto de ciberseguridad aplicada a dispositivos IoT residenciales, se realizó el siguiente proceso iterativo por cada fuente consultada:

1. Se utilizó la opción de búsqueda avanzada o general disponible en cada base de datos académica seleccionada.
2. Se aplicaron cadenas de búsqueda bilingües (español e inglés) relacionadas con IoT, tráfico de red, eventos de seguridad y arquitectura cloud.
3. Si la cantidad de resultados superó los 50 documentos, se aplicaron filtros adicionales como rangos de fechas, tipo de publicación y área temática.
4. Se evaluaron los resultados mediante la lectura del abstract y las palabras clave, aplicando los criterios de exclusión definidos previamente.
5. Se seleccionaron los estudios considerados relevantes y se repitió el proceso en las demás fuentes disponibles.

Figura 1. Procedimiento para la selección de los estudios (Elaboración Propia).

1.9.2 Ejecución de la revisión.

Luego de finalizar la búsqueda se encuentran 4 trabajos con contenido muy relevante para el desarrollo de la investigación, los mismos aportan conceptos y métodos utilizados en sistemas relacionados con dispositivos IoT y ciberseguridad. En la tabla 4 se detallan los artículos seleccionados:

Tabla 4. Estudios encontrados

#	Título	Autores	Año	URL
1	Ciberseguridad del internet de las cosas (IoT) en el sector doméstico y su estado actual en Colombia	Eugenio José Gómez Mendoza	2024	https://repository.unad.edu.co/bitstream/handle/10596/63551/ejgomezmen.pdf?isAllowed=y&sequence=1
2	Pruebas de Seguridad Aplicadas a Infraestructura IoT	Kevin Andrés Babativa Santos	2023	https://repositorio.uniandes.edu.co/bitstreams/348a8317-b588-43a4-80d1-4cf93a4dd21c/download
3	Ciberseguridad en los dispositivos IoT de uso doméstico: una revisión sistemática de la literatura	Xavier Alexander Gálvez Cisnero	2025	https://editorialalema.org/index.php/pentaciencias/article/view/1371/1846

4	Implementación de IoT y ciberseguridad caso de estudio conjunto residencial “Campos de Arcadia”	Anthony Mihael Rivadeneira Erazo Javier y Alejandro Zúñiga Salvador	**2022**	https://repositorio.puce.edu.ec/server/api/core/bitstreams/98fc1111-b2c8-4125-851b-3227dada24b1/content

Fuente: Elaboración propia.

1.9.2.1 Revisión de la selección

Luego de realizar una revisión general del contenido de los trabajos se logra seleccionar los estudios más relevantes los cuales fueron ordenados con base en el aporte que ofrecen a la investigación y al cumplimiento de los objetivos planteados.

1.9.2.2 Extracción de información

Con el objetivo de lograr extraer toda la información relevante de cada estudio se realizaron resúmenes, nube de conceptos y extracción de conceptos claves. A continuación, se detallan los estudios:

Tabla 5. Extracción fuente 1

Elemento	**Elemento**
Repositorio	Universidad Nacional Abierta y a Distancia – UNAD
Publicación	Proyecto de grado – Monografía, Montería, 2024
Autor	Eugenio José Gómez Mendoza
Título	Ciberseguridad del Internet de las Cosas (IoT) en el sector doméstico y su estado actual en Colombia.
Área	Ciberseguridad, Internet de las Cosas (IoT), Seguridad Informática, Tecnología Domótica
Resumen	El documento analiza el crecimiento del IoT en Colombia, sus aplicaciones en los hogares, los riesgos de seguridad asociados, y propone medidas para mitigar las vulnerabilidades. Se destaca la necesidad de fortalecer la

	infraestructura, la adopción de buenas prácticas y la concientización del usuario.
Aspectos a destacar	- Evaluación del estado del arte de IoT en Colombia - Identificación de riesgos en hogares inteligentes - Recomendaciones de seguridad para dispositivos IoT - Análisis de infraestructura tecnológica nacional - Propuesta de medidas preventivas y correctivas - Inclusión de estándares como ZigBee, MQTT, WirelessHART - Referencias extensas y actualizadas sobre tecnologías emergentes

Fuente: Elaboración propia.

Tabla 6. Extracción fuente 2

Elemento	**Elemento**
Repositorio	Universidad de los Andes
Publicación	Tesis de grado – Ingeniería de Sistemas y Computación, Bogotá D.C., 2023
Autor	Kevin Andrés Babativa Santos
Título	Pruebas de Seguridad Aplicadas a Infraestructura IoT
Área	Ciberseguridad, Internet de las Cosas (IoT), LoRa, LoRaWAN, Seguridad en redes
Resumen	El trabajo analiza las vulnerabilidades de la tecnología LoRa en entornos IoT, mediante pruebas prácticas en infraestructura controlada. Se identifican ataques como sniffing, replay y jamming, y se proponen soluciones como cifrado simétrico y autenticación por clave única. Se destaca la importancia de la seguridad en redes de bajo consumo y largo alcance.

Aspectos a destacar	- Evaluación de vulnerabilidades en redes LoRa - Pruebas prácticas de ataques en entorno controlado - Implementación de sniffing, replay y ataques híbridos - Propuesta de soluciones como cifrado y autenticación - Uso de dispositivos especializados como CatSniffer y HackRF - Relevancia de LoRa en aplicaciones industriales y estatales

Fuente: Elaboración propia.

Tabla 7. Extracción fuente 3

Elemento	**Elemento**
Repositorio	Revista Científica Arbitrada Multidisciplinaria PENTACIENCIAS
Publicación	Artículo científico – Vol. 7, Núm. 1, enero-marzo 2025
Autores	Xavier Alexander Gálvez Cisneros y Darío Javier Robayo Jacome
Título	Ciberseguridad en los dispositivos IoT de uso doméstico: una Revisión Sistemática de la Literatura
Área	Ciberseguridad, Internet de las Cosas (IoT), Automatización del hogar, Privacidad de datos, Interoperabilidad
Resumen	El estudio realiza una revisión sistemática sobre la seguridad de dispositivos IoT en hogares, identificando vulnerabilidades como contraseñas débiles, falta de cifrado y actualizaciones. Se proponen medidas como cifrado, autenticación fuerte y estándares comunes. También se analizan beneficios como eficiencia energética y limitaciones como costos, privacidad y falta de interoperabilidad.

Aspectos a destacar	- Aplicación del método PRISMA para revisión sistemática - Vulnerabilidades comunes: contraseñas débiles, firmware inseguro, falta de cifrado - Medidas de mitigación: cifrado, actualizaciones, autenticación 2FA - Beneficios: automatización, eficiencia energética, seguridad - Limitaciones: privacidad, costos, dependencia de conectividad - Análisis de técnicas de anonimización y pseudonimización - Recomendaciones para trabajos futuros en regulación, resiliencia y plataformas inteligentes

Fuente: Elaboración propia.

Tabla 8. Extracción fuente 4

Elemento	**Elemento**
Repositorio	Pontificia Universidad Católica del Ecuador
Publicación	Proyecto de titulación – Ingeniería en Sistemas y Computación, Quito DM, 2022
Autores	Anthony Mihael Rivadeneira Erazo, Javier Alejandro Zúñiga Salvado
Título	Implementación de IoT y Ciberseguridad: Caso de Estudio Conjunto Residencial “Campos de Arcadia”
Área	Ciberseguridad, Internet de las Cosas (IoT), Domótica, Redes, Aplicaciones Android
Resumen	El proyecto implementa dispositivos IoT para automatización y seguridad en un conjunto residencial, integrando sensores, cámaras, aspersores y controles de luz, gestionados mediante una aplicación Android. Se aplican medidas de ciberseguridad

	a nivel de red, dispositivos y aplicación, incluyendo protocolos seguros, autenticación y prevención de ataques.
Aspectos a destacar	- Arquitectura completa de solución IoT + ciberseguridad - Desarrollo de aplicación Android con login seguro - Prevención de ataques DNS, ARP Spoofing y DoS - Diseño e instalación de hardware personalizado (PCB, sensores, cámaras) - Evaluación de satisfacción de usuarios y escalabilidad del sistema

Fuente: Elaboración propia.

Capítulo 2. Marco conceptual

Basándose en los datos obtenidos luego de realizar el estado de la cuestión; se identificaron las palabras claves y se generó una nube de conceptos que permite visualizar de forma visual las palabras más relevantes mencionadas en los diferentes artículos incluidos en el artículo anterior.

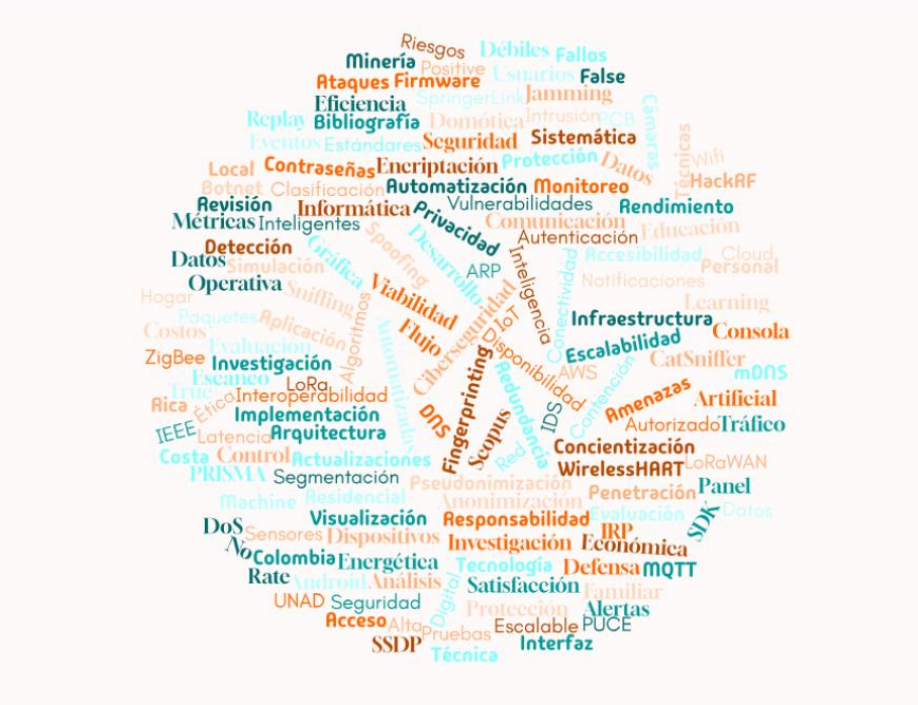

Figura 2. Nube de palabras generado (Elaboración propia.)

2.1 Conceptos sobre contenido

Con base en los hallazgos del estado de la cuestión y la nube de conceptos generada, se pretende aclarar los conceptos clave relacionados con la ciberseguridad en entornos residenciales, el monitoreo de tráfico de red, y el uso de arquitecturas cloud para la detección de eventos de seguridad en dispositivos IoT.

Tomando como punto de partida el contexto de lo que significa red doméstica; el contenido digital está continuamente generando y transmitiendo entre todos los dispositivos IoT de una casa, entre plataformas cloud y entre usuarios de la misma red.

Al referirse a contenido en redes IoT residenciales se tienen 4 tipos de contenido los cuales podemos observar en la figura 3 y son:

- **Contenido estructurado**: es fundamental para realizar el monitoreo, análisis y correlación de los diferentes eventos de seguridad que se pueden presentar en un ambiente doméstico. Su formato es tabular, basado en registros y es fácilmente almacenable en bases de datos; además permite utilizar técnicas de minería de datos. Algunos ejemplos de este tipo de contenido son:
    - Logs de eventos: son registros cronológicos generados por dispositivos IoT que documentan acciones como encendido/apagado, cambios de configuración, accesos remotos, etc.
    - Métricas de rendimiento: corresponde a datos como temperatura, consumo energético, tiempo de actividad, latencia de respuesta, número de paquetes enviados/recibidos, etc.
    - Registros de conexión: Información sobre las conexiones de red establecidas por los dispositivos, incluyendo IP de origen/destino, puertos utilizados, duración de la sesión, protocolo empleado (TCP, UDP), y frecuencia de conexión.
- **Contenido no estructurado**: utilizamos este concepto para datos que no siguen un formato predefinido ni una estructura fija, lo que dificulta su procesamiento automático sin técnicas avanzadas de análisis también se debe considerar que estos datos no se almacenan en tablas ni bases de datos relacionales. En redes domésticas este tipo de contenido es generado por los dispositivos inteligentes que interactúan con el entorno físico y con los usuarios de la vivienda. Algunos ejemplos de este tipo de contenido son:
    - Imágenes: entre ellas están Capturas de cámaras de seguridad, visualizaciones térmicas, fotos tomadas por sensores visuales, etc.
    - Audio: Grabaciones de asistentes virtuales, comandos de voz, sonidos ambientales, etc.
    - Video: Transmisiones en vivo desde cámaras IP, grabaciones de vigilancia, etc.
    - Comandos en texto libre: son enviados por los usuarios, mensajes de error, respuestas de dispositivos que no siguen un formato estándar, etc.
- **Contenido semi estructurado:** este tipo de datos no siguen una estructura rígida como las bases de datos relacionales, pero sí contiene algunas etiquetas, delimitadoras o formatos que permiten cierta organización y análisis

automatizado. Para la parte residencial se pueden obtener estos datos en la configuración, comunicación y reporte de dispositivos inteligentes. Algunos ejemplos de este tipo de contenido son:

- JSON: por sus siglas en inglés significa JavaScript Object Notation y es utilizado en transmisiones de configuración, estados de dispositivos, respuestas de APIs, y eventos de seguridad. Es legible tanto por humanos como por máquinas.
- XML: por sus siglas en inglés significa “Extensible Markup Language". Corresponde a un protocolo de comunicación, reportes de estado y configuración de dispositivos. Aunque es más pesado que JSON, es ampliamente usado en sistemas.
- YAML: es utilizado en configuraciones de sistemas cloud o scripts de automatización.
- CSV: si bien sus encabezados son técnicamente tabulares se considera semiestructurados si los datos varían en formato o longitud.
- AWS: al utilizar una estructura cloud se tiene una parte de Core que es la que recibe mensajes en formato JSON desde dispositivos; una parte de Lambda que ejecuta funciones para interpretar y actuar sobre datos semiestructurados; luego lo relacionado a DynamoDB o S3 es el encargado de almacenar configuraciones y eventos en formatos flexibles y por último se cuenta con CloudWatch que es lo que recibe logs en JSON para visualización y análisis.

- **Contenido de red**: se refiere a los datos que se transmiten entre dispositivos IoT, routers, servidores cloud y otros elementos conectados dentro de una red doméstica. Este contenido está compuesto por paquetes de datos que viajan a través de protocolos de comunicación, y su análisis es esencial para detectar eventos de seguridad, anomalías y comportamientos maliciosos. Algunos ejemplos de este tipo de contenido son:
  - Paquetes TCP/IP: Son la base de la comunicación en redes. Incluyen información sobre origen, destino, puertos, tamaño, y tipo de protocolo.
  - Protocolos IoT: incluyen los principales protocolos utilizados en redes residenciales como: SSDP que por sus siglas en inglés significa Simple Service Discovery Protocol, es utilizado por dispositivos para descubrir servicios en la red local. mDNS que significa Multicast DNS y permite la

resolución de nombres sin necesidad de un servidor DNS central. ARP que significa Address Resolution Protocol el cual es el encargado de traducir direcciones IP a direcciones MAC; puede ser explotado en ataques. MQTT que significa Message Queuing Telemetry Transport el cual es el protocolo ligero usado en IoT para enviar mensajes entre dispositivos y servidores.

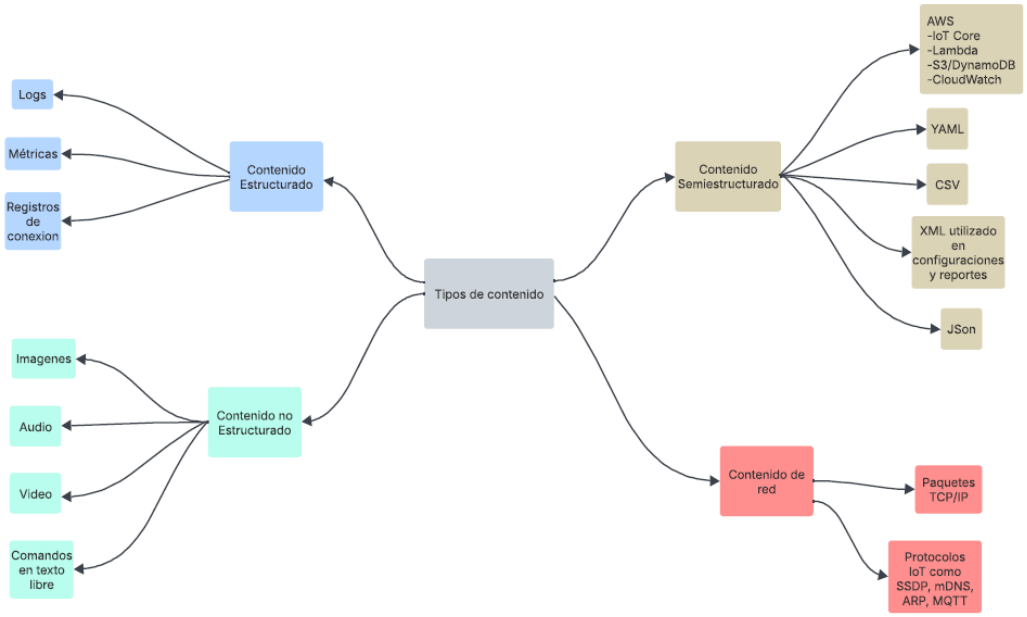

Figura 3. Marco conceptual según clasificación del contenido (LucidChart, s.f.)

2.1.1 Definición de eventos de seguridad

Se define como cualquier actividad o comportamiento detectado en la red que pueda comprometer la confidencialidad, integridad o disponibilidad de los datos, dispositivos o servicios conectados. Estos eventos pueden ser el resultado de acciones maliciosas, fallos técnicos, configuraciones inseguras o vulnerabilidades explotadas por atacantes.

Dentro de las vulnerabilidades más comunes a nivel domésticos están:

- Escaneos de red no autorizados o ARP masivo: actividad que busca identificar dispositivos conectados en la red, común en fases de reconocimiento de un ataque. Consiste en que los hackers envían mensajes ARP falsos a una red objetivo en un intento de vincular su propia dirección MAC con una dirección IP legítima de la red (Duò, 2025).

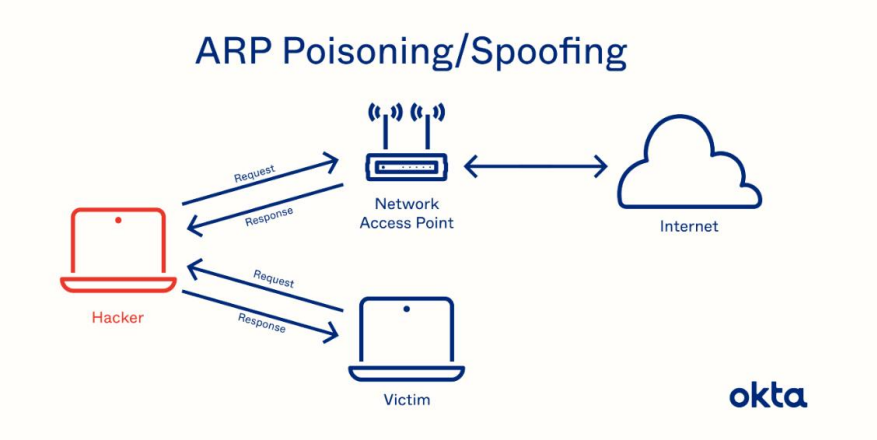

Figura 4. ARP Scanning (OKTA, 2024)

- Exfiltración de datos vía SSDP/mDNS: Uso de protocolos de descubrimiento para enviar información fuera de la red sin autorización. Este es un método que utilizan los ciberdelincuentes para robar datos de una red explotando el protocolo del sistema de nombres de dominio (DNS). En este tipo de ataques se eliminan los datos confidenciales del interior de una red doméstica protegida integrándose en paquetes DNS y dado que este tipo de tráfico se suele permitir a través de los firewall o sistemas de seguridad, no son fácilmente detectables y pueden salir de la red.

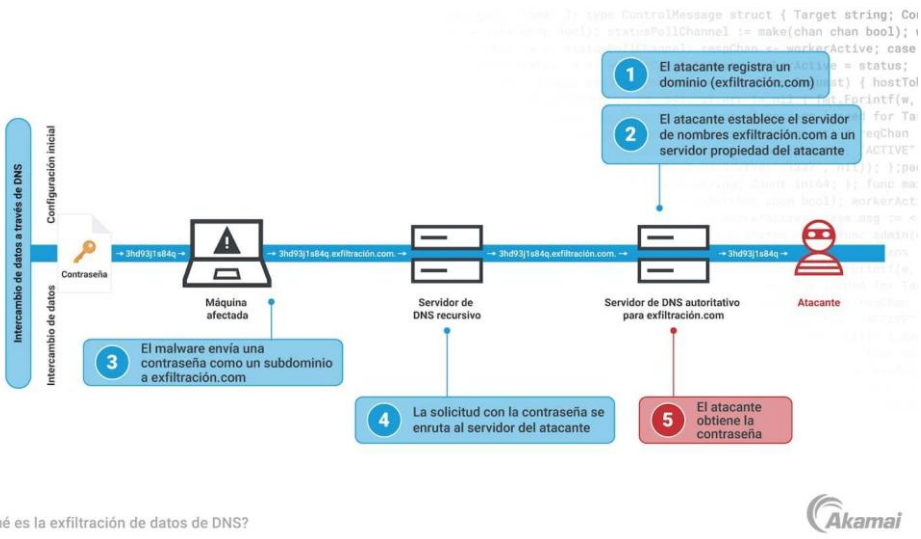

Figura 5. Ejemplo de exfiltración de datos DNS. (Akamai, s.f.)

- Servicios HTTP sin cifrado: esto abarca la comunicación entre dispositivos IoT y servidores que utilizan el protocolo HTTP en lugar de HTTPS, lo que significa que los datos transmitidos no están protegidos mediante TLS es decir Transport Layer Security. Los datos viajan en texto plano por lo que es relativamente fácil leer, modificar o interceptar la información. Esto representa una vulnerabilidad crítica en redes domésticas ya que los usuarios suelen confiar en que toda la red opera de forma segura. Un hacker podría obtener fácilmente credenciales de inicio de sesión y datos personales (Crudu, 2025).

