Seguridad de aplicaciones IoT para móviles

Análisis de la seguridad de los datos que comparten las aplicaciones móviles que controlan dispositivos inteligentes

Paola Beltrán Vázquez

Seguridad en la internet of things

Tutor del TFM
Carlos Hernández Gañán
Profesora responsable de la asignatura
Helena Rifà Pous

Enero-2023

Universitat Oberta de Catalunya

Ficha del Trabajo Final

Título del trabajo:	Seguridad de aplicaciones IoT para móviles
Nombre del autor/a:	Paola Beltrán Vázquez
Nombre del Tutor/a de TF:	Carlos Hernández Gañan
Nombre del/de la PRA:	Helena Rifà Pous
Fecha de entrega:	ENERO/2023
Titulación o programa:	Máster Universitario en Ciberseguridad y Privacidad
Área del Trabajo Final:	Seguridad en la internet of things
Idioma del trabajo:	Español
Palabras clave	IoT, seguridad, aplicaciones móviles

Resumen del Trabajo

Actualmente, nos encontramos en una era totalmente digitalizada en el que la tecnología y los dispositivos IoT forman parte de nuestra vida diaria; y el avance agigantado en la variedad de dispositivos tecnológicos con nuevas y mejores prestaciones para ayudar en el hogar de las personas y optimizar las tareas. Desde esta perspectiva, se hace necesario poder investigar y analizar los riesgos que estos implican en la privacidad de los usuarios al emplear estos dispositivos.

Por lo tanto, el presente trabajo se centra en el análisis del tráfico de red que generan los asistentes inteligentes al usar de forma continua estos; además, analizar si existe filtración de información sensible del usuario o si estos datos se envían en claro exponiendo la privacidad de este. De igual manera, estos dispositivos inteligentes disponen de aplicaciones propias para su uso, en el que es importante realizar un escaneo de las vulnerabilidades presentes en las aplicaciones móviles para el sistema operativo Android y posteriormente analizar estos datos obtenidos y verificar los riesgos que representan para el usuario referente a su privacidad; de tal manera que el usuario sea consiente al momento de emplear estos dispositivos y aplicaciones en su entorno. Además, es importante tener en cuenta como usuarios que al estar conectados en internet nuestra información es recopilada y que siempre al navegar se dejan huellas digitales acerca de nuestras preferencias y usos; por lo tanto, es importante conocer sobre estos riesgos y tratar de mitigar la exposición de datos que proporcionamos.

Abstract

Currently, we are in a fully digitized era in which technology and IoT devices are part of our daily lives; and the huge progress in the variety of technological devices with new and better features to help people at home and optimize tasks. From this perspective, it becomes necessary to be able to investigate and analyze the risks that these imply in the privacy of users when using these devices.

Therefore, this work is focused on the analysis of the network traffic generated by intelligent assistants when using them continuously; in addition, to analyze if there is leakage of sensitive user information or if these data are sent in clear exposing the user's privacy. In the same way, these smart devices have their own applications for use, in which it is important to perform a scan of the vulnerabilities present in mobile applications for the Android operating system and then analyze the data obtained and verify the risks they represent for the user regarding their privacy; so that the user is aware when using these devices and applications in their environment. Moreover, it is important to keep in mind as users that when we are connected to the Internet our information is collected and that always when browsing we leave digital footprints about our preferences and uses; therefore, it is important to know about these risks and try to mitigate the exposure of the data we provide.

Contenido

Lista de Figuras

Lista de Tablas

Siglas y Acrónimos

A

APIS
Application Programming Interfaces, 23
APK
Android Application Package, 34, 42
AVS
Alexa Voice Server, 24
AWS
Amazon Web Services, 8, 33, 34, 48

C

CALO
Cognitive Assistant that Learns and Organizes, 1
CMD
Command Prompt, 22

D

DARPA
Defense Advanced Research Projects Agency, 1

H

HTTP
Hypertext Transfer Protocol, 23, 39, 40

I

IBM
International Business Machines, 1, 44
IBSG
Internet Business Solutions Group, 13
IoT
Internet Of Things, 1, 4, 5, 6, 9, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 41, 45, 46, 47, 63
IP
Internet Protocol, 6, 21, 25, 26, 30
ITU
Unión Internacional de Telecomunicaciones, 13

J

JSON
JavaScript Object Notation, 23, 30

M

MDNS
Multicast Domain Name System, 30, 32
MITM
Men In The Middle, 38, 40

O

ODS
Objetivo de Desarrollo Sostenible, 3, 4
OEM
Original Equipment Manufacturer, 20
OWASP
Open Web Application Security Project, 7, 22, 35, 38, 61

P

P&G
Procter & Gamble, 13
PECs
Pruebas de Evaluación Continua, 4

Q

QoS
Quality of Service, 14

R

REST
Representational State Transfer, 24

S

SDK
Software Development Kit, 2, 16
SQL
Structured Query Language, 38, 40
SRI
Stanford Research Institute International, 1
SSDP
Simple Service Discovery Protocol, 30, 31

SSL
Secure Sockets Layer, 38, 40
SSMASHE
Secured Smart Mobile App for Smart Home Environment, 19

T

TFM
Trabajo Final del Máster, 2, 1, 3, 4, 6, 7, 8, 13, 19, 21, 34, 40, 41, 42
TLS
Transport Layer Security, 38, 40

U

UDP
User Datagram Protocol, 30
UOC
Universitat Oberta de Catalunya, 3

W

WSN
Wireless Sensor Networks, 15

CAPITULO I

Introducción

En este capítulo, se aborda el contexto y justificación del TFM (Trabajo Final del Máster), los objetivos, el impacto en sostenibilidad, ético-social y de diversidad, el enfoque y la metodología a seguir en el desarrollo del trabajo final del máster, la planificación del trabajo, sumario de productos obtenidos y finalmente una descripción de la estructura del trabajo.

1.1. Contexto y justificación del trabajo

La primera máquina que permitió ejecutar el reconocimiento de voz fue Shoebox diseñado por IBM (International Business Machines) en 1961 y presentada al público en general en la Feria Mundial de Seattle de 1962 capaz de reconocer 16 palabras y dígitos del 0 al 9 [1], en 1970 en la Universidad Carnegie Mellon con el apoyo del Departamento de Defensa de los Estados Unidos y DARPA (Defense Advanced Research Projects Agency) desarrollaron la herramienta Harpy capaz de reconocer más de mil palabras y 10 años después lograron desarrollar un sistema que entienda oraciones completas [2]. Para la década de 1990 el reconocimiento de voz se convirtió en una característica de las computadoras personales [3] y en 1994 IBM fundamentó el concepto de asistentes virtuales inteligentes a través de Simon el primer teléfono inteligente [4]. El asistente inteligente tiene origen en un proyecto militar artificial llamado CALO (Cognitive Assistant that Learns and Organizes), su director Adam Cheyer junto con SRI (Stanford Research Institute International) empezaron con la creación de Siri conocido como el primer asistente inteligente [5].

Con el avance de la tecnología y el auge de disponer en los hogares dispositivos inteligentes en el que una persona habla y el asistente virtual lo procesa, interpreta y responde [6] y que puedan ser controlados fácilmente por una aplicación móvil se ha hecho cada vez más habitual [7]. En ese contexto, de acuerdo con un estudio publicado en Global Smart Speaker Market [8] el mercado de estos dispositivos crece exponencialmente y se indica que crecerá anualmente un 26% hasta el año 2025 como se puede ver en la Figura 1.

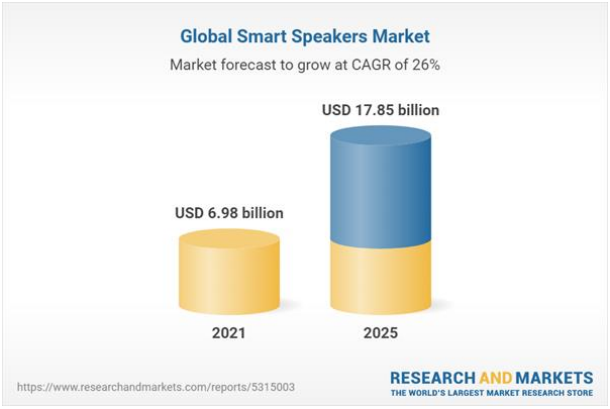


Figura 1. Pronóstico del mercado. Fuente: Imagen tomada de [8]

Además, según [9] se prevé que el número de asistentes inteligentes en uso a nivel mundial para el 2024 alcance los 8400 millones de asistentes, es decir el doble de lo que había en el 2019 como se presenta en la Figura 2. Por lo tanto, [10] predice que la tasa de aceptación de estos será más rápida que la televisión, internet y los teléfonos inteligentes.

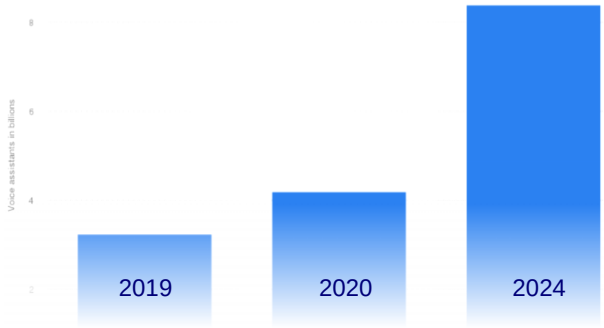


Figura 2. Número de asistentes inteligentes. Fuente: Statista [9]

Las aplicaciones móviles que empleamos en la vida diaria recolectan la información del usuario acerca de la actividad, de igual manera al instalar y para poder usar muchas aplicaciones requieren que se realice un registro proporcionando los datos del usuario. Esta información se categoriza para analítica de las empresas que procesan la información. Los datos que recolectan las aplicaciones móviles no se envían inmediatamente a la herramienta de analítica ya que los dispositivos pueden no tener conexión a red y por lo tanto el SDK[1] (Software Development Kit) se encarga de recopilar los datos de la aplicación, lo que el usuario ve, el sistema operativo del móvil, tiempo de uso de la aplicación instalada, entre otra información [7].

En este contexto, en el presente trabajo se analizarán las aplicaciones móviles para controlar los dispositivos inteligentes; entre los más populares se tienen: Asistente Siri que fue lanzado el 14 de octubre de 2011 desarrollado por Apple y su dispositivo el HomePod, Asistente Cortana que fue lanzado el 2 de abril de 2014 desarrollado por Microsoft y su dispositivo Invoke, Asistente Alexa que fue lanzado en noviembre de 2014

[1] Conjunto de herramientas software que sirven para crear aplicaciones mediante un compilador, depurador o framework.

desarrollado por Amazon y su dispositivo Amazon Echo y el Asistente Google Assistant que fue lanzado el 18 de mayo de 2016 desarrollado por Google y su dispositivo Google Home entre los más populares [11].

Por lo tanto, se plantea el análisis del tráfico de información enviada por los dispositivos inteligentes identificando los puntos clave que afecten a la privacidad de los usuarios. Además, se requiere obtener un proceso para poder minimizar la exposición de su privacidad. Finalmente, se presentan las conclusiones obtenidas del proceso realizado y se consideran para futuros trabajos a ser cubiertos.

1.2. Objetivos del Trabajo

El objetivo principal del presente trabajo se centra en el estudio y análisis de los dispositivos IoT (Internet Of Things) que son empleados como asistentes virtuales en diferentes entornos cotidianos con el objetivo de identificar las vulnerabilidades de seguridad presentes en los mismos y como afectan estos a la privacidad de las personas. Para ello se plantea los siguientes objetivos específicos:

- Identificar los conceptos procedentes de los dispositivos IoT y sus aplicaciones móviles que permiten controlar los mismos.
- Diseñar y configurar un ambiente de análisis de tráfico de la información que nos permita recolectar los datos procedentes de estos dispositivos IoT
- Identificar los permisos y accesos concedidos a estas aplicaciones
- Detallar los datos obtenidos, identificando como pueden afectar la privacidad de los usuarios
- Describir un conjunto de buenas prácticas que el usuario debe tener en cuenta al hacer uso de estos dispositivos con el objetivo de mitigar la exposición de su privacidad.
- Exponer las conclusiones del presente trabajo; así como un conjunto de trabajos futuros que extiendan el mismo.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

A continuación, se identifica los impactos positivos y/o negativos del presente trabajo final de máster en las tres dimensiones de la competencia transversal UOC “Compromiso ético y global”.

- Dimensión de sostenibilidad: el presente TFM no tiene ningún impacto referente en aspectos de sostenibilidad ya que no representa ni está alineado con ningún ODS (Objetivo de Desarrollo Sostenible).
- Dimensión de comportamiento ético y de responsabilidad social: uno de los objetivos del TFM es analizar la seguridad que proporcionan las aplicaciones y verificar que tipo de información se comparten sin consentimiento del usuario lo cual implica una violación a la privacidad y este se considera un

impacto negativo ya que los propietarios de estas aplicaciones lucran con la información personal del usuario final, por lo tanto, el objetivo es obtener la información necesaria y saber que aplicación es la más indicada para el uso del usuario final sin que se vea afectada su propagación de información personal sin consentimiento.

- Dimensión de diversidad, género y derechos humanos: no tiene ningún impacto referente en aspectos de sostenibilidad ya que no representa ni esta alineado con ningún ODS.

1.4. Enfoque y método seguido

La metodología aplicada es de tipo cuantitativo planteada por [12], la cual será adaptada para el presente trabajo ya que no se aplicarán todas las fases por la extensión de esta y en la cual se definen las siguientes fases:

1.4.1. **Idea**: se generan las ideas de investigación, el titulo y que se quiere alcanzar.

1.4.2. **Planteamiento del problema**: se plantea la problemática de origen de la investigación, el contexto y justificación, los objetivos y las tareas necesarias, impacto en sostenibilidad, ético-social y de diversidad, planificación del trabajo.

1.4.3. **Desarrollo de la base tecnológica y revisión del estado del arte**: se requiere realizar una búsqueda avanzada de la literatura con el fin de realizar un TFM autocontenido.

1.4.4. **Visualización del alcance del estudio**: se requiere diseñar y configurar el ambiente para analizar el tráfico de los dispositivos inteligentes y el escaneo de las vulnerabilidades en las aplicaciones móviles.

1.4.5. **Recolección de los datos**: al completar la fase anterior se requiere seguir un proceso para recolectar la información procedente de los dispositivos IoT.

1.4.6. **Análisis de los datos**: se analizan los datos obtenidos identificando los puntos clave que afectan la privacidad.

1.4.7. **Elaboración del reporte de resultados**: se procesa la información obtenida y se elaboran los reportes para presentar los resultados y de igual manera el proceso para mitigar la exposición de la privacidad de los usuarios.

1.5. Planificación del trabajo

Con la finalidad de cumplir las tareas se crea un cronograma en un diagrama de Gantt como se presenta en la Figura 3 donde se agregan todas las tareas a realizar para cada entrega de las PECs (Pruebas de Evaluación Continua) se presenta el desglose de cada una con la estimación de los tiempos empleados para cada tarea a desarrollar.

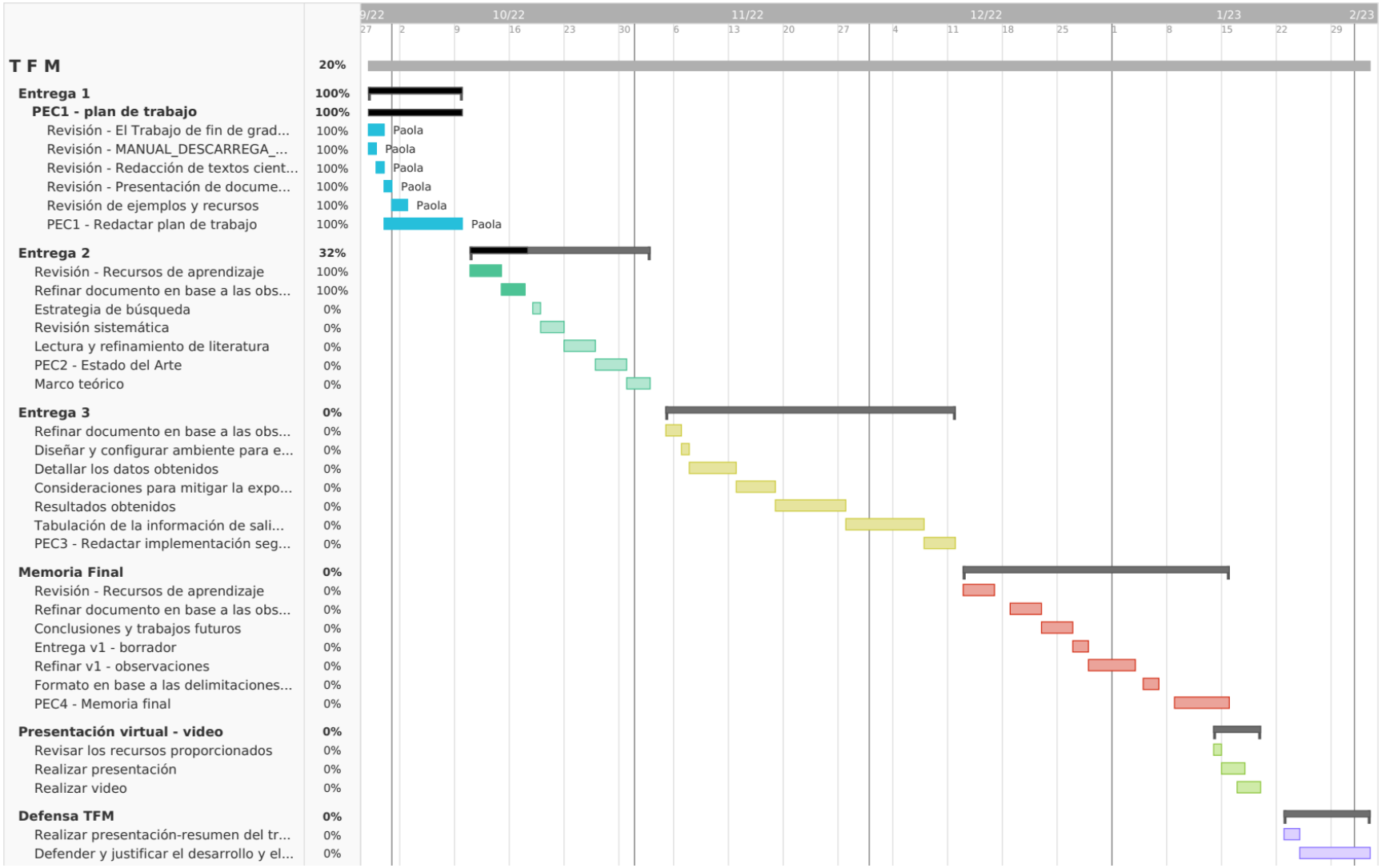


Figura 3. Cronograma de planificación. Fuente: Elaboración propia

1.6. Breve sumario de productos obtenidos

Para el desarrollo total del TFM se deben completar las pautas indicadas en las pruebas de evaluación continua las cuales se dividen de la siguiente manera:

- PEC1: Se requiere realizar el plan de trabajo, se describe el problema que se pretende resolver.
- PEC2: Depende de la definición realizada en el entregable anterior PEC1, se realiza el proceso de investigación para resolver la problemática planteada; es decir, el desarrollo del estado del arte.
- PEC3: Se describe la implementación para solventar el problema planteado y obtener los resultados para su interpretación garantizando el cumplimiento de las especificaciones descritas como es el flujo del diseño y configuración del ambiente para obtener los datos y puntos clave que afectan la privacidad de los usuarios y describir el flujo de buenas prácticas para mitigar su exposición.
- Memoria Final: Se mejora todo el trabajo que se ha venido realizando en las etapas anteriores con el fin de sintetizar el trabajo realizado y validar que se han alcanzado los objetivos propuestos y presentar las conclusiones y trabajos futuros.
- Presentación virtual: Se procede a realizar un video presentando la síntesis del trabajo donde se presente el proceso, desarrollo y resultados obtenidos del TFM.
- Defensa del TFM: Se prepara como etapa final el proceso de defender y justificar el desarrollo y resultado del trabajo realizado ante un comité de evaluación presentando los conocimientos adquiridos.

1.7. Breve descripción de otros capítulos de la memoria

A continuación, se describe de manera general cada uno de los capítulos a desarrollarse:

- **Capítulo II: Estado del Arte**
  Se plantea un método de búsqueda de estudios que analicen las seguridades y vulnerabilidades existentes en las aplicaciones móviles para controlar dispositivos inteligentes, de igual manera se verifican que datos comparten estas aplicaciones con y sin consentimiento del usuario que pueden llegar a afectar la privacidad de los usuarios.
- **Capítulo III: Marco teórico**
  Se plantean conceptos clave para la elaboración del TFM como definir los conceptos procedentes de los dispositivos IoT y sus aplicaciones móviles para controlar estos dispositivos propios y de terceros para el sistema operativo Android y el uso en la domótica.

- **Capítulo IV: Diseño y configuración del ambiente**
  Se identifican los paquetes que salen por la red con el objetivo de poder observar y analizar el contenido que se proporciona al hacer uso de estas aplicaciones para controlar los dispositivos inteligentes; con el objetivo de recolectar los datos procedentes de los dispositivos IoT y se analizan las aplicaciones Android para verificar las vulnerabilidades.
- **Capítulo V: Proceso y resultados obtenidos**
  Se presenta los datos recopilados con el objetivo de evaluar lo que implica el uso de estas aplicaciones que controlan los dispositivos inteligentes, que riesgos se asumen al usarlos, conocer su seguridad y poder identificar que datos estamos proporcionando con y sin consentimiento y para que se emplea esta información recopilada en las aplicaciones o en otros casos identificar con que fines se está empleando la información privada del usuario hacia terceros y como mitigar la exposición de su privacidad.
- **Capítulo VI: Conclusiones y trabajos futuros**
  Se presentan los resultados que se han obtenido al realizar el proceso de TFM y se proporciona un punto de vista desde el lado investigativo y concientización al hacer uso de la tecnología en nuestra vida diaria. De igual manera se plantea las propuestas de trabajos futuros que complementen el presente trabajo.

CAPITULO II

Estado del arte

Se aborda la revisión sistemática de la literatura con el objetivo de buscar, identificar y analizar varios estudios relacionados con el TFM a realizar, con la finalidad de conocer las investigaciones que se han llevado a cabo hasta ahora y que sean la base para abordar el tema a profundidad y la investigación de supuestos aún no considerados.

2.1. Estrategia de búsqueda

Para la búsqueda de artículos relacionados se han considerado las siguientes bibliotecas virtuales para llevar a cabo la revisión.

- ProQuest[2]
- IEEE Xplore[3]
- ScienceDirect[4]

Para realizar el proceso de búsqueda y recuperar los artículos científicos relacionados se plantea la siguiente cadena de búsqueda con las palabras clave y textos alternativos empleando operadores lógicos como se presenta en la Tabla 1.

Palabra completa	Conector
Smart assistant	OR
IoT	AND
Mobile app	OR
Companion app	AND
Security	OR
Privacy	
Cadena de búsqueda	(((smart assistant OR IoT) AND (mobile app OR companion app)) OR ((smart assistant OR IoT) AND security AND privacy AND companion app))

Tabla 1. Cadena de búsqueda. Fuente: Elaboración propia

Se plantea el periodo de búsqueda desde el 2014 a la actualidad, se consideran estos años de revisión ya que desde ese año aparecieron algunos asistentes inteligentes para plataformas Android dando fuerza la integración de estos dispositivos en los hogares por su facilidad de acceso.

[2] https://www.proquest.com/
[3] https://ieeexplore.ieee.org/Xplore/home.jsp
[4] https://www.sciencedirect.com/

2.2. Selección de estudios primarios

Se realizó la búsqueda avanzada en las librerías digitales siguiendo la cadena de búsqueda propuesta en el apartado anterior; esta búsqueda se realizó sobre la metadata (resumen, título, palabras clave), revistas científicas, libros y artículos científicos y que estén en inglés como se presenta en la Tabla 2.

Campos	Descripción
Campos de búsqueda	Título, resumen y palabras clave
Rango de año	2014-2022
Fecha de búsqueda	Octubre de 2022
Idioma	Inglés
Tipo de fuente	Revistas, artículos y libros

Tabla 2. Proceso de búsqueda. Fuente: Elaboración propia

A continuación, en la Tabla 3 se presenta el resultado del total de documentos obtenidos al aplicar los criterios de búsqueda avanzada en las diferentes librerías digitales.

Librería	Cadena de búsqueda	Campos	Total
IEEE Xplore *filtrado editor IEEE	("Document Title":smart assistant) OR ("Document Title":IoT) AND ("Document Title":mobile app) OR ("Document Title":companion app) OR ("Document Title":smart assistant) OR ("Document Title":IoT) AND ("Document Title":security) AND ("Document Title":privacy) AND ("Document Title":companion app)	Título	9
	("Abstract":smart assistant) OR ("Abstract":IoT) AND ("Abstract":mobile app) OR ("Abstract":companion app) OR ("Abstract":smart assistant) OR ("Abstract":IoT) AND ("Abstract":security) AND ("Abstract":privacy) AND ("Abstract":companion app)	Resumen	94
	("Author Keywords":smart assistant) OR ("Author Keywords":IoT) AND ("Author Keywords":mobile app) OR ("Author	Palabras clave	12

	Keywords":companion app) OR ("Author Keywords":smart assistant) OR ("Author Keywords":IoT) AND ("Author Keywords":security) AND ("Author Keywords":privacy) AND ("Author Keywords":companion app)		
	TOTAL		**115**
ProQuest *se excluye duplicados en el filtro de búsqueda	title(smart assistant) OR title(IoT) AND title(mobile app) OR title(companion app) OR title(smart assistant) OR title(IoT) AND title(security) AND title(privacy) AND title(companion app)	Título	18
	abstract(smart assistant) OR abstract(IoT) AND abstract(mobile app) OR abstract(companion app) OR abstract(smart assistant) OR abstract(IoT) AND abstract(security) AND abstract(privacy) AND abstract(companion app)	Resumen	110
	TOTAL		**128**
ScienceDirect	(((smart assistant OR IoT) AND (mobile app OR companion app)) OR ((smart assistant OR IoT) AND security AND privacy AND companion app))	Título Resumen Palabras clave	44

Tabla 3. Resultados de la búsqueda. Fuente: Elaboración propia

Una vez realizada la búsqueda se obtienen varios artículos para su revisión y poder realizar un proceso de filtrado considerando si el artículo se incluye o no en base a su título, resumen y palabras clave.