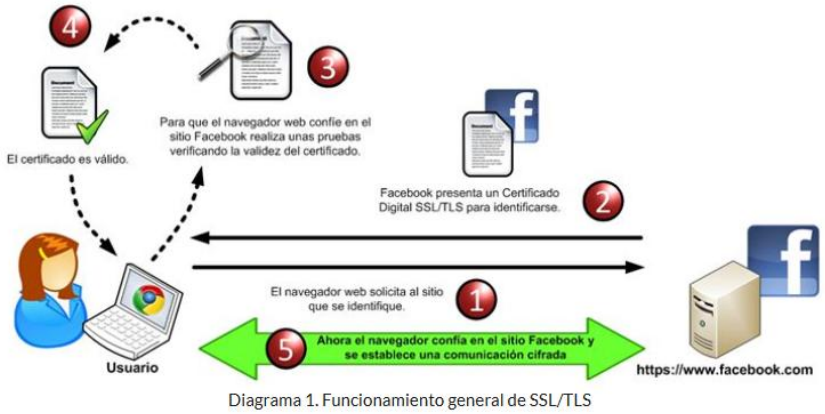

Figura 6. Funcionamiento general de SSL/TLS (Madrigal, 2025)

- Comportamiento anómalo asociado a SDKs de tercero: un SDK por sus siglas en inglés significa Software Development Kits y son bibliotecas o conjuntos de herramientas que los fabricantes de dispositivos IoT integran para facilitar funciones como conectividad, control remoto, integración con plataformas cloud, o interacción con aplicaciones móviles. Si bien estos SDK agilizan el desarrollo pueden representar una amenaza a la seguridad del hogar. Algunos ejemplos de un uso anómalo se pueden observar en la figura 7

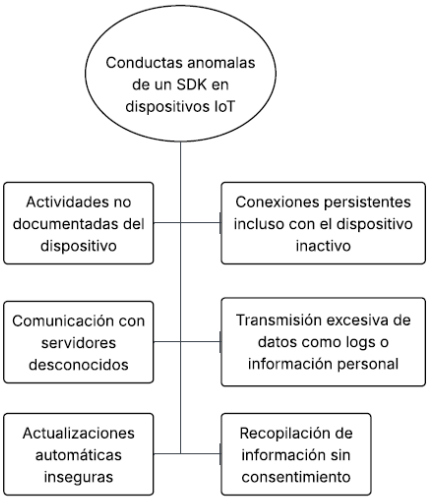

Figura 7. Conductas anómalas de un SDK (Elaboración propia).

2.1.2 Redes domésticas y sociales

Las redes domésticas son el núcleo de la conectividad en los hogares inteligentes. Permiten la interconexión de múltiples dispositivos como computadoras personales, teléfonos inteligentes, asistentes virtuales, cámaras de seguridad, electrodomésticos inteligentes y sistemas de entretenimiento. Esta conectividad facilita la automatización, el control remoto y la eficiencia energética, pero también introduce nuevas vulnerabilidades que pueden ser explotadas por actores maliciosos. Las redes sociales juegan un papel indirecto en la seguridad de los hogares inteligentes porque pueden difundir contenido malicioso o exponer información personal sensible. Además, muchos dispositivos IoT actualmente tienen una integración con las redes sociales lo que puede abrir accesos no autorizados (Vardakis, 2024).

2.1.3 Componentes de un evento en IoT

Un evento de seguridad en IoT es una manifestación observable en el tráfico de red o en el comportamiento de un dispositivo que puede indicar una amenaza, vulnerabilidad o actividad anómala. Para que un sistema de monitoreo pueda detectar, clasificar y responder a estos eventos, es necesario descomponerlos en sus componentes fundamentales que son:

- Origen del evento: el dispositivo genera el tráfico o puede ser una acción sospechosa.
- Destino del evento: se refiere al servidor o IP al que se dirige el tráfico.
- Tipo de tráfico: se refiere al protocolo utilizado
- Frecuencia y volumen: número de paquetes que fueron enviados, así como su tamaño y periodicidad.
- Contenido del tráfico: son los datos transmitidos: encabezados, payload, comandos, configuraciones, etc.
- Contexto temporal: hora y duración del evento.

2.1.4 Captura y transformación de tráfico IoT mediante gateway local

En el contexto de redes domésticas inteligentes, una estrategia efectiva para el monitoreo de eventos de seguridad consiste en implementar un gateway local, encargado de capturar el tráfico de red generado por dispositivos IoT. Esta captura puede realizarse mediante herramientas como Wireshark o TCPdump, que permiten registrar paquetes en tiempo real y almacenarlos en formato .pcap. Ambos programas son ampliamente utilizados para la captura de paquetes en redes IP capturando protocolos como TCP, UDP, HTTP, RTSP y MQTT, etc.

En la figura 8 se observa un ejemplo de traza típica de Wireshark donde se tienen detalles como:

- Paquetes numerados en orden de captura.
- Tiempos relativos desde el inicio de la captura.
- IP de origen y destino.
- Protocolos como TCP y UDP.
- Información adicional como flags TCP (ej. RST para reset de conexión).

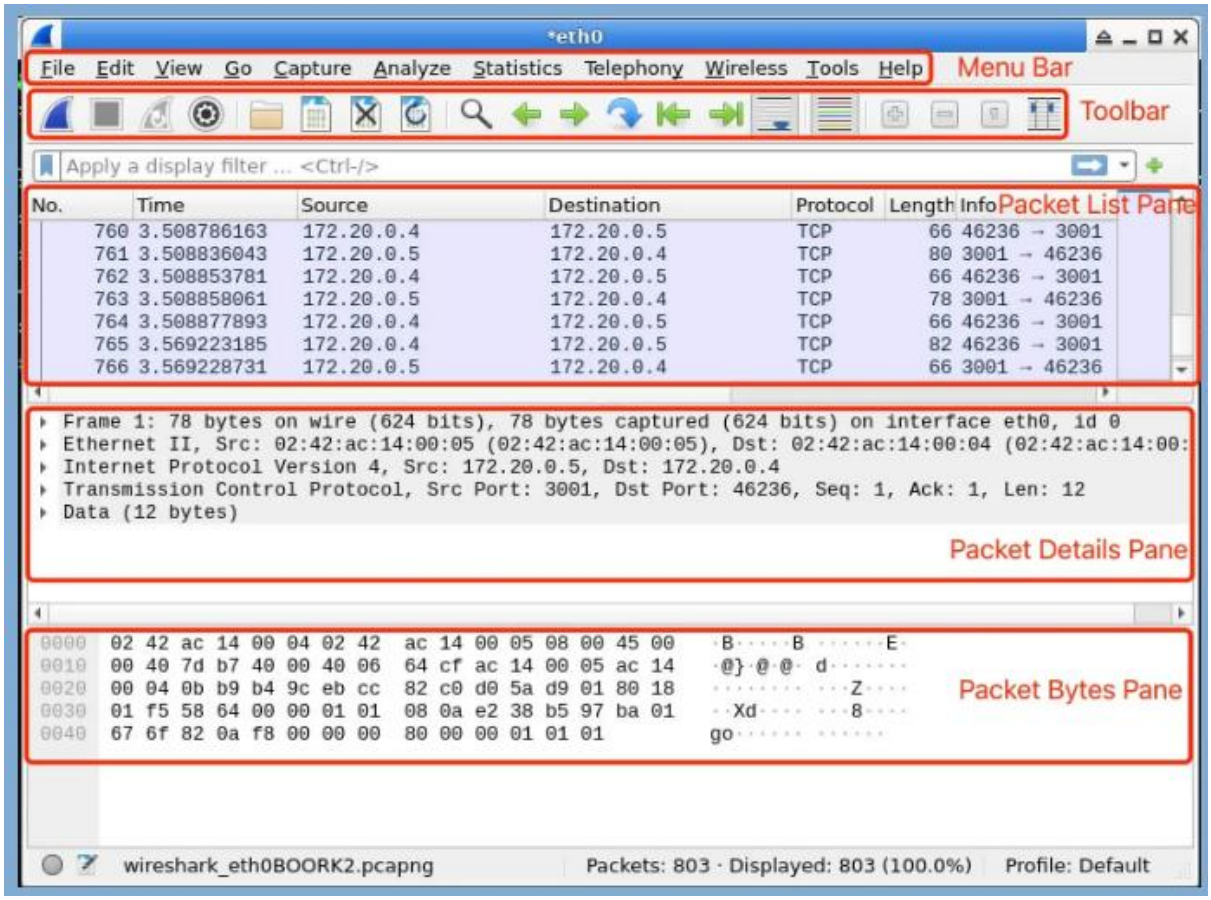

Figura 8. Ejemplo de traza Wireshark (LabEx, n.d.)

En la figura 9 se puede observar un ejemplo de traza TCPdump donde se tienen detalles como:

- Timestamp: Hora exacta en que se capturó el paquete.
- IP origen y destino: Incluye puertos (ej. 10.0.0.50.80 > 10.0.0.1.53181).
- Flags TCP: [F.] indica que es un paquete de finalización de conexión (FIN + ACK).
- seq / ack: Números de secuencia y confirmación.
- win: Tamaño de la ventana TCP.
- options: Opciones TCP como timestamps.
- length: Tamaño del payload (en este caso, 0).

```
20:58:26.765637 IP 10.0.0.50.80 > 10.0.0.1.53181: Flags [F.], seq 1, ack 2, win 453, options
[nop,nop,TS val 3822939 ecr 249100129], length 0
```

Figura 9. Ejemplo de traza TCPdump (Khurana, 2024)

2.2 Conceptos sobre minería de datos

En el contexto de ciberseguridad residencial, la minería de datos es una herramienta clave para analizar el tráfico de red generado por dispositivos IoT. Este proceso permite extraer información útil a partir de grandes volúmenes de datos, identificar patrones anómalos, y correlacionar eventos que podrían representar amenazas. Al aplicar técnicas estadísticas y analíticas sobre los datos recolectados por el sistema propuesto en AWS, se mejora la capacidad de detección temprana de intrusiones, se reduce la tasa de falsos positivos y se fortalece la toma de decisiones automatizada ante incidentes de seguridad en redes domésticas inteligentes (ENISA, 2023).

2.2.1 Minería de datos en ciberseguridad

Analizando el contexto de esta investigación, la minería de datos nos va a permitir transformar grandes volúmenes de información generada por dispositivos IoT en datos útiles para la detección temprana de amenazas y también permite clasificar el tipo de comportamiento de los dispositivos y así poder tomar las mejores decisiones y a la vez automatizarlas.

Las principales aplicaciones de la minería para este proyecto son:

- Clasificación de tráfico legítimo vs sospechoso

Existen algoritmos de clasificación que analizan el comportamiento del tráfico generado por los dispositivos inteligentes de la casa. Con esto se busca distinguir entre patrones normales de operación y actividades que podrían considerarse como una amenaza. Para esto se pueden determinar características como frecuencias de conexión, protocolos utilizados, volumen de datos, destino de tráfico IP, etc.

El árbol de decisión que se va a utilizar para este trabajo se puede observar en la figura 10.

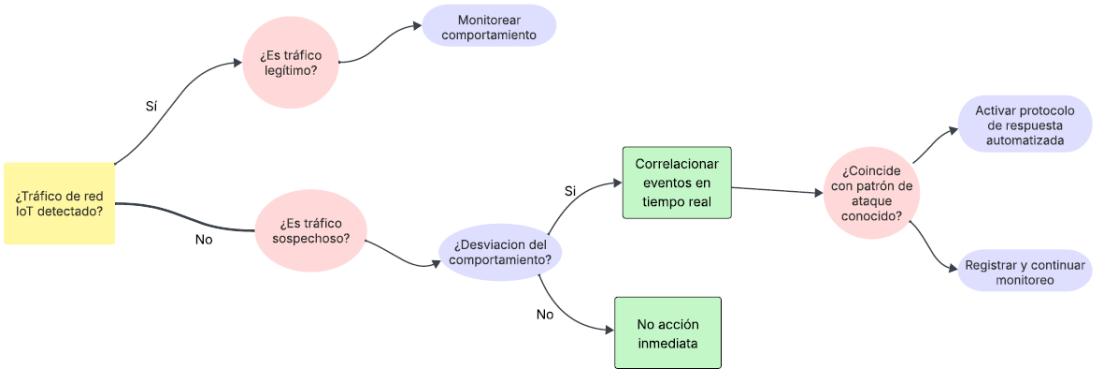

Figura 10. Árbol de decisión (Elaboración propia).

- Detectar desviaciones en el comportamiento de dispositivos

Una vez teniendo el tráfico clasificado se pueden establecer perfiles de comportamiento para cada dispositivo IoT con el que se vaya a realizar el estudio. Estos perfiles se van a realizar teniendo en cuenta varios rubros establecidos previamente y luego de tenerlos determinados se les aplicarán las técnicas de detección de anomalías para identificar desviaciones significativas como uso indebido de SDK a terceros o actividades no autorizadas entre otras.

- Correlación de eventos

Como último paso se incorpora la funcionalidad de AWS IoT Core, que permite correlacionar eventos de seguridad mediante el análisis continuo de métricas de comportamiento de los dispositivos IoT. Esta herramienta facilita la identificación de patrones de ataque al detectar desviaciones respecto al comportamiento esperado, incluso cuando los eventos individuales podrían parecer aislados. Para lograrlo, el sistema propuesto utiliza un gateway local que captura el tráfico de red de los dispositivos IoT, lo transforma en métricas estructuradas (como conexiones activas, puertos utilizados, volumen de datos transmitidos) y las publica en los topics definidos. Este servicio en la nube aplica reglas de auditoría y detección para correlacionar eventos, generar alertas ante anomalías y fortalecer la postura de seguridad del entorno residencial. Se puede observar el diagrama general en la figura 11

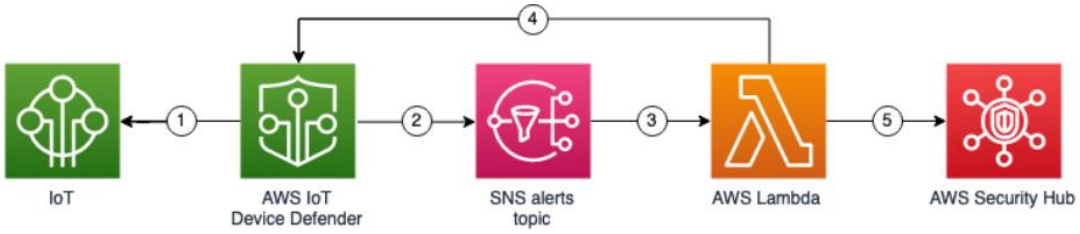

Figura 11. Sistema AWS Device Defender (Amazon Web Services, 2021).

2.3 Procesamiento de Trazas con Python

Una vez capturado el tráfico de red generado por los diferentes dispositivos de IoT utilizando las herramientas mencionadas anteriormente, se requiere un proceso de transformación de datos que permita convertir las trazas en información útil para la detección de eventos de seguridad. Esto se va a realizar por medio de un script en Python; este actuará como puente entre la captura de la traza recolectada en el Gateway local y la plataforma de monitoreo en la nube.

Este script permitirá transformar los archivos en formato JSON para que puedan ser interpretados por las plataformas de AWS. Una vez transformados los archivos se publicarán utilizando el protocolo MQTT.

AWS Lambda analizará las métricas recibidas y hará una comparación de comportamientos normales los cuales deben ser previamente definidos; todo lo que detecten como anomalías serán notificadas al usuario permitiendo una integración eficiente sin necesidad de desarrollo de software y facilitando la generación de alarmas.

Capítulo 3. Marco metodológico

3.1 Tipo de Investigación

La presente investigación busca resolver un problema concreto de ciberseguridad en entornos residenciales mediante la integración de los dispositivos IoT con tecnologías basadas en la nube, para así realizar captura y análisis del tráfico de red. Se orienta a generar una solución práctica basada en tecnologías existentes como AWS IoT Core.

Como parte del enfoque aplicado, se desarrolla un script en Python que cumple la función de leer trazas de tráfico de red capturadas mediante herramientas como Wireshark o TCPdump, para homologar los datos obtenidos. Una vez los datos estén en el formato requerido por AWS se podrán publicar dichos eventos a través del protocolo MQTT en AWS IoT Core, que se encarga de analizar el comportamiento del dispositivo y detectar posibles anomalías.

3.2 Alcance Investigativo

El alcance de esta investigación es exploratorio y descriptivo dado que aborda un problema emergente en el ámbito de la ciberseguridad residencial, donde aún existe una limitada documentación académica y técnica, especialmente en el contexto costarricense.

- **Exploratorio**: esto porque se busca identificar y comprender las características del tráfico de red generado por dispositivos IoT en entornos domésticos, así como las vulnerabilidades asociadas a su comportamiento. Se examinan protocolos de comunicación, patrones de conexión, y eventos de seguridad que podrían pasar desapercibidos en redes residenciales. Este nivel permite descubrir elementos clave que no han sido suficientemente estudiados, como el uso de SDKs de terceros o la exfiltración de datos vía protocolos de descubrimiento.
- **Descriptivo**: por el detalle de la documentación sobre el funcionamiento del sistema propuesto, incluyendo la arquitectura cloud basada en AWS, el uso de una PC como gateway local, la captura de tráfico mediante Wireshark y TCPdump, y el procesamiento de trazas con Python para generar métricas compatibles con AWS IoT Core. Se describen los componentes del sistema,

los flujos de datos, los tipos de eventos detectados y las respuestas automatizadas ante incidentes.

3.3 Enfoque

Esta investigación plantea un enfoque mixto porque se utilizan métodos cuantitativos y cualitativos para abordar de forma integral el problema de monitoreo y detección de eventos de seguridad en redes domésticas con dispositivos IoT. Este enfoque permite validar la efectividad del sistema propuesto mediante pruebas controladas, simulaciones de ataques y análisis de métricas de rendimiento.

- Enfoque Cuantitativo

Se centra en la medición, análisis estadístico y evaluación de datos generados por dispositivos IoT en redes residenciales y permite reducir el tiempo de detección de amenazas y disminuye la tasa de falsos positivos en entornos domésticos.

Entre los elementos clave están:

  - Tasa de detección verdadera
  - Tasa de falsos positivos
  - Volumen de tráfico procesado
  - Frecuencia de los eventos
  - Número de dispositivos
  - Comparación de escenarios normales versus ataques

- Enfoque cualitativo

Se centra en la interpretación, comprensión y análisis contextual de los eventos de seguridad en redes domésticas con dispositivos IoT, así como en la evaluación de las prácticas actuales de ciberseguridad en entornos residenciales.

Entre los elementos clave están:

  - Revisión sistemática de literatura
  - Interpretación de patrones de tráfico
  - Análisis de eventos de seguridad

3.4 Diseño de la investigación

El diseño de esta investigación es de tipo no experimental, transversal y aplicado, orientado al desarrollo y validación de una solución tecnológica para la detección de eventos de seguridad en redes domésticas con dispositivos IoT, utilizando una arquitectura cloud en AWS.

- No experimental: La investigación no manipula deliberadamente las variables independientes, sino que observa y analiza el comportamiento del tráfico de red generado por dispositivos IoT en entornos residenciales reales. Se capturan datos en condiciones naturales, sin alterar el funcionamiento normal de los dispositivos.
- Transversal: El estudio se realiza en un periodo determinado, recolectando datos en momentos específicos para evaluar el desempeño del sistema propuesto.
- Aplicado: El diseño busca resolver un problema concreto mediante la implementación de una solución práctica. Se desarrolla un sistema funcional que puede ser replicado y adaptado a diferentes hogares, contribuyendo a mejorar la seguridad digital en el entorno residencial.

3.5 Población y Muestreo

La población objetivo de esta investigación está conformada por usuarios residenciales en Costa Rica que poseen dispositivos IoT conectados a redes WiFi domésticas. Esta población representa un segmento creciente de hogares que han adoptado tecnologías inteligentes para automatización, vigilancia, eficiencia energética y conectividad, pero que enfrentan desafíos importantes en materia de ciberseguridad.

La población incluye:

- Hogares con dispositivos como cámaras IP, enchufes inteligentes, asistentes virtuales, termostatos, sensores, etc.
- Ambientes residenciales donde se puede simular tráfico de red y escenarios de ataque para validar el sistema propuesto.

El muestreo utilizado es no probabilístico por conveniencia, ya que se seleccionan hogares accesibles para el investigador, incluyendo su propia residencia y la de familiares que cuentan con dispositivos IoT. Esta estrategia permite realizar pruebas

controladas, implementar el sistema propuesto y recolectar datos de forma segura y eficiente.

Dado que el objetivo es validar el funcionamiento técnico del sistema en condiciones reales, se considera que este tipo de muestreo es adecuado para la etapa experimental del proyecto.

3.6 Instrumentos de Recolección de Datos

Para la presente investigación se emplean instrumentos tecnológicos y documentales que permiten recolectar datos de tráfico de red generado por dispositivos IoT en entornos residenciales, así como información contextual relevante para el análisis de eventos de seguridad.

- Wireshark y TCPdump.
- Gateway local (PC dedicada).
- Script en Python.
- AWS IoT Core.
- Revisión sistemática de literatura.
- Bitácora de pruebas.

3.7 Técnicas de Análisis de Información

La presente investigación emplea técnicas de análisis de información tanto cuantitativas como cualitativas, en concordancia con el enfoque mixto adoptado como se puede observar en la figura 12. Estas técnicas permiten evaluar el comportamiento del tráfico de red generado por dispositivos IoT residenciales, identificar eventos de seguridad, y validar la efectividad del sistema propuesto.

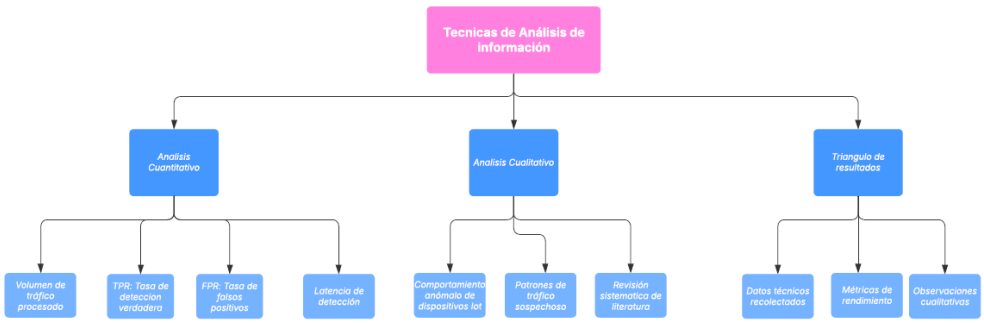

Figura 12. Técnicas de análisis de información (Elaboración propia).

3.8 Estrategia de Desarrollo de la Propuesta

La estrategia de desarrollo de la propuesta se estructura en fases progresivas que permiten proponer un sistema de monitoreo y detección de eventos de seguridad en redes domésticas con dispositivos IoT, utilizando una arquitectura cloud en AWS. Esta estrategia combina elementos técnicos, metodológicos y operativos, garantizando la coherencia entre los objetivos planteados y los resultados esperados.

Fase 1. Diagnóstico y revisión documental: se realiza una revisión sistemática de literatura para identificar vulnerabilidades comunes en dispositivos IoT residenciales. Se analizan enfoques existentes de monitoreo y detección de eventos de seguridad en redes domésticas. Se definen las categorías de análisis y los criterios técnicos que guiarán el diseño del sistema.

Fase 2. Diseño de la arquitectura: Se seleccionan los servicios de AWS que conformarán la solución. Se define el modelo de datos, los flujos de comunicación y los componentes del sistema (gateway local, script de transformación, canal MQTT). Se establecen las reglas de detección y correlación de eventos en AWS.

Fase 3. Implementación técnica: Se configura un gateway local para la captura de tráfico de red mediante Wireshark. Se desarrolla un script en Python para transformar las trazas en eventos estructurados (JSON). Se publica el tráfico procesado en AWS IoT Core y se activa la detección de anomalías en AWS Lambda.

Fase 4. Simulación de escenarios de ataque: Se ejecutan pruebas controladas en redes residenciales reales, simulando eventos. Se recolectan métricas de rendimiento del sistema.

Fase 5. Evaluación y validación: se analizan los resultados obtenidos mediante técnicas cuantitativas y cualitativas. Se realiza una triangulación entre los datos técnicos, las observaciones contextuales y la revisión documental. Se valida la efectividad del sistema y se proponen recomendaciones para su mejora y escalabilidad.

Capítulo 4. Análisis del diagnóstico

4.1 Implementación de la infraestructura

En este capítulo, se desarrollan los componentes utilizados para la elaboración de la infraestructura, sus componentes, funcionalidad y se analiza los datos obtenidos de manera profunda, con el fin de realizar una propuesta de solución acorde a las necesidades del proyecto.

4.1.1 Componentes de una infraestructura IoT residencial

Basándonos en una infraestructura típica residencial, se procede a identificar los elementos principales que se utilizan en los hogares inteligentes.

- **Punto de Acceso**: crea la red inalámbrica local que permite que los dispositivos IoT (sensores, cámaras, enchufes inteligentes, asistentes) se conecten a la LAN y, si procede, a Internet; actúa como “puerta” entre los dispositivos WiFi y la red cableada o el router principal.
- **Enrutador principal**: Conecta a los dispositivos enlazados en el AP con el internet, define las redes a ser utilizadas por los dispositivos y permite delimitar el acceso no autorizado de terceros.
- **Dispositivo IoT**: objeto físico equipado con sensores, actuadores, software y conectividad de red que le permite recopilar, enviar y recibir datos a través de Internet o redes locales, automatizar acciones y comunicarse con otros sistemas.
- **Plataforma de acceso de fabricante**: se refiere al conjunto integrado de servicios, interfaces y software provistos por un fabricante para permitir el acceso, gestión y operación de sus dispositivos IoT; incluye tanto las aplicaciones cliente como los servicios cloud, APIs y mecanismos de

aprovisionamiento que permiten controlar y obtener telemetría de los dispositivos.

4.1.2 Diagrama de una infraestructura IoT residencial

Las capturas de tráfico son tomadas de una red de tipo residencial, donde se cuenta con un enrutador principal y un AP, se utilizará una cámara web, un bombillo inteligente y un tomacorriente inteligente. Como se puede observar en la figura 13, estos elementos se conectan por medio del AP a la red y este permite acceso al internet pasando por el enrutador principal, lo cual permite la gestión remota de estos dispositivos utilizando aplicaciones del fabricante.

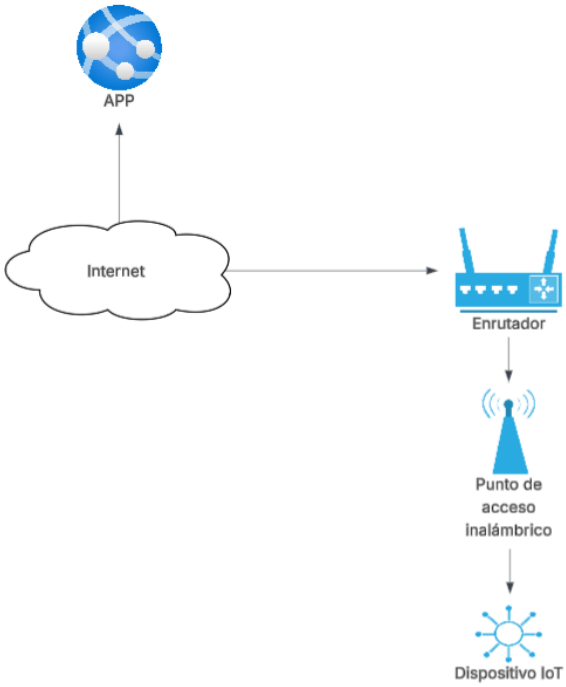

Figura 13. Diagrama de estructura IoT residencial (Elaboración propia).

4.2 Selección de equipos de prueba

Se estará identificando cada uno de los elementos que se utilizará para definir sus alcances tanto de software como de hardware.

4.2.1 Equipo de red

- **Enrutador principal**: se utiliza un módem Huawei OptiXstar EG8147X6, el cual cumple la función de enrutador en la red interna del hogar.
- **Punto de Acceso**: para lograr capturar el tráfico hacia los dispositivos IoT, se utiliza una computadora Dell Latitude 7410 con Windows 10, funcionando como

punto de acceso inalámbrico mediante la función “Zona de cobertura inalámbrica móvil”. De esta manera, el tráfico que vaya dirigido hacia los dispositivos IoT será capturado con Wireshark instalado en esta computadora.

4.2.2 Dispositivos IoT

- **Cámara IP**: se utiliza una cámara HSKAH modelo WH33, la cual es una cámara oculta en un reloj de escritorio, muy utilizada para el cuidado de niños.
- **Bombillo inteligente**: el bombillo utilizado en esta infraestructura es marca Tuya Smart, modelo E27 9-15W RGBW.
- **Tomacorriente inteligente**: para este escenario, el tomacorriente usado es el enchufe inteligente WiFi Nexxt modelo AHIWPSO4U1.

4.3 Captura de tráfico IoT

En esta fase, procedemos a analizar los datos obtenidos por medio de Wireshark, corriendo en una PC con sistema operativo Windows 11. La data obtenida se refiere a una cámara IP, así como un bombillo inteligente y un tomacorriente inteligente.

4.3.1 Captura de tráfico de bombillo inteligente

La figura 14 nos demuestra el comportamiento que tiene el tráfico del bombillo inteligente en reposo y la figura 15 nos demuestra un comportamiento cuando se envía una acción al bombillo.

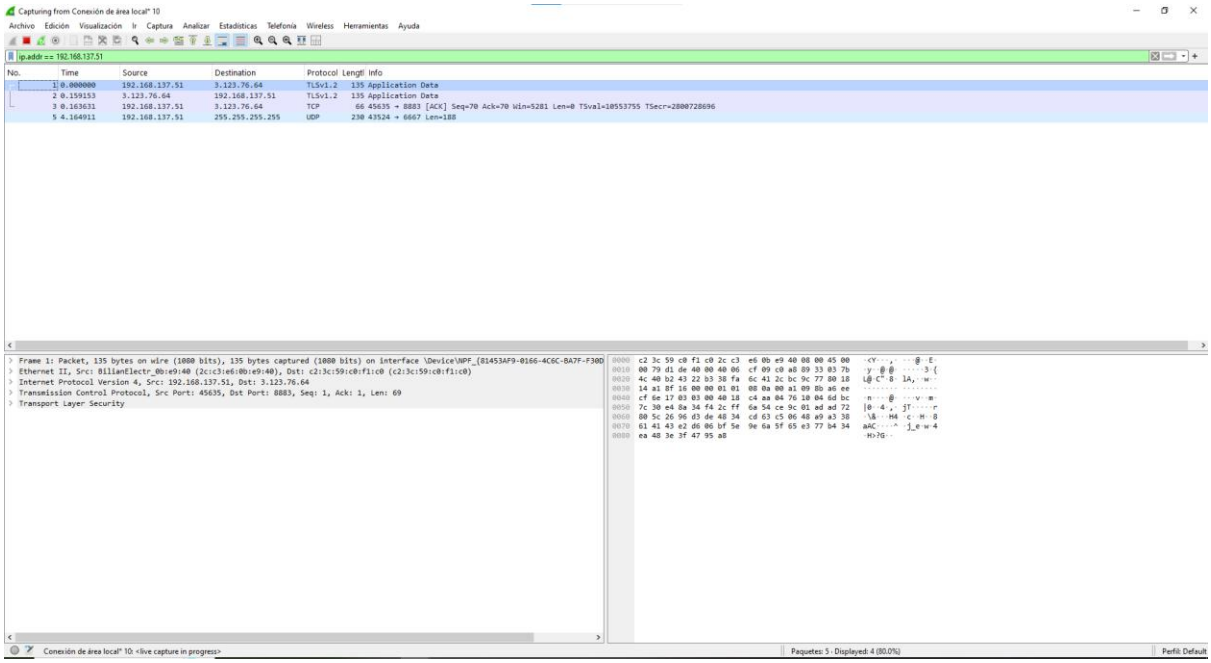

Figura 14. Captura de tráfico de bombillo inteligente en reposo (Elaboración propia)

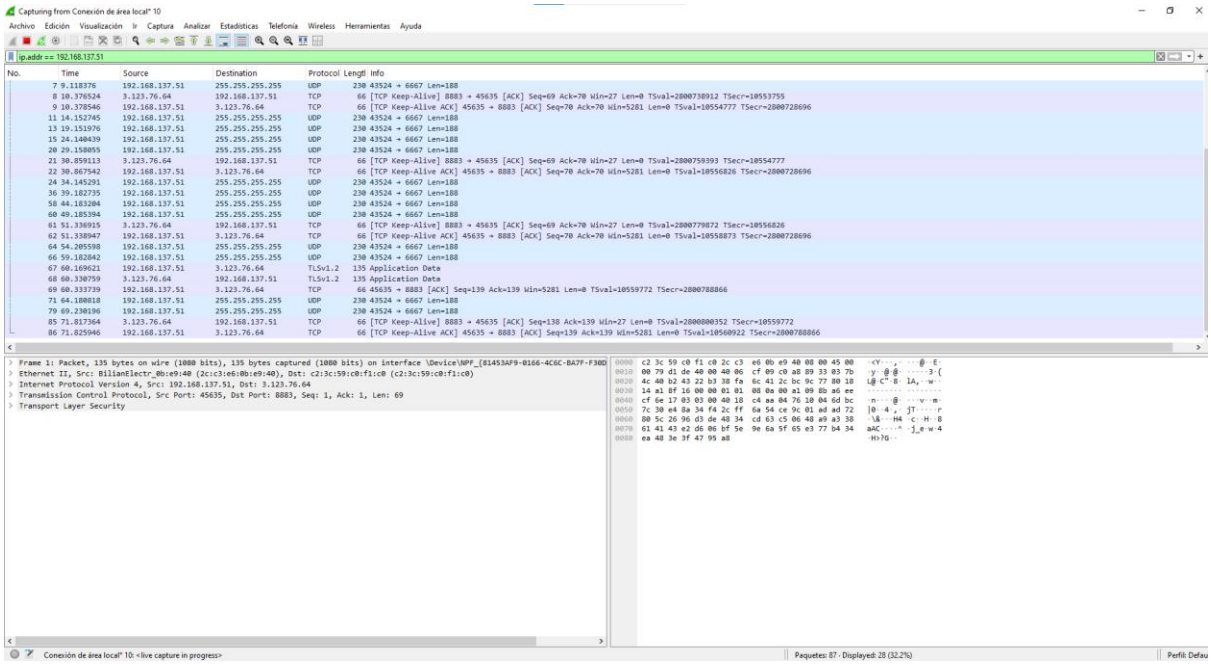

Figura 15. Captura de tráfico de bombillo inteligente recibiendo comandos (Elaboración propia)

4.3.2 Captura de tráfico en tomacorriente inteligente

La figura 16 nos demuestra el comportamiento que tiene el tráfico del tomacorriente inteligente en reposo y la figura 17 nos demuestra un comportamiento cuando se envía una acción al tomacorriente.

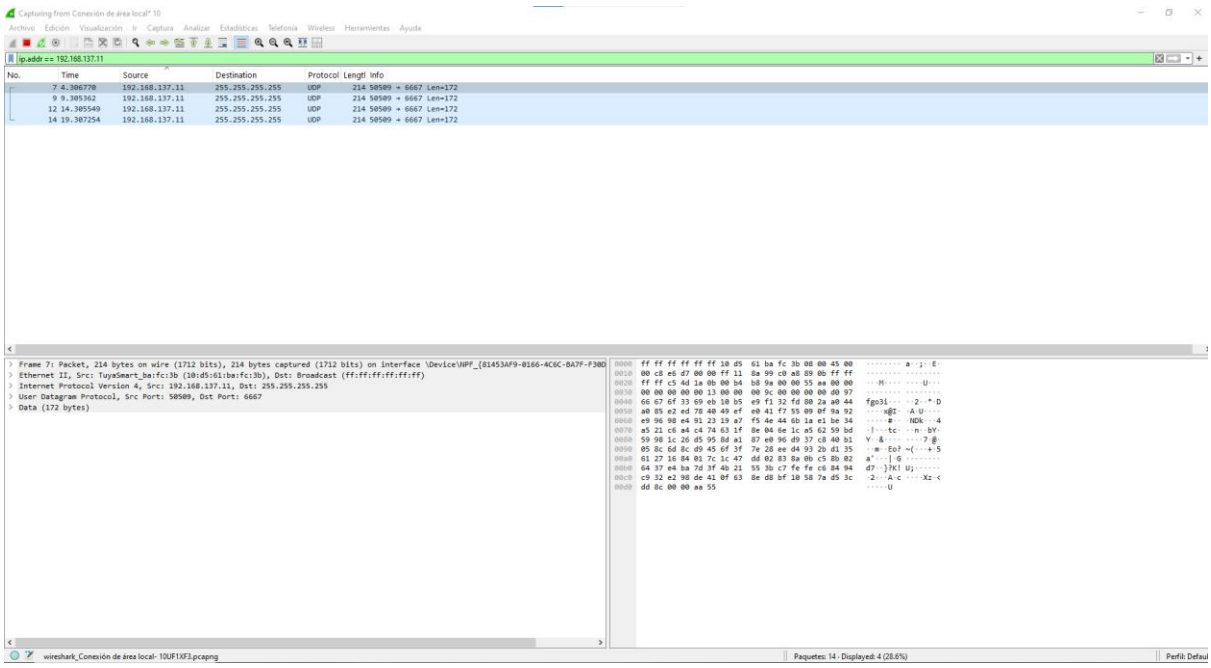

Figura 16. Captura de tráfico de tomacorriente en reposo (Elaboración propia)

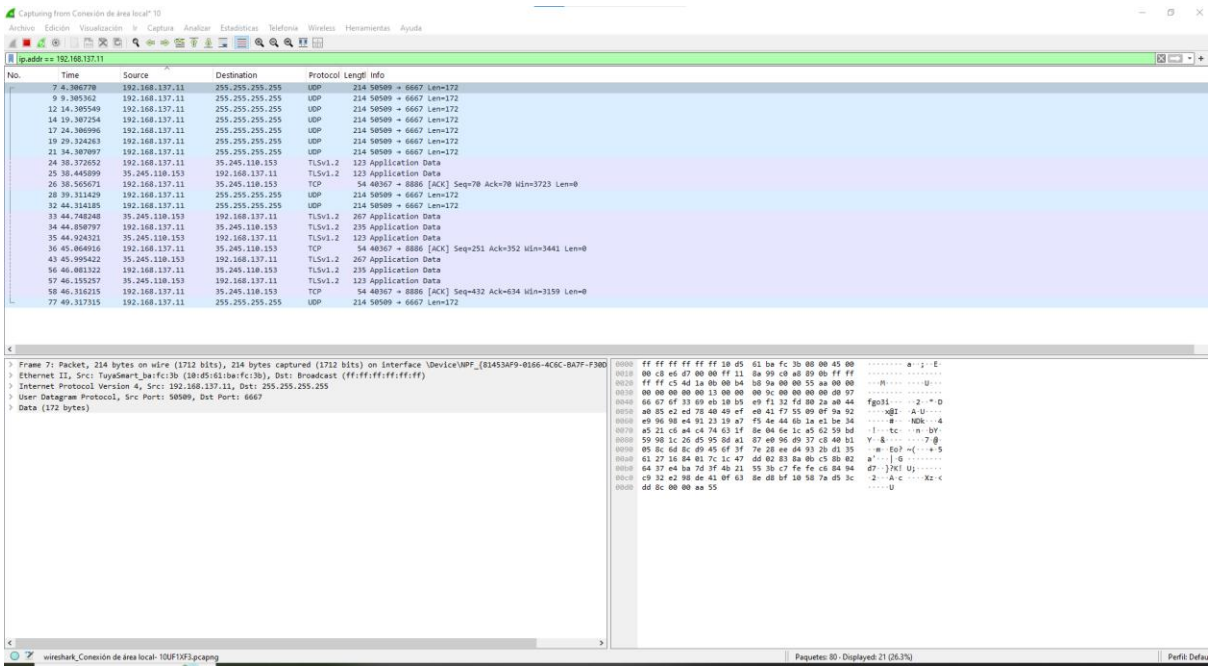

Figura 17. Captura de tráfico de tomacorriente recibiendo comandos (Elaboración propia)

4.3.3 Captura de tráfico de cámara IP

Para el caso de la cámara IP, al ser video lo que se transmite, se observa una gran cantidad de tráfico UDP , podemos ver la diferencia en el flujo de datos en las figuras 18 y 19.

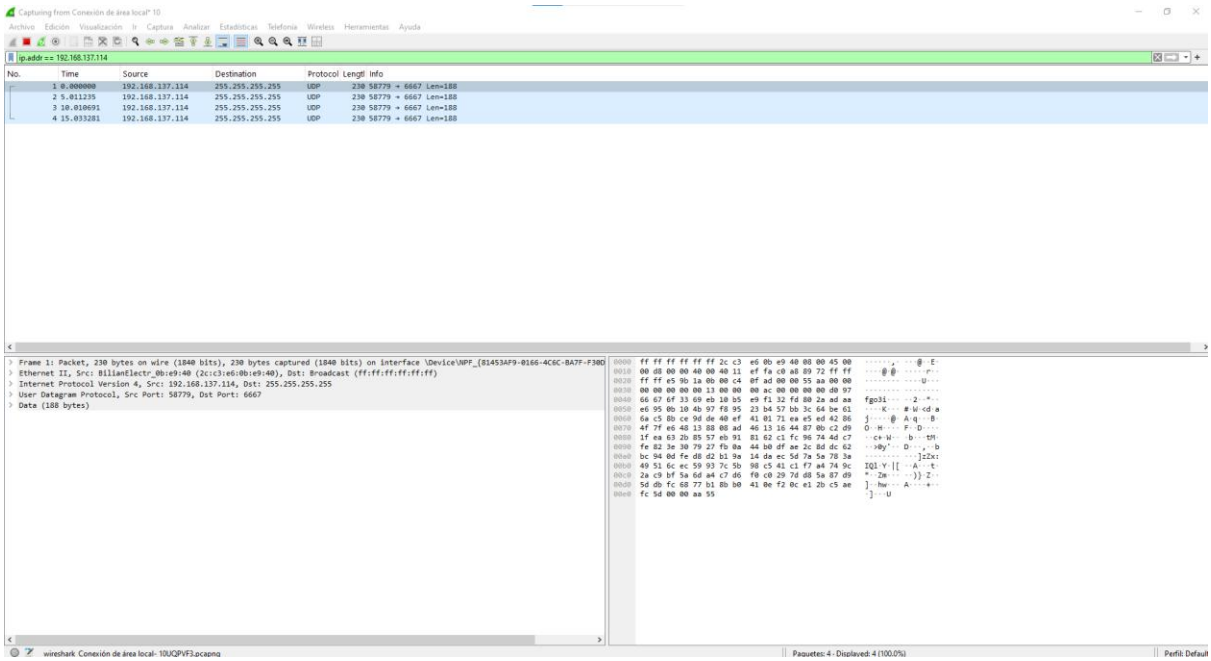

Figura 18. Captura de tráfico de cámara IP en reposo (Elaboración propia)

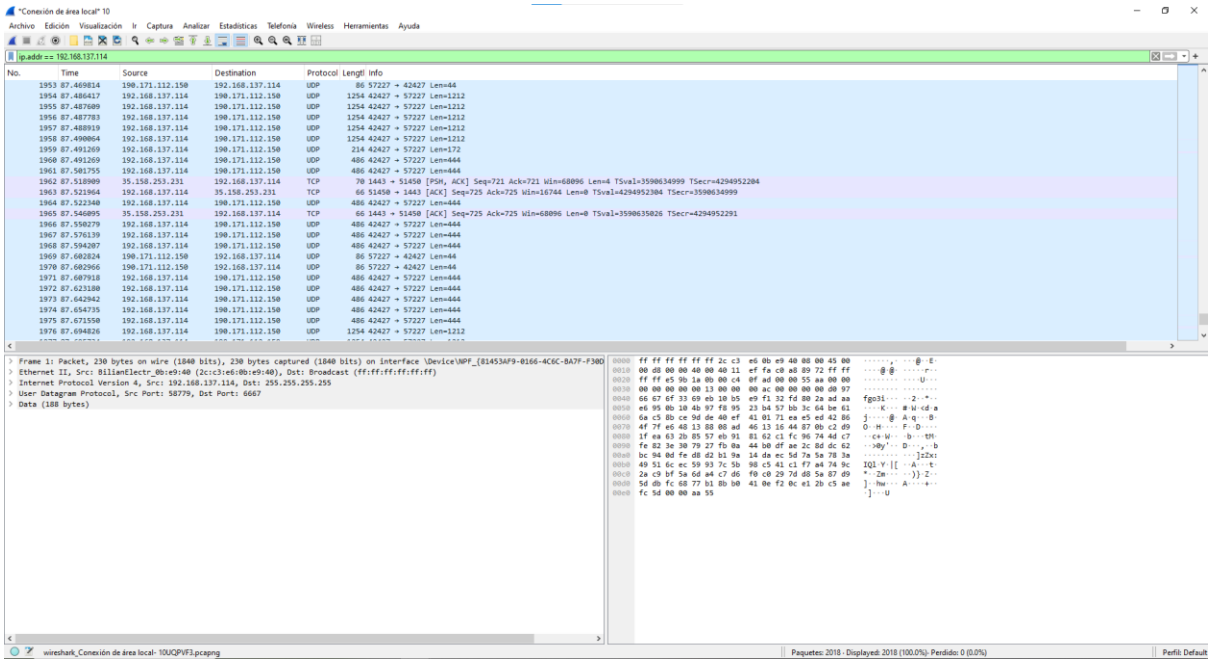

Figura 19. Captura de tráfico de cámara IP transmitiendo video (Elaboración propia)

4.4 Interpretación de datos obtenidos con las capturas de tráfico

4.4.1 Interpretación de los protocolos obtenidos en las capturas de tráfico

Como se puede ver en las diferentes capturas del tráfico que se realizaron a los 3 dispositivos, incluso siendo funciones diferentes, hay protocolos que se utilizan para todos estos y que representan la mayoría del tráfico que pasa por la red hacia los equipos IoT.