Se incluyen los documentos en base a los siguientes criterios de inclusión:

- Estudios acerca de los asistentes inteligentes, seguridad y privacidad
- Estudios acerca de los asistentes inteligentes y aplicaciones móviles que se emplean para su control y
- Estudios acerca de los asistentes inteligentes y su uso en la domótica

Se excluyen los documentos en base a los siguientes criterios de exclusión:

- Estudios duplicados, similares y
- Estudios introductorios para otros temas específicos

Por lo tanto, al finalizar el proceso de filtrado a cada uno de los documentos obtenidos en la búsqueda avanzada en las diferentes librerías digitales se tiene los siguientes resultados presentados en la Tabla 4.

Librería digital	No relacionado	Duplicados	Total, Inicial	Total, Final
IEEE Xplore	41	6	115	68
ProQuest	58	10	128	60
ScienceDirect	19	0	44	25
TOTAL	**118**	**16**	**287**	**153**

Tabla 4. Resultados de los documentos. Fuente: Elaboración propia

Además, se incluye un gráfico con el porcentaje del total de documentos finales para un mejor entendimiento visual como se puede ver en la Figura 4.

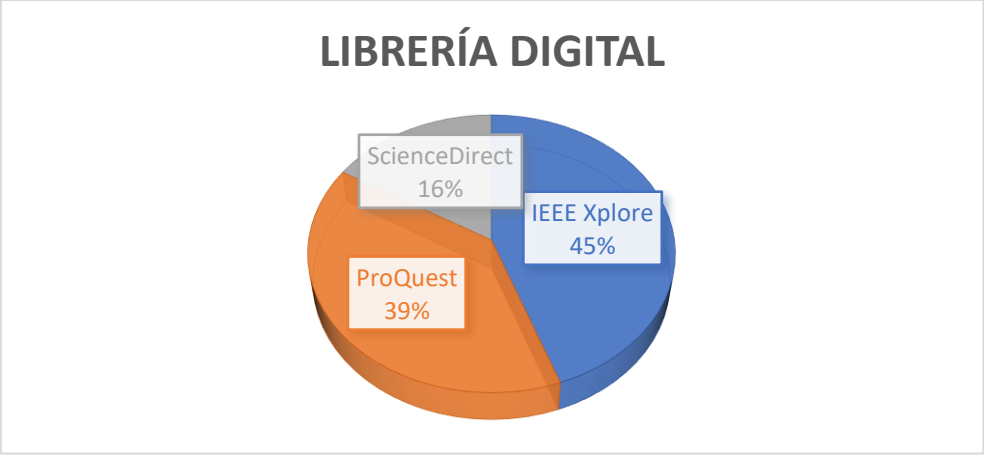


Figura 4. Porcentaje de resultados finales. Fuente: Elaboración propia

Finalmente, en esta etapa se obtuvo varios estudios que se han publicado en más de una conferencia o revista y de igual manera se ha podido verificar estudios similares, por lo tanto, se ha tomado en cuenta solo uno de los estudios considerando el documento más completo. Además, se han podido evidenciar estudios acerca del área de desarrollo, pero enfocados a otros ámbitos/paradigmas por lo cual no se han considerado como parte de nuestro estudio.

2.3. Discusión de resultados

De acuerdo con la literatura revisada se tienen resultados acerca de estudios introductorios sobre IoT, latencia y consumo de energía, seguridad, privacidad y domótica; algunos estudios sobre análisis, exploración del uso y la privacidad de altavoces virtuales inteligentes.

Al revisar los estudios obtenidos de las diferentes librerías digitales; algunos de estos indican el propósito de recopilar la información de los usuarios en el que se expone que el motivo principal de la recopilación de cierta información se debe principalmente para poder proporcionar un servicio personalizado en base a las preferencias del usuario.

En las siguientes investigaciones más relevantes [13] [14] [15] [16] [17] [18] se presentan temas acerca de IoT, conceptos base, el uso y su clasificación. En [19] [20] [21] [22] se abarcan temas relacionados a la seguridad y privacidad y en los siguientes artículos [23] [24] se presenta la integración de estos dispositivos inteligentes en los hogares inteligentes dando lugar a lo que se conoce como domótica.

Además, se puede verificar que en los estudios revisados no se abarca un análisis acerca de que datos de los usuarios que emplean asistentes virtuales inteligentes se comparten, esto se puede entender ya que el avance y el uso actual de la tecnología en los hogares ha crecido de manera significativa dejando de lado la importancia de la privacidad y seguridad. Por lo tanto, las empresas proveedoras de estos dispositivos no ven la utilidad de mitigar estas brechas de información ya que se sirven de dicha información para mejorar y personalizar su asistencia a cada uno de los usuarios. Sin embargo, es importante concientizar a los usuarios acerca de la exposición que conlleva usar estos asistentes virtuales inteligentes.

CAPITULO III

Marco Teórico

Es la base para el desarrollo y entendimiento del TFM en el que se abordan los temas clave como es la conceptualización de IoT, dispositivos IoT y aplicaciones móviles que permiten controlar sus dispositivos y de terceros para el sistema operativo Android y su uso en la domótica.

3.1. Qué es IoT

Mark Weiser en 1991 describe la visión del futuro de internet como “Ubiquitous Computing” y se focaliza en entender cómo se centra un entorno inteligente en presencia de un teléfono móvil que proporciona un poderoso sistema multimedia [13]. Kevin Ashton, es uno de los pioneros al introducir el término IoT (Internet of Things, en español internet de las cosas) en una presentación realizada en P&G (Procter & Gamble) en 1999 [25]. Posteriormente, en el 2005, IoT entró en un nuevo nivel cuando se publicó el primer informe de la ITU (Unión Internacional de Telecomunicaciones) y en 2008/2009 IoT "nace" por Cisco IBSG (Internet Business Solutions Group) [13].

IoT, describe los objetos físicos que poseen sensores y actuadores que permiten la comunicación con sistemas informáticos mediante redes inalámbricas o cableadas, de esta manera se monitorea y controla digitalmente el mundo físico [14]. Además, [13] define IoT como un conjunto de cosas u objetos inteligentes como dispositivos de hogar, móviles, computadores, entre otros direccionados por un esquema único y conectados a internet ya sea computación en la nube.

Para conseguir la conexión digital y física IoT emplea una variedad de tecnologías; los dispositivos físicos están integrados con sensores con lo cual permite monitorear cosas como temperatura, movimiento o cambios en el entorno y actuadores que son los que reciben las señales de los sensores y envían una respuesta. Por lo tanto, los sensores y actuadores se comunican a través de redes cableadas o inalámbricas con sistemas informáticos que permiten administrar y monitorizar estos objetos conectados [14].

Actualmente, si se dispone de un rastreador de actividad física, un termostato o asistentes inteligentes se forma parte del internet de las cosas ya que IoT se ha integrado en la vida cotidiana [14]. De acuerdo con [26] en su estudio realizado a un grupo de 12000 usuarios se obtiene que un 74% de los usuarios había interactuado con un asistente de voz o chat para buscar información, estado de compras y servicios; un 58% de los usuarios ha empleado para actividades recreativas como escuchar música, ver direcciones y realizar reservas y un 53% de los usuarios ha empleado para obtener información de banca y facturas.

3.1.1. Aplicaciones IoT

Según [13] [15], las aplicaciones IoT se clasifican en cuatro categorías:

- Personal y hogar, está enfocada a nivel del usuario u hogar y la información recopilada del sensor es utilizada por las personas propietarias de la red. Su conectividad a través de Wifi permite una mayor transferencia de datos y tasas de muestreo [15] .
- Empresa, está enfocada a nivel de comunidad y la información recopilada es utilizada por los propietarios y los datos se procesan selectivamente [15].
- Utilidades, está enfocada a nivel nacional o regional y la información es utilizada para la optimización de los servicios para la gestión de recursos con el fin de optimizar la relación costo beneficio [15].
- Móvil, se extiende a otros dominios debido a la naturaleza de la conectividad y escala que permite el uso de sensor de redes inalámbricas para monitorear los tiempos de viaje, comportamiento y elección de ruta de origen y destino [15].

A continuación, en la Tabla 5 se puede ver sus servicios, tipo de conectividad y los retos o desafíos de cada categoría. Para conocer más a detalle sobre la conectividad y retos expuestos en la Tabla 5 se agrega la información en el Anexo 1.

Categorías	Servicios	Conectividad	Retos
Personal/hogar	Cuidado de la salud	Wifi, 3G, 4GLTE	Heterogeneidad Interoperabili-dad Métodos de control Tiempo real Seguridad QoS (Quality of Service)
Empresa	Ciudades inteligen-tes	Wifi, 3G, 4GLTE, Satélite	Escalabilidad Identificación y descubrimiento Heterogeneidad Virtualización Métodos de control Tiempo real Big data Consumo de energía Seguridad
	Entornos inteligen-tes		
	Video vigilancia	Wifi, Satélite	

Utilidades	Red inteligente Energía inteligente Agua inte-ligente	Wifi, Satélite y Celular	Consumo de energía Balance de carga Tiempo real Escalabilidad Métodos de control Costo
Móvil	Transporte/Tráfico inteligente	WSN (Wireless Sensor Net-works), Satélite	Identificación y descubrimiento Métodos de control Escalabilidad WSN Arquitectura y diseño IoT Tiempo real Costo

Tabla 5. Categorías de las aplicaciones IoT. Fuente: Tabla tomada de [13]

Sin embargo, la cantidad de aplicaciones IoT es más amplia y se utilizan en innumerables entornos que ayudan en la mejora de los procesos ya sea en el ámbito de uso personal, negocios, salud, movilidad, etc. [14]

3.2. Dispositivos IoT

Es una pieza de hardware que a través de un sensor transmite información de un sitio a otro mediante el uso de una red inalámbrica o cableada [14]. Los dispositivos IoT se limitaban a simples comandos para completar algunas tareas como encender, apagar, recordatorios, alarmas, entre otras; en la actualidad estos dispositivos son capaces de controlar una variedad de productos inteligentes [27]. El último estudio de Edison Research “The smart audio report” obtiene que alrededor de 43 millones de personas en Estados Unidos poseen un asistente de voz [10].

Los productos de IoT se clasifican por segmentos de mercado: bienes de consumo, salud, transporte inteligente, distribución de energía, ciudades inteligentes, distribución y logística, seguridad pública, industrial y manufactura, agricultura y manejo de recursos naturales y análisis de big data [18]. Las áreas de mayor crecimiento para los productos de IoT son los bienes de consumo y salud electrónica; para el presente trabajo nos enfocamos en productos de consumo específicamente los asistentes de voz inteligentes como Alexa y Google Assistant.

Un estudio realizado por Nae [28] en Estados Unidos evidencia los beneficios percibidos por el usuario respecto a los asistentes virtuales; un 59% de los usuarios entrevistados destacan el beneficio de los asistentes inteligentes en la vida diaria; un 40% resaltó la

optimización del tiempo empleado frente a una pantalla; un 39% sugerencias y servicios proporcionados y un 28% menciona el tiempo de espera en atención al cliente. Además, se evidencia los aspectos negativos; un 50% indican la dificultad para interactuar con otras personas; un 50% indica la falta de personalización; un 39% indica la falta de interpretación y un 30% indica el exceso de publicidad recibida.

3.2.1. Asistentes de voz inteligentes

Son sistemas que le permiten al usuario interactuar a través de comandos de voz, gestos, textos o diferentes acciones y estos responden a los comandos indicados ejecutando o dando respuesta a las diferentes acciones solicitadas [29] [30]. Este tipo de asistentes son capaces de realizar múltiples tareas como buscar información, realizar compras, efectuar reservas y conectarse a otros dispositivos que se encuentren en la red para poder administrar o controlar su funcionamiento como el manejo de un sistema de iluminación, control de temperatura, rutinas, control de electrodomésticos, entre otras [26] [29] [31].

La capacidad de procesamiento de la información combinada con el aprendizaje automático permite a estos dispositivos aprender sobre las preferencias del usuario y brindarle una asistencia personalizada al usuario en base a su perfil de usuario [29] [32].

Un altavoz inteligente es un dispositivo IoT que integra un asistente virtual inteligente que le permite interactuar al usuario a través de comandos de voz; a continuación, de acuerdo con la clasificación de la firma Voice.ai se listan los asistentes de voz inteligentes que han tenido una mayor aceptación por parte de los usuarios: Alexa, Google Assistant y Siri [33].

3.2.1.1. Alexa

Es un asistente virtual desarrollado por Amazon; lanzado en 2014 junto a su línea de altavoces inteligente Echo [11]. Inicialmente, el asistente estaba vinculado solo a los asistentes inteligentes y poco tiempo después abren su SDK para que otros desarrolladores y fabricantes puedan trabajar sobre este logrando incluir en una gran cantidad de dispositivos [34].

Alexa depende de dos elementos clave. En primer lugar, el comando de voz integrado que permite ejecutar varias peticiones por parte del usuario; en segundo lugar, las skills que son complementos que se pueden instalar para dotar de mayor funcionalidad al asistente [11].

Los dispositivos echo vienen con el hub smart home integrado, esto permite detectar los dispositivos compatibles de forma automática. Tiene una conectividad Wifi de doble banda compatible con redes 802.11 a/b/g/n [34].

3.2.1.2. Google Assistant

Es un asistente virtual desarrollado por Google; lanzado en 2016 [11]. Los dispositivos Google Nest vienen con un Chromecast integrado para poder

enviar órdenes a varios dispositivos de la marca e integra dos funciones para ajustar el volumen y mejorar la calidad del sonido en función al ruido del fondo (Media EQ y Ambient IQ) [35]. Tiene una conectividad Wifi 802.11b/g/n/ac y bluetooth 5.0 [35].

3.2.1.3. Siri

Es un asistente virtual desarrollado por Apple; fue lanzado en 2011 [11]. Los HomePod cancelan el ruido ambiental y el eco para que sea posible permitir escuchar los comandos de voz y solo es compatible con dispositivos de la marca Apple relativamente modernos; permite una interacción con la voz y también permite utilizar la interfaz táctil alojada en la parte superior del dispositivo [36] . Tiene una conectividad Wifi 802.11 ac y bluetooth 5.0 y es compatible con AirPlay 2; además permite la opción de acceso directo para invitados [36].

Además, de los asistentes propios se tienen varios altavoces inteligentes de terceros, como Sonos One, que vienen integrados con el asistente de voz de Alexa y Google Assistant [33].

A continuación, en la Figura 5 se presenta la tasa de crecimiento por geografía de la adopción de los asistentes virtuales inteligentes comprendido entre los años de 2020 hasta el 2025. Centro América, América latina y Asia presenta un crecimiento bajo en relación con la adopción de los asistentes virtuales inteligentes; en Europa, Oriente Medio y África la tasa de crecimiento es alta y en Norteamérica la tasa de adopción es medio y de acuerdo con [33] Estados Unidos es un mercado clave para la adopción de altavoces inteligentes.

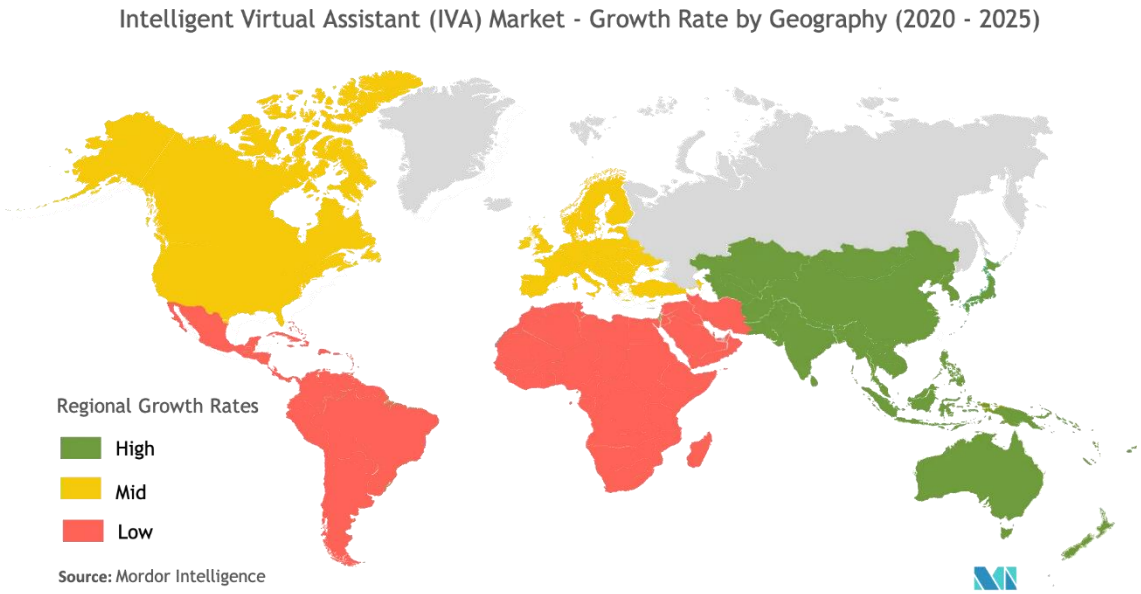


Figura 5. Tasa de crecimiento. Fuente: Imagen tomada de [33]

3.2.2. Aplicaciones móviles disponibles para el sistema operativo Android

Se tiene una variedad de aplicaciones disponibles para el control de los dispositivos inteligentes propios y de terceros.

A continuación, [37] presenta un listado de las aplicaciones más importantes para el control de dispositivos inteligentes:

- Amazon Alexa: aplicación para el asistente virtual Alexa
- Google Home: aplicación para el asistente virtual Google Assistant
- Aplicaciones de terceros como:
  - Smart Life
  - kasa Smart
  - Houseinhand KNX
  - Nexho
  - TaHoma by Somfy
  - Home Connect App
  - Philips Hue
  - Mi Home de Xiaomi
  - Wink Smart Home
  - Intesis AC Cloud
  - SmartThings de Samsung
  - openHAB Foundation

Estas aplicaciones permiten controlar los dispositivos inteligentes que se encuentran conectados en la misma red y ayudan a optimizar el consumo energético [37].

3.3. Seguridad y Privacidad IoT

Los dispositivos inteligentes de hogar se han convertido en un objetivo principal por parte de los atacantes para vulnerar la seguridad de estos; ya sea por falta de conocimiento de los usuarios respecto a la falta de seguridad de los dispositivos o por un mal uso por parte del usuario [38]. A medida que incrementan los dispositivos IoT aumenta la cantidad de formas de atacarlas, por lo tanto, es necesario promover la ciberseguridad y abordar la privacidad de los usuarios en relación con los dispositivos conectados [14].

El objetivo de la seguridad es protegerse ante las amenazas. Según [13] las amenazas se clasifican en dos tipos i) amenazas externas que son ataques al sistema por parte de los atacantes y ii) amenazas internas que se presentan por el mal uso del sistema o información. De acuerdo con [39] hay tres factores de seguridad i) confidencialidad de los datos, que garantiza que solo los usuarios autorizados accedan y modifiquen los datos a través de un mecanismo de control de acceso y un proceso de autenticación ii)

privacidad, permite mantener cierta información en secreto mediante un control de acceso, las características de la privacidad son secreto, anonimato y soledad [16] y iii) veracidad, que garantiza la aplicación de reglas de seguridad en el sistema [39].

La seguridad y privacidad son aspectos importantes para garantizar la interacción entre el mundo físico y digital [39].

Los asistentes virtuales inteligentes se activan mediante la voz; es decir siempre están a la escucha, pero solo se activan al escuchar el comando que los activa ya sea Alexa, Ok Google, Hey Siri, etc. Estos asistentes graban todo lo que escuchan para detectar la palabra de activación y transferir datos, caso contrario descartan el texto escuchado [40]. La voz no se almacena en el dispositivo local, sino que se almacena en el servidor de los fabricantes y desarrolladores de estos asistentes inteligentes [40].

Pero ¿Qué datos realmente se transfieren a esos servidores? ¿Qué hacen las empresas con esos datos? Y ¿Qué podemos hacer como usuarios? Es lo que se desea validar y conocer en el presente TFM.

Google home, no graba todas nuestras conversaciones; si no que escucha pequeños fragmentos para detectar si se ha pronunciado la frase de activación; recolecta la duración de sesiones multimedia y las aplicaciones empleadas, pero, no guarda información sobre el contenido que reproduce [40].

Alexa, en cambio recolecta conversaciones, peticiones y órdenes de voz y Amazon procesa y registra esa información que puede ser compartida con terceros; lo cual indican que se graba para mejorar los servicios y precisión de los resultados [40]. Estas grabaciones son posibles de administrar desde el menú de configuración.

Latha, M et al. [19] propone un método de una aplicación móvil inteligente segura para el entorno doméstico SSMASHE (Secured Smart Mobile App for Smart Home Environment) que consta de tres módulos i) cosas inteligentes, ii) aplicación móvil inteligente y iii) plataforma de big data.

Xu, W et al. [41] propone un sistema escalable llamado SoProtector, que evita que se filtre la información a través de un análisis de flujo de datos entre las capas de java y nativo que detecta funciones maliciosas implantadas en el sistema operativo. Para detectar las funciones maliciosas SoProtector realiza un motor en tiempo real i) presenta archivos binarios en la familia nativa como una imagen en escala de grises, ii) uso de ARM configuradas a la inversa para que se obtenga el código del sistema operativo y iii) se transforma el archivo a lenguaje ensamblador que incluye un archivo gdl. El experimento se plantea con 3400 aplicaciones en el que se demuestra que el sistema puede detectar sumideros; además de inspeccionar y bloquear de manera efectiva al menos el 82% de las aplicaciones que contienen sistema operativo malicioso por parte de terceros.

Sun, A et al. [20] analiza un conjunto de electrodomésticos inteligentes como luces, interruptor, sensores de movimiento, cámara de seguridad y un asistente doméstico y se pone a prueba las vulnerabilidades para analizar a que información podría tener acceso un atacante. Demostrando que los dispositivos IoT inalámbricos son

extremadamente vulnerables a los ataques de canal lateral ya que estos dispositivos filtran una gran cantidad de datos y permiten que el atacante use la red de sensores IoT para sus propios fines.

Youn, M et al. [42] en su estudio identifica un ecosistema que consiste en altavoces de pantalla y recopila datos del sistema y teléfono inteligente. De igual manera se recopila datos almacenados en la nube de aplicaciones para teléfonos inteligentes que funcionan con pantallas inteligentes.

Los dispositivos IoT tienden a reutilizar y personalizar los componentes de otros, por lo que las vulnerabilidades que se encuentran en un dispositivo suelen estar presente en otros y por lo tanto los dispositivos renombrados heredan una vulnerabilidad del OEM (Original Equipment Manufacturer en español el fabricante original de equipo), pero no el parche de seguridad que corrige la vulnerabilidad [38]. Durante el proceso de validación se evidenció que, aunque las vulnerabilidades son antiguas se han evidenciado un gran conjunto de dispositivos potencialmente vulnerables y muchos proveedores pequeños no se preocupan por mantener la seguridad después de venderlo. Además, que terceros personalizan los dispositivos IoT de los OEM y los revenden con su propia marca, esto complica la gestión de la seguridad del producto y pone a los clientes en peligro, ya que las vulnerabilidades de los proveedores ascendentes tienden a propagarse a un conjunto más amplio de proveedores descendentes, pero los parches de seguridad no lo hacen [41].

3.4. Uso en la domótica

De acuerdo con [23] se define los entornos inteligentes como la optimización energética ajustando dinámicamente el comportamiento de los diferentes actuadores que controlan las condiciones ambientales a través de la detección y adaptación a la presencia humana; la arquitectura de estos sistemas suele basarse en redes de sensores especializada que realiza un análisis en tiempo real de los datos recopilados, ajustando el comportamiento de los diferentes actuadores en base a una situación específica.

De igual manera [43] considera un hogar inteligente como una aplicación específica del concepto de ambiente inteligente que se define como un entorno digital que apoya a las personas en su vida diaria.

Finalmente [24] define un hogar inteligente como una aplicación de internet de las cosas que utiliza el internet para monitorear y controlar los dispositivos mediante un sistema de automatización para el hogar; Waheb A. Jabbar, desarrolla un prototipo denominado IoT@HoMe que permite el monitoreo de las condiciones del hogar y automatización del control de los electrodomésticos a través de internet en cualquier momento y lugar.

Por lo tanto, se entiende por domótica el proceso de automatizar y controlar los sistemas electrónicos de una vivienda mediante la gestión del control de software con el fin de proporcionar bienestar a los usuarios.

CAPITULO IV

Diseño y Configuración del ambiente

En este capítulo se presenta un análisis de las aplicaciones que se van a emplear para el proceso y obtención de los resultados aplicados en los dispositivos inteligentes y aplicaciones móviles; su arquitectura y funcionamiento de los dispositivos inteligentes para el análisis de las vulnerabilidades y riesgos y el monitoreo del tráfico de red.

4.1. Dispositivos

Para el presente TFM, se va a realizar el proceso de análisis en los siguientes dispositivos IoT:

- Google Nest mini 2da generación
- Echo Dot 3ra generación y
- Echo Dot 4ta generación

Estos dispositivos se emplean en el hogar para tareas cotidianas como encender y apagar las luces, reproducir música y actividades de preguntas realizadas por el usuario mediante comandos de voz y el asistente virtual inteligente lo procesa, interpreta y responde a las consultas. Estos dispositivos siempre se encuentran escuchando a la espera de la palabra de activación y poder empezar a grabar el audio que indica el usuario para procesarlo y dar una respuesta.

4.2. Descripción de las herramientas a emplear

Para el análisis del tráfico de red de los dispositivos inteligentes se ha optado por emplear la herramienta Wireshark[5] versión 4.0.1 que es un analizador de paquetes de código abierto y libre, multiplataforma y con un amplio soporte para cientos de protocolos de red. Con esta herramienta se logra capturar y analizar en detalle todo el tráfico de la red de manera gráfica, presenta un esquema de colores lo cual permite identificar el escaneo de manera óptima, dar énfasis a los paquetes más relevantes y es capaz de descifrar los paquetes que han sido enviados por medio de un protocolo seguro permitiendo analizar el contenido de estos.

Conjuntamente, para identificar la IP (Internet Protocol) de cada uno de los dispositivos inteligentes a analizar se emplea la aplicación Fing[6] para Android versión 12.0.3 el cual se ha tomado la decisión de emplear este escáner ya que permite identificar dispositivos, escanear puertos, entre otros; se visualiza de manera gráfica todos los dispositivos

[5] https://www.wireshark.org/
[6] https://www.fing.com/

conectados a la red y de igual manera permite realizar un escaneo de los puertos abiertos; por otro lado, también se puede obtener a través de la consola CMD (Command Prompt) de Windows en el que se realiza el ping para verificar la solicitud de respuesta.

Finalmente, para realizar el escaneo de vulnerabilidades en las aplicaciones se realiza un análisis con varias herramientas y se puede visualizar los resultados en la Tabla 6. Por lo tanto, en base al previo análisis se opta por emplear la herramienta ImmuneWeb[7] en la versión Community Edition. Esta herramienta permite obtener los siguientes resultados de la aplicación móvil: permisos y privacidad, TOP 10 de pruebas de seguridad OWASP (Open Web Application Security Project), comunicaciones externas y el análisis de la composición del software.