- **UDP**: El Protocolo de Datagrama de Usuario (UDP) es un protocolo de transporte sin conexión que envía datagramas sin garantizar entrega, orden o corrección de errores; es ligero y de baja latencia, usado en streaming, voz en tiempo real, juegos y protocolos que gestionan la fiabilidad a nivel de aplicación (Cloudflare, 2024).
- **TCP**: se refiere al conjunto integrado de servicios, interfaces y software provistos por un fabricante para permitir el acceso, gestión y operación de sus dispositivos IoT; incluye tanto las aplicaciones cliente como los servicios cloud, APIs y mecanismos de aprovisionamiento que permiten controlar y obtener telemetría de los dispositivos (Cloudflare, 2024).
- **TLS**: Transport Layer Security (TLS) es un protocolo criptográfico que proporciona confidencialidad, integridad y autenticación en comunicaciones de red mediante cifrado y firmas; se emplea para asegurar HTTPS, correo, VPNs y otras aplicaciones que requieren privacidad y protección frente a manipulación (Cloudflare, 2024).

Analizando a profundidad estos protocolos y su frecuencia, vemos que la solución a plantear, debe tomar en cuenta que los paquetes se encuentran cifrados de extremo a extremo, dando un punto de seguridad extra a los dispositivos IoT administrados de manera remota.

4.4.2 Interpretación de las direcciones IP obtenidas en las capturas

A manera de análisis de tráfico, se puede observar que las direcciones IP que se encuentran registradas en las trazas varían dependiendo del servicio, la aplicación y el lugar geográfico de origen de los datos. Dentro de las recomendaciones se debe de tomar en cuenta estos datos, ya que, si se llegara a registrar un flujo de datos de un origen diferente al normal, debe ser visto como tráfico anómalo por parte de la

herramienta de monitoreo. La herramienta Wireshark nos permite realizar un gráfico con las zonas geográficas de donde se origina el tráfico hacia nuestros dispositivos IoT, tal como se muestra en la figura 20.

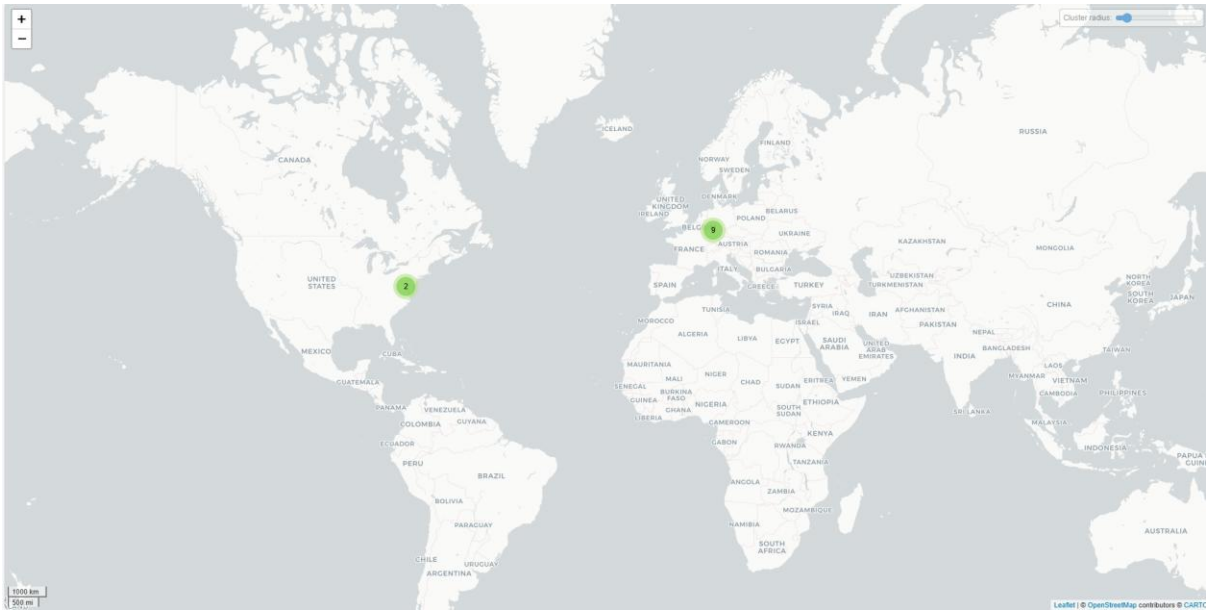

Figura 20. Zonas geográficas de las direcciones IP capturadas en Wireshark (Elaboración propia)

Dentro de la herramienta se tiene la opción de realizar un análisis profundo en el porcentaje del tráfico que se origina desde cada dirección IP, con lo que se pueden tomar referencias de un flujo de tráfico normal para estos dispositivos IoT. La figura 21 muestra que, a pesar de tener muchas direcciones de origen capturadas, el tráfico usualmente se genera desde 3 posibles orígenes distintos.

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
IPv4 Statistics/Destinations and Ports	28631				0,0007	100%	0,1400	42320,646
255.255.255.255	17225				0,0004	60,16%	0,0200	840,764
3.69.125.150	3530				0,0001	12,33%	0,0200	1702,775
192.168.137.36	3405				0,0001	11,89%	0,0400	275,674
35.245.110.153	1767				0,0000	6,17%	0,0300	42590,979
192.168.137.104	1160				0,0000	4,05%	0,0300	1033,296
35.212.87.166	305				0,0000	1,07%	0,0300	1033,630
18.199.123.50	221				0,0000	0,77%	0,0200	275,512
18.192.81.114	195				0,0000	0,68%	0,0200	1422,462
239.255.255.250	172				0,0000	0,60%	0,0300	42304,962
224.0.0.22	141				0,0000	0,49%	0,0600	42320,646
224.0.0.251	135				0,0000	0,47%	0,0700	42320,698
18.153.224.52	135				0,0000	0,47%	0,0200	5025,003
192.168.137.1	117				0,0000	0,41%	0,0500	2,446
18.156.58.194	54				0,0000	0,19%	0,0100	42482,094
3.120.166.59	20				0,0000	0,07%	0,0100	42468,643
3.121.244.214	16				0,0000	0,06%	0,0100	43196,339
18.195.44.247	16				0,0000	0,06%	0,0100	42382,404
3.123.76.64	12				0,0000	0,04%	0,0100	42396,558
224.0.0.252	5				0,0000	0,02%	0,0200	42304,731

Figura 21. Zonas geográficas de las direcciones IP capturadas en Wireshark (Elaboración propia)

Analizando la utilización y solicitudes capturadas con Wireshark podemos encontrar patrones de comportamiento típicos de estos dispositivos IoT. En la figura 22 podemos ver más a detalle las IP que se relacionan al tráfico de la red del hotspot, tanto direcciones privadas como públicas.

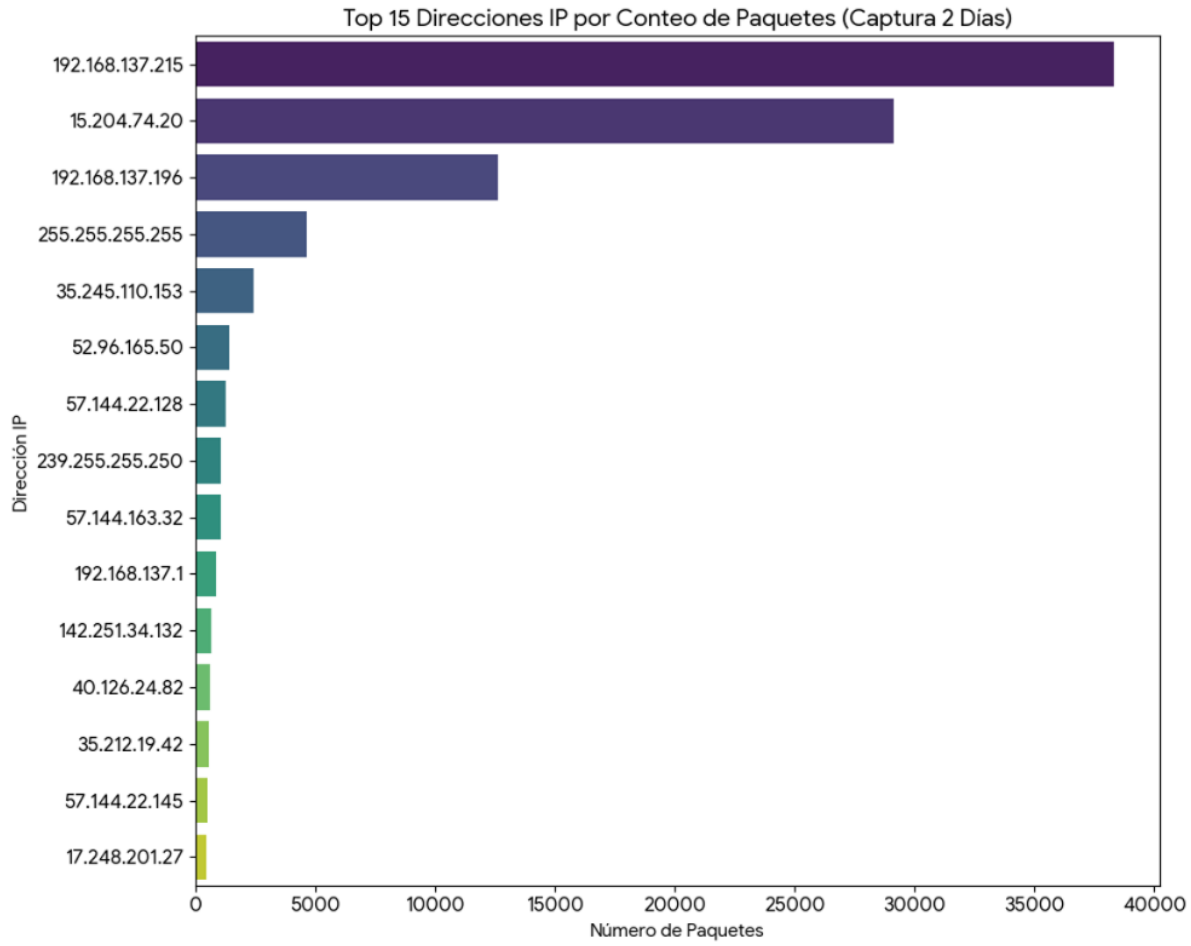

Figura 22. Gráfica de las IP por conteo de paquetes (Elaboración propia)

Según la información recolectada, podemos identificar los bloques de direcciones IP que utilizan los diferentes servicios que se comunican con nuestra infraestructura IoT, permitiéndonos hacer un filtro basado en direccionamiento IP.

4.5 Análisis comparativo de proveedores de servicios en la nube

De acuerdo con las secciones previas, este proyecto se desarrolla sobre una infraestructura en la nube para aprovechar su flexibilidad y el catálogo de servicios

disponibles. A continuación, se presenta un cuadro comparativo con las ventajas y desventajas de los tres proveedores líderes en la actualidad:

Tabla 9. Comparación de proveedores de servicios en la nube

Proveedor	**Características**
Amazon Web Services (AWS)	-Cuenta con servicios como IoT Core, Lamdba, SNS, etc. -Soporta mTLS con certificados X.509 para comunicaciones seguras -Ofrece periodo de prueba por 1 año de sus servicios principales. -Ampliamente adoptado en la industria y cuenta con múltiples plataformas de aprendizaje
Microsoft Azure	-Cuenta con servicios como Azure IoT Hub, Azure Functions, Event Grid y Device Provisioning Service (DPS). -Integración nativa con Azure Active Directory y soporte sólido para certificados X.509 y TPM. -Excelente ecosistema para empresas que ya utilizan soluciones de Microsoft y herramientas de desarrollo .NET.
Google Cloud Platform (GCP)	-Integra servicios como Cloud IoT Core (aunque en transición a socios), Cloud Functions y Pub/Sub. -Enfoque en el cifrado por defecto y el modelo "BeyondCorp" para acceso seguro. -Liderazgo en análisis de datos masivos y Machine Learning (BigQuery/Vertex AI) -No muy adaptado en la industria como las otras dos opciones.

Fuente: Elaboración propia.

La elección de Amazon Web Services (AWS) sobre sus competidores no solo va respaldada por sus servicios y su orientación a la seguridad, sino también a su alta adopción en la industria y a que, a la hora de realizar este trabajo, el investigador cuenta con amplia experiencia en el manejo de los servicios de este proveedor.

4.6 Análisis de viabilidad económica, costos de operación y escalabilidad del sistema

Uno de los elementos más importantes para esta investigación es que, al ser enfocada en un ambiente residencial, esta tiene que ser viable, no solo a nivel técnico, sino que debe de ser económicamente sostenible para los usuarios.

Esta solución implementada en AWS utiliza 4 servicios principales: ingreso de datos con AWS IoT Core, procesamiento lógico con AWS Lambda, notificación de alertas con AWS SNS y administración de logs con AWS CloudWatch.

4.6.1 Estimación de costos por servicio

En el siguiente análisis de costos, se tomarán en cuenta los servicios principales utilizados, esto partiendo de un flujo de datos constante cada 60 segundos, las 24 horas del día durante 30 días:

- AWS IoT Core: Se factura por el tiempo que los dispositivos permanecen conectados al broker MQTT, el cual es de $0.08 por cada millón de minutos de conexión; además se factura por cada mensaje enviado o recibido en bloques de 5 KB, este costo es de $1 por cada millón de mensajes. En el nivel gratuito, se tiene 1 millón de solicitudes con 400 mil GB/segundo sin costo adicional (AWS, 2026).
- AWS Lambda: Se factura basado en 2 rubros: solicitudes y duración de ejecución. Con las solicitudes se toma la referencia de $0.2 por cada millón y con el tiempo de ejecución se toma $0.0000166667 como costo por GB/segundo, en una arquitectura x86 (AWS, 2026).
- AWS SNS: El servicio de notificaciones de AWS tiene como costo $2 por cada 100 mil correos, en la capa gratuita se tiene 1000 correos sin costo
- AWS CloudWatch: Este servicio cobra basado en dos variables: ingesta de logs y almacenamiento. El costo de la ingesta es de $0.5 por cada GB de datos y el almacenamiento es de $0.03 por cada GB guardado al mes (AWS, 2026).

4.6.2 Análisis de costos de operación

Una vez aclarados los costos de los servicios podemos tener un aproximado de costos por mes de nuestro sistema. Se toma como referencia que hay 3 dispositivos actualmente conectados, con reportes cada 60 segundos.

- Mensajes enviados: 43200 reportes por mes.
- Minutos de conexión: 129600 minutos al mes.
- Tamaño promedio del mensaje: 1.1KB.

Con la referencia de consumo total por mes podemos definir el costo aproximado por servicio.

- AWS IoT Core: Calculamos por conectividad y por cantidad de mensajes:
  - Conectividad: $\frac{129600\ min}{1000000} * \$0.08 = \$0.0104$

- Mensajería: $43200 * \$0.000001 = \$0.043$
- Subtotal: $0.0557

- AWS Lambda: Cada reporte activa una ejecución de 128MB de RAM con una duración de 200 ms aproximadamente:
  - Solicitudes: $\frac{43200}{1000000} * \$0.2 = \$0.0086$
  - Tiempo de ejecución:
    $\big((0.025GB * s) * 43200\big) * \$0.0000166667 = \$0.018$
  - Subtotal: $0.02664
- AWS SNS: Asumimos 100 brechas por mes para efectos de cálculo de costos ya que están incluidos en el umbral de 1000 por mes.
- AWS CloudWatch: Tamaño de los logs:
  - Logs: $\frac{(1.1KB*43200)}{1024} = 46.4MB * \$0.5 = \$0.025$

Ya definidos los costos por mensajes, tiempo de ejecución, y tamaño de mensaje por mes, podemos tener la proyección final del costo operativo mensual y anual de nuestro sistema:

- Costo mensual: $0.1074
- Costo anual: $1.2888

4.6.3 Análisis de escalabilidad por costo de operación

Esta propuesta no solo debe ser viable para 3 dispositivos, puesto que se tiene constancia de que algunos hogares, se cuenta con 10 y hasta más equipos. En esta sección procederemos a hacer una proyección de costos en el caso que se tuvieran 10, 25, 50 o 100 elementos IoT presentes.

Tabla 10. Proyección de costos con escalabilidad del sistema

Dispositivos en el Hogar	Peso del Resumen (Aprox)	Mensajes Facturados por AWS IoT	Costo Operativo Mensual Total	Costo Operativo Anual

3 equipos (Base)	1.1 KB	43,200 (1 bloque)	~$0.10	$1.20
10 equipos	3.6 KB	43,200 (1 bloque)	~$0.16	$1.92
25 equipos	9.0 KB	86,400 (2 bloques)	~$0.31	$3.72
50 equipos	18.0 KB	172,800 (4 bloques)	~$0.60	$7.20
100 equipos	36.0 KB	345,600 (8 bloques)	~$1.22	$14.64

Fuente: Elaboración propia

Con el anterior análisis podemos comprobar que el sistema es viable y que además presenta una facilidad para escalar a más dispositivos por hogar.

Capítulo 5. Propuesta de Solución

La solución que se propone a continuación se fundamenta en un modelo híbrido que integra un gateway local con la infraestructura de AWS IoT, permitiendo una visibilidad granular del tráfico de red y una detección de anomalías determinadas.

5.1 Arquitectura Conceptual y Flujo de Datos

El diseño del sistema se estructura en una canalización de datos (data pipeline) que abarca desde la captura del paquete en la capa física hasta la ejecución de acciones de mitigación en la nube. Esta arquitectura se divide en cuatro fases críticas: recolección y preprocesamiento en el borde, ingestión y mensajería segura, análisis de comportamiento y detección de eventos, y finalmente, la capa de visualización y respuesta orquestada.

La necesidad de esta segmentación radica en el cumplimiento de los principios de diseño de AWS para soluciones IoT, donde se busca que el dispositivo de borde realice tareas de transformación para evitar cuellos de botella, delegando la correlación compleja a servicios gestionados como AWS IoT Core. El flujo se inicia cuando los dispositivos IoT (cámaras, bombillos, enchufes) se conectan al punto de acceso inalámbrico configurado en el gateway local. Este gateway intercepta el tráfico y extrae los metadatos necesarios.

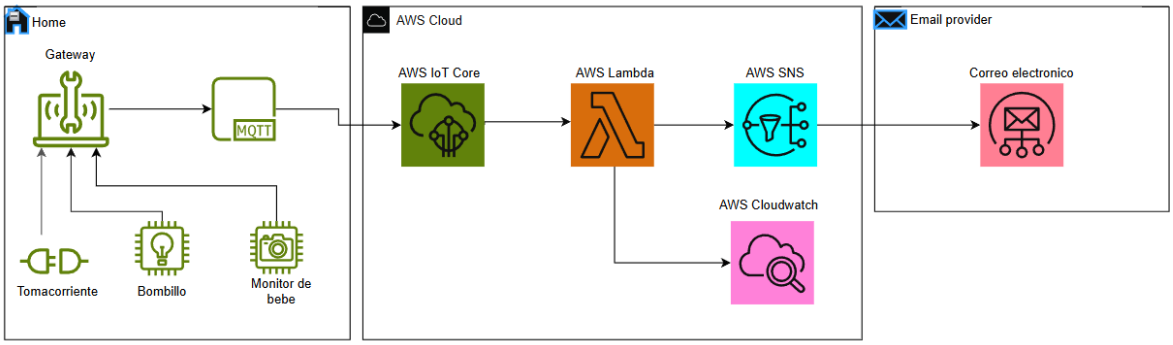

Figura 23. Diagrama para la solución propuesta (Elaboración propia)

5.2 Diseño de la Infraestructura en la Nube

La arquitectura IDS basada en AWS constituye el núcleo de inteligencia del sistema. Se utiliza AWS IoT Core como el punto de entrada seguro, donde cada dispositivo es registrado como un "Thing" (Objeto) con un certificado único que garantiza la autenticación mutua y el cifrado de los datos en tránsito. Luego la data es enviada a AWS Lambda donde, mediante un código de Python, se estará realizando la inteligencia y toma de decisiones del sistema; esta es enviada a su vez a AWS CloudWatch para registro de eventos. Finalmente, dependiendo de las condiciones en AWS Lambda se estará enviando un correo electrónico al usuario con el servicio de AWS SNS, donde contenga la información de la amenaza detectada.

5.2.1 Configurar dispositivos en AWS IoT Core

La configuración inicial se realiza desde la consola de AWS

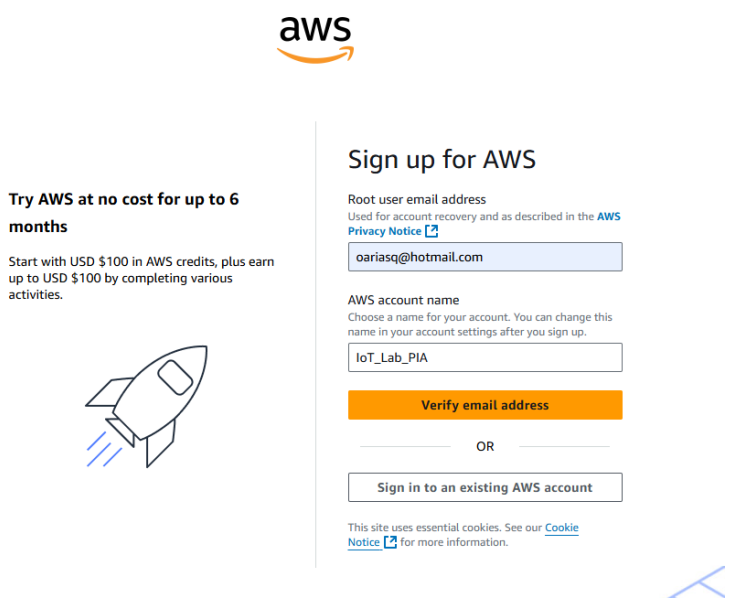

Figura 24. Creación de cuenta en AWS (Elaboración propia)

Posteriormente, es necesario acceder al servicio AWS IoT Core, el cual se puede localizar mediante la barra de búsqueda de la consola

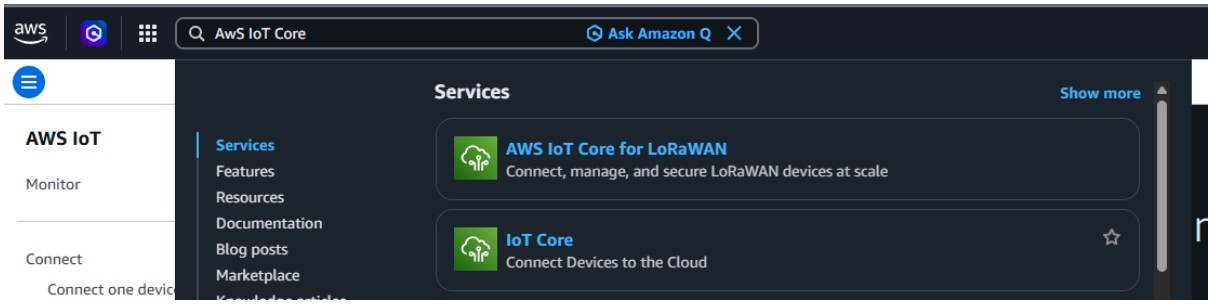

Figura 25. Servicio de AWS IoT Core en AWS (Elaboración propia)

En IoT Core se agregaran los dispositivos, a los cuales se les llama “things” a nivel de la plataforma. En la siguiente imagen se visualiza la consola para la creación de dispositivos:

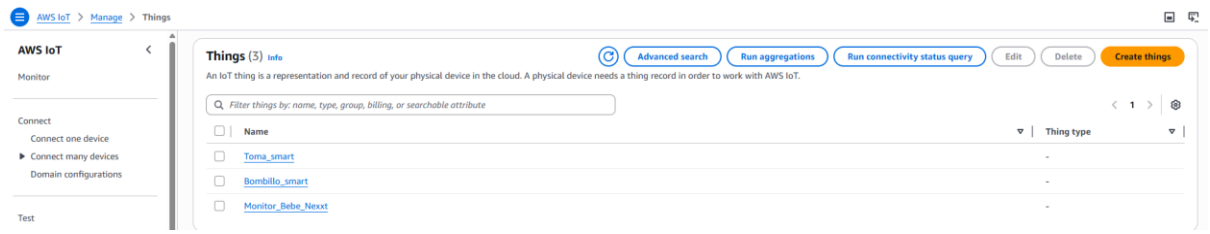

Figura 26. Agregar dispositivos a AWS IoT Core (Elaboración propia)

Posteriormente, se requiere la creación de una política de seguridad orientada a restringir los permisos de los dispositivos. Esta configuración permite que, ante una eventual vulnerabilidad o el compromiso de la llave privada, el acceso se limite exclusivamente al entorno predeterminado, impidiendo la interacción con otros servicios de AWS. El procedimiento para la configuración de estas políticas se detalla a continuación:

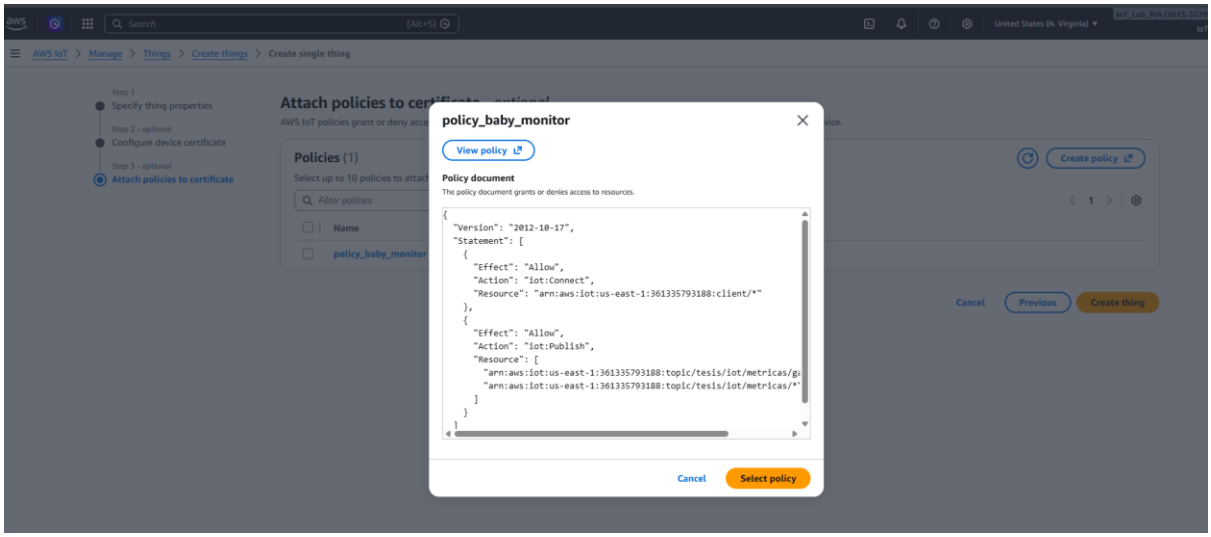

Figura 27. Definir políticas para los dispositivos IoT (Elaboración propia)

Para finalizar se descargan las llaves pública y privada, así como el certificado raíz de AWS para establecer la conexión segura desde el Gateway hasta AWS, dicha configuración se desarrolla según los pasos que se detallan a continuación::

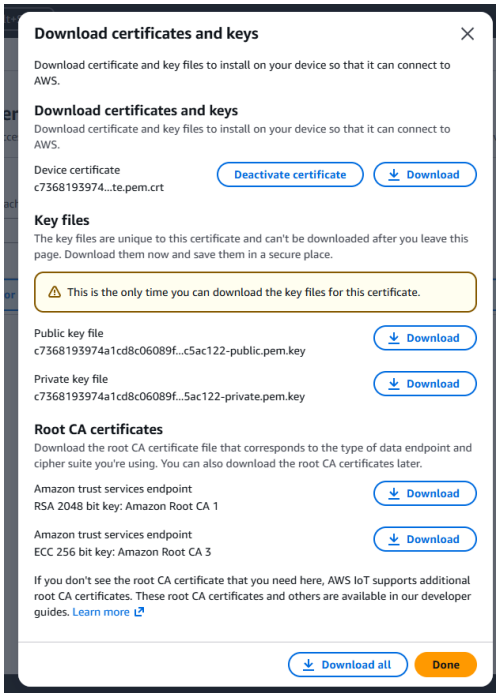

Figura 28. Descargar certificados y llaves de AWS (Elaboración propia)

5.2.2 Implementación de módulo de análisis y toma de decisiones con AWS Lambda

El “cerebro” detrás de la operación del IDS se encuentra configurado de manera serverless con AWS Lambda. A continuación, se muestra la consola en la cual se podrá crear la nueva función:

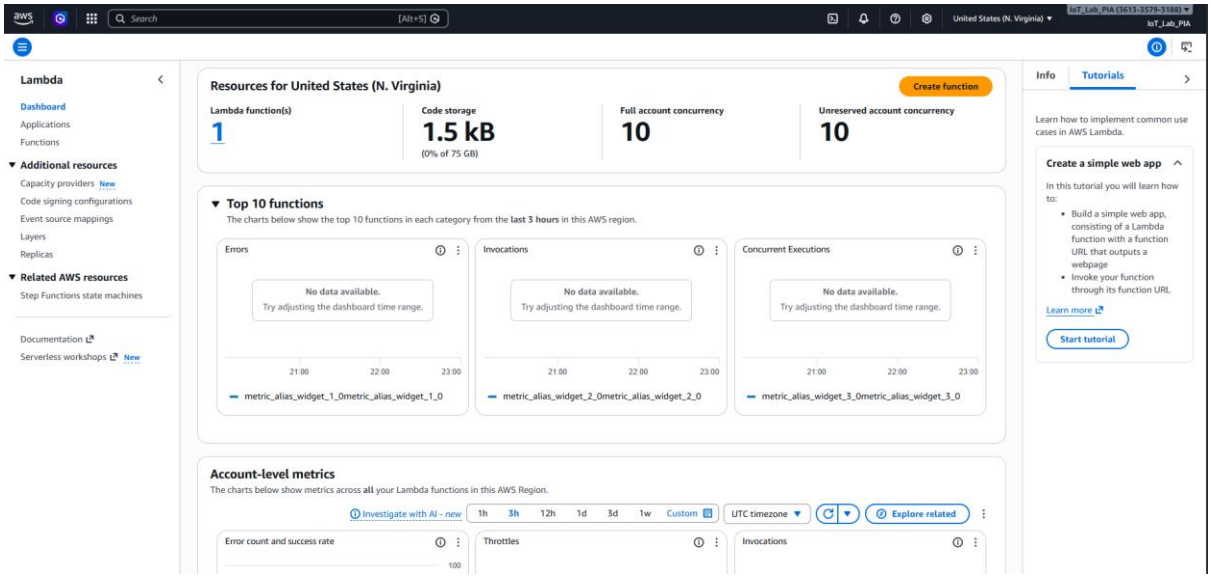

Figura 29. Consola de AWS Lambda (Elaboración propia)

En este servicio, se crea una función que corra en Python 3.14, A continuación, el procedimiento para crearla:

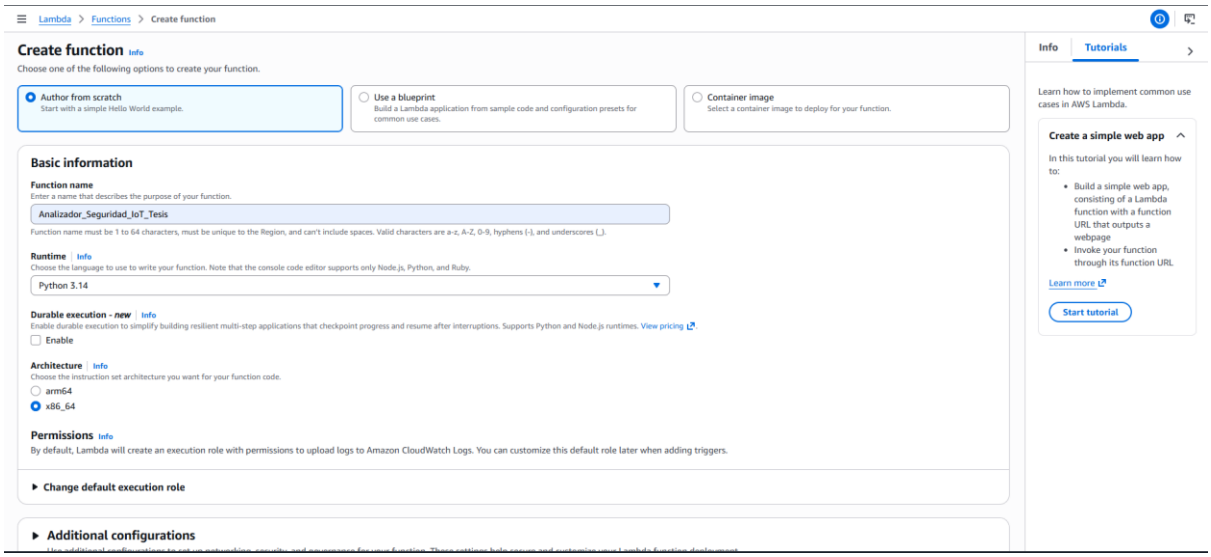

Figura 30. Crear nueva función en AWS Lambda (Elaboración propia)

Al tener la función definida, se creará el código de Python que realizará el análisis y toma de decisiones. Este se implementa de la siguiente manera:

#Función AWS Lambda para evaluar el tráfico reportado por el Gateway IoT.

#Actúa como el cerebro en la nube que decide si los eventos de red locales son una amenaza real.

```
import json

import boto3

import ipaddress


#========================================

#1. LISTA BLANCA DE REDES (WHITELIST)

#========================================
```

```
#Lista de rangos de red (CIDR) conocidos y confiables.
#Si un dispositivo se comunica con una IP fuera de estos rangos, se asume
#que podría estar comprometido o enviando datos a un servidor malicioso.
NETWORKS_WHITELIST = [
    #--- INFRAESTRUCTURA LOCAL Y CONTROL ---
    '192.168.137.0/24', '192.168.1.0/24',   #Tus subredes locales
    '224.0.0.0/4',                      #Multicast (IGMP/mDNS)
    '239.255.255.250/32',                   #SSDP Discovery
    '255.255.255.255/32',                   #DHCP/Broadcast

    #--- AWS (Rangos específicos detectados) ---
    '3.64.0.0/12',      #AWS Frankfurt
    '54.180.0.0/14',    #AWS IoT Core / Services
    '44.192.0.0/10',    #AWS US Regions
    '34.192.0.0/10',    #AWS Cloudfront/Global

    #--- GOOGLE & CLOUD COMPLEMENTARIOS ---
    '34.0.0.0/8', '35.0.0.0/8',       #Google Cloud Services
    '142.251.0.0/16',                 #Google Infrastructure
    '104.16.0.0/12',                  #Cloudflare (Updates/Firmware)
    '15.204.0.0/16',                  #OVH SAS (Backend del fabricante)

    #--- MICROSOFT / AZURE ---
    '52.96.0.0/12',     #Azure / Outlook
```

```
    '20.96.0.0/12',     #Azure IoT
    '40.124.0.0/14',    #Azure Services

    #--- OTROS (Apple & Carriers) ---
    '17.248.0.0/13',    #Apple HomeKit
    '8.240.0.0/13',     #Lumen (Tránsito de red)
    '57.144.0.0/16'     #Meta (Notificaciones/Integración)
]

#========================================
#2. CONFIGURACIÓN DE FIRMAS Y ALERTAS
#========================================
#Huella digital esperada en el payload (hexadecimal) para tráfico legítimo de los dispositivos Tuya/Nexxt.
FIRMA_LEGITIMA = "000055aa"

#Cliente de AWS Simple Notification Service (SNS) para enviar correos o SMS al administrador.
sns = boto3.client('sns')
TOPIC_ARN = "arn:aws:sns:us-east-1:361335793188:Alerta_Seguridad_IoT"

#========================================
#3. MOTOR DE EVALUACIÓN LAMBDA
#========================================
def lambda_handler(event, context):
```

#Punto de entrada principal que ejecuta AWS cuando llega un nuevo JSON desde el Gateway local.

```
print(f"Evento: {json.dumps(event)}")
devices_data = event.get('devices', [])
if not devices_data: devices_data = [event]

alertas_globales = []
hubo_brecha = False

#Evaluamos cada dispositivo reportado de manera individual
for item in devices_data:
    device = item.get('device', 'Desconocido')
    m = item.get('metrics', {})
    alertas_este = []

    #--- Validación Capa 3 (Control de Destinos) ---
    #Revisamos si el dispositivo se comunicó con IPs de internet que no están en nuestra lista blanca.
    ips_intrusas = [ip for ip in m.get('ips_externas', []) if not any(ipaddress.ip_address(ip) in ipaddress.ip_network(net) for net in NETWORKS_WHITELIST)]
    if ips_intrusas:
        hubo_brecha = True
        alertas_este.append(f"  [!] IPs no autorizadas: {ips_intrusas}")
```

```
    #--- Validación Capa 7 (Inspección Profunda - DPI) ---
    #Si el puerto 6667 está activo, verificamos que el contenido del paquete tenga la firma de la marca.
    #Si no coincide, es probable que alguien esté inyectando tráfico falso.
    puertos = m.get('puertos', {}).get('lista_permitidos', []) + m.get('puertos', {}).get('lista_no_permitidos', [])
    if 6667 in puertos:
        for f in m.get('firmas_detectadas', []):
            if f != FIRMA_LEGITIMA:
                hubo_brecha = True
                alertas_este.append(f"  [!] DPI: Firma anómala '{f}' en puerto 6667.")

    #--- Auditoría de Puertos Críticos ---
    #Los puertos del 0 al 1024 son "privilegiados" (ej. HTTP, SSH, Telnet).
    #Un dispositivo IoT genérico no debería intentar usar estos puertos en tu red interna.
    criticos = [p for p in m.get('puertos', {}).get('lista_no_permitidos', []) if p <= 1024]
    if criticos:
        hubo_brecha = True
        alertas_este.append(f"  [!] Puertos críticos detectados: {criticos}")

    #--- Heurística (Banderas rojas levantadas por el Gateway local) ---
    anomalias = m.get('anomalias', [])
```

```
        #Alertas de escaneo masivo (Típico de la fase de reconocimiento de un atacante)
        if any("ALERTA_ESCANEADOR" in a or "ARP_SCAN_SUSPECT" in a for a in anomalias):
            hubo_brecha = True
            alertas_este.append("  [!] CRÍTICO: Escaneo masivo detectado (Posible reconocimiento).")

        #Alerta de Ping (Comportamiento inusual para un IoT)
        if "ICMP_REQUEST_DETECTADO" in anomalias:
            hubo_brecha = True
            alertas_este.append("  [!] Advertencia: Tráfico ICMP (Ping) detectado.")

        #Alerta de Ráfagas (Podría indicar un intento de DDoS o exfiltración masiva de datos)
        for anom in anomalias:
            if "EXFIL_SUSPECT" in anom:
                hubo_brecha = True
                alertas_este.append(f"  [!] Advertencia: Ráfaga anómala en puerto {anom.split('_')[-1]}.")

        #--- Construcción del Reporte Individual ---
        if alertas_este:
            alertas_globales.append(f"\n--- DISPOSITIVO: {device} [🔴 VULNERADO] ---")
            alertas_globales.extend(alertas_este)
        else:
```

```
            alertas_globales.append(f"\n--- DISPOSITIVO: {device} [🟢 SEGURO] ---")
            alertas_globales.append("  [✓] Tráfico bajo norma.")

    #--- Envío de Notificaciones ---
    #Si se detectó al menos un comportamiento peligroso, se dispara la alerta por Amazon SNS.
    if hubo_brecha:
        cuerpo = "--- REPORTE CONSOLIDADO IDS ---\n" + "\n".join(alertas_globales)
        sns.publish(TopicArn=TOPIC_ARN, Message=cuerpo, Subject="Evento Crítico IoT")
        print("Notificación enviada.")

    return {"status": "procesado", "hubo_brecha": hubo_brecha}
```

Tras completar las etapas descritas, el sistema genera automáticamente una regla de enrutamiento para dirigir los eventos hacia AWS CloudWatch. La evidencia de esta integración y el flujo de los datos se presentan en el siguiente apartado:

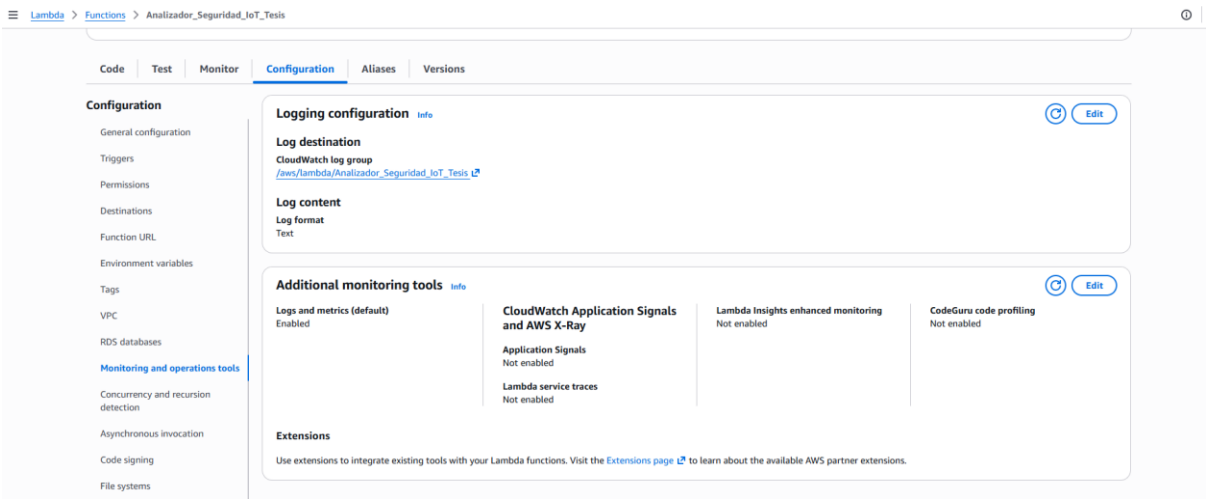

Figura 31. Envío de eventos a AWS CloudWatch (Elaboración propia)

5.2.3 Conectar AWS IoT Core con AWS Lambda y AWS CloudWatch

El envío de eventos desde IoT Core a Lambda y CloudWatch debe configurarse como una regla de reenvío de mensajes desde IoT Core, esto de la siguiente manera:

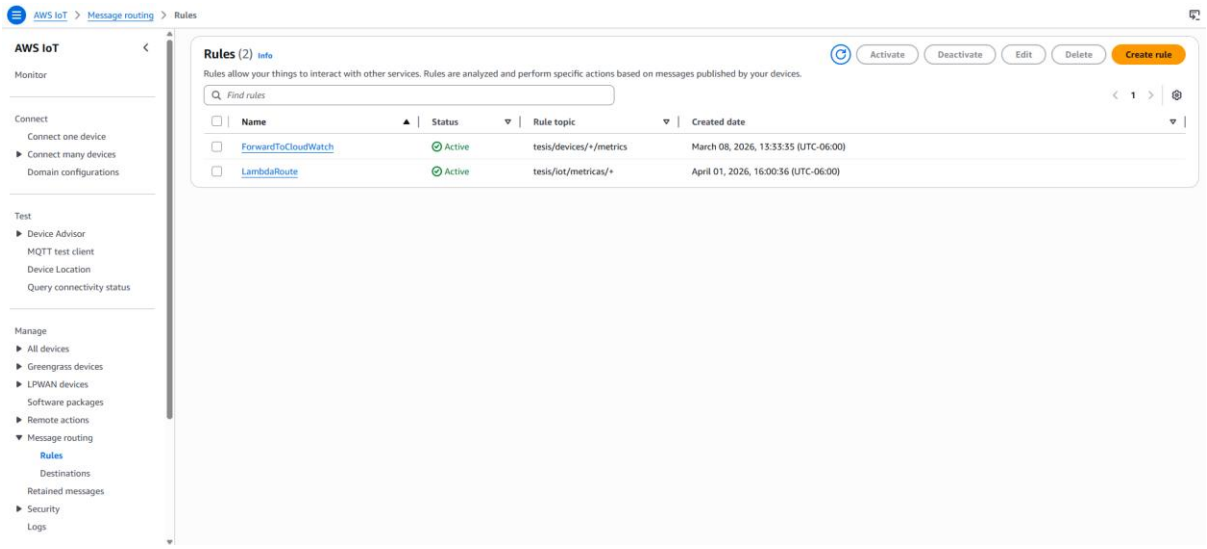

Figura 32. Envío de mensajes desde AWS IoT Core (Elaboración propia)