Herramientas	Pros	Contra	Observación
AppCensus	Gratuita Comportamiento de privacidad de las aplicaciones Cumplimiento de RGPD, CCPA y COPA	Actualmente conjunto de datos obsoleto	No está disponible hasta una próxima actualización
MobSF	Libre y de pago Análisis de malware	Soporte limitado al usar versión libre Análisis dinámico, no funciona si se configura en una máquina virtual Requiere realizar la configuración del ambiente	Resultados en tiempo real (se presentan en función de procesos de la estructura de una aplicación)
SonarQube	Prueba gratuita Integración DevOps Análisis rápido	Límite de funcionalidades al usar versión gratuita	Análisis a código fuente
ImmuneWeb	Versión Community y Enterprise Pruebas de seguridad de aplicaciones basadas en riesgos Audita OWASP, móvil TOP 10 y	Tiempo de resultados	Generación en PDF de los resultados

[7] https://www.immuniweb.com/

	otras vulnerabilidades Comprobación de privacidad		

Tabla 6. Resultado de análisis de herramientas. Fuente: Elaboración propia

4.3. Proceso de implementación

A continuación, se presenta la arquitectura y funcionamiento de los dispositivos inteligentes y el proceso seguido para la obtención de los datos necesarios para poder realizar el análisis en los dispositivos descritos inicialmente.

4.3.1. Arquitectura y funcionamiento

A continuación, se presenta la arquitectura y el funcionamiento de los asistentes virtuales inteligentes; que están compuestos por componentes de hardware y software.

4.3.1.1. Arquitectura de Google Nest mini

El proceso de interactuar con el asistente virtual se lo realiza a través de comandos de voz al activar el asistente mediante la palabra de activación más populares como: “Ok, Google”, “Hey, Google”, entre otras palabras de activación que no son muy conocidos y que interactúan con los componentes del sistema como se presenta en la Figura 6.

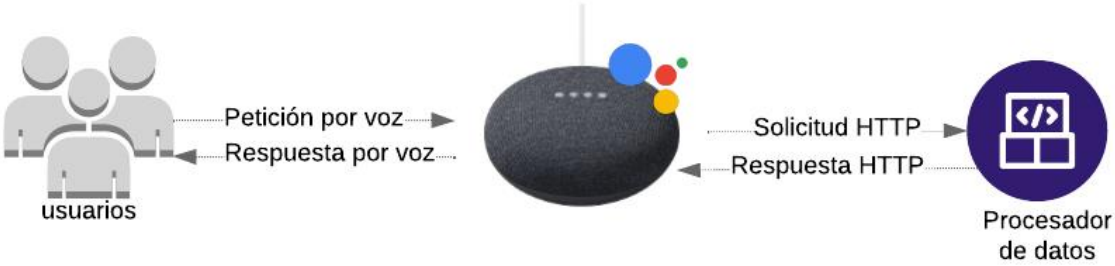


Figura 6. Arquitectura Google Assistant. Fuente: Imagen basada de [44]

A continuación, se detalla el proceso que se presenta en la Figura 6; en el que el usuario realiza una petición a través de comandos de voz hacia el dispositivo inteligente la cual llega al dispositivo inteligente que siempre está a la escucha de la palabra de activación y el cual transforma el lenguaje hablado en archivos que puedan tratarse empleando tramas de tipo JSON (JavaScript Object Notation) para el almacenamiento de datos y estas se envían mediante el protocolo HTTP (Hypertext Transfer Protocol) para su procesamiento en el código, al ejecutarse el código se genera la respuesta HTTP hacia el dispositivo inteligente en el que se convierte a lenguaje hablado para dar respuesta al usuario a través del asistente virtual para su entendimiento. En la sección del proceso de datos al ser capaces de ejecutar código tienen la capacidad de interactuar con la mayoría de APIS

(Application Programming Interfaces) del mercado con lo que se consigue la conexión entre distintas partes añadiendo la funcionalidad mediante la voz a través de los asistentes inteligentes [44].

4.3.1.2. Arquitectura de Alexa

El proceso de usar un asistente virtual a través de un altavoz inteligente o smartphone se realiza mediante la interacción del usuario vía comandos de voz activando con la palabra “Alexa” u otras palabras configuradas que se pueden activar para el dispositivo y que interactúan con los componentes del sistema de Amazon-Alexa y las skills de terceros las cuales conforman el ecosistema de Amazon Alexa como se presenta en la Figura 7.

Figura 7. Arquitectura Alexa. Fuente: Imagen basada de [45]

A continuación, se detalla el flujo de la Figura 7 en el que el usuario envía una petición de comando de voz al dispositivo inteligente el cual se caracteriza por estar siempre escuchando y a la espera de la palabra de activación en el que el dispositivo graba el audio y envía ese comando de voz a Alexa Voice Server (AVS) el cual procesa mediante el reconocimiento automático de voz y la comprensión del lenguaje natural de Alexa y se obtiene la skill necesaria del servicio de Alexa para realizar la tarea que solicita el usuario; seguidamente se delega el comando al servidor de la skill que el usuario requiere ejecutar ya sea propia del servicio de Amazon o de terceros (al ser una skill de terceros se necesita acceder a un servicio web vía HTTPS mediante una llamada REST (Representational State Transfer)). Finalmente, al obtener el servidor las skills envía una respuesta al servicio en la nube de Alexa y esta envía una respuesta en texto al dispositivo inteligente el cual reproduce el audio de respuesta solicitado por el usuario a través del asistente virtual. Además, se envía la información del comando en formato de tarjetas informativas a la aplicación de Alexa las cuales contienen el comando de voz realizado por el usuario y la respuesta proporcionada por el dispositivo inteligente.

4.3.1.3. *Flujo del tráfico de red*

A continuación, en la Figura 8 se presenta un diagrama del tráfico de red de los asistentes inteligentes de cómo se comunican los dispositivos con los servidores de cada uno.

El proceso del flujo inicia cuando el usuario envía una petición de comando de voz hacia el asistente virtual inteligente el cual se inicia con la palabra de activación, al finalizar la grabación se encripta la información y se envía los datos a los servidores asociados (Alexa Voice Service para el dispositivo Alexa y Voice Match para el dispositivo Google Nest mini); estos servidores reciben los datos, desencriptan, procesan y construyen la respuesta. Posteriormente, el servidor encripta la respuesta y lo envía al dispositivo inteligente el cual desencripta y reproduce el comando a través del asistente virtual inteligente.

Figura 8. Tráfico de red. Fuente imagen basada de [46]

4.3.2. Identificar la IP de los dispositivos inteligentes

Inicialmente, se identifica la IP de cada uno de los dispositivos inteligentes a analizar empleando la aplicación Fing como se presenta en la Figura 9.

- Se presenta la IP 192.168.1.2 correspondiente al dispositivo Alexa Echo Dot 4ta generación
- Se presenta la IP 192.168.1.3 correspondiente al dispositivo Alexa Echo Dot 3ra generación
- Se presenta la IP 192.168.1.9 correspondiente al dispositivo Google Nest mini 2da generación

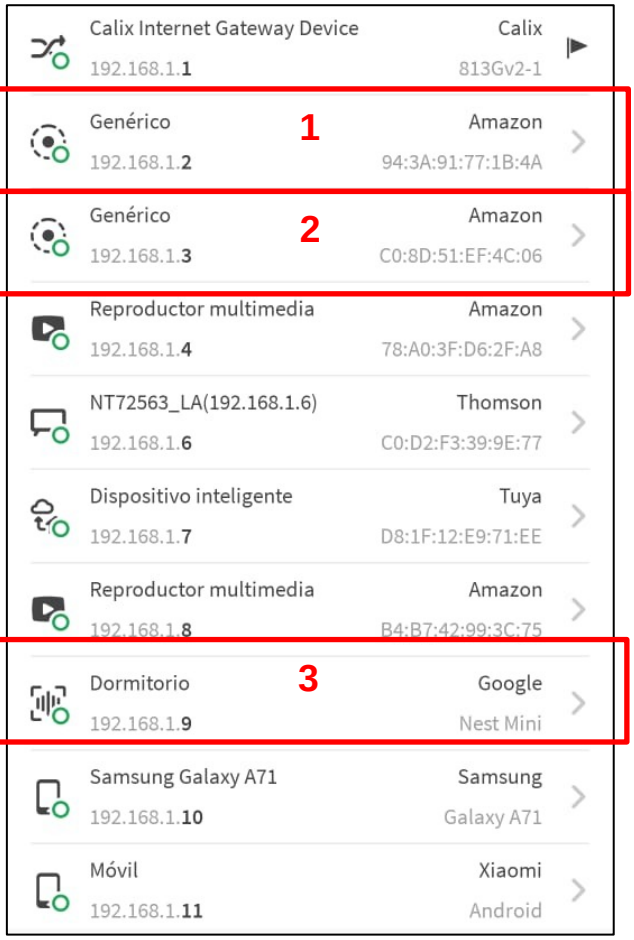


Figura 9. Identificación de dispositivos en la red. Fuente: Elaboración propia

Seguidamente, al tener la IP correspondiente de cada dispositivo inteligente se realiza un ping a cada uno de los dispositivos como se indica a continuación y poder verificar la solicitud de respuesta:

- En la Figura 10 se presenta la respuesta del ping al dispositivo Alexa Echo Dot 4ta generación y se tiene como respuesta “Tiempo de espera agotado para esta solicitud” ya que el dispositivo se encuentra protegido para no responder la solicitud de ping. De igual manera se valida con la aplicación Fing.

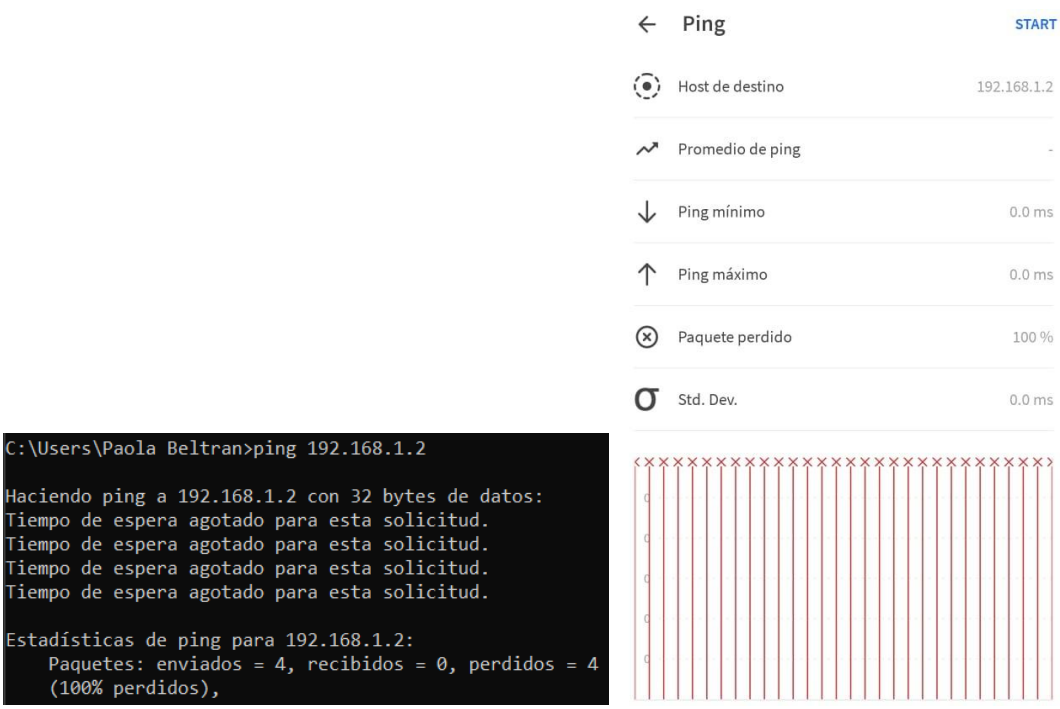


Figura 10. Ping dispositivo Alexa 4ta Generación. Fuente: Elaboración propia

- En la Figura 11 se presenta la respuesta del ping al dispositivo Alexa Echo Dot 3ra generación y se tiene como respuesta “Tiempo de espera agotado para esta solicitud” ya que el dispositivo se encuentra protegido para no responder la solicitud de ping. De igual manera se valida con la aplicación Fing.

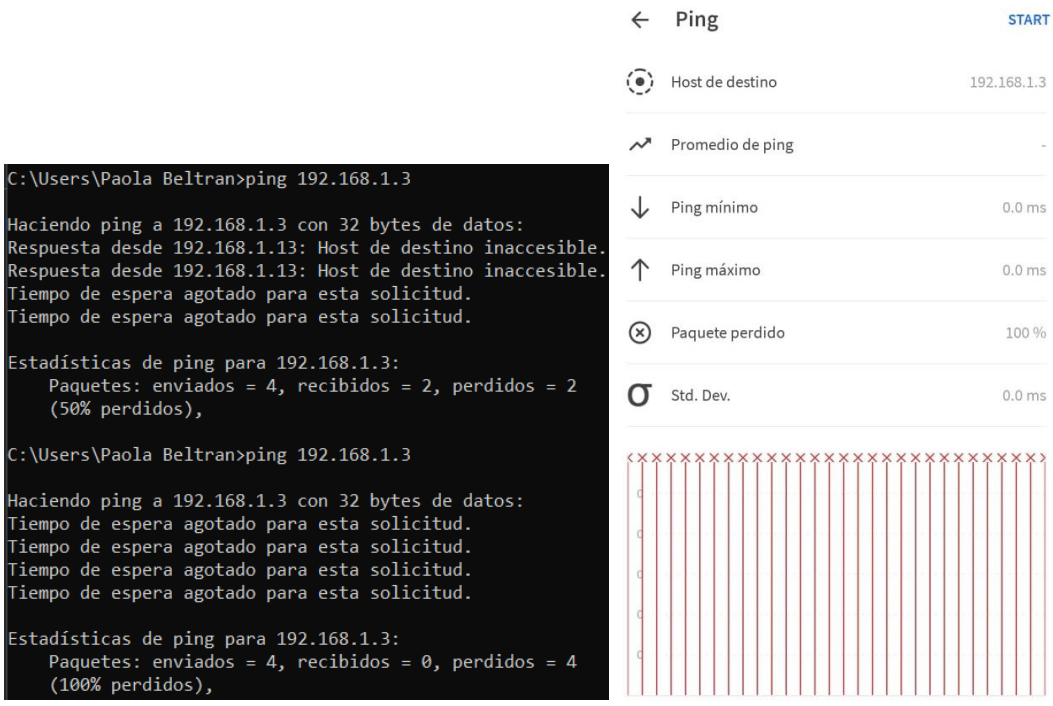


Figura 11. Ping dispositivo Alexa 3ra Generación. Fuente: Elaboración propia

- En la Figura 12 se presenta la respuesta del ping al dispositivo Google Nest mini 2da generación, y se tiene como respuesta el acceso por lo tanto este dispositivo no tiene bloqueado a las peticiones de ping. De igual manera se valida con la aplicación Fing.

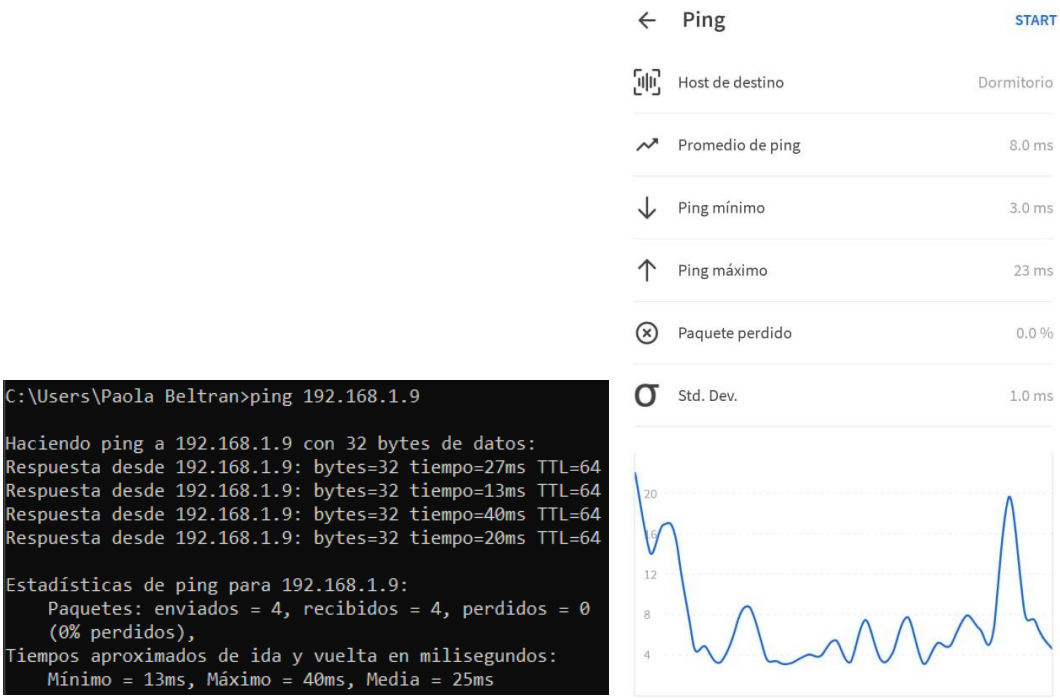


Figura 12. Ping dispositivo Google Nest mini. Fuente: Elaboración propia

4.3.3. Identificación de puertos abiertos

Tener los puertos abiertos significa un riesgo ya que si no se monitoriza se pueden ejecutar servicios no autorizados y son explotables por los atacantes, por eso se recomienda cerrar los puertos abiertos. Además, el tener uno o varios puertos abiertos no significa que se establezca un canal de comunicación, por lo tanto, al no tener un servicio escuchando en el puerto se descartan todos los paquetes enviados al puerto. Sin embargo, no todos los puertos abiertos con servicios escuchando son vulnerables [47].

Por lo tanto, para escanear los puertos abiertos se emplea la aplicación Fing obteniendo los siguientes resultados:

- En la Figura 13 se presenta el resultado de los puertos abiertos correspondiente a los dispositivos inteligentes Alexa de 3ra y 4ta generación.

Figura 13. Puertos abiertos de Alexa. Fuente: Elaboración propia

- En la Figura 14 se presenta el resultado de los puertos abiertos correspondiente al dispositivo inteligente Google Nest mini de 2da generación.

Figura 14. Puertos abiertos de Google Nest mini. Fuente: Elaboración propia

CAPITULO V

Proceso y resultados obtenidos

En esta sección se presenta el proceso seguido y los resultados obtenidos al realizar el análisis de trafico de red y de igual manera los resultados del escaneo de vulnerabilidades en las aplicaciones móviles. Además, se presenta la interpretación de los resultados obtenidos y se proporciona las recomendaciones que el usuario debe tener en cuenta para evitar la exposición de su privacidad.

5.1. Proceso y análisis del tráfico de red

Para el proceso de la captura de los paquetes se emplea la herramienta Wireshark y se toma la información de cuatro días en el cual se interactúa con los dispositivos con peticiones a través de comandos de voz como encender y apagar la luz, reproducir música y un listado de múltiples preguntas para generar peticiones y respuestas y que se procese la información.

Por el tamaño de los archivos, se adjuntan una parte de los resultados obtenidos de los dispositivos inteligentes identificados por su IP en el Apéndice 1, Apéndice 2, Apéndice 3, Apéndice 4, Apéndice 5, Apéndice 6, Apéndice 7, Apéndice 8, Apéndice 9, Apéndice 10, Apéndice 11 y Apéndice 12.

A continuación, se presentan los resultados más relevantes obtenidos en el análisis del tráfico de red.

Al analizar el tráfico de la red de los dispositivos se puede evidenciar resultados comunes como:

- Protocolos de descubrimiento como SSDP (Simple Service Discovery Protocol) que emplea UDP (User Datagram Protocol), MDNS (Multicast Domain Name System)
- Protocolo TPLINK-SMARTHOME/JSON que obtiene la información del sistema con /system/get_sysinfo
- Protocolo MDNS verifica que se realizan consultas en las cuales al seleccionar cada uno de los paquetes se verifica en los resultados que la información está encriptada y que la IP de destino es 224.0.0.251 la cual al verificar su origen se presenta que es un tipo de dirección multicast reservada

Y resultados específicos de cada dispositivo como:

Alexa 3ra y 4ta generación

- El protocolo SSDP consulta el dispositivo con el que interactúa y se obtiene la marca Sengled (fabricante de productos de iluminación) con el que se puede identificar que pertenece a una bombilla inteligente conectada como se presenta en la Figura 15.

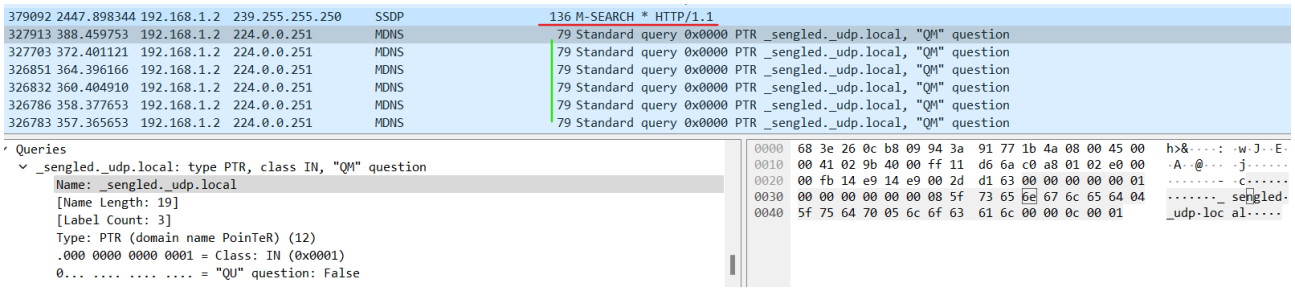


Figura 15. Resultados del análisis del tráfico. Fuente: Elaboración propia

- De igual manera como se puede apreciar en la Figura 16 se presenta que el dispositivo hace uso de viziocast (permite transmitir de forma inalámbrica programas) para ejecutar el comando solicitado; el cual se presenta de manera encriptada, pero se puede interpretar la acción ya que se muestra el dispositivo.

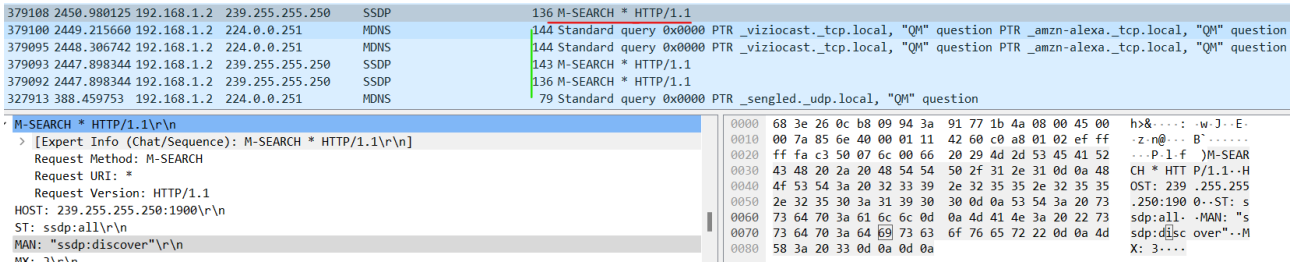


Figura 16. Resultados del análisis del tráfico. Fuente: Elaboración propia

Google Nest mini 2da generación

- En el análisis del tráfico de red, se puede verificar muchas peticiones que se realizan aun cuando el dispositivo está inactivo como se presenta en la Figura 17. De igual manera, se puede verificar que se consulta frecuentemente googlezone, get_sysinfo, entre otros.

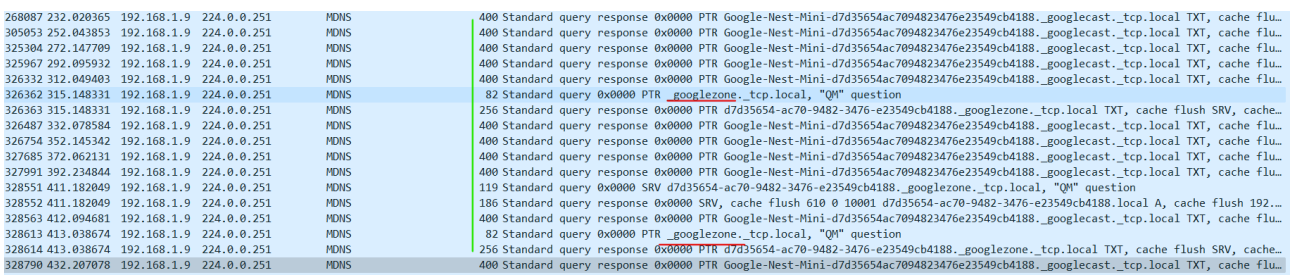


Figura 17. Resultados del análisis del tráfico. Fuente: Elaboración propia.

- En el siguiente protocolo MDNS se consultan los datos del dispositivo; en el que se puede ver la marca/modelo, se obtiene el nombre configurado en el dispositivo, el tiempo de actividad, servicio que se está ejecutando como Spotify entre otros como se puede ver en la Figura 18.

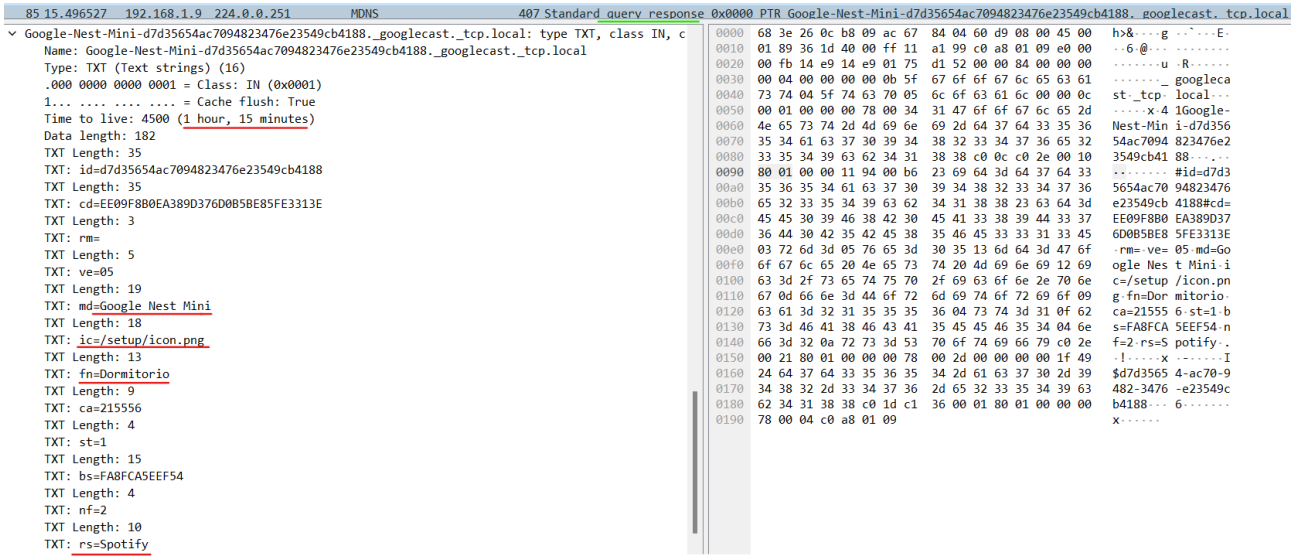


Figura 18. Resultados del paquete. Fuente: Elaboración propia

- En la Figura 19 se presenta un ejemplo de un comando *"rs=Enviando contenido: Ojos Marrones"* y de igual manera se han podido validar otros contenidos de respuesta a la petición de reproducir música.

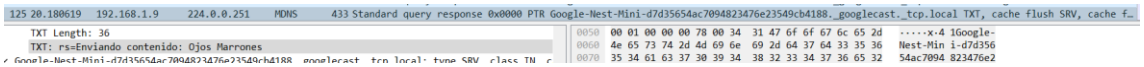


Figura 19. Resultado de un paquete específico. Fuente: Elaboración propia

Por lo tanto, al analizar el tráfico de red se puede evidenciar que los comandos e interacciones con los dispositivos se envían y se reciben de manera encriptada; por lo cual no se ha podido analizar su contenido. Además, al verificar los puertos se identifica que estos proveen un servicio poco fidedigno y los datagramas pueden llegar duplicados, con errores o se pierden sin aviso.

Por otro lado, los datos sobre las interacciones con el asistente de Google se almacenan en sus servidores y de igual manera si se desea borrar las interacciones se puede realizar en la sección "Mi actividad" [48].

A continuación, en la Figura 20 se presenta la ubicación de los servidores de Google: 14 centros en Norteamérica, 1 en Sudamérica, 6 en Europa y 2 en Asia. Al tener varios servidores la información puede estar en cualesquiera de estos; por lo tanto, la protección de la información varía de acuerdo con los países y de igual manera Google indica que aplican las mismas protecciones y cumplen con los marcos legales relacionados con la transferencia de datos [49].

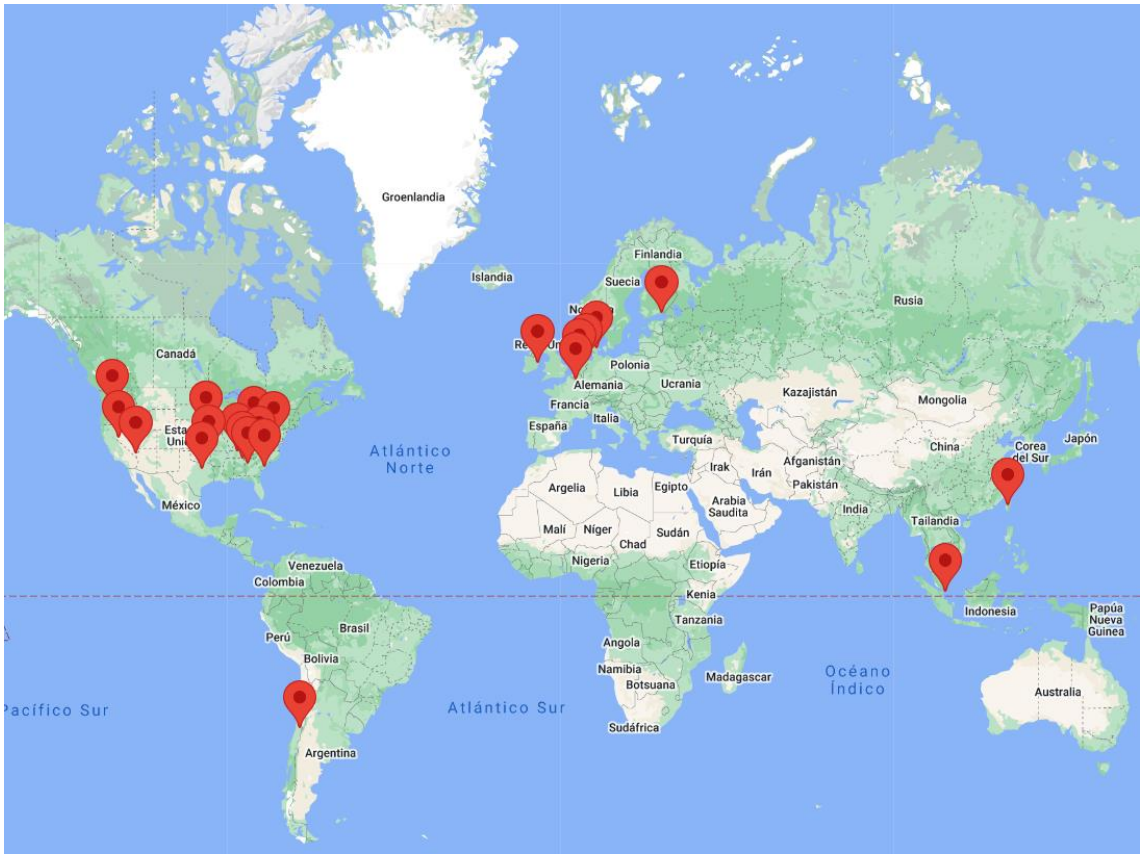


Figura 20. Ubicaciones de los centros de datos de Google. Fuente: Imagen tomada de [49]

De igual manera, los datos sobre las interacciones con el asistente de Amazon se almacenan en sus servidores y de igual manera si se desea borrar las interacciones se puede realizar en la sección “Privacidad de Alexa” en el que se puede revisar el historial de voz, dispositivos, permisos de las skills y administrar los datos.

Con respecto a la recopilación, el uso y la retención de información personal Amazon participa en el marco de protección de Privacidad UE-EEUU y Suiza-EEUU [50]. Además, Alexa opera en EEUU por lo que pueden restringir el acceso a Alexa o a determinadas funciones desde otros sitios.

A continuación, en la Figura 21 se presenta la ubicación de los servidores en el que la nube de AWS (Amazon Web Services) abarca un total de 96 zonas distribuidas en 30 regiones de todo el mundo [51].

Figura 21. Ubicaciones de los centros de datos de AWS. Fuente: Imagen tomada de [51]

5.2. Proceso y análisis del escaneo de vulnerabilidades

En primer lugar se identifica las aplicaciones disponibles en Android correspondientes para cada dispositivo inteligente como es Alexa Amazon y Google Home.

Seguidamente se obtiene el APK (Android Application Package) de cada una de las aplicaciones para el proceso del escaneo de las vulnerabilidades de las aplicaciones móviles de Android y se emplea la herramienta ImmuniWeb Community Edition.

A continuación, en la Tabla 7 se presentan las características de las aplicaciones móviles a escanear:

Aplicación	Dispositivo	Versión
Amazon Alexa	Alexa Echo Dot	2.2.487227.0
Google Home	Google Nest mini	2.60.1.19

Tabla 7. Características aplicaciones Android. Fuente: Elaboración propia

Al realizar el proceso de escaneo con la herramienta se obtienen los siguientes resultados los cuales se presentan a nivel general en la Tabla 8. Además, cada uno de los resultados expuestos en la tabla se dividen en subcategorías, prioridad y nivel de criticidad; de los cuales se van a detallar los 2 primeros resultados presentados en la tabla analizando sus resultados más relevantes en base a su prioridad; ya que por su extensión y objetivos del TFM no es posible abarcar todos y detallar cada uno de ellos por lo cual se indicarán de manera general en la siguiente sección.

Resultados	Total Amazon Alexa	Total Google home
Permisos y privacidad de aplicaciones móviles	48	25
Prueba de seguridad OWASP Mobile Top 10	23	19
Comunicaciones externas de la aplicación móvil	0	0
Análisis de composición de software	60	18

Tabla 8. Resultado proceso de escaneo. Fuente: Elaboración propia

5.2.1. Permisos y privacidad de aplicaciones móviles

En la Tabla 9 se presenta los permisos de la aplicación que tienen un nivel de criticidad peligroso. La aplicación móvil solicita acceso a las siguientes funcionalidades que pueden poner en peligro la privacidad del usuario bajo ciertas circunstancias.

Permisos y privacidad de aplicaciones móviles	Descripción	Amazon Alexa	Google Home
ACCESS_BACKGROUND_ LOCATION	Permite que una aplicación acceda a la ubicación en segundo plano.	X	
ACCESS_COARSE_ LOCATION	Acceder a fuentes de ubicación aproximadas, como la base de datos de la red móvil, para determinar una ubicación aproximada del teléfono, donde esté disponible. Las aplicaciones maliciosas pueden usar esto para determinar aproximadamente dónde se encuentra.	X	X
ACCESS_FINE_ LOCATION	Acceder a fuentes de ubicación precisas, como el sistema de posicionamiento global en el teléfono, donde esté disponible. Las aplicaciones maliciosas pueden usar esto para determinar dónde se encuentra y pueden consumir energía adicional de la batería.	X	X
ACCESS_MEDIA_ LOCATION	Permite que una aplicación acceda a cualquier ubicación	X	

	geográfica persistente en la colección compartida del usuario.		
ANSWER_PHONE_ CALLS	Permite que la aplicación responda una llamada telefónica entrante.	X	
AUTHENTICATE_ ACCOUNTS	Permite que una aplicación use las capacidades de autenticación de cuentas del administrador de cuentas, incluida la creación de cuentas, así como la obtención y configuración de sus contraseñas.	X	
CALL_PHONE	Permite que la aplicación llame a números de teléfono sin su intervención. Las aplicaciones maliciosas pueden causar llamadas inesperadas en su factura telefónica. No permite que la aplicación llame a números de emergencia.	X	X
CAMERA	Permite que la aplicación tome fotos y videos con la cámara. Esto permite que la aplicación recopile imágenes que la cámara está viendo en cualquier momento.	X	X
GET_ACCOUNTS	Permite el acceso a la lista de cuentas en el servicio de cuentas.	X	X
MANAGE_ACCOUNTS	Permite que una aplicación realice operaciones como agregar y eliminar cuentas y eliminar su contraseña.	X	X
READ_CONTACTS	Permite que una aplicación lea todos los datos de contactos (direcciones) almacenados en su teléfono. Las aplicaciones maliciosas pueden usar esto para enviar sus datos a otras personas.	X	
READ_EXTERNAL_ STORAGE	Permite que una aplicación lea desde un almacenamiento externo.	X	

READ_PHONE_STATE	Permite que la aplicación acceda a las funciones del teléfono del dispositivo. Una aplicación con este permiso puede determinar el número de teléfono y el número de serie de este teléfono, si una llamada está activa, el número al que está conectada esa llamada, etc.	X	
READ_SMS	Permite que la aplicación lea mensajes SMS almacenados en su teléfono o tarjeta SIM. Aplicaciones maliciosas podrían leer los mensajes confidenciales.	X	
RECEIVE_MMS/SMS	Permite que la aplicación reciba y procese mensajes MMS. Las aplicaciones maliciosas pueden monitorear los mensajes o eliminarlos sin mostrárselos.	X	
RECORD_AUDIO	Permite que la aplicación acceda a la ruta de grabación de audio.	X	X
SEND_SMS	Permite que la aplicación envíe mensajes SMS. Las aplicaciones maliciosas pueden costar dinero al enviar mensajes sin su confirmación.	X	
USE_CREDENTIALS	Permite que una aplicación solicite tokens de autenticación.	X	
WRITE_EXTERNAL_ STORAGE	Permite que una aplicación escriba en el almacenamiento externo.	X	X

Tabla 9. Permisos de aplicaciones móviles. Fuente: Elaboración propia

Por lo tanto, al identificar y analizar cada uno de los permisos que estas aplicaciones móviles solicitan, se puede verificar el riesgo al que estamos expuestos día a día al emplear estas aplicaciones y que no somos conscientes de estos riesgos al momento de instalar una aplicación y otorgar los permisos que estas solicitan para su correcto funcionamiento. Por lo cual, de no ser estrictamente necesario el uso de dichas aplicaciones como usuarios debemos ser conscientes a lo que estamos expuestos; de igual manera se pueden otorgar permisos limitados a las aplicaciones o solo conceder permisos al momento de usarla.

Los permisos de aplicaciones móviles calificados como normal y otros no se abarcan pero se adjuntan en el Apéndice 13, Apéndice 14, Apéndice 15 y Apéndice 16.

5.2.2. Prueba de seguridad OWASP Mobile Top 10

En la Tabla 10 se presenta el resultado de la prueba de seguridad OWASP Mobile Top 10 en el que se muestra las siguientes fallas y debilidades de seguridad que pueden afectar la aplicación.

Prueba de seguridad OWASP Mobile Top 10	Descripción	Amazon Alexa	Google Home
Base de datos SQLITE en claro	La aplicación móvil utiliza una base de datos SQLite sin cifrar. A esta base de datos puede acceder un atacante con acceso físico al dispositivo móvil o una aplicación maliciosa con acceso de root al dispositivo. La aplicación no debe almacenar información confidencial en texto claro.	X	X
Exposición de datos potencialmente sensibles	La aplicación móvil puede exponer información potencialmente confidencial durante su tiempo de ejecución.	X	X
Datos sensibles codificados	La aplicación móvil contiene datos codificados potencialmente confidenciales. Un atacante con acceso a la aplicación móvil puede extraer fácilmente estos datos de la aplicación y utilizarlos en cualquier otro ataque.	X	
Datos externos en consultas SQL (Structured Query Language)	La inclusión de entradas en consultas SQL sin procesar puede generar una vulnerabilidad de inyección de SQL local en la aplicación móvil, lo que podría comprometer cualquier información confidencial almacenada en los archivos de la base de datos. El enfoque correcto es usar sentencias SQL preparadas más allá del control del usuario.	X	X
Posible ataque de hombre en el medio	La verificación incorrecta o deshabilitada del nombre de host de los certificados SSL(Secure Sockets	X	X

	Layer)/TLS(Transport Layer Security) de back-end puede exponer a los usuarios de aplicaciones móviles a ataques MITM (men in the middle) bajo ciertas condiciones.		
Claves de cifrado codificadas	Las claves de cifrado codificadas pueden poner en peligro el almacenamiento y la transmisión seguros de datos dentro de la aplicación móvil al permitir que un atacante descifre cualquier información potencialmente confidencial.	X	
Uso de protocolo HTTP no cifrado	La aplicación móvil utiliza el protocolo HTTP para enviar o recibir datos. El diseño del protocolo HTTP no proporciona ninguna encriptación de los datos transmitidos y puede ser interceptado fácilmente si un atacante se encuentra en la misma red o tiene acceso al canal de datos de la víctima.	X	X
Algoritmos hash débiles	La aplicación móvil utiliza algoritmos hash débiles. Los algoritmos hash débiles (por ejemplo, MD2, MD4, MD5 o SHA-1) pueden ser vulnerables a colisiones y otras debilidades de seguridad, y no deben usarse cuando se requiere un hash de datos confiable.	X	X

Tabla 10. Prueba de seguridad de alto riesgo. Fuente: Elaboración propia

Por lo tanto, al identificar solo las fallas y debilidades de seguridad que pueden afectar la aplicación con un nivel de criticidad alto, es evidente que es un proceso necesario para considerar y tener en cuenta en las etapas del desarrollo al momento de desarrollar estas aplicaciones y poder agregar y solventar estas brechas de seguridad y mantener la información de los usuarios seguros; agregando de tal manera un cierto nivel de seguridad en las aplicaciones para evitar el acceso a terceros o atacantes y que puedan tener acceso y control de las mismas para fines malintencionados.

Las fallas y debilidades de seguridad con un nivel de criticidad medio, bajo y de advertencia no se abarcan pero se adjuntan en el Apéndice 17 y Apéndice 18.

CAPITULO VI

Conclusiones y trabajos futuros

En el presente TFM se ha realizado el análisis de los dispositivos inteligentes y el proceso de escaneo de vulnerabilidades de las aplicaciones Android propias de los asistentes inteligentes.

En primer lugar, en el proceso de análisis del tráfico de red se pudo evidenciar que en los paquetes obtenidos de la interacción con los asistentes inteligentes la información que se envía y recibe está totalmente encriptada; por lo cual, no se pudo analizar su contenido y por lo tanto, el flujo de datos generados en la interacción con el asistente no representa un riesgo para el usuario. Sin embargo, se pudieron identificar algunos paquetes en los que los dispositivos obtienen información como descubrir los dispositivos inteligentes conectados, obtener la zona horaria, información del dispositivo como el nombre, marca, modelo, el tiempo de actividad y servicio que se está ejecutando. Además, con el análisis del tráfico de red se pudo evidenciar que el asistente de Google a pesar de no estar activo este genera mucho tráfico; en el que se identificó los protocolos de descubrimiento obteniendo información acerca del dispositivo, zona horaria, entre otros. Pero, que esta información no representa un riesgo para los usuarios ya que no se identifica o expone información sensible.

En segundo lugar, al completar el proceso de escaneo de vulnerabilidades en las aplicaciones Android se pudo analizar los resultados de manera general; en el que se observan los permisos que se otorgan a estas aplicaciones y los riesgos que estas significan si un atacante toma control ya que tiene acceso a toda la información del usuario como se menciona a continuación: acceso a la ubicación, autenticación de cuentas del administrador y obtención de contraseñas, acceso y llamadas a los contactos, acceso a la cámara para fotos y videos lo cual permite recopilar datos que está pasando en ese momento, obtención de información del teléfono y servicios activos y acceso a audios registrados.
De igual manera, se pudo obtener las fallas y debilidades de seguridad que estas aplicaciones presentan y que pueden afectar a los usuarios que hacen uso de estas aplicaciones, como es que las aplicaciones emplean una base de datos sin cifrar, exposición de información confidencial, consultas SQL sin procesar lo cual puede comprometer la información y además puede generar una vulnerabilidad de inyección SQL, certificados SSL/TLS deshabilitados y el protocolo HTTP no proporciona encriptación por lo cual podría generar un ataque MITM.

Finalmente, al realizar este TFM se pudo evidenciar los riesgos que implican hacer uso de estos dispositivos inteligentes y aplicaciones móviles en nuestra vida cotidiana, los cuales están inmersos y son parte de nuestra vida diaria y que por desconocimiento o falta de interés o tiempo no somos conscientes de las mínimas configuraciones a tener

en cuenta o a un monitoreo constante como: revisar políticas de privacidad, proporcionar solo permisos estrictamente necesarios, eliminar de forma periódica información que estas aplicaciones recopilan y almacenan con el objetivo de mejorar y brindar un servicio más personalizado; por lo tanto, es importante tener estas consideraciones antes de emplearlos ya que estos dispositivos IoT forman parte de nuestro entorno en el hogar.

Conclusiones por capítulo

En el capítulo 2, se proporciona una base de investigación, para poder cumplir los objetivos planteados relacionados al TFM. Identificando y buscando información relacionada que nos sirva de guía y alcanzar los objetivos.

En el capítulo 3, se cumple el objetivo planteado de identificar los conceptos relacionados a los dispositivos IoT, revisión de los dispositivos inteligentes a analizar, las aplicaciones móviles que permiten controlar estos dispositivos, revisión acerca de la seguridad y privacidad que conllevan el uso de estos y su uso en la domótica.

En el capítulo 4, se da cumplimiento al objetivo planteado sobre diseñar y configurar un ambiente para obtener el tráfico de datos y además de buscar una herramienta que nos permita analizar las aplicaciones móviles que controlan los dispositivos inteligentes.

Por último, en el capítulo 5 se da cumplimiento a varios objetivos planteados como es el identificar los permisos y accesos que se otorgan a estas aplicaciones móviles, de igual manera se analizan estos hallazgos obtenidos tanto en el tráfico como en el escaneo de vulnerabilidades en los que se identifican los riesgos que implican y como afectan a la privacidad de los usuarios y se exponen las consideraciones a tener en cuenta por parte de los usuarios al momento de hacer uso de estos dispositivos para poder mitigar la exposición de la privacidad.

Metodología e impacto en sostenibilidad, ético-social y diversidad

En base a la metodología planteada esta ha sido muy importante ya que fue la guía para poder dar seguimiento y cumplimiento a todos los objetivos planteados.

Con lo referente a las tres dimensiones se pudo evidenciar que en la dimensión de sostenibilidad se tiene un impacto ecológico positivo ya que los dispositivos inteligentes ayudan a optimizar el consumo energético. En la dimensión de comportamiento ético y de responsabilidad social se identificó que estos asistentes inteligentes recolectan información de preferencias del usuario para poder proporcionar un servicio personalizado y de igual manera estos dispositivos indican que recolectan toda la información que el usuario mismo proporcione y otro punto a indicar es que en sus políticas de uso se mencionan que para un correcto funcionamiento de los dispositivos estos recolectan información para mejoras, atención personalizada, proporcionar sugerencias y publicidad enfocada. Además, indican que se cumplen las normativas de privacidad dispuestas en cada país con el fin de precautelar la información.

Discusión y Trabajos Futuros

A continuación, se presentan algunas limitaciones que se han tenido en el desarrollo del presente TFM. En primer lugar, cabe mencionar que por el tiempo limitado no se ha

podido recolectar por un periodo mayor el tráfico de red lo cual ha condicionado la información obtenida; obteniendo peticiones y respuestas generadas en los dispositivos al interactuar con preguntas previamente seleccionadas para generar un flujo de datos en los asistentes inteligentes. Por otro lado, para el escaneo de las vulnerabilidades se requiere el APK de las aplicaciones a analizar el cual se ha tomado una versión previa a las últimas versiones disponibles en Play Store. Finalmente, se ha limitado el análisis a un nivel de granularidad alto de cada uno de los permisos y vulnerabilidades expuestas en la sección 5.2.1 y 5.2.2.

Como trabajo futuro, se plantea importante analizar los riesgos y vulnerabilidades que implica tener los puertos abiertos; de igual manera una vez identificado es importante proporcionar el proceso para configurar los dispositivos y evitar tener puertos expuestos a atacantes que se aprovecharían de estas exposiciones para cometer actos malintencionados.

Por otro lado, es importante de igual manera poder realizar el análisis de las vulnerabilidades que tienen un nivel de criticidad normal y que no se han podido analizar y detallar por el alcance del TFM. Además, como trabajo futuro se plantea analizar las fallas de seguridad que tienen un nivel medio, bajo y de advertencia.

De igual manera, se requiere cubrir el proceso de análisis de: Comunicaciones externas de la aplicación móvil y Análisis de composición de software que han sido obtenidos en el proceso del escaneo de vulnerabilidades.

Finalmente, se requiere poder cubrir el escaneo de las vulnerabilidades para las aplicaciones disponibles en la plataforma iOS.

Glosario de términos

Asistente virtual inteligente: es un agente de software que puede realizar tareas o servicios para un individuo.

Siri, Cortana, Alexa, Google Assistant: asistentes virtuales controlados por voz.

Aplicación móvil: es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles.

Sistema operativo: conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación de software.

Privacidad: es el derecho que tiene cualquier usuario de la web a decidir cuáles datos personales desea compartir.

Internet de las cosas: proceso que permite conectar los elementos físicos cotidianos al Internet.

Vulnerabilidades: brechas de seguridad presentes en cualquier software.

Seguridad: protección de la información y, especialmente, al procesamiento que se hace de la misma, con el objetivo de evitar la manipulación de datos y procesos por personas no autorizadas.

Tráfico de la red/información: hace referencia a los datos que se desplazan por una red en un momento determinado.

Metodología: conjunto de métodos que se siguen en una investigación científica, un estudio o una exposición doctrinal.

Memoria final: es un documento que se realiza al finalizar el curso.

Android: sistema operativo móvil basado en el núcleo Linux y otro software de código abierto.

Domótica: implementaciones tecnológicas de control y automatización orientadas a espacios residenciales.

Red: es un conjunto de equipos conectados por medio de cables, señales, ondas o cualquier otro método de transporte de datos, que comparten información.

Sensor: son herramientas que detectan y responden a algún tipo de información del entorno físico.

Actuador: se encargan de convertir las señales eléctricas de control en otro tipo de señales o en señales eléctricas de mayor potencia.

Redes inalámbricas: permiten que los dispositivos permanezcan vinculados a una red y que puedan movilizarse sin estar conectados a un cable.

Redes cableadas: se comunican a través de cables de datos , generalmente basada en Ethernet.

Computación en la nube: es una tecnología que permite acceder remotamente, de cualquier lugar del mundo y en cualquier momento.

Companion app: aplicación móvil complementaria.

Metadata: datos que describen otros datos o "datos sobre datos".

Bibliografía

[1] I. Archives, «(1961) Shoebox - IBM Archives (78-013),» [En línea]. Available: https://mediacenter.ibm.com/media/(1961)+Shoebox+-+IBM+Archives+(78-013)/0_4m2ynnkk. [Último acceso: 15 octubre 2022].

[2] B. T. Lowerre, «The HARPY speech recognition system,» Pittsburgh, Pennsylvania, 1976.

[3] V. Tunon, M. Villarreal, J. Morán, G. Cubilla y S. GregPTY, «Evolución de la Interacción Humano Computadora,» [En línea]. Available: https://www.sutori.com/es/historia/evolucion-de-la-interaccion-humano-computadora--R9Djgnt1gDriEzjxSFWuSSmr.

[4] «Historia de la informática,» 3 diciembre 2012. [En línea]. Available: https://histinf.blogs.upv.es/2012/12/03/smartphones/. [Último acceso: 15 octubre 2022].

[5] E. Naone, «MIT TEchnology Review,» 24 febrero 2009. [En línea]. Available: http://www2.technologyreview.com/news/412191/tr10-intelligent-software-assistant/. [Último acceso: 7 octubre 2022].

[6] Equipo BLOG Grupo Cajamar, «Qué son los Asistentes Virtuales Inteligentes,» [En línea]. Available: https://blog.grupocajamar.com/que-son-los-asistentes-virtuales-inteligentes/. [Último acceso: 15 octubre 2022].

[7] R. Merkle, «Merkle,» 2 septiembre 2020. [En línea]. Available: https://www.merkle.com/es/es/blog/recopilar-datos-aplicacion-movil. [Último acceso: 7 octubre 2022].

[8] «Smart Speakers Global Market Report 2021: COVID-19 Growth and Change to 2030,» abril 2021. [En línea]. Available: https://www.researchandmarkets.com/reports/5315003/smart-speakers-global-market-report-2021-covid. [Último acceso: 16 octubre 2022].