Cuando se crea esta regla, se puede configurar en AWS Lambda el disparador (trigger) que iniciará la ejecución del código en AWS Lambda, este disparador se configura de la siguiente manera:

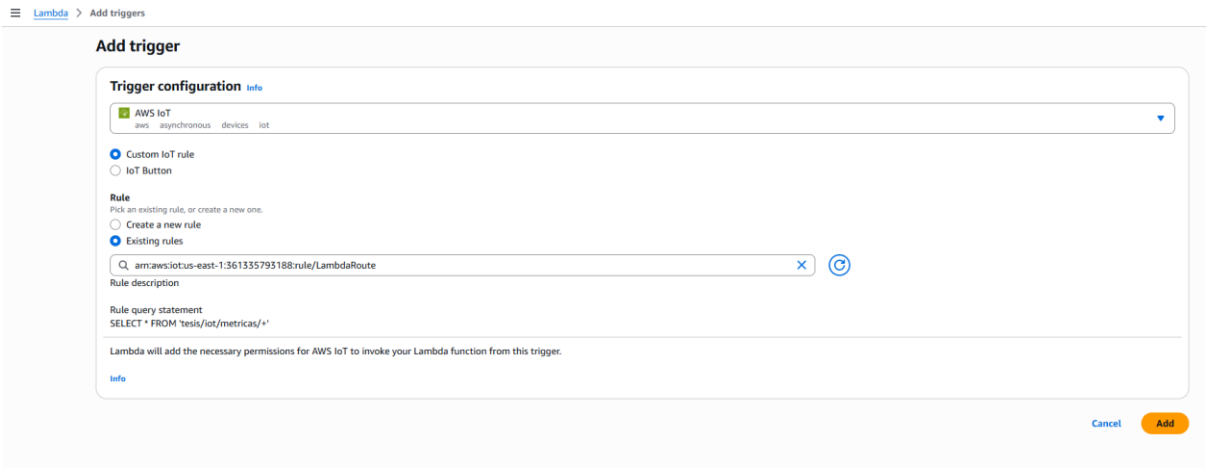

Figura 33. Creación de regla en AWS Lambda (Elaboración propia)

5.2.4 Configurar cuenta y suscripción en Amazon SNS

Para el envío de correos electrónicos una vez se detecten las alertas de seguridad se utilizará Amazon SNS (Simple Notification Service), el cual requiere de forma simple asociar un tema y una suscripción al correo electrónico donde se enviarán las notificaciones. Esto se logra desde la consola de SNS de la siguiente manera:

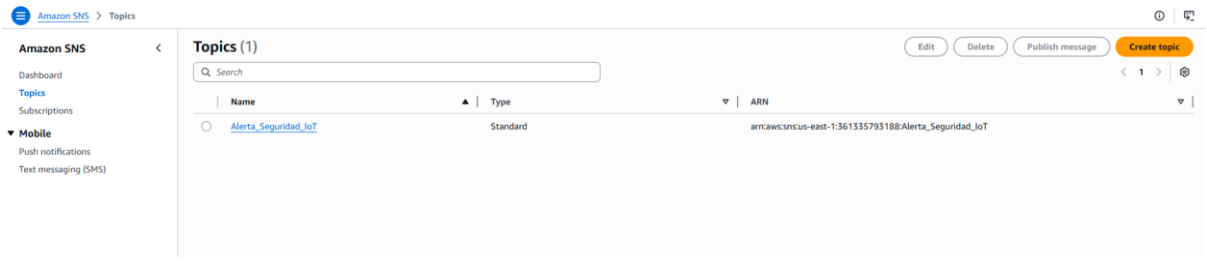

Figura 34. Consola de AWS SNS (Elaboración propia)

5.3 Implementación del Módulo de Captura y Preprocesamiento Ligero

Esta investigación requiere la implementación de un módulo capaz de recolectar y anonimizar el tráfico residencial. Para ello, se propone el uso de una computadora dedicada que actúa como gateway local, utilizando herramientas de manipulación de paquetes como Scapy. La elección de estas herramientas se justifica por su capacidad para manejar flujos de datos en tiempo real y su extensibilidad mediante scripts de Python, lo que facilita la creación de un agente de métricas personalizado.

5.3.1 Creación de código de Python para el análisis del tráfico

La implementación del Gateway local se realiza para recolectar la información necesaria a la hora del análisis de seguridad por parte de AWS, donde se realiza la mayor carga analítica. Dicho esto, el código contiene múltiples filtros, conversiones y reglas para capturar el tráfico estrictamente necesario, sin comprometer la carga de RAM o CPU del hardware ni saturar la red con cantidades masivas de datos subidos hacia la nube. El código de Python se puede observar a continuación:

#Sistema de Detección de Intrusiones (IDS) basado en arquitectura Edge computing para redes IoT.

#Implementación con Interfaz Gráfica de Usuario (GUI) y procesamiento concurrente (Multithreading).

```
import os
import time
import json
import logging
import threading
import tkinter as tk
from tkinter import ttk, scrolledtext, messagebox
from scapy.all import sniff, IP, TCP, UDP, ICMP, ARP
from scapy.interfaces import get_working_ifaces
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient

#========================================
#1. PARÁMETROS DE CONFIGURACIÓN Y RUTAS
#========================================
ENDPOINT = "a3hag5eik9z9fz-ats.iot.us-east-1.amazonaws.com"
PATH_BASE = r"C:\Users\Oscar\Documents\Tesis\iot_core"
```

#Ruta del certificado raíz (Root CA) requerido para la autenticación TLS mutua con AWS IoT Core.

```
ROOT_CA = os.path.join(PATH_BASE, "cert_camara_monitor", "AmazonRootCA1.pem")
```

#Lista blanca de puertos autorizados. El tráfico fuera de este espectro se clasificará como anomalía de puerto.

```
PUERTOS_PERMITIDOS_REF = [53, 80, 443, 554, 1443, 1900, 3478, 5353, 5355, 6666, 6667, 8554, 8883, 8886]


#Estructuras de datos en memoria para el monitoreo de estado y detección de patrones anómalos a corto plazo.
contadores_deteccion = {
    "arp_requests": {},
    "discovery_burst": {}
}


def crear_metricas_vacias():
    #Inicialización de la estructura de datos que albergará las métricas de tráfico por ciclo de evaluación.
    return {
        "paquetes_totales": 0,
        "ips_externas_detectadas": set(),
        "puertos_permitidos_detectados": set(),
        "puertos_no_permitidos_detectados": set(),
        "firmas_payload_udp": set(),
        "anomalias_protocolo": set()
    }


#Diccionario de registro de dispositivos IoT, incluyendo parámetros de autenticación criptográfica y métricas individuales.
DISPOSITIVOS = {
```

```
    "192.168.137.58": {
        "type": "sensor", "thing_name": "Monitor_Bebe_Nexxt",
        "cert": os.path.join(PATH_BASE, "cert_camara_monitor",
"eed11e1cf7746e870586f72aeac0876a700d970b344ef357d5dc9182cffddf26-
certificate.pem.crt"),
        "key": os.path.join(PATH_BASE, "cert_camara_monitor",
"eed11e1cf7746e870586f72aeac0876a700d970b344ef357d5dc9182cffddf26-
private.pem.key"),
        "metricas": crear_metricas_vacias(), "mqtt_client": None
    },
    "192.168.137.247": {
        "type": "actuator", "thing_name": "Bombillo_smart",
        "cert": os.path.join(PATH_BASE, "cert_bombillo",
"40a0c5f827895b77c7ae845c7cd6b80384f19fa2e62a69245281f93c592847ea-
certificate.pem.crt"),
        "key": os.path.join(PATH_BASE, "cert_bombillo",
"40a0c5f827895b77c7ae845c7cd6b80384f19fa2e62a69245281f93c592847ea-
private.pem.key"),
        "metricas": crear_metricas_vacias(), "mqtt_client": None
    },
    "192.168.137.223": {
        "type": "actuator", "thing_name": "Toma_smart",
        "cert": os.path.join(PATH_BASE, "cert_tomacorriente",
"e015b132157c6219c695c2fd0f9291f5e574ef32610812a6f402d532a85619b9-
certificate.pem.crt"),
        "key": os.path.join(PATH_BASE, "cert_tomacorriente",
"e015b132157c6219c695c2fd0f9291f5e574ef32610812a6f402d532a85619b9-
private.pem.key"),
```

```
        "metricas": crear_metricas_vacias(), "mqtt_client": None
    }
}

def setup_mqtt():
    #Inicializa y configura los clientes MQTT implementando seguridad mTLS para cada dispositivo registrado.
    for ip, info in DISPOSITIVOS.items():
        if info["mqtt_client"] is None:
            client = AWSIoTMQTTClient(info["thing_name"])
            client.configureEndpoint(ENDPOINT, 8883)
            client.configureCredentials(ROOT_CA, info["key"], info["cert"])
            client.connect()
            info["mqtt_client"] = client
            logging.info(f"Conexión mTLS establecida exitosamente con AWS IoT Core: {info['thing_name']}")

#========================================
#2. MOTOR DE ANÁLISIS DE TRÁFICO (SNIFFER DE RED)
#========================================
def procesar_paquete(packet):
    #Función de retrollamada (callback) ejecutada por cada paquete capturado en la interfaz de red.
```

#Detección de reconocimiento de red (Capa 2). Identifica ráfagas de solicitudes ARP que indican un posible escaneo.

if ARP in packet and packet[ARP].op == 1:

src_ip = packet[ARP].psrc

contadores_deteccion["arp_requests"][src_ip] = contadores_deteccion["arp_requests"].get(src_ip, 0) + 1

if contadores_deteccion["arp_requests"][src_ip] > 10:

target_dev = src_ip if src_ip in DISPOSITIVOS else list(DISPOSITIVOS.keys())[0]

DISPOSITIVOS[target_dev]["metricas"]["anomalias_protocolo"].add(f"ALERTA_ESCANEADOR_EXTERNO_{src_ip}")

#Inspección de las capas de Red y Transporte (Capa 3 a 7).

if IP in packet:

src_ip, dst_ip = packet[IP].src, packet[IP].dst

dispositivo_ip = src_ip if src_ip in DISPOSITIVOS else (dst_ip if dst_ip in DISPOSITIVOS else None)

#Filtrado espacial: el análisis procede únicamente si el paquete pertenece a un dispositivo registrado en el sistema.

if dispositivo_ip:

metricas = DISPOSITIVOS[dispositivo_ip]["metricas"]

metricas["paquetes_totales"] += 1

externa = src_ip if src_ip != dispositivo_ip else dst_ip

metricas["ips_externas_detectadas"].add(externa)

#Identificación de tráfico ICMP, el cual es anómalo en el perfil de operación estándar de estos dispositivos IoT.

if ICMP in packet:

metricas["anomalias_protocolo"].add("ICMP_REQUEST_DETECTADO")

puerto = None

if TCP in packet: puerto = packet[TCP].dport

elif UDP in packet:

puerto = packet[UDP].dport

#Inspección Profunda de Paquetes (DPI). Verificación de la firma hexadecimal en la carga útil (payload) UDP en el puerto 6667.

if puerto == 6667 and packet[UDP].payload:

payload_hex = bytes(packet[UDP].payload)[:4].hex()

if payload_hex: metricas["firmas_payload_udp"].add(payload_hex)

#Análisis de heurística por volumen. Identifica picos anómalos en puertos de descubrimiento (SSDP/mDNS) que sugieren exfiltración.

if puerto in [1900, 5353]:

if dispositivo_ip not in contadores_deteccion["discovery_burst"]:

contadores_deteccion["discovery_burst"][dispositivo_ip] = {}

actual = contadores_deteccion["discovery_burst"][dispositivo_ip].get(puerto, 0) + 1

contadores_deteccion["discovery_burst"][dispositivo_ip][puerto] = actual

if actual > 50: metricas["anomalias_protocolo"].add(f"EXFIL_SUSPECT_PORT_{puerto}")

```
        #Clasificación de puertos de acuerdo con las directivas de seguridad establecidas en la lista blanca.
        if puerto:
            if puerto in PUERTOS_PERMITIDOS_REF: metricas["puertos_permitidos_detectados"].add(puerto)
            else: metricas["puertos_no_permitidos_detectados"].add(puerto)


def consolidar_y_enviar():
    #Consolidación de las métricas temporales en formato JSON y transmisión segura hacia la infraestructura en la nube.
    reporte = {"gateway_id": "Hotspot_IDS_Tesis", "timestamp": time.strftime("%Y-%m-%d %H:%M:%S"), "devices": []}
    for ip, info in DISPOSITIVOS.items():
        m = info["metricas"]
        if m["paquetes_totales"] > 0 or m["anomalias_protocolo"]:
            reporte["devices"].append({
                "device": info["thing_name"], "type": info["type"],
                "metrics": {
                    "paquetes_totales": m["paquetes_totales"],
                    "ips_externas": list(m["ips_externas_detectadas"]),
                    "firmas_detectadas": list(m["firmas_payload_udp"]),
                    "anomalias": list(m["anomalias_protocolo"]),
                    "puertos": {
                        "lista_permitidos": list(m["puertos_permitidos_detectados"]),
                        "lista_no_permitidos": list(m["puertos_no_permitidos_detectados"])
```

```
                }
            }
        })
        #Restablecimiento de las estructuras de datos para iniciar un nuevo ciclo de evaluación temporal.
        info["metricas"] = crear_metricas_vacias()

    if reporte["devices"]:
        client = list(DISPOSITIVOS.values())[0]["mqtt_client"]
        client.publish("tesis/iot/metricas/gateway_summary", json.dumps(reporte), 1)
        logging.info("Matriz de métricas consolidada y transmitida exitosamente a AWS IoT Core.")

    contadores_deteccion["arp_requests"].clear()
    contadores_deteccion["discovery_burst"].clear()

#=========================================
#3. ARQUITECTURA DE LA INTERFAZ DE USUARIO Y GESTIÓN DE HILOS
#=========================================
class TextHandler(logging.Handler):
    #Clase manejadora para redirigir el flujo estándar de registro (logging) hacia el widget de texto de la interfaz gráfica.
    def __init__(self, text_widget):
        logging.Handler.__init__(self)
        self.text_widget = text_widget
```

```
    def emit(self, record):
        msg = self.format(record)
        def append():
            self.text_widget.configure(state='normal')
            self.text_widget.insert(tk.END, msg + '\n')
            self.text_widget.configure(state='disabled')
            self.text_widget.yview(tk.END)
        self.text_widget.after(0, append)

class IDSApplication:
    def __init__(self, root):
        self.root = root
        self.root.title("Gateway IDS - Sistema de Monitoreo de Seguridad IoT")
        self.root.geometry("700x500")

        self.is_running = False
        self.interfaces = get_working_ifaces()

        #Inicialización y distribución espacial de los componentes gráficos (widgets) en la ventana principal.
        frame_top = tk.Frame(root, pady=10, padx=10)
        frame_top.pack(fill=tk.X)
```

```
tk.Label(frame_top, text="Interfaz de Red Activa:", font=("Arial", 10, "bold")).pack(side=tk.LEFT)

self.combo_ifaces = ttk.Combobox(frame_top, width=50, state="readonly")
self.combo_ifaces['values'] = [f"{i.description} (IP: {getattr(i, 'ip', 'N/A')})" for i in self.interfaces]
if self.interfaces:
    self.combo_ifaces.current(0)
self.combo_ifaces.pack(side=tk.LEFT, padx=10)

frame_buttons = tk.Frame(root, pady=5)
frame_buttons.pack(fill=tk.X)

self.btn_start = tk.Button(frame_buttons, text="▶ INICIAR CAPTURA", bg="#4CAF50", fg="white", font=("Arial", 10, "bold"), command=self.start_monitoring)
self.btn_start.pack(side=tk.LEFT, padx=20)

self.btn_stop = tk.Button(frame_buttons, text="■ INTERRUMPIR EJECUCIÓN", bg="#f44336", fg="white", font=("Arial", 10, "bold"), state=tk.DISABLED, command=self.stop_monitoring)
self.btn_stop.pack(side=tk.LEFT)

tk.Label(root, text="Consola de Eventos y Auditoría:", font=("Arial", 10, "bold")).pack(anchor="w", padx=10)
self.log_text = scrolledtext.ScrolledText(root, state='disabled', bg="black", fg="#00FF00", font=("Consolas", 9))
```

```
        self.log_text.pack(fill=tk.BOTH, expand=True, padx=10, pady=10)


        #Configuración del nivel y formato de registro para el flujo de salida hacia la GUI.
        text_handler = TextHandler(self.log_text)
        text_handler.setFormatter(logging.Formatter('%(asctime)s - [%(levelname)s] %(message)s'))
        logging.getLogger().addHandler(text_handler)
        logging.getLogger().setLevel(logging.INFO)


    def sniff_worker(self, interfaz, filtro):
        #Función delegada al hilo de captura de red. La declaración del timeout facilita la salida ordenada del bucle (graceful exit).
        logging.info(f"Módulo de inspección inicializado en la interfaz: {interfaz.description}")
        while self.is_running:
            sniff(iface=interfaz, filter=filtro, prn=procesar_paquete, store=False, timeout=2)


    def report_worker(self):
        #Función delegada al hilo de transmisión temporal. Garantiza la ejecución periódica sin utilizar llamadas recursivas.
        ciclo_espera = 60
        while self.is_running:
            for _ in range(ciclo_espera):
                if not self.is_running: return
                time.sleep(1)
```

```
            consolidar_y_enviar()


    def start_monitoring(self):
        #Validación de precondiciones y alteración de los estados de los widgets preventivos ante condiciones de carrera.
        idx = self.combo_ifaces.current()
        if idx == -1:
            messagebox.showwarning("Error de Parámetros", "Es mandatorio seleccionar una interfaz de red válida.")
            return

        self.is_running = True
        self.btn_start.config(state=tk.DISABLED)
        self.btn_stop.config(state=tk.NORMAL)
        self.combo_ifaces.config(state=tk.DISABLED)

        interfaz = self.interfaces[idx]
        filtro = f"arp or ({' or '.join([f'host {ip}' for ip in DISPOSITIVOS.keys()])})"

        #Despliegue del proceso de conexión en un hilo secundario (daemon thread) para evitar el bloqueo del bucle principal de eventos de la GUI.
        threading.Thread(target=self.startup_worker, args=(interfaz, filtro), daemon=True).start()

    def startup_worker(self, interfaz, filtro):
```

```
        logging.info("Negociando protocolos de seguridad y conexión con infraestructura en la nube...")
        try:
            setup_mqtt() #Ejecuta el protocolo de enlace (handshake) TLS con AWS de manera asíncrona.
        except Exception as e:
            logging.error(f"Excepción en la negociación mTLS con AWS: {e}")
            #Recuperación de estado seguro en la Interfaz Gráfica ante un fallo de conectividad.
            self.root.after(0, self.reset_gui_on_error)
            return

        logging.info("Canales seguros establecidos. Activando módulos de análisis de paquetes...")

        #Asignación de recursos e instanciación de los procesos concurrentes para el núcleo del IDS.
        self.thread_sniff = threading.Thread(target=self.sniff_worker, args=(interfaz, filtro), daemon=True)
        self.thread_report = threading.Thread(target=self.report_worker, daemon=True)

        self.thread_sniff.start()
        self.thread_report.start()
        logging.info("--- MOTOR IDS DESPLEGADO Y OPERATIVO ---")

    def reset_gui_on_error(self):
```

#Función de retrollamada para la restauración del estado base de la GUI posterior a una condición de error.

self.is_running = False

self.btn_start.config(state=tk.NORMAL)

self.btn_stop.config(state=tk.DISABLED)

self.combo_ifaces.config(state="readonly")

messagebox.showerror("Excepción de Red", "Fallo en la resolución o conectividad hacia AWS IoT Core. Verifique los certificados criptográficos y la resolución DNS.")

def stop_monitoring(self):

#Modificación de la variable global de estado para inducir la terminación controlada (graceful shutdown) de los hilos subyacentes.

self.is_running = False

self.btn_start.config(state=tk.NORMAL)

self.btn_stop.config(state=tk.DISABLED)

self.combo_ifaces.config(state="readonly")

logging.info("--- SECUENCIA DE APAGADO INICIADA --- (Liberando recursos e hilos en segundo plano...)")

if __name__ == "__main__":

root = tk.Tk()

app = IDSApplication(root)

root.mainloop()

5.3.2 Configuración de red como Zona de cobertura Inalámbrica Móvil

El punto crítico para que todo el sistema funcione se basa originalmente en que los dispositivos IoT puedan conectarse a un Gateway local, el cual provee acceso a internet, así como acceso a los servicios configurados en la nube específicos para las tareas necesarias a la hora de realizar el análisis del tráfico. Esta configuración se lleva a cabo de la siguiente manera: Configuración - Red e Internet - Zona con cobertura inalámbrica móvil, de la siguiente manera:

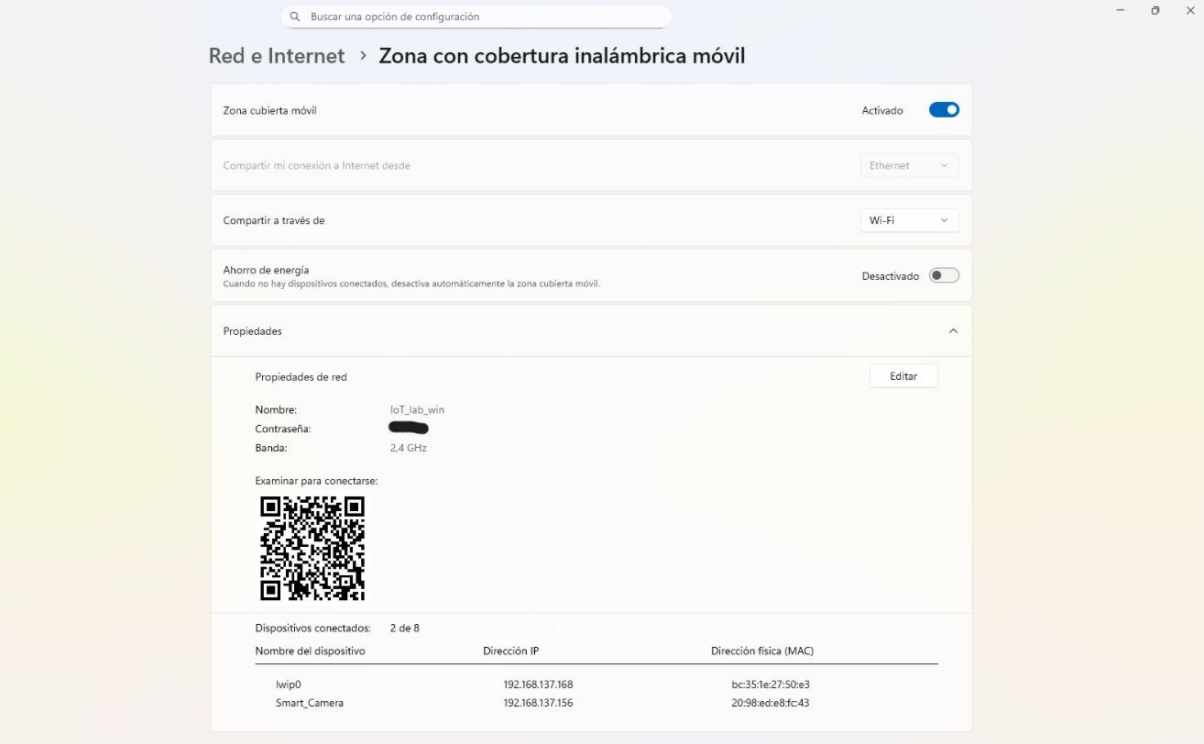

Figura 35. Configuración de Zona con cobertura (Elaboración propia)

5.3.3 Comunicación mediante MQTT y Gestión de Temas

La comunicación entre el gateway local y la nube se realiza a través del protocolo MQTT, esto permite publicar mensajes desde el hogar hasta AWS. Los reportes de métricas se publican en temas específicos de IoT Core. Esta estructura de temas permite que AWS reconozca el formato del mensaje y lo dirija al motor de detección. La manera en la que se visualizan los mensajes en el cliente de pruebas se muestra en la siguiente imagen:

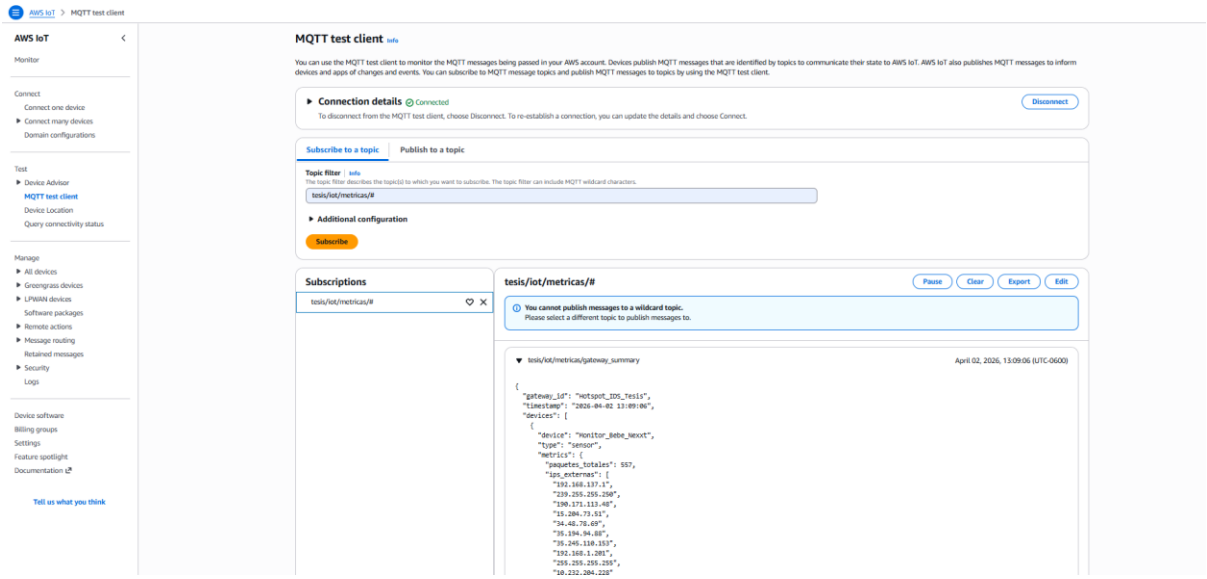

Figura 36. Cliente MQTT para pruebas de conexión (Elaboración propia)

La seguridad de esta capa se refuerza mediante políticas de IoT de "mínimo privilegio", donde el gateway solo tiene permisos para publicar en sus temas específicos. Esto previene que un compromiso en el gateway local sea utilizado para realizar ataques de suplantación contra otros servicios de la cuenta de AWS.

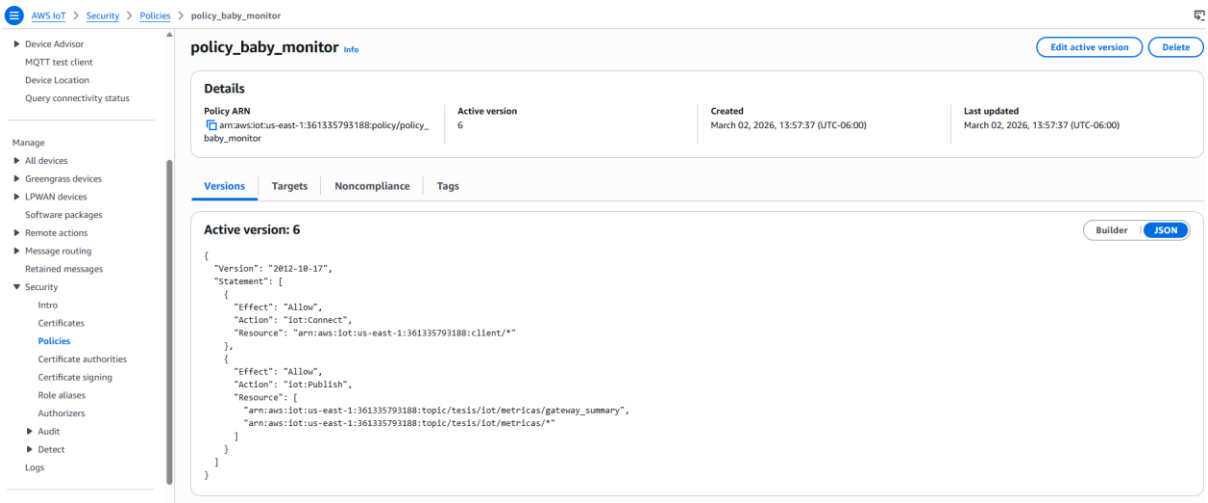

Figura 37. Políticas de seguridad para los dispositivos IoT (Elaboración propia)

5.4 Fase de pruebas y Validación del Sistema

Una vez integrados los servicios de AWS con el Gateway, se procede a iniciar el sistema de detección de intrusos (IDS), el cual estará notificando en el caso de detectar alguna anomalía de seguridad. Para las pruebas de funcionalidad, se utilizará

una máquina virtual, la cual se encuentra en el mismo segmento de red y brindará el acceso a herramientas de pruebas de seguridad, para corroborar que los eventos se están detectando correctamente.

5.4.1 Ejecución del programa de monitoreo

Para esta prueba se creó un archivo ejecutable para mayor facilidad del usuario final. El cual debe correr el programa para iniciar el sistema de monitoreo. Se puede ver el archivo y la interfaz gráfica en las figuras 44 y 45:

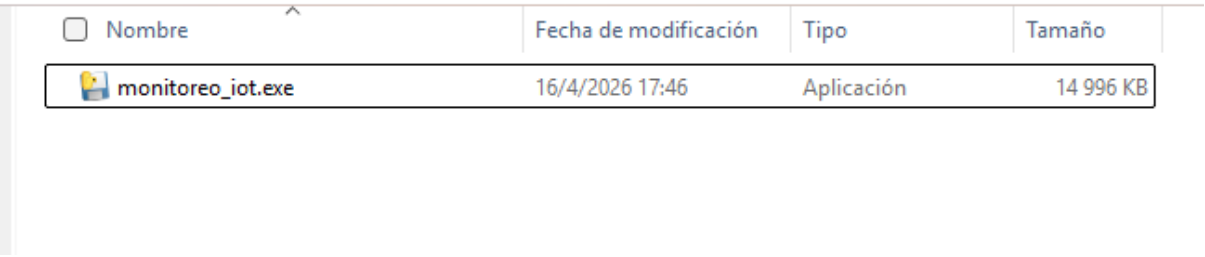

Figura 38. Archivo ejecutable para iniciar el programa (Elaboración propia)

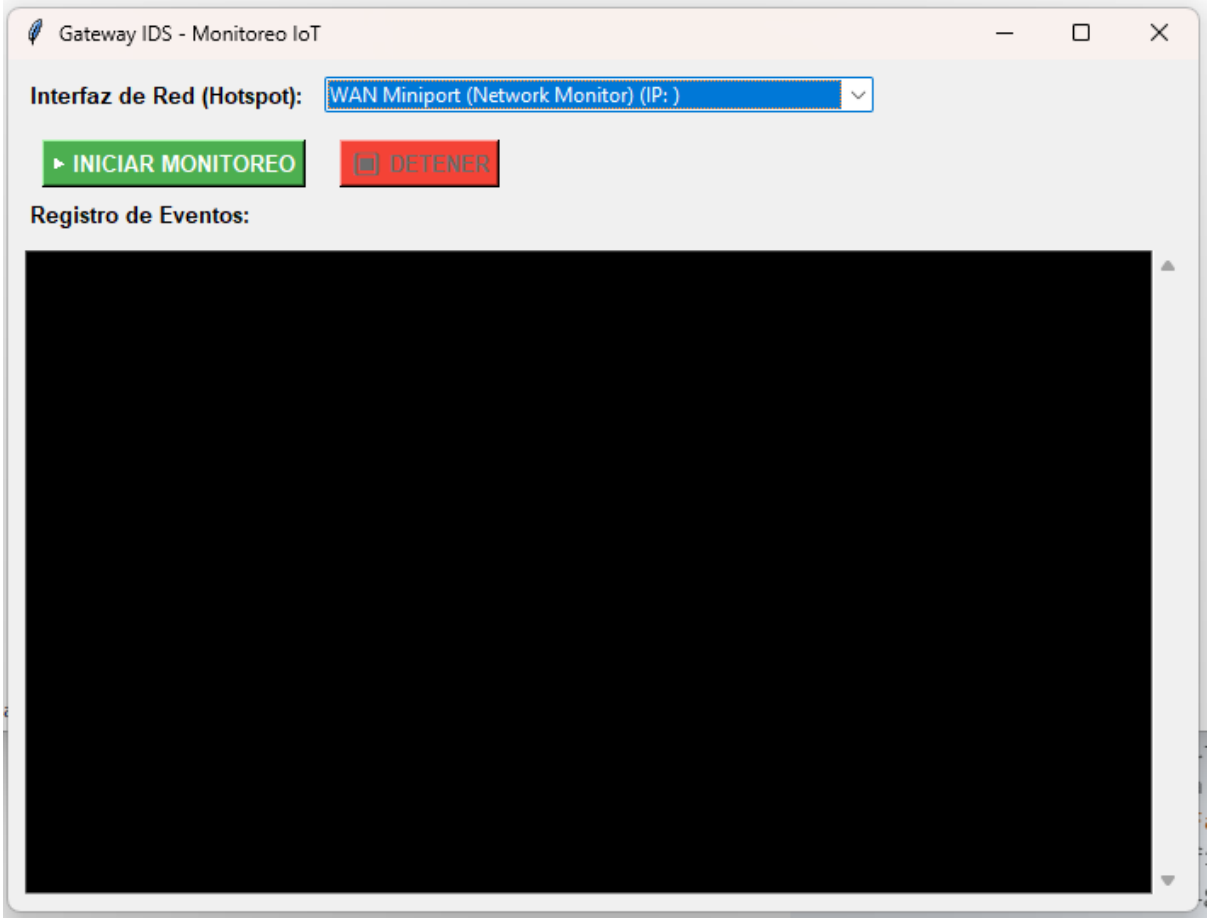

Figura 39. Archivo ejecutable para iniciar el programa (Elaboración propia)

Una vez ejecutado el programa, se debe seleccionar la interfaz donde se encuentran los dispositivos IoT conectados, en la figura 46 se muestra las interfaces disponibles en el gateway:

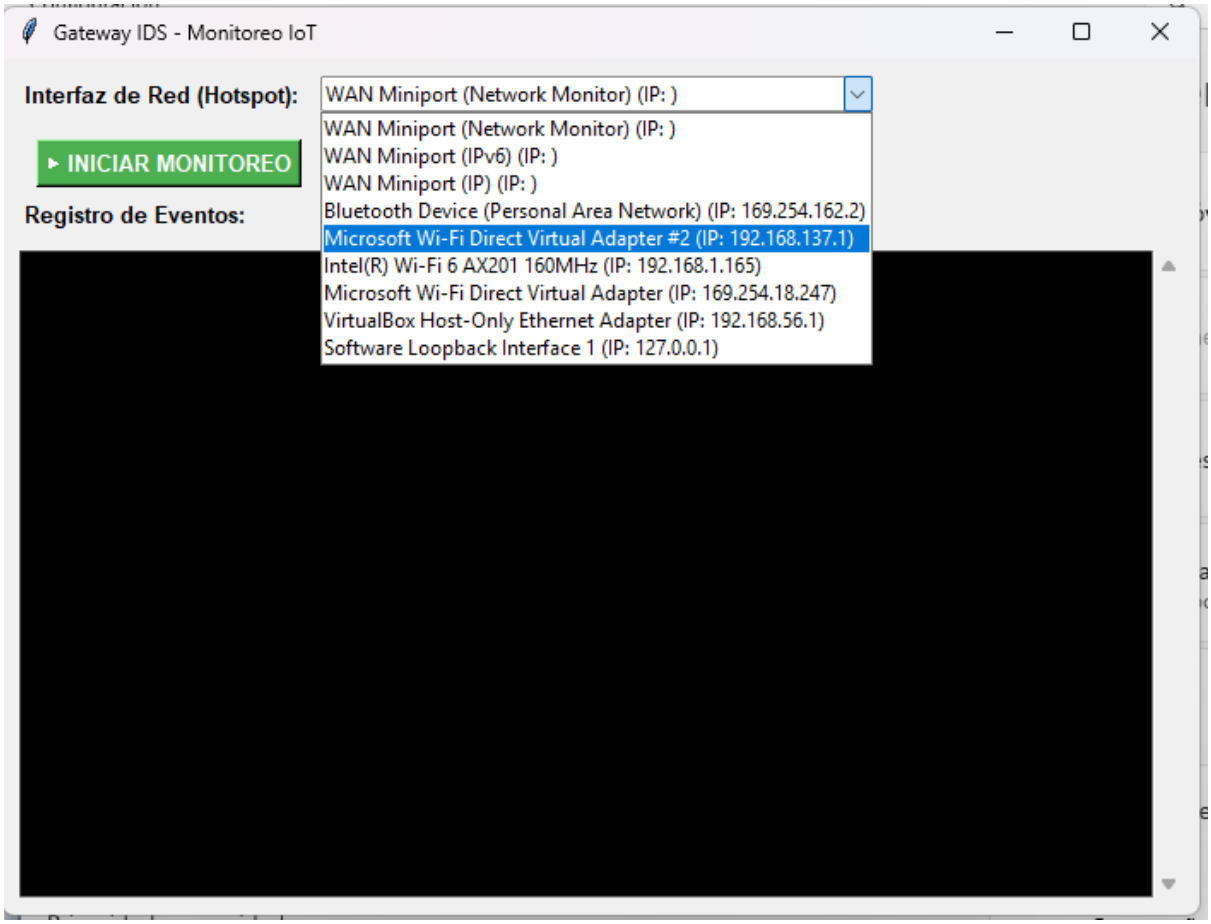

Figura 40. Interfaces disponibles para monitoreo (Elaboración propia)

Al seleccionar la interfaz e iniciar el monitoreo se confirma que se inician los servicios locales necesarios, se puede observar además en la figura 47 que el programa realiza una serie de chequeos para validar que las conexiones se establezcan correctamente:

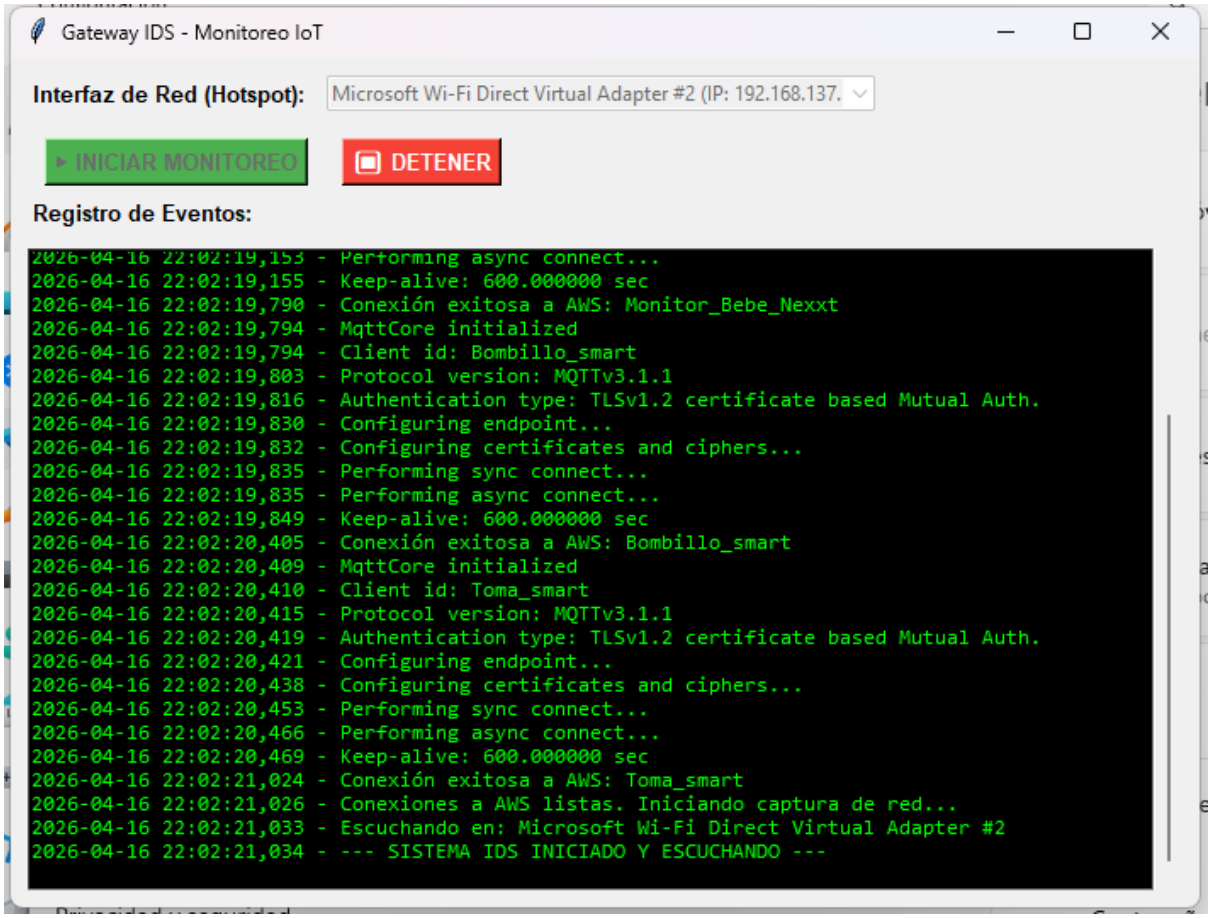

Figura 41. Sistema de monitoreo operativo (Elaboración propia)

5.4.2 Entorno de Pruebas y Herramientas

La estación para simular ataques es una máquina virtual creada en VirtualBox, en la cual se instaló previamente una imagen de Kali Linux, está a su vez se agregó al segmento de red donde se encuentran los dispositivos IoT. Para las pruebas se utilizarán las herramientas de nmap y nping, las cuales permitirán simular ataques sobre los dispositivos en prueba.

5.4.3 Escenario 1: Prueba de escaneo ARP

Se procede a ejecutar el comando sudo nmap -sn -PR 192.168.137.0/24 -e eth0, forzando la generación masiva de paquetes ARP. En la figura 44 podemos observar la salida del comando una vez ejecutado.

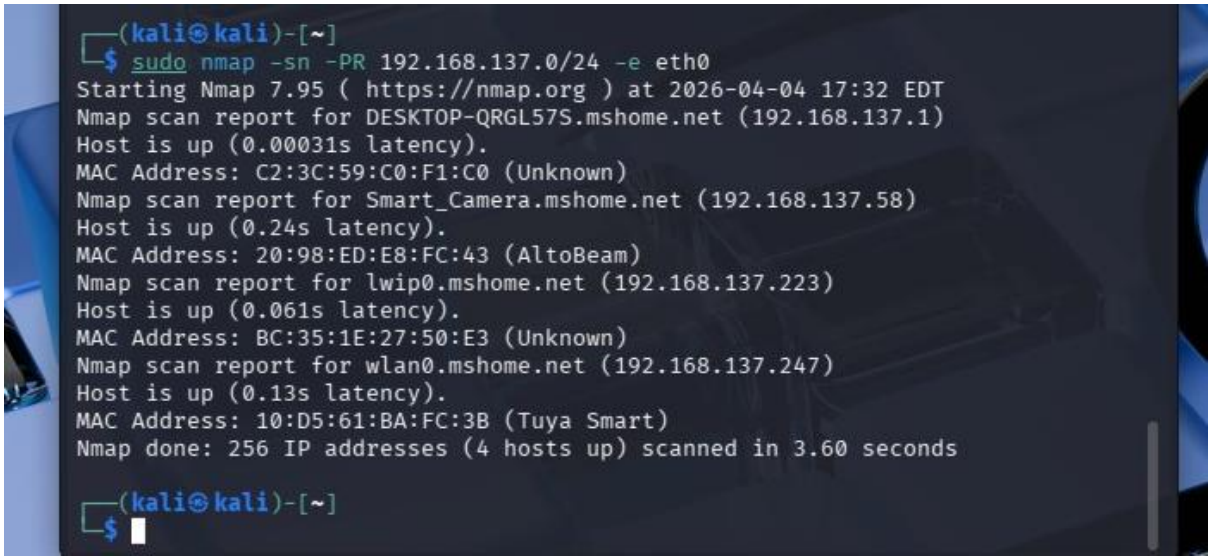

Figura 42. Prueba de escaneo ARP desde Kali Linux (Elaboración propia)

Este evento es enviado desde el gateway hasta AWS, donde el valor ARP_SCAN_SUSPECT supera el umbral y dispara una alerta, la cual es enviada por correo al usuario. En la figura 45 vemos la manera en la que se recibe el correo con el diagnóstico encontrado.