[9] R. Fernández, «Statista,» 7 diciembre 2021. [En línea]. Available: https://es.statista.com/estadisticas/972995/asistentes-virtuales-en-uso-en-el-mundo/. [Último acceso: 17 octubre 2022].

[10] R. Carreras, «Datos y estadísticas sobre el crecimiento del mercado de asistentes de voz,» 31 enero 2019. [En línea]. Available: https://robertocarreras.es/datos-y-estadisticas-sobre-el-crecimiento-del-mercado-de-asistentes-de-voz/. [Último acceso: 17 octubre 2022].

[11] L. Baltazar, «Crehana,» 14 abril 2022. [En línea]. Available: https://www.crehana.com/blog/marketing-digital/asistentes-de-voz/. [Último acceso: 7 octubre 2022].

[12] R. Hernández Sampieri, C. Fernández Collado y M. d. P. Baptista Lucio, Metodología de la Investigación, Mc Graw Hill Education, 2014.

[13] Zainab H. Ali, Hesham A. Ali y Mahmoud M. Badawy, «Internet of Things (IoT): Definitions, Challenges and Recent Research Directions,» de *International Journal of Computer Applications*, 2015.

[14] McKinsey Insights, «What is the Internet of Things?,» de *McKinsey & Company*, New York, 2022.

[15] J. Gubbi, R. Buyya, S. Marusic y M. Palaniswami, «Internet of Things (IoT): A vision, architectural elements, and future directions,» de *Future Generation Computer Systems*, 2013.

[16] H. Ning, «Unit and Ubiquitous Internet of Things,» nº https://doi.org/10.1201/b14742, 2018.

[17] N. Ahmad, P. A. Laplante y J. F. DeFranco, «Life, IoT, and the Pursuit of Happiness,» *IT Professiona,* nº 10.1109/MITP.2019.2949944, pp. 4-7, 2020.

[18] J. Ritz y Z. Knaack, «Internet of Things,» *Technology and Engineering Teacher,* vol. 76, pp. 28-33, 2017.

[19] M. Latha Challa y K. Soujanya, «Secured smart mobile app for smart home environment,» *Dept. of Computer Science & Engineering, CMR College of Engineering & Technology,* nº https://doi.org/10.1016/j.matpr.2020.07.536, 2020.

[20] A. Sun, W. Gong, R. Shea y J. Liu, «A Castle of Glass: Leaky IoT Appliances in Modern Smart Homes,» *IEEE Wireless Communications,* vol. 25, nº 10.1109/MWC.2017.1800120, pp. 32 - 37, 2018.

[21] X. Zhang, S. Kuenzel, J.-R. Córdoba y C. Watkins, «Privacy-Functionality Trade-Off: A Privacy- Preserving Multi-Channel Smart Metering System,» nº 10.3390/en13123221, junio 2020.

[22] Y. Park, H. Choi, S. Cho y Y.-G. Kim, «Security Analysis of Smart Speaker: Security Attacks and Mitigation,» *Computers, Materials and Continua,* nº 10.32604/cmc.2019.08520, pp. 1075-1090, 2019.

[23] J. Palacín, E. Clotet, D. Martínez, J. Moreno y M. Tresanchez, «Automatic Supervision of Temperature, Humidity, and Luminance with an Assistant Personal Robot,» nº https://doi.org/10.1155/2017/1480401, 2017.

[24] W. A. J. Al-Areeqi, T. K. Kian, R. M. Ramli y S. N. Zubir, «Design and Fabrication of Smart Home With Internet of Things Enabled Automation System,» *IEEE Access,* nº 10.1109/ACCESS.2019.2942846, 2019.

[25] K. Ashton, «That 'Internet of Things' Thing,» 22 junio 2009. [En línea]. Available: https://www.rfidjournal.com/that-internet-of-things-thing. [Último acceso: 24 octubre 2022].

[26] Capgemini Research Institute, «Smart Talk: How organizations and consumers are embracing voice and chat assistants,» abril-mayo 2019. [En línea]. Available: https://www.capgemini.com/wp-content/uploads/2019/09/Report_Conversational-Interfaces-1.pdf. [Último acceso: octubre 2022].

[27] Innovación y Tecnología, «Dispositivos IoT,» 18 agosto 2020. [En línea]. Available: https://www.innovacion-tecnologia.com/iot/dispositivos-iot/. [Último acceso: octubre 2022].

[28] Nae, «Tendencias globales para el mercado de los asistentes virtuales,» 29 enero 2018. [En línea]. Available: https://nae.global/es/tendencias-globales-para-el-mercado-de-los-asistentes-virtuales/. [Último acceso: octubre 2022].

[29] D. Belanche y C. Flavián, «Retos de la adopción de los nuevos sistemas de recomendación basados en inteligencia artificial,» de *Marketing digital y big data*, 2022, pp. 67-83.

[30] V. Tiwari, M. Farukh Hashmi, A. Keskar y N. C. Shivaprakash , «Virtual home assistant for voice based controlling and scheduling with short speech speaker identification,» *Multimedia Tools and Applications,* vol. 79, nº https://doi.org/10.1007/s11042-018-6358-x, p. 5243–5268, 2020.

[31] D. Pal, M. Dawood Babakerkhell y X. Zhang, «Exploring the Determinants of Users' Continuance Usage Intention of Smart Voice Assistants,» *IEEE Access ,* vol. (Volume: 9), nº 10.1109/ACCESS.2021.3132399, pp. 162259 - 162275, 2021.

[32] J. Lemley, S. Bazrafkan y P. Corcoran, «Deep Learning for Consumer Devices and Services: Pushing the limits for machine learning, artificial intelligence, and computer vision.,» *IEEE Consumer Electronics Magazine,* vol. 6, nº 10.1109/MCE.2016.2640698, pp. 48 - 56, 2017.

[33] voicebot.ai, «MERCADO DE ASISTENTE VIRTUAL INTELIGENTE (IVA): CRECIMIENTO, TENDENCIAS, IMPACTO DE COVID-19 Y PRONÓSTICOS (2022 - 2027),» Voicebot Research, 2022.

[34] Amazon.com, Inc, «Alexa,» 2010-2022. [En línea]. Available: https://developer.amazon.com/es-ES/alexa. [Último acceso: octubre 2022].

[35] Google, «Speakers,» [En línea]. Available: https://assistant.google.com/. [Último acceso: octubre 2022].

[36] Apple Inc., «HomePod mini,» 2022. [En línea]. Available: https://www.apple.com/es/homepod-mini/. [Último acceso: 2022].

[37] N. Ponce, «Domótica en el hogar: las aplicaciones más útiles para una casa inteligente,» 25 marzo 2022. [En línea]. Available: https://tuapppara.com/aplicaciones/android/domotica/. [Último acceso: noviembre 2022].

[38] X. Wang, Y. Sun, S. Nanda y X. Wang, «Looking from the Mirror: Evaluating IoT Device Security through Mobile Companion Apps,» de *28th USENIX Security Symposium*, Santa Clara, CA, USA, 2019.

[39] R. H. Weber y R. Weber, «Internet of Things,» *Springer Berlin, Heidelberg,* nº https://doi.org/10.1007/978-3-642-11710-7.

[40] J. Pastor, «Qué es lo que realmente escuchan, almacenan y procesan Alexa, Assistant y Siri,» 30 noviembre 2018. [En línea]. Available: https://www.xataka.com/robotica-e-ia/que-que-realmente-escuchan-almacenan-procesan-alexa-assistant-siri. [Último acceso: octubre 2022].

[41] G. Xu, W. Wang, L. Jiao, X. Li, K. Liang, X. Zheng, W. Lian y H. Xian, «SoProtector: Safeguard Privacy for Native SO Files in Evolving Mobile IoT Applications,» *IEEE Internet of Things Journal,* vol. 7, nº 10.1109/JIOT.2019.2944006, pp. 2539 - 2552, 2019.

[42] M.-A. Youn, Y. Lim, K. Seo, H. Chung y S. Lee, «Forensic analysis for AI speaker with display Echo Show 2nd generation as a case study,» vol. 38, nº https://doi.org/10.1016/j.fsidi.2021.301130, 2021.

[43] D. J. Cook, J. C. Augusto y V. R. Jakkula, «Ambient intelligence: Technologies, applications, and opportunities,» *Pervasive and Mobile Computing,* vol. 5, nº https://doi.org/10.1016/j.pmcj.2009.04.001, pp. 277-298, 2009.

[44] G. D. Fentanes, Utilizando los asistentes virtuales de voz para promover la autonomía y el envejecimiento activo y saludable de las personas, Madrid: Universidad Politécnica Madrid, 2020.

[45] R. Barceló-Armada, I. Castell-Uroz y P. Barlet-Ros, «Amazon Alexa traffic traces,» *Computer Networks,* vol. 205, nº 108782, 2022.

[46] R. B. Armada, «Disección y análisis del tráfico de red de Amazon Alexa,» 2021.

[47] S. Abraham, «Descubra las amenazas de los puertos abiertos y mejore la seguridad con las herramientas de análisis de puertos,» 7 julio 2021. [En línea]. Available: https://blogs.manageengine.com/espanol/2021/07/07/descubrimiento-

amenazas-puertos-abiertos-htas-analisis-puertos.html. [Último acceso: diciembre 2022].

[48] Google, «Seguridad y privacidad de los datos de dispositivos que funcionan con Asistente,» [En línea]. Available: https://support.google.com/googlenest/answer/7072285?hl=es-419#zippy=%2Cd%C3%B3nde-se-guardan-mis-datos%2Cqu%C3%A9-informaci%C3%B3n-recopila-google-cuando-interact%C3%BAo-con-asistente-de-google%2Casistente-de-google-puede-obtener-informaci%C3%B3n-de-mis-. [Último acceso: 3 enero 2023].

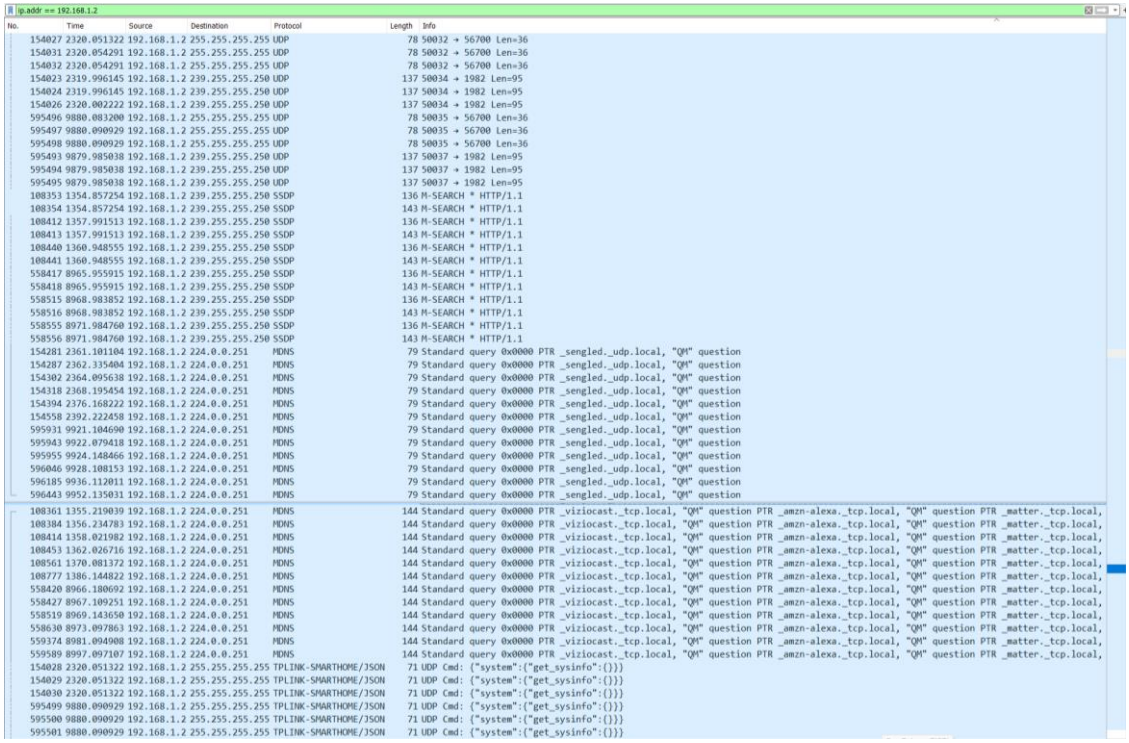
[49] Google, «Centros de datos,» [En línea]. Available: https://www.google.com/about/datacenters/locations/. [Último acceso: 3 enero 2023].

[50] Amazon, «Aviso de privacidad de Amazon,» 1 enero 2023. [En línea]. Available: https://www.amazon.com/-/es/gp/help/customer/display.html?nodeId=468496. [Último acceso: 3 enero 2023].

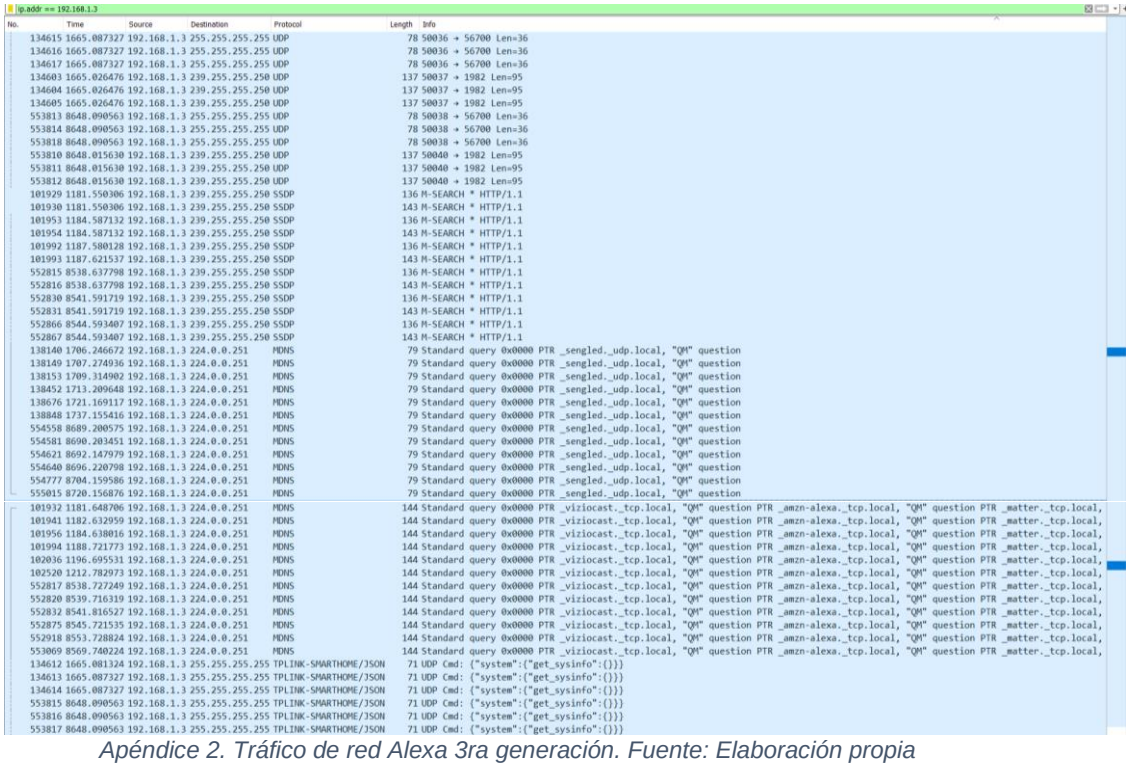
[51] Amazon, «Infraestructura global de AWS,» [En línea]. Available: https://aws.amazon.com/es/about-aws/global-infrastructure/?p=ngi&loc=1. [Último acceso: 3 enero 2023].

Apéndice

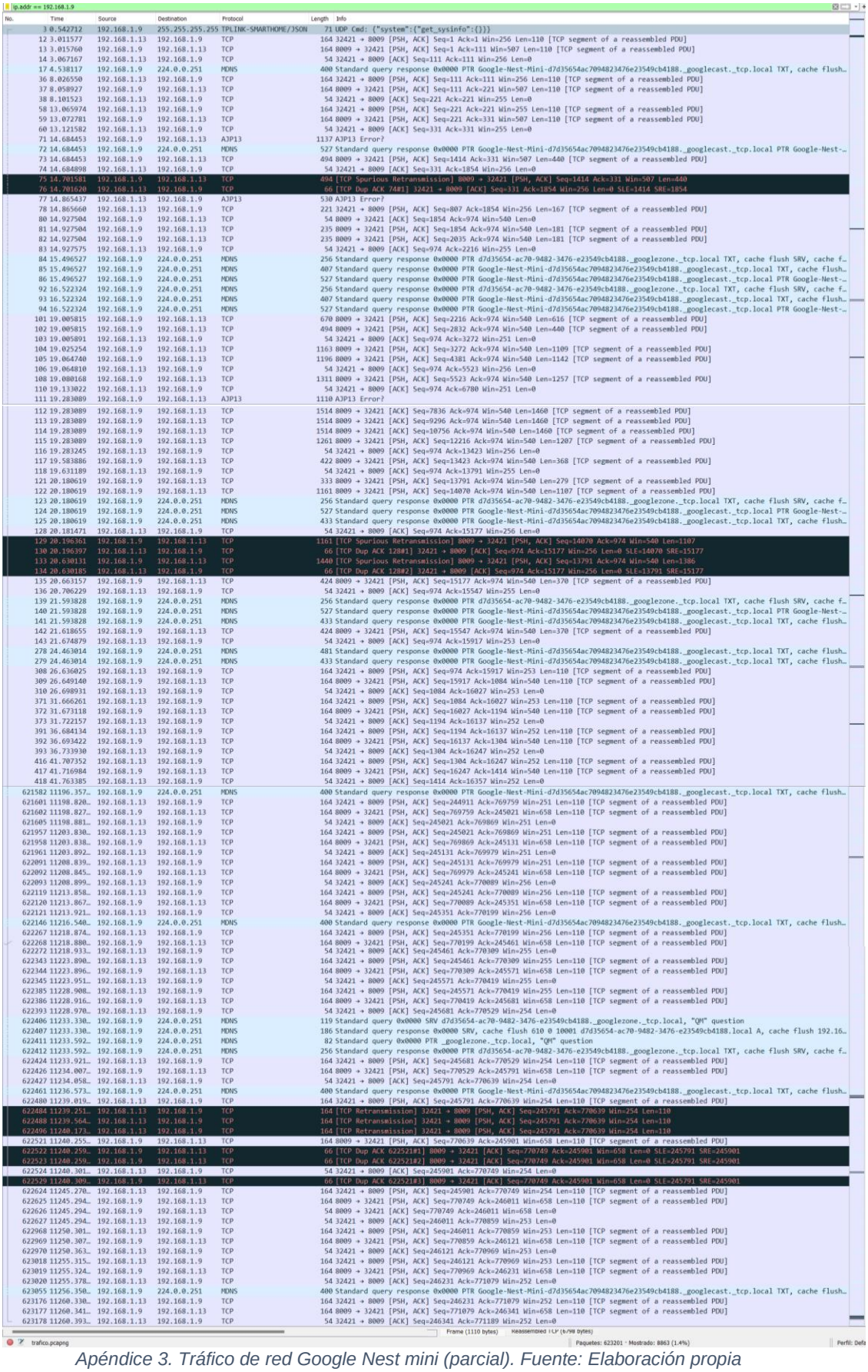
Apéndice A: Tráfico de red día 1



Apéndice 1. Tráfico de red Alexa 4ta generación. Fuente: Elaboración propia

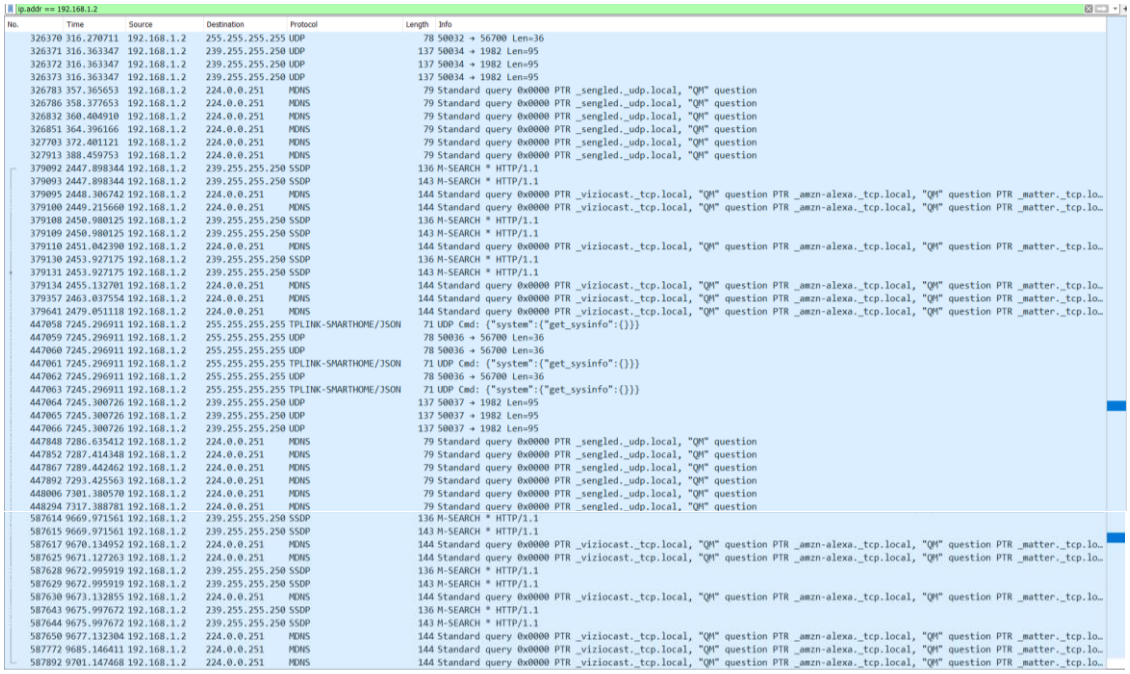


Apéndice 2. Tráfico de red Alexa 3ra generación. Fuente: Elaboración propia



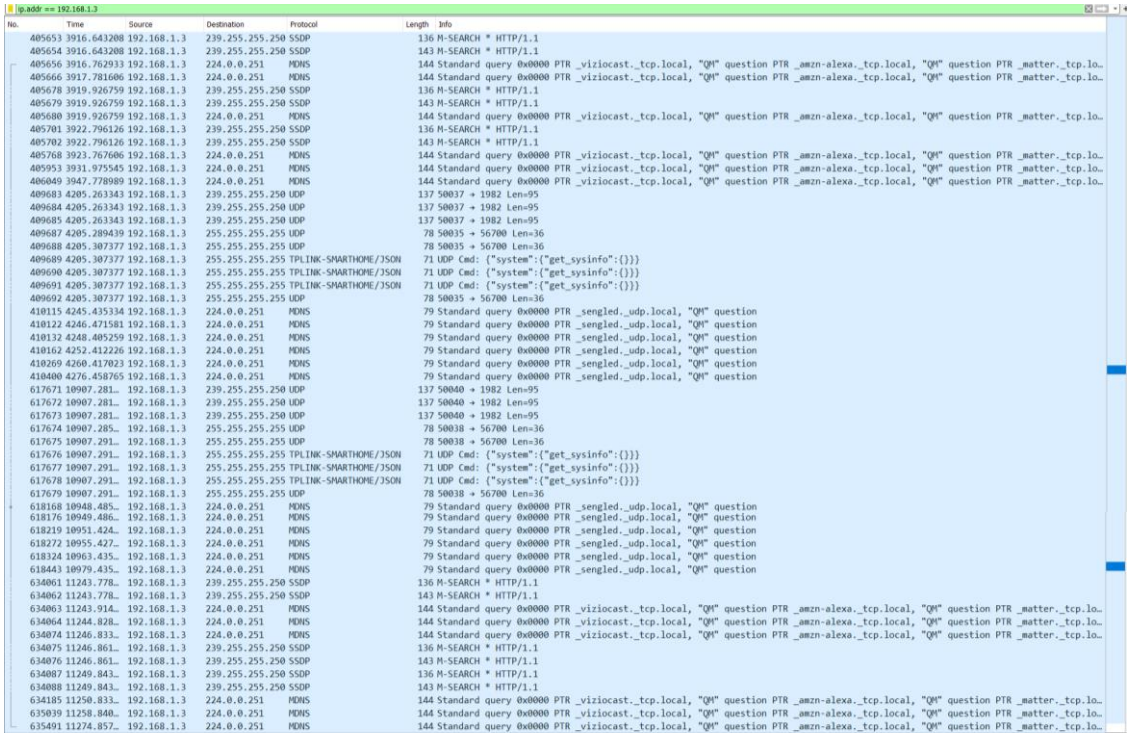
Apéndice 3. Tráfico de red Google Nest mini (parcial). Fuente: Elaboración propia

Apéndice B: Tráfico de red día 2



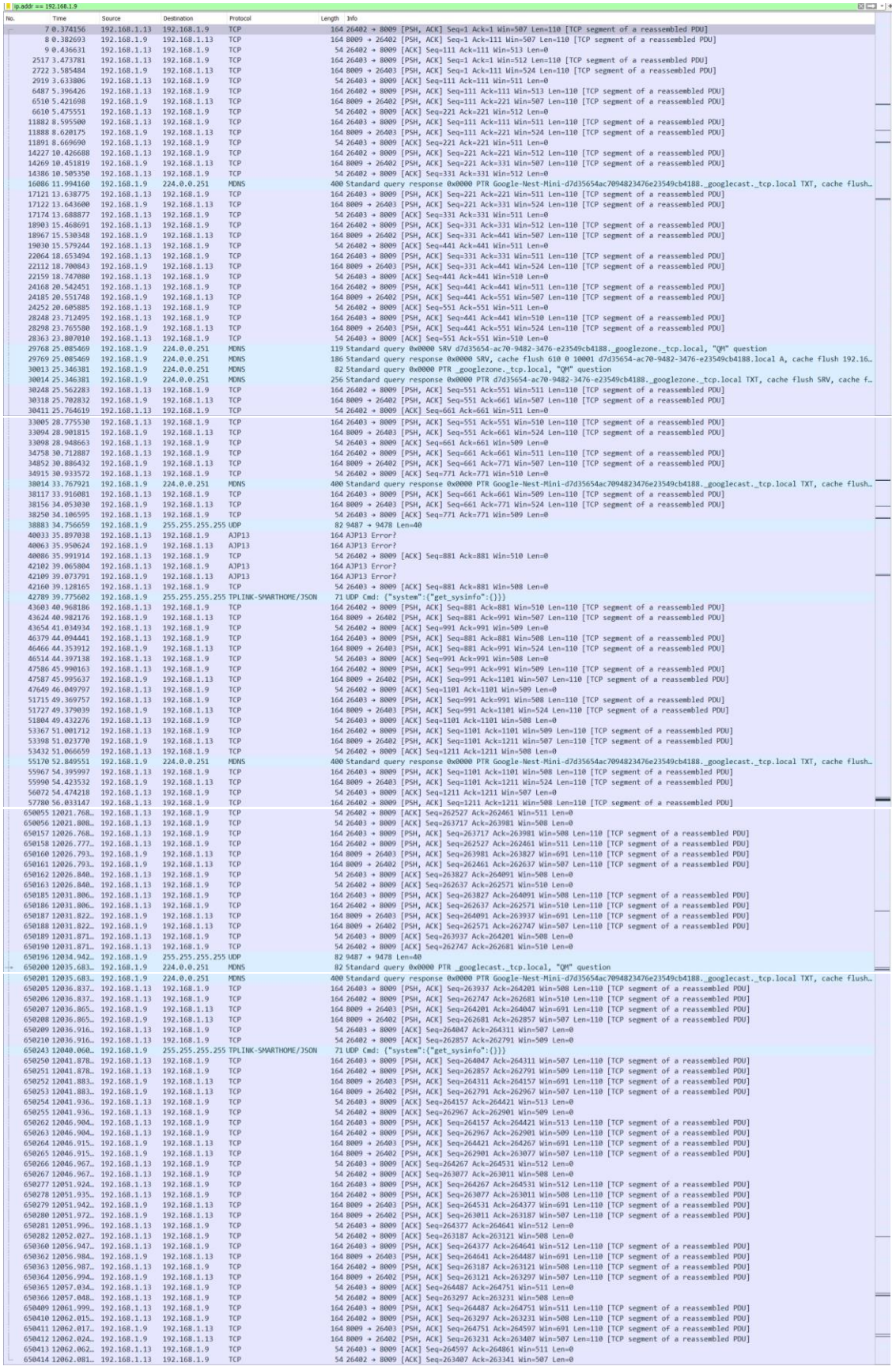
No.	Time	Source	Destination	Protocol	Length	Info
326370	316.270711	192.168.1.2	255.255.255.255	UDP	78	50032 → 56700 Len=36
326371	316.363347	192.168.1.2	239.255.255.250	UDP	137	50034 → 1982 Len=95
326372	316.363347	192.168.1.2	239.255.255.250	UDP	137	50034 → 1982 Len=95
326373	316.363347	192.168.1.2	239.255.255.250	UDP	137	50034 → 1982 Len=95
326783	357.365653	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
326786	358.377653	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
326832	360.404910	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
326851	364.396166	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
327703	372.401121	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
327913	388.459753	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
379092	2447.898344	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
379093	2447.898344	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
379095	2448.306742	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
379100	2449.215660	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
379108	2450.980125	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
379109	2450.980125	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
379110	2451.042390	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
379130	2453.927175	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
379131	2453.927175	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
379134	2455.132701	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
379357	2463.037554	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
379641	2479.051118	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
447058	7245.296911	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
447059	7245.296911	192.168.1.2	255.255.255.255	UDP	78	50036 → 56700 Len=36
447060	7245.296911	192.168.1.2	255.255.255.255	UDP	78	50036 → 56700 Len=36
447061	7245.296911	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
447062	7245.296911	192.168.1.2	255.255.255.255	UDP	78	50036 → 56700 Len=36
447063	7245.296911	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
447064	7245.300726	192.168.1.2	239.255.255.250	UDP	137	50037 → 1982 Len=95
447065	7245.300726	192.168.1.2	239.255.255.250	UDP	137	50037 → 1982 Len=95
447066	7245.300726	192.168.1.2	239.255.255.250	UDP	137	50037 → 1982 Len=95
447848	7286.635412	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
447852	7287.414348	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
447867	7289.442462	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
447892	7293.425563	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
448006	7301.380570	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
448294	7317.388781	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
587614	9669.971561	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
587615	9669.971561	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
587617	9670.134952	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
587625	9671.127263	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
587628	9672.995919	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
587629	9672.995919	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
587630	9673.132855	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
587643	9675.997672	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
587644	9675.997672	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
587650	9677.132304	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
587772	9685.146411	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
587892	9701.147468	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…

Apéndice 4. Tráfico de red Alexa 4ta generación. Fuente: Elaboración propia



No.	Time	Source	Destination	Protocol	Length	Info
405653	3916.643208	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
405654	3916.643208	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
405656	3916.762933	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
405666	3917.781606	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
405678	3919.926759	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
405679	3919.926759	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
405680	3919.926759	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
405701	3922.796126	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
405702	3922.796126	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
405768	3923.767606	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
405953	3931.975545	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
406049	3947.778989	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
409683	4205.263343	192.168.1.3	239.255.255.250	UDP	137	50037 → 1982 Len=95
409684	4205.263343	192.168.1.3	239.255.255.250	UDP	137	50037 → 1982 Len=95
409685	4205.263343	192.168.1.3	239.255.255.250	UDP	137	50037 → 1982 Len=95
409687	4205.289439	192.168.1.3	255.255.255.255	UDP	78	50035 → 56700 Len=36
409688	4205.307377	192.168.1.3	255.255.255.255	UDP	78	50035 → 56700 Len=36
409689	4205.307377	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
409690	4205.307377	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
409691	4205.307377	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
409692	4205.307377	192.168.1.3	255.255.255.255	UDP	78	50035 → 56700 Len=36
410115	4245.435334	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
410122	4246.471581	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
410132	4248.405259	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
410162	4252.412226	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
410269	4260.417023	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
410400	4276.458765	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
617671	10907.281…	192.168.1.3	239.255.255.250	UDP	137	50040 → 1982 Len=95
617672	10907.281…	192.168.1.3	239.255.255.250	UDP	137	50040 → 1982 Len=95
617673	10907.281…	192.168.1.3	239.255.255.250	UDP	137	50040 → 1982 Len=95
617674	10907.285…	192.168.1.3	255.255.255.255	UDP	78	50038 → 56700 Len=36
617675	10907.291…	192.168.1.3	255.255.255.255	UDP	78	50038 → 56700 Len=36
617676	10907.291…	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
617677	10907.291…	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
617678	10907.291…	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
617679	10907.291…	192.168.1.3	255.255.255.255	UDP	78	50038 → 56700 Len=36
618168	10948.485…	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
618176	10949.486…	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
618219	10951.424…	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
618272	10955.427…	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
618324	10963.435…	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
618443	10979.435…	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
634061	11243.778…	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
634062	11243.778…	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
634063	11243.914…	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
634064	11244.828…	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
634074	11246.833…	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
634075	11246.861…	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
634076	11246.861…	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
634087	11249.843…	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
634088	11249.843…	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
634185	11250.833…	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
635039	11258.840…	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
635491	11274.857…	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…

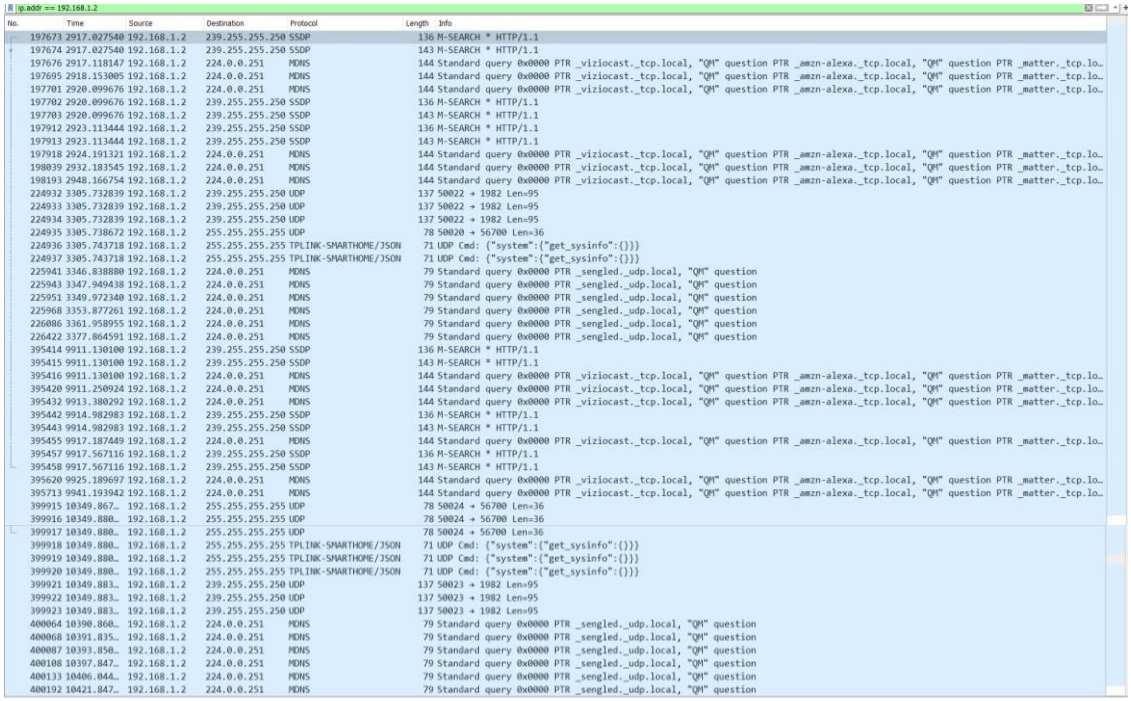
Apéndice 5. Tráfico de red Alexa 3ra generación. Fuente: Elaboración propia



No.	Time	Source	Destination	Protocol	Length	Info
7	0.374156	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=1 Ack=1 Win=507 Len=110 [TCP segment of a reassembled PDU]
8	0.382693	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=1 Ack=111 Win=507 Len=110 [TCP segment of a reassembled PDU]
9	0.436631	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=111 Ack=111 Win=513 Len=0
2517	3.473781	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=110 [TCP segment of a reassembled PDU]
2722	3.585484	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=1 Ack=111 Win=524 Len=110 [TCP segment of a reassembled PDU]
2919	3.633806	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=111 Ack=111 Win=511 Len=0
6487	5.396426	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=111 Ack=111 Win=513 Len=110 [TCP segment of a reassembled PDU]
6510	5.421698	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=111 Ack=221 Win=507 Len=110 [TCP segment of a reassembled PDU]
6610	5.475551	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=221 Ack=221 Win=512 Len=0
11882	8.595500	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=111 Ack=111 Win=511 Len=110 [TCP segment of a reassembled PDU]
11888	8.620175	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=111 Ack=221 Win=524 Len=110 [TCP segment of a reassembled PDU]
11891	8.669690	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=221 Ack=221 Win=511 Len=0
14227	10.426688	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=221 Ack=221 Win=512 Len=110 [TCP segment of a reassembled PDU]
14269	10.451819	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=221 Ack=331 Win=507 Len=110 [TCP segment of a reassembled PDU]
14386	10.505350	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=331 Ack=331 Win=512 Len=0
16086	11.994160	192.168.1.9	224.0.0.251	MDNS	400	Standard query response 0x0000 PTR Google-Nest-Mini-d7d35654ac7094823476e23549cb4188._googlecast._tcp.local TXT, cache flush…
17121	13.638775	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=221 Ack=221 Win=511 Len=110 [TCP segment of a reassembled PDU]
17122	13.643600	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=221 Ack=331 Win=524 Len=110 [TCP segment of a reassembled PDU]
17174	13.688877	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=331 Ack=331 Win=511 Len=0
18903	15.468691	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=331 Ack=331 Win=512 Len=110 [TCP segment of a reassembled PDU]
18967	15.530348	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=331 Ack=441 Win=507 Len=110 [TCP segment of a reassembled PDU]
19030	15.579244	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=441 Ack=441 Win=511 Len=0
22064	18.653494	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=331 Ack=331 Win=511 Len=110 [TCP segment of a reassembled PDU]
22112	18.700843	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=331 Ack=441 Win=524 Len=110 [TCP segment of a reassembled PDU]
22159	18.747080	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=441 Ack=441 Win=510 Len=0
24168	20.542451	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=441 Ack=441 Win=511 Len=110 [TCP segment of a reassembled PDU]
24185	20.551748	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=441 Ack=551 Win=507 Len=110 [TCP segment of a reassembled PDU]
24252	20.605885	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=551 Ack=551 Win=511 Len=0
28248	23.712495	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=441 Ack=441 Win=510 Len=110 [TCP segment of a reassembled PDU]
28298	23.765580	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=441 Ack=551 Win=524 Len=110 [TCP segment of a reassembled PDU]
28363	23.807010	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=551 Ack=551 Win=510 Len=0
29768	25.085469	192.168.1.9	224.0.0.251	MDNS	119	Standard query 0x0000 SRV d7d35654-ac70-9482-3476-e23549cb4188._googlezone._tcp.local, "QM" question
29769	25.085469	192.168.1.9	224.0.0.251	MDNS	186	Standard query response 0x0000 SRV, cache flush 610 0 10001 d7d35654-ac70-9482-3476-e23549cb4188.local A, cache flush 192.16…
30013	25.346381	192.168.1.9	224.0.0.251	MDNS	82	Standard query 0x0000 PTR _googlezone._tcp.local, "QM" question
30014	25.346381	192.168.1.9	224.0.0.251	MDNS	256	Standard query response 0x0000 PTR d7d35654-ac70-9482-3476-e23549cb4188._googlezone._tcp.local TXT, cache flush SRV, cache f…
30248	25.562283	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=551 Ack=551 Win=511 Len=110 [TCP segment of a reassembled PDU]
30318	25.702832	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=551 Ack=661 Win=507 Len=110 [TCP segment of a reassembled PDU]
30411	25.764619	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=661 Ack=661 Win=511 Len=0
33005	28.775530	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=551 Ack=551 Win=510 Len=110 [TCP segment of a reassembled PDU]
33094	28.901815	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=551 Ack=661 Win=524 Len=110 [TCP segment of a reassembled PDU]
33098	28.948663	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=661 Ack=661 Win=509 Len=0
34758	30.712887	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=661 Ack=661 Win=511 Len=110 [TCP segment of a reassembled PDU]
34852	30.886432	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=661 Ack=771 Win=507 Len=110 [TCP segment of a reassembled PDU]
34915	30.933572	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=771 Ack=771 Win=510 Len=0
38014	33.767921	192.168.1.9	224.0.0.251	MDNS	400	Standard query response 0x0000 PTR Google-Nest-Mini-d7d35654ac7094823476e23549cb4188._googlecast._tcp.local TXT, cache flush…
38117	33.916081	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=661 Ack=661 Win=509 Len=110 [TCP segment of a reassembled PDU]
38156	34.053030	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=661 Ack=771 Win=524 Len=110 [TCP segment of a reassembled PDU]
38250	34.106595	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=771 Ack=771 Win=509 Len=0
38883	34.756659	192.168.1.9	255.255.255.255	UDP	82	9487 → 9478 Len=40
40033	35.897038	192.168.1.13	192.168.1.9	AJP13	164	AJP13 Error?
40063	35.950624	192.168.1.9	192.168.1.13	AJP13	164	AJP13 Error?
40086	35.991914	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=881 Ack=881 Win=510 Len=0
42102	39.065804	192.168.1.13	192.168.1.9	AJP13	164	AJP13 Error?
42109	39.073791	192.168.1.9	192.168.1.13	AJP13	164	AJP13 Error?
42160	39.128165	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=881 Ack=881 Win=508 Len=0
42789	39.775602	192.168.1.9	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
43603	40.968186	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=881 Ack=881 Win=510 Len=110 [TCP segment of a reassembled PDU]
43624	40.982176	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=881 Ack=991 Win=507 Len=110 [TCP segment of a reassembled PDU]
43654	41.034934	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=991 Ack=991 Win=509 Len=0
46379	44.094441	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=881 Ack=881 Win=508 Len=110 [TCP segment of a reassembled PDU]
46466	44.353912	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=881 Ack=991 Win=524 Len=110 [TCP segment of a reassembled PDU]
46514	44.397138	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=991 Ack=991 Win=508 Len=0
47586	45.990163	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=991 Ack=991 Win=509 Len=110 [TCP segment of a reassembled PDU]
47587	45.995637	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=991 Ack=1101 Win=507 Len=110 [TCP segment of a reassembled PDU]
47649	46.049797	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=1101 Ack=1101 Win=509 Len=0
51715	49.369757	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=991 Ack=991 Win=508 Len=110 [TCP segment of a reassembled PDU]
51727	49.379039	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=991 Ack=1101 Win=524 Len=110 [TCP segment of a reassembled PDU]
51804	49.432276	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=1101 Ack=1101 Win=508 Len=0
53367	51.001712	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=1101 Ack=1101 Win=509 Len=110 [TCP segment of a reassembled PDU]
53398	51.023770	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=1101 Ack=1211 Win=507 Len=110 [TCP segment of a reassembled PDU]
53432	51.066659	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=1211 Ack=1211 Win=508 Len=0
55170	52.849551	192.168.1.9	224.0.0.251	MDNS	400	Standard query response 0x0000 PTR Google-Nest-Mini-d7d35654ac7094823476e23549cb4188._googlecast._tcp.local TXT, cache flush…
55967	54.395997	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=1101 Ack=1101 Win=508 Len=110 [TCP segment of a reassembled PDU]
55990	54.423532	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=1101 Ack=1211 Win=524 Len=110 [TCP segment of a reassembled PDU]
56072	54.474218	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=1211 Ack=1211 Win=507 Len=0
57780	56.033147	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=1211 Ack=1211 Win=508 Len=110 [TCP segment of a reassembled PDU]
650055	12021.768…	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=262527 Ack=262461 Win=511 Len=0
650056	12021.808…	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=263717 Ack=263981 Win=508 Len=0
650157	12026.768…	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=263717 Ack=263981 Win=508 Len=110 [TCP segment of a reassembled PDU]
650158	12026.777…	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=262527 Ack=262461 Win=511 Len=110 [TCP segment of a reassembled PDU]
650160	12026.793…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=263981 Ack=263827 Win=691 Len=110 [TCP segment of a reassembled PDU]
650161	12026.793…	192.168.1.13	192.168.1.9	TCP	164	8009 → 26402 [PSH, ACK] Seq=262461 Ack=262637 Win=507 Len=110 [TCP segment of a reassembled PDU]
650162	12026.840…	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=263827 Ack=264091 Win=508 Len=0
650163	12026.840…	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=262637 Ack=262571 Win=510 Len=0
650185	12031.806…	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=263827 Ack=264091 Win=508 Len=110 [TCP segment of a reassembled PDU]
650186	12031.806…	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=262637 Ack=262571 Win=510 Len=110 [TCP segment of a reassembled PDU]
650187	12031.822…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=264091 Ack=263937 Win=691 Len=110 [TCP segment of a reassembled PDU]
650188	12031.822…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=262571 Ack=262747 Win=507 Len=110 [TCP segment of a reassembled PDU]
650189	12031.871…	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=263937 Ack=264201 Win=508 Len=0
650190	12031.871…	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=262747 Ack=262681 Win=510 Len=0
650196	12034.942…	192.168.1.9	255.255.255.255	UDP	82	9487 → 9478 Len=40
650200	12035.683…	192.168.1.9	224.0.0.251	MDNS	82	Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
650201	12035.683…	192.168.1.9	224.0.0.251	MDNS	400	Standard query response 0x0000 PTR Google-Nest-Mini-d7d35654ac7094823476e23549cb4188._googlecast._tcp.local TXT, cache flush…
650205	12036.837…	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=263937 Ack=264201 Win=508 Len=110 [TCP segment of a reassembled PDU]
650206	12036.837…	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=262747 Ack=262681 Win=510 Len=110 [TCP segment of a reassembled PDU]
650207	12036.865…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=264201 Ack=264047 Win=691 Len=110 [TCP segment of a reassembled PDU]
650208	12036.865…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=262681 Ack=262857 Win=507 Len=110 [TCP segment of a reassembled PDU]
650209	12036.916…	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=264047 Ack=264311 Win=507 Len=0
650210	12036.916…	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=262857 Ack=262791 Win=509 Len=0
650243	12040.060…	192.168.1.9	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
650250	12041.878…	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=264047 Ack=264311 Win=507 Len=110 [TCP segment of a reassembled PDU]
650251	12041.878…	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=262857 Ack=262791 Win=509 Len=110 [TCP segment of a reassembled PDU]
650252	12041.883…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=264311 Ack=264157 Win=691 Len=110 [TCP segment of a reassembled PDU]
650253	12041.883…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=262791 Ack=262967 Win=507 Len=110 [TCP segment of a reassembled PDU]
650254	12041.936…	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=264157 Ack=264421 Win=513 Len=0
650255	12041.936…	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=262967 Ack=262901 Win=509 Len=0
650262	12046.904…	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=264157 Ack=264421 Win=513 Len=110 [TCP segment of a reassembled PDU]
650263	12046.904…	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=262967 Ack=262901 Win=509 Len=110 [TCP segment of a reassembled PDU]
650264	12046.915…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=264421 Ack=264267 Win=691 Len=110 [TCP segment of a reassembled PDU]
650265	12046.915…	192.168.1.13	192.168.1.9	TCP	164	8009 → 26402 [PSH, ACK] Seq=262901 Ack=263077 Win=507 Len=110 [TCP segment of a reassembled PDU]
650266	12046.967…	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=264267 Ack=264531 Win=512 Len=0
650267	12046.967…	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=263077 Ack=263011 Win=508 Len=0
650277	12051.924…	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=264267 Ack=264531 Win=512 Len=110 [TCP segment of a reassembled PDU]
650278	12051.935…	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=263077 Ack=263011 Win=508 Len=110 [TCP segment of a reassembled PDU]
650279	12051.942…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=264531 Ack=264377 Win=691 Len=110 [TCP segment of a reassembled PDU]
650280	12051.972…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=263011 Ack=263187 Win=507 Len=110 [TCP segment of a reassembled PDU]
650281	12051.996…	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=264377 Ack=264641 Win=512 Len=0
650282	12052.027…	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=263187 Ack=263121 Win=508 Len=0
650360	12056.947…	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=264377 Ack=264641 Win=512 Len=110 [TCP segment of a reassembled PDU]
650362	12056.984…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=264641 Ack=264487 Win=691 Len=110 [TCP segment of a reassembled PDU]
650363	12056.987…	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=263187 Ack=263121 Win=508 Len=110 [TCP segment of a reassembled PDU]
650364	12056.994…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=263121 Ack=263297 Win=507 Len=110 [TCP segment of a reassembled PDU]
650365	12057.034…	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=264487 Ack=264751 Win=511 Len=0
650366	12057.048…	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=263297 Ack=263231 Win=508 Len=0
650409	12061.999…	192.168.1.13	192.168.1.9	TCP	164	26403 → 8009 [PSH, ACK] Seq=264487 Ack=264751 Win=511 Len=110 [TCP segment of a reassembled PDU]
650410	12062.015…	192.168.1.13	192.168.1.9	TCP	164	26402 → 8009 [PSH, ACK] Seq=263297 Ack=263231 Win=508 Len=110 [TCP segment of a reassembled PDU]
650411	12062.017…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26403 [PSH, ACK] Seq=264751 Ack=264597 Win=691 Len=110 [TCP segment of a reassembled PDU]
650412	12062.024…	192.168.1.9	192.168.1.13	TCP	164	8009 → 26402 [PSH, ACK] Seq=263231 Ack=263407 Win=507 Len=110 [TCP segment of a reassembled PDU]
650413	12062.062…	192.168.1.13	192.168.1.9	TCP	54	26403 → 8009 [ACK] Seq=264597 Ack=264861 Win=511 Len=0
650414	12062.081…	192.168.1.13	192.168.1.9	TCP	54	26402 → 8009 [ACK] Seq=263407 Ack=263341 Win=507 Len=0

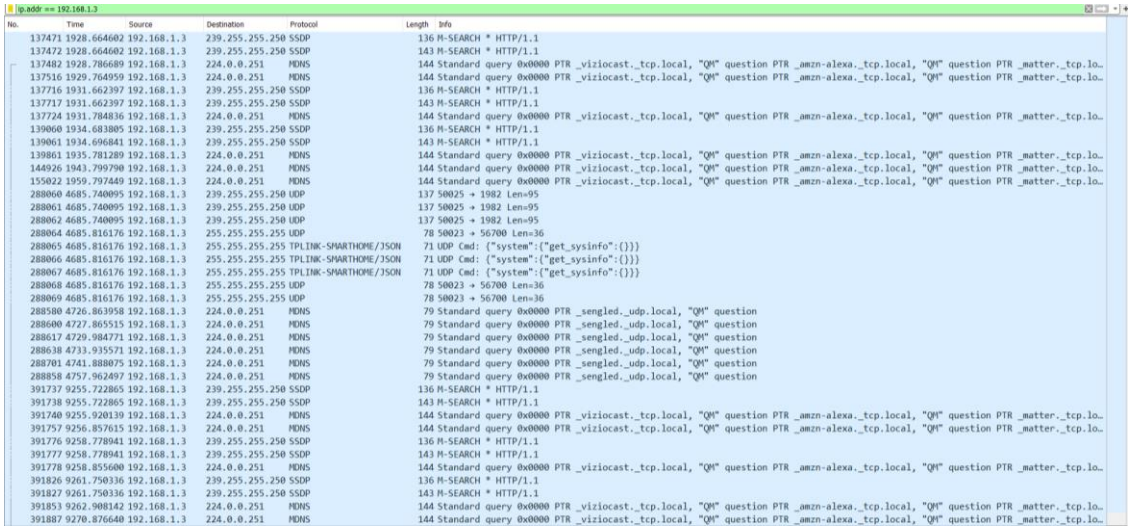
Apéndice 6. Tráfico de red Google Nest mini (parcial). Fuente: Elaboración propia