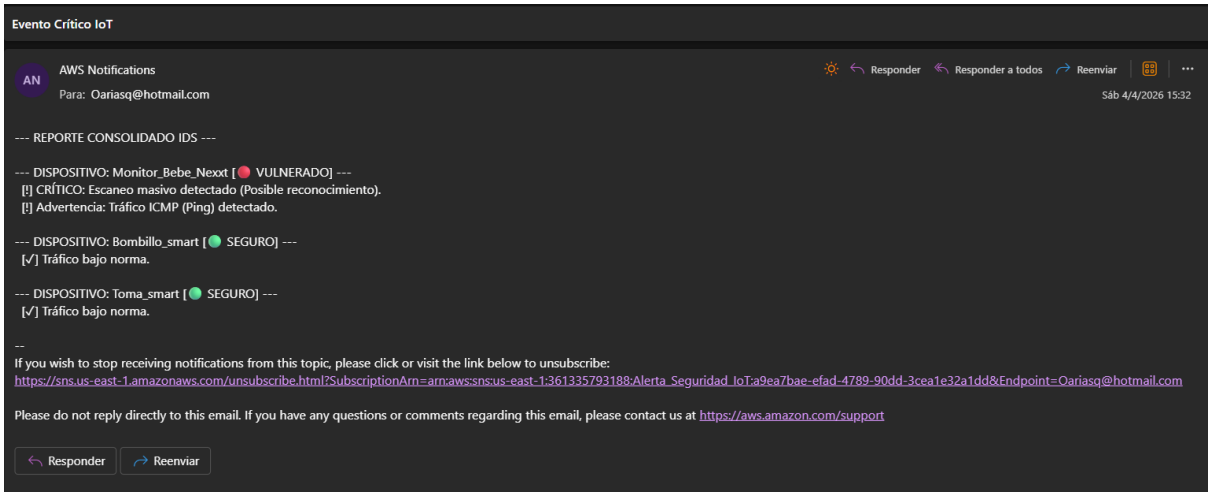

Figura 43. Correo recibido con la alerta por escaneo ARP (Elaboración propia)

5.4.4 Escenario 2: Escaneo de puertos críticos

En esta prueba se utilizará el comando nmap -sS -p 22 192.168.137.0/24, generando un escaneo masivo al puerto 22 (SSH). En la figura 46 se muestra la respuesta del comando.

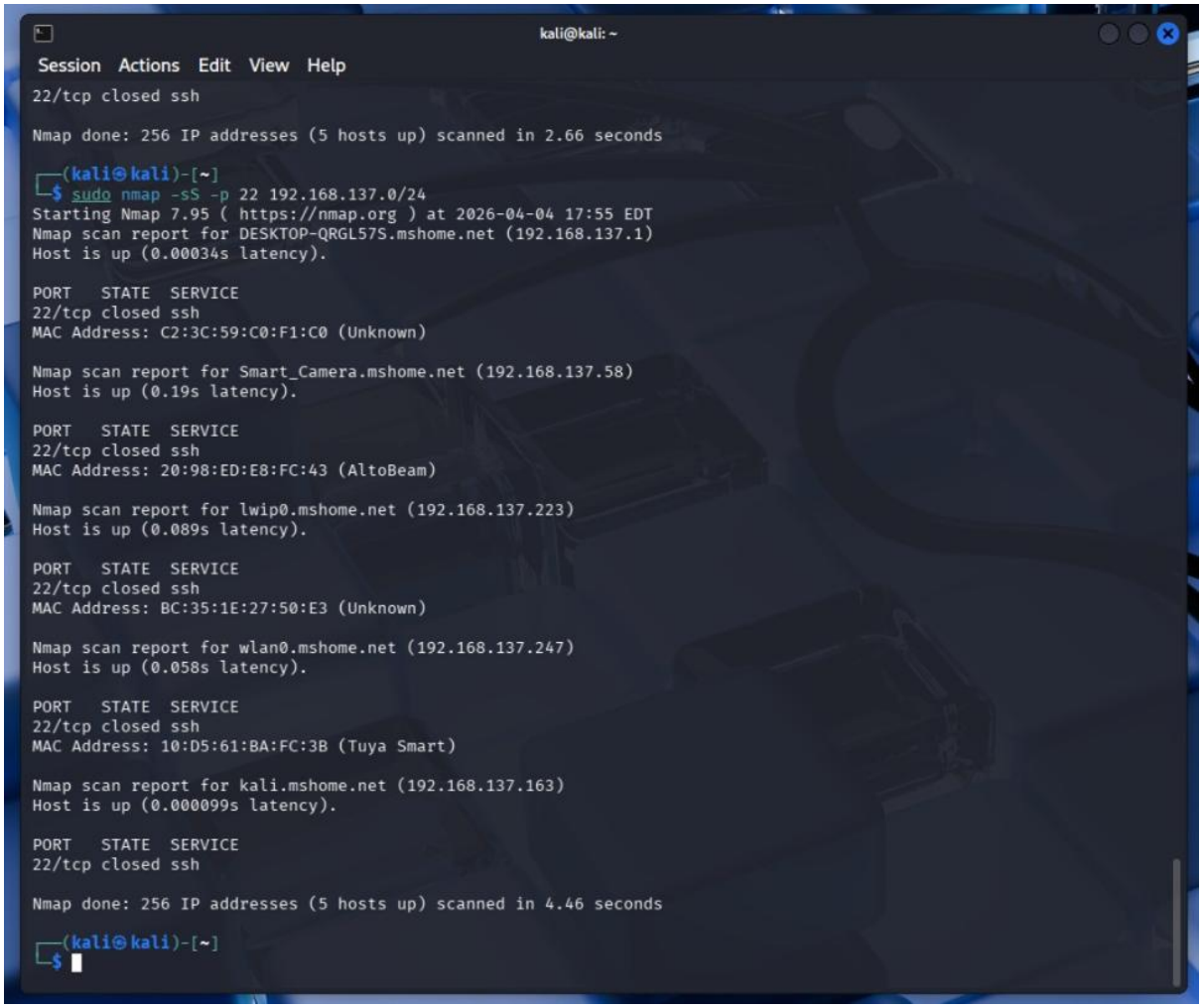

Figura 44. Prueba de escaneo de puertos desde Kali Linux (Elaboración propia)

En este caso, el puerto 22 no es parte de la lista PUERTOS_PERMITIDOS_REF, por lo que procederá a generar una alerta de seguridad, esta alerta la se muestra a detalle en la figura 47.

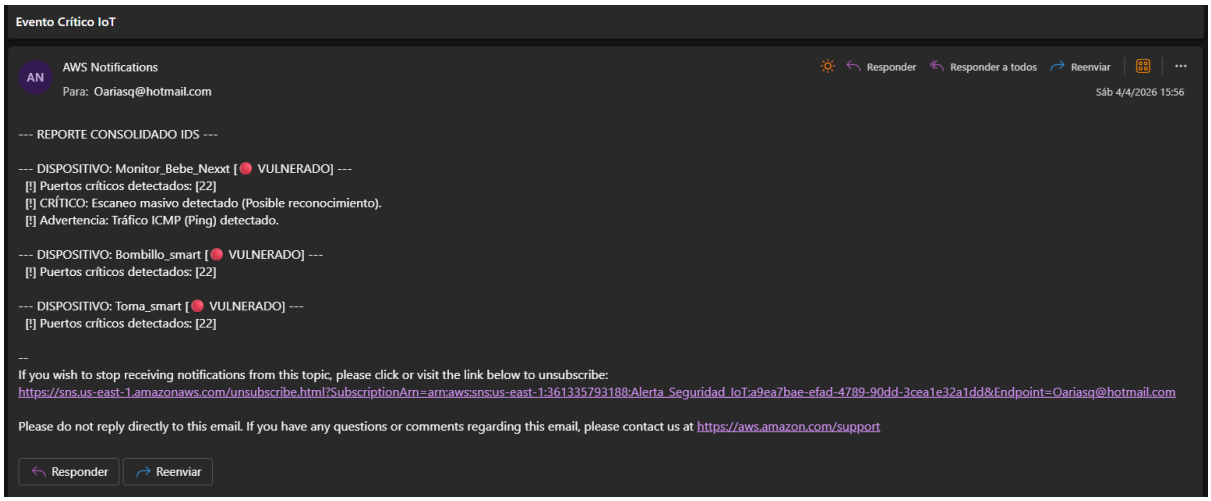

Figura 45. Correo recibido con la alerta por escaneo de puertos críticos (Elaboración propia)

5.4.5 Escenario 3: Conexión desde redes no reconocidas

Para validar este escenario, se realizó una conexión a la aplicación Nexxt mediante un dispositivo móvil conectado a la red del ICE, cuya dirección IP no se encuentra registrada en la lista de acceso permitido (whitelist). En la Figura 48, se detalla la dirección IP asignada al dispositivo utilizado para la prueba.

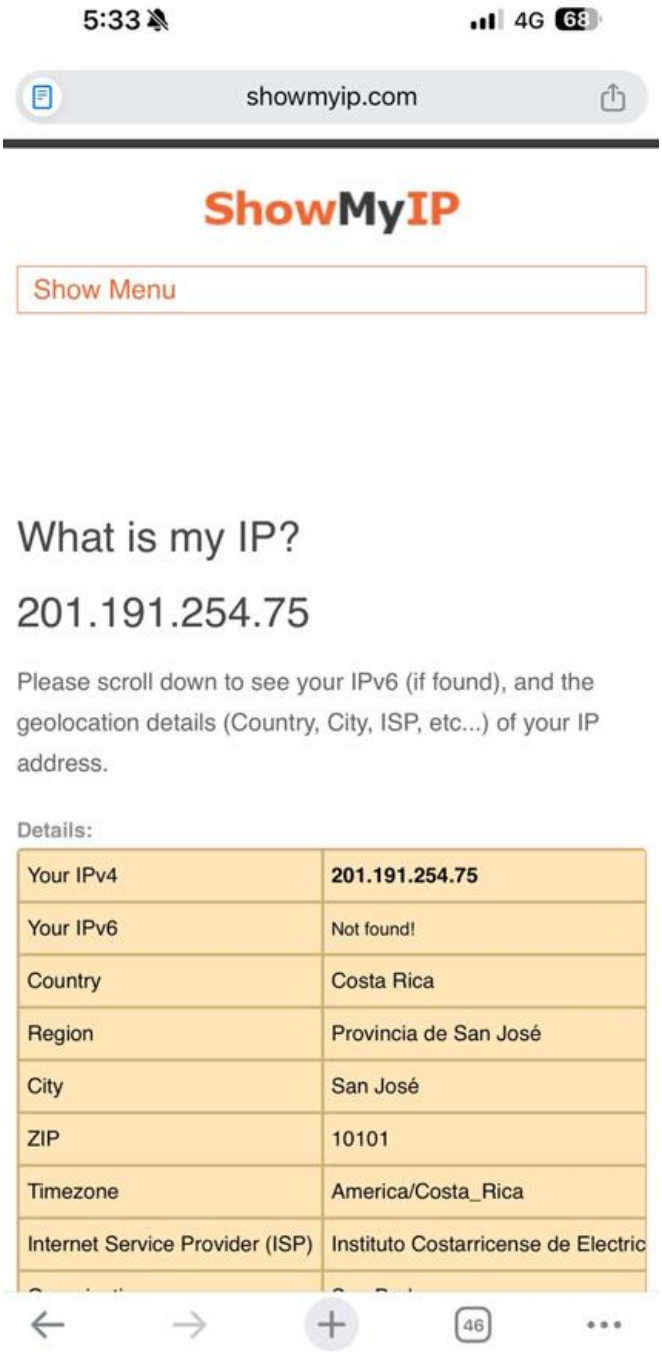

Figura 46. Dirección IP del celular de pruebas (Elaboración propia)

Al realizar la conexión desde este dispositivo, se puede visualizar que se dispara una alerta por dirección IP no autorizada, lo cual se detalla en la figura 49.

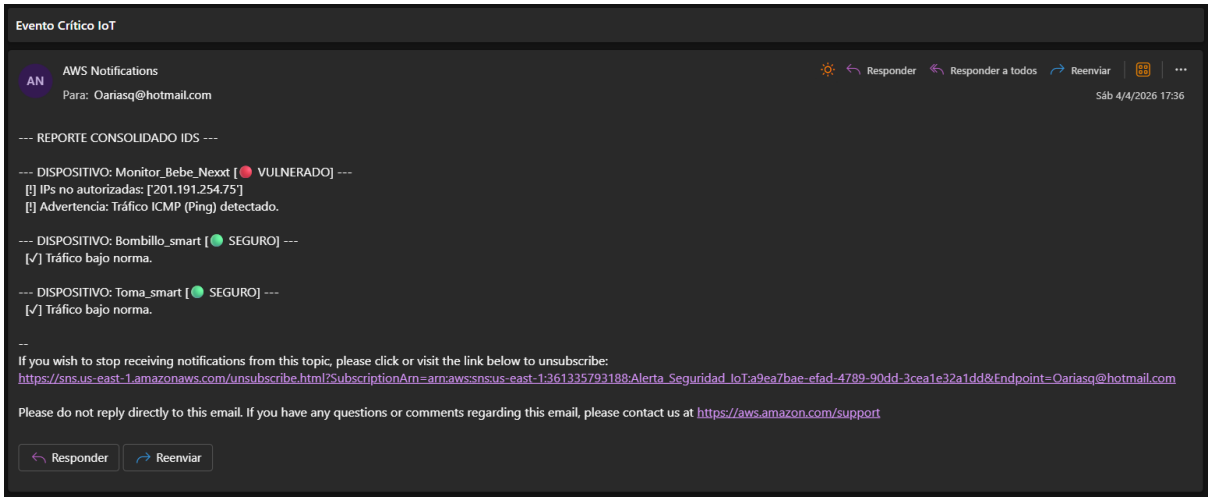

Figura 47. Correo recibido con la alerta por dirección IP no autorizada (Elaboración propia)

Capítulo 6. Conclusiones y Recomendaciones

El análisis de los resultados obtenidos permite validar la arquitectura propuesta como una solución integral para los desafíos de seguridad en entornos IoT domésticos. A continuación, se sintetizan los hallazgos principales en términos de eficiencia operativa y capacidad de respuesta ante amenazas, demostrando la viabilidad técnica y económica del proyecto.

6.1 Conclusiones sobre la Infraestructura y Detección

- El uso de un gateway local basado en Scapy mitiga las limitaciones técnicas de los enrutadores domésticos para realizar inspección granular. La transformación de trazas PCAP a JSON en el origen permite que solo la información relevante (metadatos) sea transmitida, garantizando la eficiencia del ancho de banda residencial.
- Consolidación de Datos (3-a-1): La arquitectura de Edge Computing implementada valida un ahorro significativo en costos operativos. Al consolidar las métricas de tres dispositivos en un único mensaje resumen hacia AWS IoT Core, se logra una reducción del 66% en los costos de ingestión, haciendo el sistema financieramente sostenible para un despliegue masivo.

6.2 Conclusiones sobre la Postura de Defensa Proactiva

- El flujo de remediación mediante AWS Lambda eleva el sistema de un estado de monitoreo pasivo a una defensa activa. La capacidad de automatizar el

aislamiento de dispositivos sospechosos en grupos de cuarentena dentro de la consola de AWS evita el movimiento lateral de amenazas hacia activos críticos de la red (computadoras personales o servidores de archivos).

- La integración con Amazon CloudWatch facilita la gestión de ciberseguridad residencial. Al traducir eventos técnicos complejos en paneles visuales intuitivos, se empodera al residente para tomar decisiones informadas sobre la salud digital de su hogar.

6.3 Recomendaciones para la Implementación Técnica

- Segmentación de Red: Se sugiere la implementación de redes virtuales (VLAN) o redes de invitados para lograr un aislamiento lógico y físico de los dispositivos IoT. En esta arquitectura, el gateway local debe configurarse como el único nodo con capacidad de enrutamiento y visibilidad inter-segmento.
- Fortalecimiento de la Identidad Digital: Como evolución crítica de seguridad, se propone transicionar de un certificado global para el gateway hacia el uso de certificados X.509 individuales por cada dispositivo. Esta medida asegura el principio de identidad única y garantiza que el compromiso de una credencial no comprometa la integridad de la red doméstica completa.

6.4 Recomendaciones para la Gestión Operativa y Usuario

- Protocolo de Remediación: Ante la detección de una alerta de severidad alta, se recomienda realizar una inspección física del dispositivo previo a la reversión de las medidas de aislamiento. Asimismo, es imperativo establecer un ciclo de retroalimentación en la consola de AWS (clasificación de falsos positivos) para optimizar la precisión de los algoritmos de detección..
- Mantenimiento Preventivo: Es fundamental asegurar la actualización constante del firmware en los dispositivos finales. Debe entenderse que el sistema propuesto actúa como una capa de detección perimetral, pero la corrección de vulnerabilidades de origen reside en los parches suministrados por los fabricantes.

6.5 Recomendaciones para Investigaciones Futuras

- Inteligencia de Amenazas: Se plantea la integración de fuentes externas de Threat Intelligence para la identificación en tiempo real de conexiones hacia infraestructuras de botnets conocidas.

- Interfaz de Usuario Avanzada: Resulta de interés académico explorar el uso de Amazon Bedrock (IA Generativa) para transformar eventos técnicos en reportes de lenguaje natural, facilitando la toma de decisiones para usuarios no técnicos.
- Evaluar el despliegue de AWS IoT Greengrass para mover la detección de anomalías localmente, permitiendo que el sistema responda incluso ante una desconexión total de Internet.

Bibliografía

Amazon Web Services. (2021, May 24). *How to import AWS IoT Device Defender audit findings into Security Hub*. Amazon Web Services. https://aws.amazon.com/es/blogs/security/how-to-import-aws-iot-device-defender-audit-findings-into-security-hub/

AWS. (2026). *AWS: Pricing*. Amazon Web Services. https://aws.amazon.com/es/pricing/

Bitdefender. (2025, octubre 21). *The 2025 IOT security landscape report*. Bitdefender. https://blogapp.bitdefender.com/hotforsecurity/content/files/2025/10/2025_iot_security_report.pdf

Cloudflare. (2024). *¿Qué es el UDP?* Cloudflare. https://www.cloudflare.com/es-es/learning/ddos/glossary/user-datagram-protocol-udp/

Cloudflare. (2024). *¿Qué es TCP/IP? Obtenido de ¿Qué son IP y TCP?* Cloudflare. https://www.cloudflare.com/es-es/learning/ddos/glossary/tcp-ip/

Cloudflare. (2024). *¿Qué es TLS (Transport Layer Security)?* Cloudflare. https://www.cloudflare.com/es-es/learning/ssl/transport-layer-security-tls/

Crudu, A. (2025, marzo 20). *The Dangers of Insecure HTTP - Key Risks and Effective Mitigations*. MoldStrud. https://moldstud.com/articles/p-the-dangers-of-insecure-http-key-risks-and-effective-mitigations

Duò, M. (2025, octubre 1). *¿Qué es el ARP? Cómo Funciona el Protocolo de Resolución de Direcciones*. Kinsta. https://kinsta.com/es/blog/que-es-arp/

ENISA. (2023, July 12). *Cybersecurity of Smart Home IoT: Challenges and good practices. European Union Agency for Cybersecurity.* ENISA. https://www.enisa.europa.eu/publications/cybersecurity-of-smart-home-iot

Flórez Gutiérrez, A. (2022, enero 15). *Vulnerabilidad de la información en los dispositivos domésticos inteligentes del hogar*. Institución Universitaria Politécnico Grancolombiano. https://dialnet.unirioja.es/descarga/articulo/8821184.pdf

Gálvez Cisneros, X. A. (2025, febrero 5). *Ciberseguridad en los dispositivos IOT de uso doméstico. Una revisión sistemática de la literatura*. Pontificia Universidad Católica del Ecuador. https://repositorio.puce.edu.ec/items/f0baf5ef-12dc-42d7-b1db-c216cbac1d35

Girish, A. (2024, octubre 24). *In the Room Where It Happens: Characterizing Local Communication and Threats in Smart Homes*. IMDEA. https://dspace.networks.imdea.org/handle/20.500.12761/1746

INEC. (2025, mayo 16). *85,4% de las viviendas del país tienen acceso a internet*. INEC. https://inec.cr/noticias/854-las-viviendas-del-pais-tienen-acceso-internet

Khurana, V. (2024, mayo 14). *¿Cómo capturar y analizar el tráfico de red con tcpdump?* Geekflare. https://geekflare.com/es/tcpdump-examples/

LabEx. (n.d.). *Explora y Personaliza la Interfaz de Wireshark*. LabEx. https://labex.io/es/tutorials/wireshark-explore-and-customize-wireshark-interface-415949

Madrigal, L. (2025, agosto 10). *El cifrado Web (SSL/TLS)*. https://revista.seguridad.unam.mx/numero-10/el-cifrado-web-ssltls. https://revista.seguridad.unam.mx/numero-10/el-cifrado-web-ssltls

OKTA. (2024, agosto 29). *ARP Poisoning: Definition, Techniques, Defense & Prevention*. OKTA. https://www.okta.com/identity-101/arp-poisoning/

Oluleke Akinpelu, Z. (2024, julio 10). *The Looming Threat of Unsecured IoT Devices: A Deep Dive*. ISACA. https://www.isaca.org/resources/news-and-trends/isaca-now-blog/2024/the-looming-threat-of-unsecured-iot-devices#:~:text=Weak%20authentication:%20Many%20IoT%20devices%20suffer%20from,security%2C%20leaving%20devices%20susceptible%20to%20unauthorized%20acces

TrinitySec. (2024, octubre 9). *IoT y Ciberseguridad: Desafíos y Estrategias para un Entorno Conectado*. TrinitySec. https://trinitysec.mx/iot-ciberseguridad-estrategias/

Vardakis, G. (2024, agosto 22). *Review of Smart-Home Security Using the Internet of Things*. MPDI. https://www.mdpi.com/2079-9292/13/16/3343