Apéndice C: Tráfico de red día 3



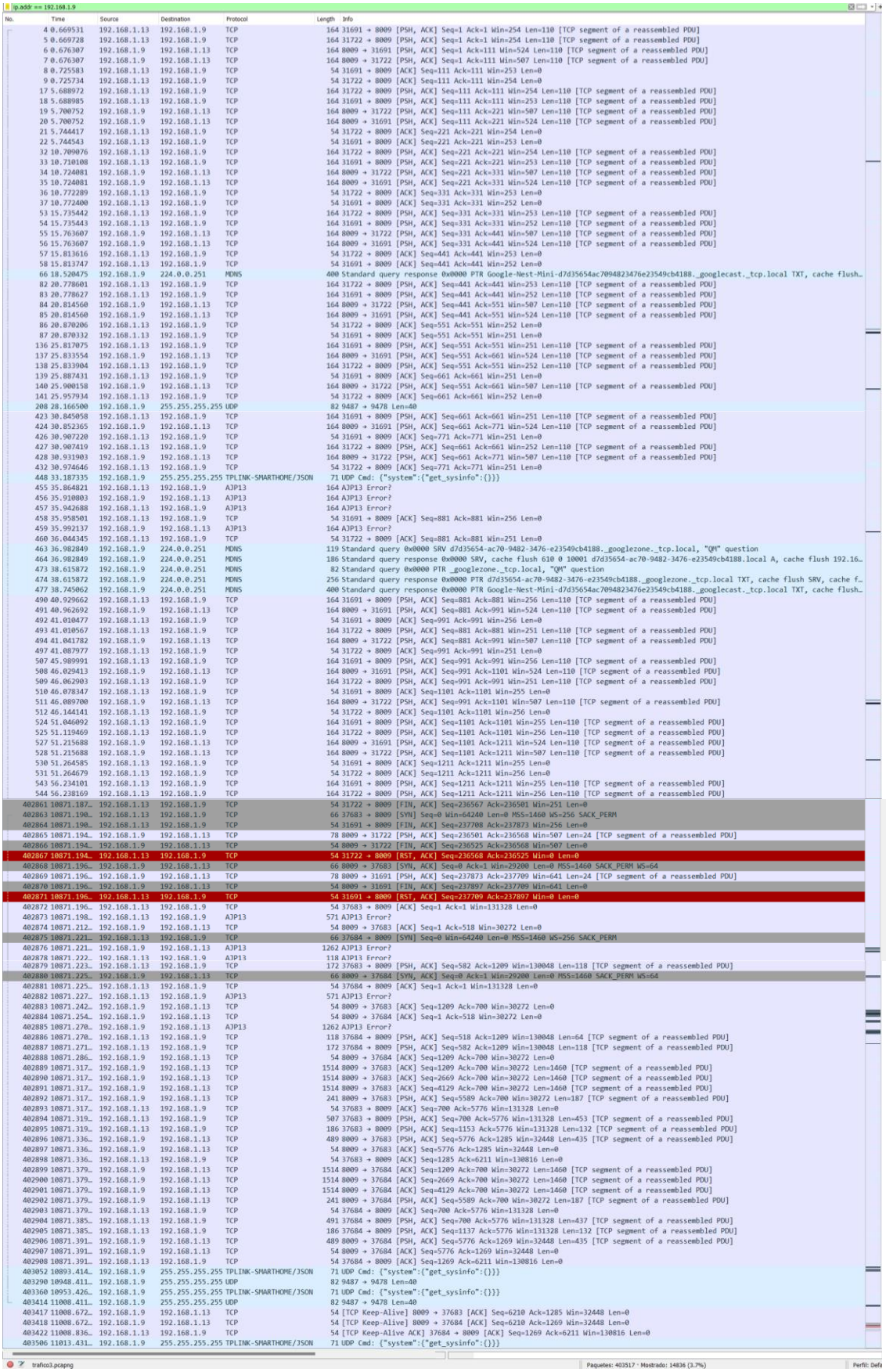
No.	Time	Source	Destination	Protocol	Length	Info
197673	2917.027540	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
197674	2917.027540	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
197676	2917.118147	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
197695	2918.153005	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
197701	2920.099676	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
197702	2920.099676	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
197703	2920.099676	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
197912	2923.113444	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
197913	2923.113444	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
197918	2924.191321	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
198039	2932.183545	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
198193	2948.166754	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
224932	3305.732839	192.168.1.2	239.255.255.250	UDP	137	50022 → 1982 Len=95
224933	3305.732839	192.168.1.2	239.255.255.250	UDP	137	50022 → 1982 Len=95
224934	3305.732839	192.168.1.2	239.255.255.250	UDP	137	50022 → 1982 Len=95
224935	3305.738672	192.168.1.2	255.255.255.255	UDP	78	50020 → 56700 Len=36
224936	3305.743718	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
224937	3305.743718	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
225941	3346.838880	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
225943	3347.949438	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
225951	3349.972340	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
225968	3353.877261	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
226086	3361.958955	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
226422	3377.864591	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
395414	9911.130100	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
395415	9911.130100	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
395416	9911.130100	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
395420	9911.250924	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
395432	9913.380292	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
395442	9914.982983	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
395443	9914.982983	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
395455	9917.187449	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
395457	9917.567116	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
395458	9917.567116	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
395620	9925.189697	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
395713	9941.193942	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
399915	10349.867…	192.168.1.2	255.255.255.255	UDP	78	50024 → 56700 Len=36
399916	10349.880…	192.168.1.2	255.255.255.255	UDP	78	50024 → 56700 Len=36
399917	10349.880…	192.168.1.2	255.255.255.255	UDP	78	50024 → 56700 Len=36
399918	10349.880…	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
399919	10349.880…	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
399920	10349.880…	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
399921	10349.883…	192.168.1.2	239.255.255.250	UDP	137	50023 → 1982 Len=95
399922	10349.883…	192.168.1.2	239.255.255.250	UDP	137	50023 → 1982 Len=95
399923	10349.883…	192.168.1.2	239.255.255.250	UDP	137	50023 → 1982 Len=95
400064	10390.860…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
400068	10391.835…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
400087	10393.850…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
400108	10397.847…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
400133	10406.044…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
400192	10421.847…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question

Apéndice 7. Tráfico de red Alexa 4ta generación. Fuente: Elaboración propia



No.	Time	Source	Destination	Protocol	Length	Info
137471	1928.664602	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
137472	1928.664602	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
137482	1928.786689	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
137516	1929.764959	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
137716	1931.662397	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
137717	1931.662397	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
137724	1931.784836	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
139060	1934.683805	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
139061	1934.696841	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
139861	1935.781289	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
144926	1943.799790	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
155022	1959.797449	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
288060	4685.740095	192.168.1.3	239.255.255.250	UDP	137	50025 → 1982 Len=95
288061	4685.740095	192.168.1.3	239.255.255.250	UDP	137	50025 → 1982 Len=95
288062	4685.740095	192.168.1.3	239.255.255.250	UDP	137	50025 → 1982 Len=95
288064	4685.816176	192.168.1.3	255.255.255.255	UDP	78	50023 → 56700 Len=36
288065	4685.816176	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
288066	4685.816176	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
288067	4685.816176	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
288068	4685.816176	192.168.1.3	255.255.255.255	UDP	78	50023 → 56700 Len=36
288069	4685.816176	192.168.1.3	255.255.255.255	UDP	78	50023 → 56700 Len=36
288580	4726.863958	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
288600	4727.865515	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
288617	4729.984771	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
288638	4733.935571	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
288701	4741.888075	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
288858	4757.962497	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
391737	9255.722865	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
391738	9255.722865	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
391740	9255.920139	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
391757	9256.857615	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
391776	9258.778941	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
391777	9258.778941	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
391778	9258.855600	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
391826	9261.750336	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
391827	9261.750336	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
391853	9262.908142	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
391887	9270.876640	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…

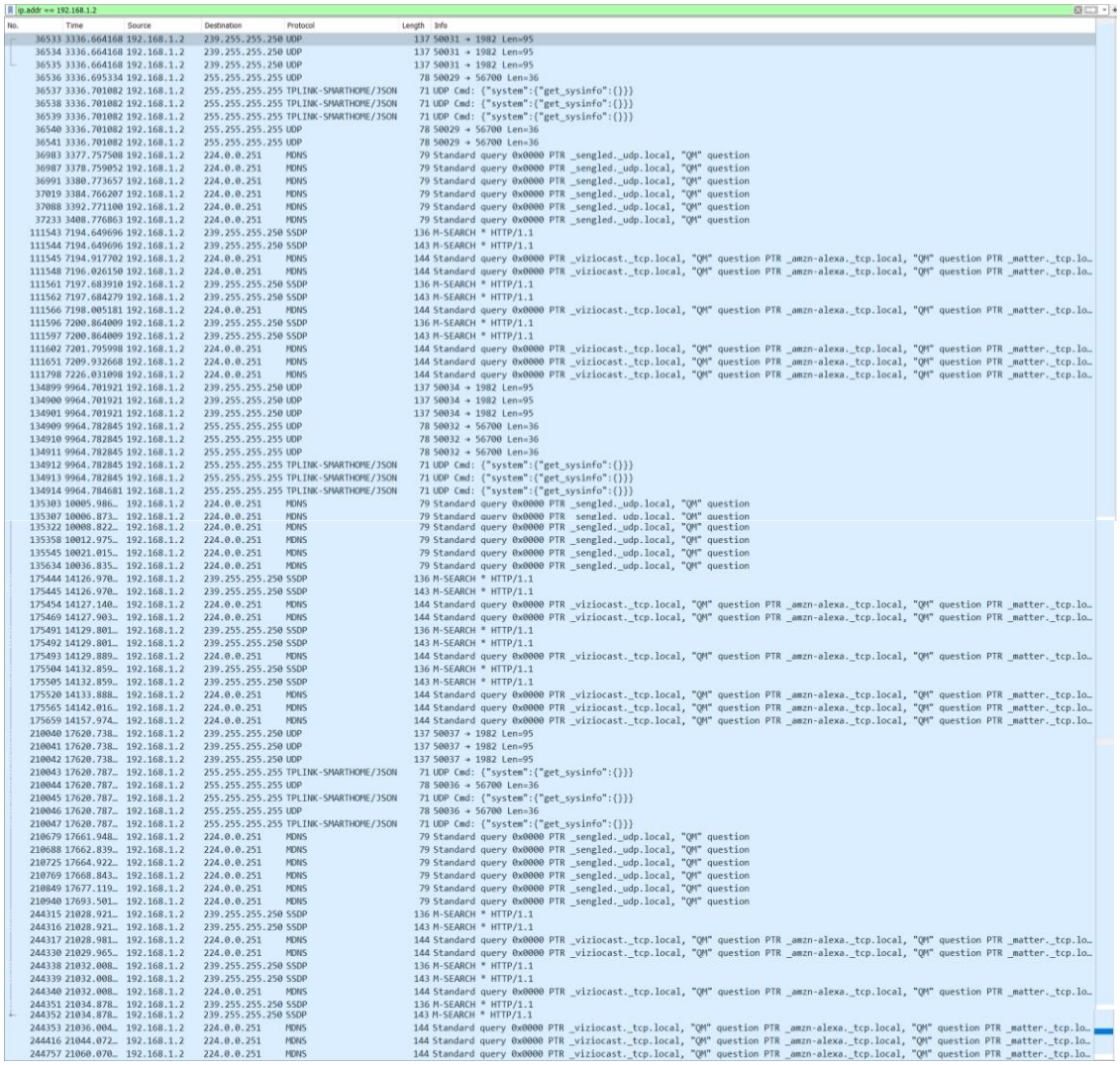
Apéndice 8. Tráfico de red Alexa 3ra generación. Fuente: Elaboración propia



No.	Time	Source	Destination	Protocol	Length	Info
4	0.669531	192.168.1.13	192.168.1.9	TCP	164	31691 → 8009 [PSH, ACK] Seq=1 Ack=1 Win=254 Len=110 [TCP segment of a reassembled PDU]
5	0.669728	192.168.1.13	192.168.1.9	TCP	164	31722 → 8009 [PSH, ACK] Seq=1 Ack=1 Win=254 Len=110 [TCP segment of a reassembled PDU]
6	0.676307	192.168.1.9	192.168.1.13	TCP	164	8009 → 31691 [PSH, ACK] Seq=1 Ack=111 Win=524 Len=110 [TCP segment of a reassembled PDU]
7	0.676307	192.168.1.9	192.168.1.13	TCP	164	8009 → 31722 [PSH, ACK] Seq=1 Ack=111 Win=507 Len=110 [TCP segment of a reassembled PDU]
8	0.725583	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [ACK] Seq=111 Ack=111 Win=253 Len=0
9	0.725734	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [ACK] Seq=111 Ack=111 Win=254 Len=0
17	5.688972	192.168.1.13	192.168.1.9	TCP	164	31722 → 8009 [PSH, ACK] Seq=111 Ack=111 Win=254 Len=110 [TCP segment of a reassembled PDU]
18	5.688985	192.168.1.13	192.168.1.9	TCP	164	31691 → 8009 [PSH, ACK] Seq=111 Ack=111 Win=253 Len=110 [TCP segment of a reassembled PDU]
19	5.700752	192.168.1.9	192.168.1.13	TCP	164	8009 → 31722 [PSH, ACK] Seq=111 Ack=221 Win=507 Len=110 [TCP segment of a reassembled PDU]
20	5.700752	192.168.1.9	192.168.1.13	TCP	164	8009 → 31691 [PSH, ACK] Seq=111 Ack=221 Win=524 Len=110 [TCP segment of a reassembled PDU]
21	5.744417	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [ACK] Seq=221 Ack=221 Win=254 Len=0
22	5.744543	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [ACK] Seq=221 Ack=221 Win=253 Len=0
32	10.709076	192.168.1.13	192.168.1.9	TCP	164	31722 → 8009 [PSH, ACK] Seq=221 Ack=221 Win=254 Len=110 [TCP segment of a reassembled PDU]
33	10.710108	192.168.1.13	192.168.1.9	TCP	164	31691 → 8009 [PSH, ACK] Seq=221 Ack=221 Win=253 Len=110 [TCP segment of a reassembled PDU]
34	10.724081	192.168.1.9	192.168.1.13	TCP	164	8009 → 31722 [PSH, ACK] Seq=221 Ack=331 Win=507 Len=110 [TCP segment of a reassembled PDU]
35	10.724081	192.168.1.9	192.168.1.13	TCP	164	8009 → 31691 [PSH, ACK] Seq=221 Ack=331 Win=524 Len=110 [TCP segment of a reassembled PDU]
36	10.772289	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [ACK] Seq=331 Ack=331 Win=253 Len=0
37	10.772400	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [ACK] Seq=331 Ack=331 Win=252 Len=0
53	15.735442	192.168.1.13	192.168.1.9	TCP	164	31722 → 8009 [PSH, ACK] Seq=331 Ack=331 Win=253 Len=110 [TCP segment of a reassembled PDU]
54	15.735443	192.168.1.13	192.168.1.9	TCP	164	31691 → 8009 [PSH, ACK] Seq=331 Ack=331 Win=252 Len=110 [TCP segment of a reassembled PDU]
55	15.763607	192.168.1.9	192.168.1.13	TCP	164	8009 → 31722 [PSH, ACK] Seq=331 Ack=441 Win=507 Len=110 [TCP segment of a reassembled PDU]
56	15.763607	192.168.1.9	192.168.1.13	TCP	164	8009 → 31691 [PSH, ACK] Seq=331 Ack=441 Win=524 Len=110 [TCP segment of a reassembled PDU]
57	15.813616	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [ACK] Seq=441 Ack=441 Win=253 Len=0
58	15.813747	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [ACK] Seq=441 Ack=441 Win=252 Len=0
66	18.520475	192.168.1.9	224.0.0.251	MDNS	400	Standard query response 0x0000 PTR Google-Nest-Mini-d7d35654ac7094823476e23549cb4188._googlecast._tcp.local TXT, cache flush…
82	20.778601	192.168.1.13	192.168.1.9	TCP	164	31722 → 8009 [PSH, ACK] Seq=441 Ack=441 Win=253 Len=110 [TCP segment of a reassembled PDU]
83	20.778627	192.168.1.13	192.168.1.9	TCP	164	31691 → 8009 [PSH, ACK] Seq=441 Ack=441 Win=252 Len=110 [TCP segment of a reassembled PDU]
84	20.814560	192.168.1.9	192.168.1.13	TCP	164	8009 → 31722 [PSH, ACK] Seq=441 Ack=551 Win=507 Len=110 [TCP segment of a reassembled PDU]
85	20.814560	192.168.1.9	192.168.1.13	TCP	164	8009 → 31691 [PSH, ACK] Seq=441 Ack=551 Win=524 Len=110 [TCP segment of a reassembled PDU]
86	20.870206	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [ACK] Seq=551 Ack=551 Win=252 Len=0
87	20.870332	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [ACK] Seq=551 Ack=551 Win=251 Len=0
136	25.817075	192.168.1.13	192.168.1.9	TCP	164	31691 → 8009 [PSH, ACK] Seq=551 Ack=551 Win=251 Len=110 [TCP segment of a reassembled PDU]
137	25.833554	192.168.1.9	192.168.1.13	TCP	164	8009 → 31691 [PSH, ACK] Seq=551 Ack=661 Win=524 Len=110 [TCP segment of a reassembled PDU]
138	25.833904	192.168.1.13	192.168.1.9	TCP	164	31722 → 8009 [PSH, ACK] Seq=551 Ack=551 Win=252 Len=110 [TCP segment of a reassembled PDU]
139	25.887431	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [ACK] Seq=661 Ack=661 Win=251 Len=0
140	25.900158	192.168.1.9	192.168.1.13	TCP	164	8009 → 31722 [PSH, ACK] Seq=551 Ack=661 Win=507 Len=110 [TCP segment of a reassembled PDU]
141	25.957934	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [ACK] Seq=661 Ack=661 Win=252 Len=0
208	28.166500	192.168.1.9	255.255.255.255	UDP	82	9487 → 9478 Len=40
423	30.845058	192.168.1.13	192.168.1.9	TCP	164	31691 → 8009 [PSH, ACK] Seq=661 Ack=661 Win=251 Len=110 [TCP segment of a reassembled PDU]
424	30.852365	192.168.1.9	192.168.1.13	TCP	164	8009 → 31691 [PSH, ACK] Seq=661 Ack=771 Win=524 Len=110 [TCP segment of a reassembled PDU]
426	30.907220	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [ACK] Seq=771 Ack=771 Win=251 Len=0
427	30.907419	192.168.1.13	192.168.1.9	TCP	164	31722 → 8009 [PSH, ACK] Seq=661 Ack=661 Win=252 Len=110 [TCP segment of a reassembled PDU]
428	30.931903	192.168.1.9	192.168.1.13	TCP	164	8009 → 31722 [PSH, ACK] Seq=661 Ack=771 Win=507 Len=110 [TCP segment of a reassembled PDU]
432	30.974646	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [ACK] Seq=771 Ack=771 Win=251 Len=0
448	33.187335	192.168.1.9	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
455	35.864821	192.168.1.13	192.168.1.9	AJP13	164	AJP13 Error?
456	35.910803	192.168.1.9	192.168.1.13	AJP13	164	AJP13 Error?
457	35.942688	192.168.1.13	192.168.1.9	AJP13	164	AJP13 Error?
458	35.958501	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [ACK] Seq=881 Ack=881 Win=256 Len=0
459	35.992137	192.168.1.9	192.168.1.13	AJP13	164	AJP13 Error?
460	36.044345	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [ACK] Seq=881 Ack=881 Win=251 Len=0
463	36.982849	192.168.1.9	224.0.0.251	MDNS	119	Standard query 0x0000 SRV d7d35654-ac70-9482-3476-e23549cb4188._googlezone._tcp.local, "QM" question
464	36.982849	192.168.1.9	224.0.0.251	MDNS	186	Standard query response 0x0000 SRV, cache flush 610 0 10001 d7d35654-ac70-9482-3476-e23549cb4188.local A, cache flush 192.16…
473	38.615872	192.168.1.9	224.0.0.251	MDNS	82	Standard query 0x0000 PTR _googlezone._tcp.local, "QM" question
474	38.615872	192.168.1.9	224.0.0.251	MDNS	256	Standard query response 0x0000 PTR d7d35654-ac70-9482-3476-e23549cb4188._googlezone._tcp.local TXT, cache flush SRV, cache f…
477	38.745062	192.168.1.9	224.0.0.251	MDNS	400	Standard query response 0x0000 PTR Google-Nest-Mini-d7d35654ac7094823476e23549cb4188._googlecast._tcp.local TXT, cache flush…
490	40.929662	192.168.1.13	192.168.1.9	TCP	164	31691 → 8009 [PSH, ACK] Seq=881 Ack=881 Win=256 Len=110 [TCP segment of a reassembled PDU]
491	40.962692	192.168.1.9	192.168.1.13	TCP	164	8009 → 31691 [PSH, ACK] Seq=881 Ack=991 Win=524 Len=110 [TCP segment of a reassembled PDU]
492	41.010477	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [ACK] Seq=991 Ack=991 Win=256 Len=0
493	41.010567	192.168.1.13	192.168.1.9	TCP	164	31722 → 8009 [PSH, ACK] Seq=881 Ack=881 Win=251 Len=110 [TCP segment of a reassembled PDU]
494	41.041782	192.168.1.9	192.168.1.13	TCP	164	8009 → 31722 [PSH, ACK] Seq=881 Ack=991 Win=507 Len=110 [TCP segment of a reassembled PDU]
497	41.087977	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [ACK] Seq=991 Ack=991 Win=251 Len=0
507	45.989991	192.168.1.13	192.168.1.9	TCP	164	31691 → 8009 [PSH, ACK] Seq=991 Ack=991 Win=256 Len=110 [TCP segment of a reassembled PDU]
508	46.029413	192.168.1.9	192.168.1.13	TCP	164	8009 → 31691 [PSH, ACK] Seq=991 Ack=1101 Win=524 Len=110 [TCP segment of a reassembled PDU]
509	46.062903	192.168.1.13	192.168.1.9	TCP	164	31722 → 8009 [PSH, ACK] Seq=991 Ack=991 Win=251 Len=110 [TCP segment of a reassembled PDU]
510	46.078347	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [ACK] Seq=1101 Ack=1101 Win=255 Len=0
511	46.089700	192.168.1.9	192.168.1.13	TCP	164	8009 → 31722 [PSH, ACK] Seq=991 Ack=1101 Win=507 Len=110 [TCP segment of a reassembled PDU]
512	46.144141	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [ACK] Seq=1101 Ack=1101 Win=256 Len=0
524	51.046092	192.168.1.13	192.168.1.9	TCP	164	31691 → 8009 [PSH, ACK] Seq=1101 Ack=1101 Win=255 Len=110 [TCP segment of a reassembled PDU]
525	51.119469	192.168.1.13	192.168.1.9	TCP	164	31722 → 8009 [PSH, ACK] Seq=1101 Ack=1101 Win=256 Len=110 [TCP segment of a reassembled PDU]
527	51.215688	192.168.1.9	192.168.1.13	TCP	164	8009 → 31691 [PSH, ACK] Seq=1101 Ack=1211 Win=524 Len=110 [TCP segment of a reassembled PDU]
528	51.215688	192.168.1.9	192.168.1.13	TCP	164	8009 → 31722 [PSH, ACK] Seq=1101 Ack=1211 Win=507 Len=110 [TCP segment of a reassembled PDU]
530	51.264585	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [ACK] Seq=1211 Ack=1211 Win=255 Len=0
531	51.264679	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [ACK] Seq=1211 Ack=1211 Win=256 Len=0
543	56.234101	192.168.1.13	192.168.1.9	TCP	164	31691 → 8009 [PSH, ACK] Seq=1211 Ack=1211 Win=255 Len=110 [TCP segment of a reassembled PDU]
544	56.238169	192.168.1.13	192.168.1.9	TCP	164	31722 → 8009 [PSH, ACK] Seq=1211 Ack=1211 Win=256 Len=110 [TCP segment of a reassembled PDU]
402861	10871.187…	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [FIN, ACK] Seq=236567 Ack=236501 Win=251 Len=0
402863	10871.190…	192.168.1.13	192.168.1.9	TCP	66	37683 → 8009 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
402864	10871.190…	192.168.1.9	192.168.1.13	TCP	54	31691 → 8009 [FIN, ACK] Seq=237708 Ack=237873 Win=256 Len=0
402865	10871.194…	192.168.1.9	192.168.1.13	TCP	78	8009 → 31722 [PSH, ACK] Seq=236501 Ack=236568 Win=507 Len=24 [TCP segment of a reassembled PDU]
402866	10871.194…	192.168.1.9	192.168.1.13	TCP	54	8009 → 31722 [FIN, ACK] Seq=236525 Ack=236568 Win=507 Len=0
402867	10871.194…	192.168.1.13	192.168.1.9	TCP	54	31722 → 8009 [RST, ACK] Seq=236568 Ack=236525 Win=0 Len=0
402868	10871.196…	192.168.1.9	192.168.1.13	TCP	66	8009 → 37683 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=64
402869	10871.196…	192.168.1.9	192.168.1.13	TCP	78	8009 → 31691 [PSH, ACK] Seq=237873 Ack=237709 Win=641 Len=24 [TCP segment of a reassembled PDU]
402870	10871.196…	192.168.1.9	192.168.1.13	TCP	54	8009 → 31691 [FIN, ACK] Seq=237897 Ack=237709 Win=641 Len=0
402871	10871.196…	192.168.1.13	192.168.1.9	TCP	54	31691 → 8009 [RST, ACK] Seq=237709 Ack=237897 Win=0 Len=0
402872	10871.196…	192.168.1.13	192.168.1.9	TCP	54	37683 → 8009 [ACK] Seq=1 Ack=1 Win=131328 Len=0
402873	10871.198…	192.168.1.13	192.168.1.9	AJP13	571	AJP13 Error?
402874	10871.212…	192.168.1.9	192.168.1.13	TCP	54	8009 → 37683 [ACK] Seq=1 Ack=518 Win=30272 Len=0
402875	10871.221…	192.168.1.13	192.168.1.9	TCP	66	37684 → 8009 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
402876	10871.221…	192.168.1.9	192.168.1.13	AJP13	1262	AJP13 Error?
402878	10871.222…	192.168.1.13	192.168.1.9	AJP13	118	AJP13 Error?
402879	10871.223…	192.168.1.13	192.168.1.9	TCP	172	37683 → 8009 [PSH, ACK] Seq=582 Ack=1209 Win=130048 Len=118 [TCP segment of a reassembled PDU]
402880	10871.225…	192.168.1.9	192.168.1.13	TCP	66	8009 → 37684 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=64
402881	10871.225…	192.168.1.13	192.168.1.9	TCP	54	37684 → 8009 [ACK] Seq=1 Ack=1 Win=131328 Len=0
402882	10871.227…	192.168.1.13	192.168.1.9	AJP13	571	AJP13 Error?
402883	10871.242…	192.168.1.9	192.168.1.13	TCP	54	8009 → 37683 [ACK] Seq=1209 Ack=700 Win=30272 Len=0
402884	10871.254…	192.168.1.9	192.168.1.13	TCP	54	8009 → 37684 [ACK] Seq=1 Ack=518 Win=30272 Len=0
402885	10871.270…	192.168.1.9	192.168.1.13	AJP13	1262	AJP13 Error?
402886	10871.270…	192.168.1.13	192.168.1.9	TCP	118	37684 → 8009 [PSH, ACK] Seq=518 Ack=1209 Win=130048 Len=64 [TCP segment of a reassembled PDU]
402887	10871.271…	192.168.1.13	192.168.1.9	TCP	172	37684 → 8009 [PSH, ACK] Seq=582 Ack=1209 Win=130048 Len=118 [TCP segment of a reassembled PDU]
402888	10871.286…	192.168.1.9	192.168.1.13	TCP	54	8009 → 37684 [ACK] Seq=1209 Ack=700 Win=30272 Len=0
402889	10871.317…	192.168.1.13	192.168.1.9	TCP	1514	8009 → 37683 [ACK] Seq=1209 Ack=700 Win=30272 Len=1460 [TCP segment of a reassembled PDU]
402890	10871.317…	192.168.1.13	192.168.1.9	TCP	1514	8009 → 37683 [ACK] Seq=2669 Ack=700 Win=30272 Len=1460 [TCP segment of a reassembled PDU]
402891	10871.317…	192.168.1.9	192.168.1.13	TCP	1514	8009 → 37683 [ACK] Seq=4129 Ack=700 Win=30272 Len=1460 [TCP segment of a reassembled PDU]
402892	10871.317…	192.168.1.13	192.168.1.9	TCP	241	8009 → 37683 [PSH, ACK] Seq=5589 Ack=700 Win=30272 Len=187 [TCP segment of a reassembled PDU]
402893	10871.317…	192.168.1.13	192.168.1.9	TCP	54	37683 → 8009 [ACK] Seq=700 Ack=5776 Win=131328 Len=0
402894	10871.319…	192.168.1.13	192.168.1.9	TCP	507	37683 → 8009 [PSH, ACK] Seq=700 Ack=5776 Win=131328 Len=453 [TCP segment of a reassembled PDU]
402895	10871.319…	192.168.1.13	192.168.1.9	TCP	186	37683 → 8009 [PSH, ACK] Seq=1153 Ack=5776 Win=131328 Len=132 [TCP segment of a reassembled PDU]
402896	10871.336…	192.168.1.9	192.168.1.13	TCP	489	8009 → 37683 [PSH, ACK] Seq=5776 Ack=1285 Win=32448 Len=435 [TCP segment of a reassembled PDU]
402897	10871.336…	192.168.1.9	192.168.1.13	TCP	54	8009 → 37683 [ACK] Seq=5776 Ack=1285 Win=32448 Len=0
402898	10871.336…	192.168.1.13	192.168.1.9	TCP	54	37683 → 8009 [ACK] Seq=1285 Ack=6211 Win=130816 Len=0
402899	10871.379…	192.168.1.9	192.168.1.13	TCP	1514	8009 → 37684 [ACK] Seq=1209 Ack=700 Win=30272 Len=1460 [TCP segment of a reassembled PDU]
402900	10871.379…	192.168.1.9	192.168.1.13	TCP	1514	8009 → 37684 [ACK] Seq=2669 Ack=700 Win=30272 Len=1460 [TCP segment of a reassembled PDU]
402901	10871.379…	192.168.1.9	192.168.1.13	TCP	1514	8009 → 37684 [ACK] Seq=4129 Ack=700 Win=30272 Len=1460 [TCP segment of a reassembled PDU]
402902	10871.379…	192.168.1.9	192.168.1.13	TCP	241	8009 → 37684 [PSH, ACK] Seq=5589 Ack=700 Win=30272 Len=187 [TCP segment of a reassembled PDU]
402903	10871.379…	192.168.1.13	192.168.1.9	TCP	54	37684 → 8009 [ACK] Seq=700 Ack=5776 Win=131328 Len=0
402904	10871.385…	192.168.1.9	192.168.1.13	TCP	491	37684 → 8009 [PSH, ACK] Seq=700 Ack=5776 Win=131328 Len=437 [TCP segment of a reassembled PDU]
402905	10871.385…	192.168.1.13	192.168.1.9	TCP	186	37684 → 8009 [PSH, ACK] Seq=1137 Ack=5776 Win=131328 Len=132 [TCP segment of a reassembled PDU]
402906	10871.391…	192.168.1.9	192.168.1.13	TCP	489	8009 → 37684 [PSH, ACK] Seq=5776 Ack=1269 Win=32448 Len=435 [TCP segment of a reassembled PDU]
402907	10871.391…	192.168.1.9	192.168.1.13	TCP	54	8009 → 37684 [ACK] Seq=5776 Ack=1269 Win=32448 Len=0
402908	10871.391…	192.168.1.13	192.168.1.9	TCP	54	37684 → 8009 [ACK] Seq=1269 Ack=6211 Win=130816 Len=0
403052	10893.414…	192.168.1.9	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
403290	10948.411…	192.168.1.9	255.255.255.255	UDP	82	9487 → 9478 Len=40
403360	10953.426…	192.168.1.9	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
403414	11008.411…	192.168.1.9	255.255.255.255	UDP	82	9487 → 9478 Len=40
403417	11008.672…	192.168.1.9	192.168.1.13	TCP	54	[TCP Keep-Alive] 8009 → 37683 [ACK] Seq=6210 Ack=1285 Win=32448 Len=0
403418	11008.672…	192.168.1.9	192.168.1.13	TCP	54	[TCP Keep-Alive] 8009 → 37684 [ACK] Seq=6210 Ack=1269 Win=32448 Len=0
403422	11008.836…	192.168.1.13	192.168.1.9	TCP	54	[TCP Keep-Alive ACK] 37684 → 8009 [ACK] Seq=1269 Ack=6211 Win=130816 Len=0
403506	11013.431…	192.168.1.9	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}

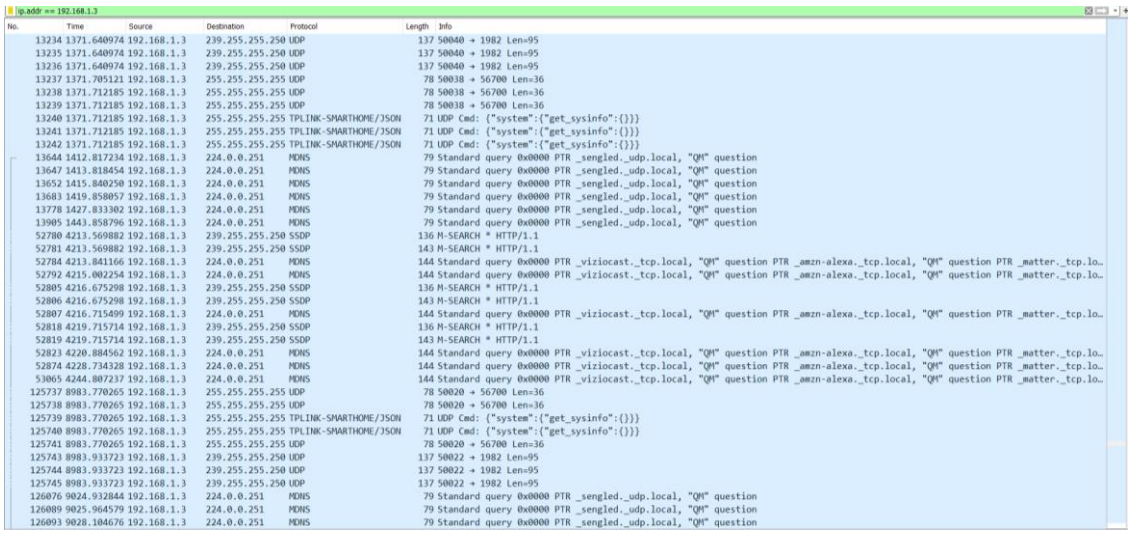
Apéndice 9. Tráfico de red Google Nest mini (parcial). Fuente: Elaboración propia

Apéndice D: Tráfico de red día 4

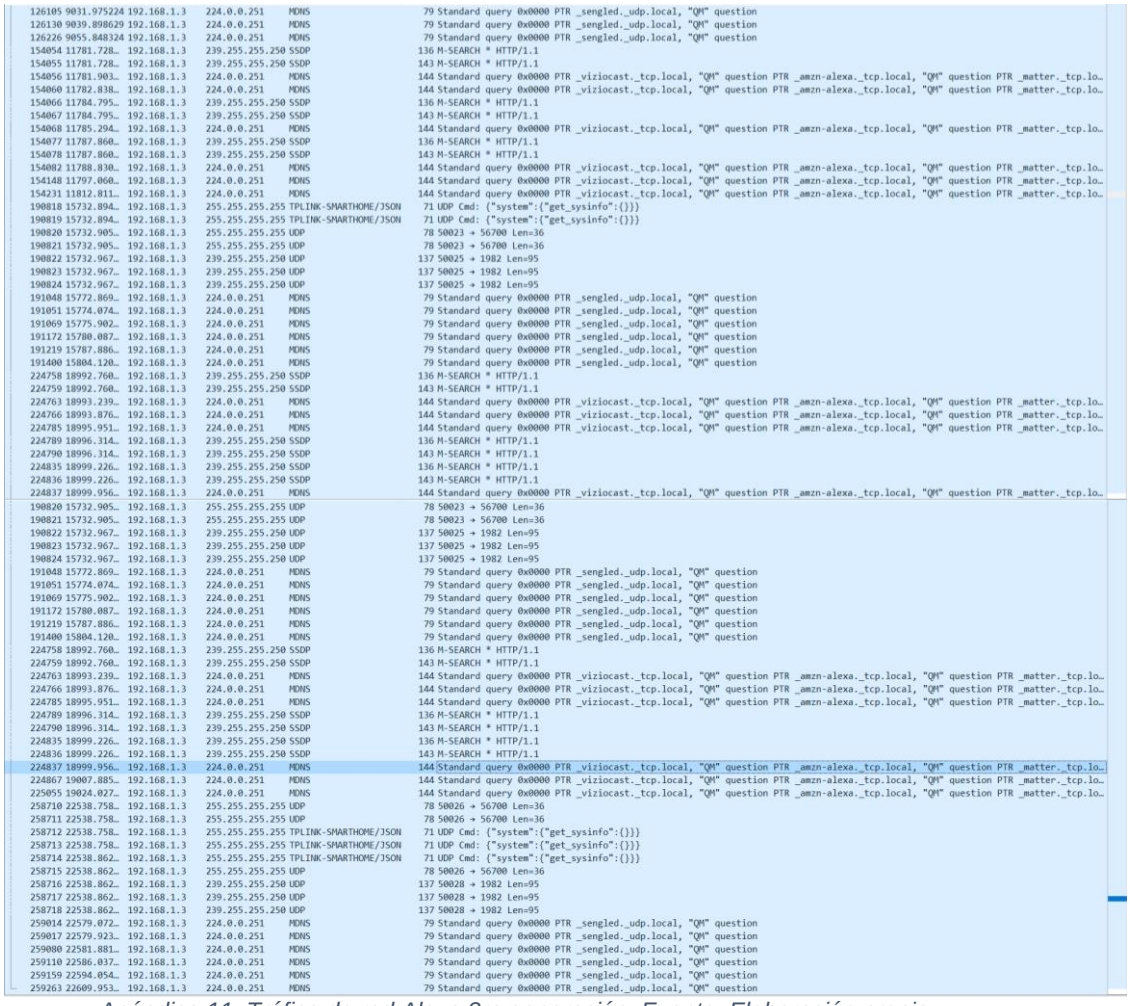


No.	Time	Source	Destination	Protocol	Length	Info
36533	3336.664168	192.168.1.2	239.255.255.250	UDP	137	50031 → 1982 Len=95
36534	3336.664168	192.168.1.2	239.255.255.250	UDP	137	50031 → 1982 Len=95
36535	3336.664168	192.168.1.2	239.255.255.250	UDP	137	50031 → 1982 Len=95
36536	3336.695334	192.168.1.2	255.255.255.255	UDP	78	50029 → 56700 Len=36
36537	3336.701082	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
36538	3336.701082	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
36539	3336.701082	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
36540	3336.701082	192.168.1.2	255.255.255.255	UDP	78	50029 → 56700 Len=36
36541	3336.701082	192.168.1.2	255.255.255.255	UDP	78	50029 → 56700 Len=36
36983	3377.757508	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
36987	3378.759052	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
36991	3380.773657	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
37019	3384.766207	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
37088	3392.771100	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
37233	3408.776863	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
111543	7194.649696	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
111544	7194.649696	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
111545	7194.917702	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
111548	7196.026150	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
111561	7197.683910	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
111562	7197.684279	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
111566	7198.005181	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
111596	7200.864009	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
111597	7200.864009	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
111602	7201.795998	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
111651	7209.932668	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
111798	7226.031098	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
134899	9964.701921	192.168.1.2	239.255.255.250	UDP	137	50034 → 1982 Len=95
134900	9964.701921	192.168.1.2	239.255.255.250	UDP	137	50034 → 1982 Len=95
134901	9964.701921	192.168.1.2	239.255.255.250	UDP	137	50034 → 1982 Len=95
134909	9964.782845	192.168.1.2	255.255.255.255	UDP	78	50032 → 56700 Len=36
134910	9964.782845	192.168.1.2	255.255.255.255	UDP	78	50032 → 56700 Len=36
134911	9964.782845	192.168.1.2	255.255.255.255	UDP	78	50032 → 56700 Len=36
134912	9964.782845	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
134913	9964.782845	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
134914	9964.784681	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
135303	10005.986…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
135307	10006.873…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR sengled. udp.local, "QM" question
135322	10008.822…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
135358	10012.975…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
135545	10021.015…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
135634	10036.835…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
175444	14126.970…	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
175445	14126.970…	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
175454	14127.140…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
175469	14127.903…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
175491	14129.801…	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
175492	14129.801…	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
175493	14129.889…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
175504	14132.859…	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
175505	14132.859…	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
175520	14133.888…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
175565	14142.016…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
175659	14157.974…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
210040	17620.738…	192.168.1.2	239.255.255.250	UDP	137	50037 → 1982 Len=95
210041	17620.738…	192.168.1.2	239.255.255.250	UDP	137	50037 → 1982 Len=95
210042	17620.738…	192.168.1.2	239.255.255.250	UDP	137	50037 → 1982 Len=95
210043	17620.787…	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
210044	17620.787…	192.168.1.2	255.255.255.255	UDP	78	50036 → 56700 Len=36
210045	17620.787…	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
210046	17620.787…	192.168.1.2	255.255.255.255	UDP	78	50036 → 56700 Len=36
210047	17620.787…	192.168.1.2	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
210679	17661.948…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
210688	17662.839…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
210725	17664.922…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
210769	17668.843…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
210849	17677.119…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
210940	17693.501…	192.168.1.2	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
244315	21028.921…	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
244316	21028.921…	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
244317	21028.981…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
244330	21029.965…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
244338	21032.008…	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
244339	21032.008…	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
244340	21032.008…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
244351	21034.878…	192.168.1.2	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
244352	21034.878…	192.168.1.2	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
244353	21036.004…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
244416	21044.072…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
244757	21060.070…	192.168.1.2	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…

Apéndice 10. Tráfico de red Alexa 4ta generación. Fuente: Elaboración propia

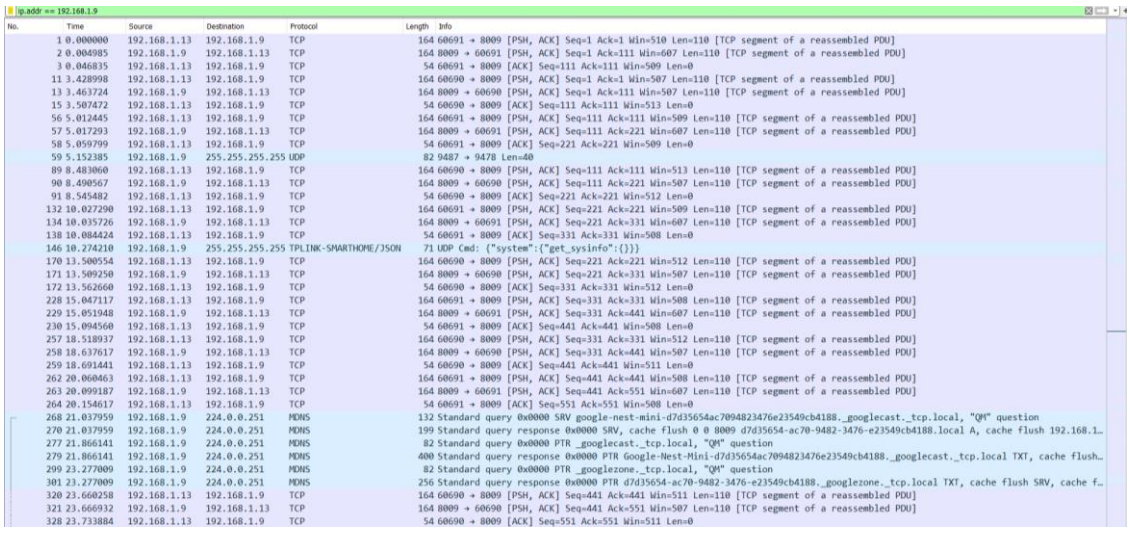


No.	Time	Source	Destination	Protocol	Length	Info
13234	1371.640974	192.168.1.3	239.255.255.250	UDP	137	50040 → 1982 Len=95
13235	1371.640974	192.168.1.3	239.255.255.250	UDP	137	50040 → 1982 Len=95
13236	1371.640974	192.168.1.3	239.255.255.250	UDP	137	50040 → 1982 Len=95
13237	1371.705121	192.168.1.3	255.255.255.255	UDP	78	50038 → 56700 Len=36
13238	1371.712185	192.168.1.3	255.255.255.255	UDP	78	50038 → 56700 Len=36
13239	1371.712185	192.168.1.3	255.255.255.255	UDP	78	50038 → 56700 Len=36
13240	1371.712185	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
13241	1371.712185	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
13242	1371.712185	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
13644	1412.817234	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
13647	1413.818454	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
13652	1415.840250	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
13683	1419.858057	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
13778	1427.833302	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
13905	1443.858796	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
52780	4213.569882	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
52781	4213.569882	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
52784	4213.841166	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
52792	4215.002254	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
52805	4216.675298	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
52806	4216.675298	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
52807	4216.715499	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
52818	4219.715714	192.168.1.3	239.255.255.250	SSDP	136	M-SEARCH * HTTP/1.1
52819	4219.715714	192.168.1.3	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
52823	4220.884562	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
52874	4228.734328	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
53065	4244.807237	192.168.1.3	224.0.0.251	MDNS	144	Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
125737	8983.770265	192.168.1.3	255.255.255.255	UDP	78	50020 → 56700 Len=36
125738	8983.770265	192.168.1.3	255.255.255.255	UDP	78	50020 → 56700 Len=36
125739	8983.770265	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
125740	8983.770265	192.168.1.3	255.255.255.255	TPLINK-SMARTHOME/JSON	71	UDP Cmd: {"system":{"get_sysinfo":{}}}
125741	8983.770265	192.168.1.3	255.255.255.255	UDP	78	50020 → 56700 Len=36
125743	8983.933723	192.168.1.3	239.255.255.250	UDP	137	50022 → 1982 Len=95
125744	8983.933723	192.168.1.3	239.255.255.250	UDP	137	50022 → 1982 Len=95
125745	8983.933723	192.168.1.3	239.255.255.250	UDP	137	50022 → 1982 Len=95
126076	9024.932844	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
126089	9025.964579	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question
126093	9028.104676	192.168.1.3	224.0.0.251	MDNS	79	Standard query 0x0000 PTR _sengled._udp.local, "QM" question

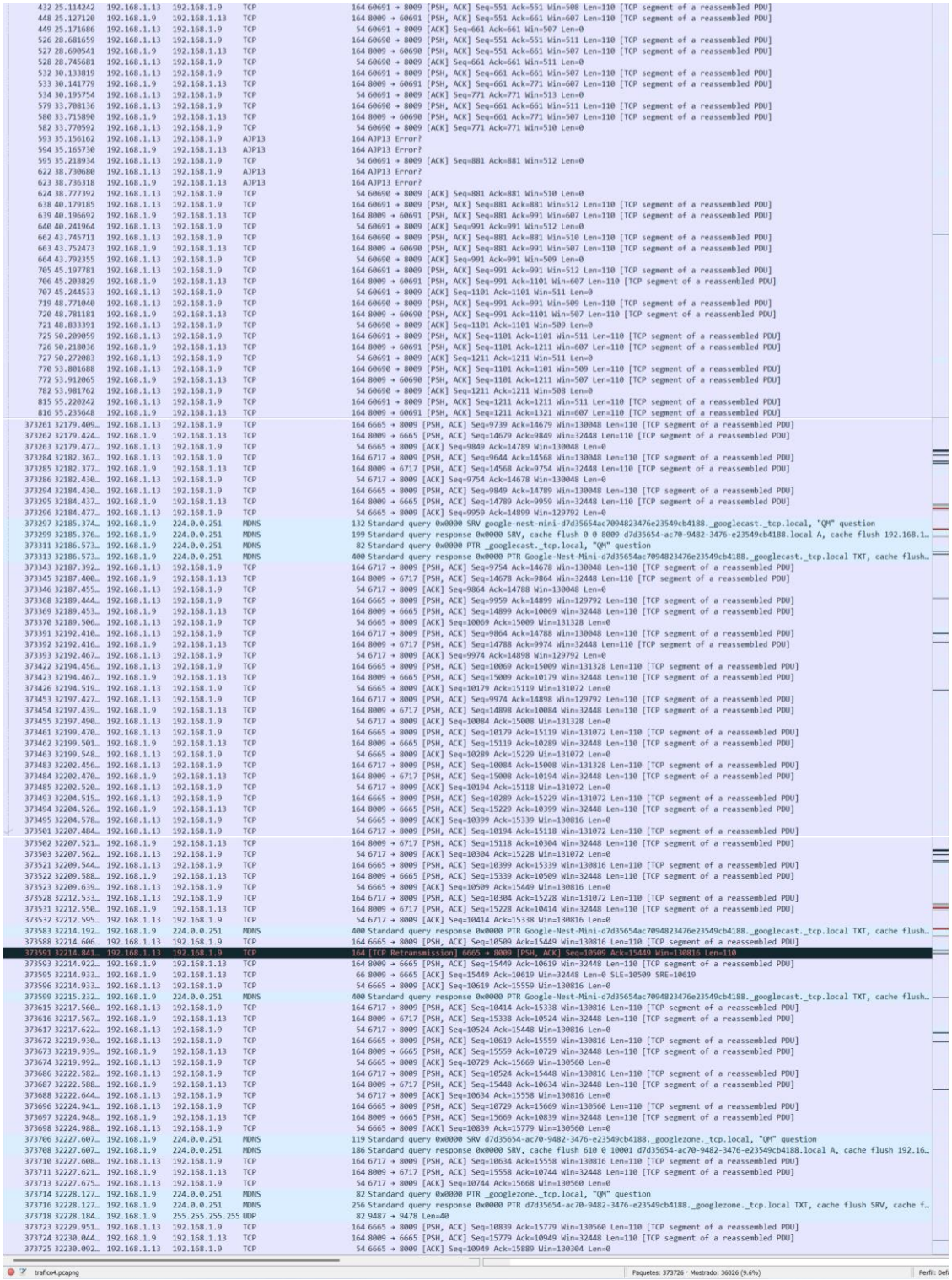


```
126105 9031.975224 192.168.1.3   224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
126130 9039.898629 192.168.1.3   224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
126226 9055.848324 192.168.1.3   224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
154054 11781.728… 192.168.1.3    239.255.255.250 SSDP                 136 M-SEARCH * HTTP/1.1
154055 11781.728… 192.168.1.3    239.255.255.250 SSDP                 143 M-SEARCH * HTTP/1.1
154056 11781.903… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
154060 11782.838… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
154066 11784.795… 192.168.1.3    239.255.255.250 SSDP                 136 M-SEARCH * HTTP/1.1
154067 11784.795… 192.168.1.3    239.255.255.250 SSDP                 143 M-SEARCH * HTTP/1.1
154068 11785.294… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
154077 11787.860… 192.168.1.3    239.255.255.250 SSDP                 136 M-SEARCH * HTTP/1.1
154078 11787.860… 192.168.1.3    239.255.255.250 SSDP                 143 M-SEARCH * HTTP/1.1
154082 11788.830… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
154148 11797.060… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
154231 11812.811… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
190818 15732.894… 192.168.1.3    255.255.255.255 TPLINK-SMARTHOME/JSON  71 UDP Cmd: {"system":{"get_sysinfo":{}}}
190819 15732.894… 192.168.1.3    255.255.255.255 TPLINK-SMARTHOME/JSON  71 UDP Cmd: {"system":{"get_sysinfo":{}}}
190820 15732.905… 192.168.1.3    255.255.255.255 UDP                   78 50023 → 56700 Len=36
190821 15732.905… 192.168.1.3    255.255.255.255 UDP                   78 50023 → 56700 Len=36
190822 15732.967… 192.168.1.3    239.255.255.250 UDP                  137 50025 → 1982 Len=95
190823 15732.967… 192.168.1.3    239.255.255.250 UDP                  137 50025 → 1982 Len=95
190824 15732.967… 192.168.1.3    239.255.255.250 UDP                  137 50025 → 1982 Len=95
191048 15772.869… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
191051 15774.074… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
191069 15775.902… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
191172 15780.087… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
191219 15787.886… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
191400 15804.120… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
224758 18992.760… 192.168.1.3    239.255.255.250 SSDP                 136 M-SEARCH * HTTP/1.1
224759 18992.760… 192.168.1.3    239.255.255.250 SSDP                 143 M-SEARCH * HTTP/1.1
224763 18993.239… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
224766 18993.876… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
224785 18995.951… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
224789 18996.314… 192.168.1.3    239.255.255.250 SSDP                 136 M-SEARCH * HTTP/1.1
224790 18996.314… 192.168.1.3    239.255.255.250 SSDP                 143 M-SEARCH * HTTP/1.1
224835 18999.226… 192.168.1.3    239.255.255.250 SSDP                 136 M-SEARCH * HTTP/1.1
224836 18999.226… 192.168.1.3    239.255.255.250 SSDP                 143 M-SEARCH * HTTP/1.1
224837 18999.956… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
190820 15732.905… 192.168.1.3    255.255.255.255 UDP                   78 50023 → 56700 Len=36
190821 15732.905… 192.168.1.3    255.255.255.255 UDP                   78 50023 → 56700 Len=36
190822 15732.967… 192.168.1.3    239.255.255.250 UDP                  137 50025 → 1982 Len=95
190823 15732.967… 192.168.1.3    239.255.255.250 UDP                  137 50025 → 1982 Len=95
190824 15732.967… 192.168.1.3    239.255.255.250 UDP                  137 50025 → 1982 Len=95
191048 15772.869… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
191051 15774.074… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
191069 15775.902… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
191172 15780.087… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
191219 15787.886… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
191400 15804.120… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
224758 18992.760… 192.168.1.3    239.255.255.250 SSDP                 136 M-SEARCH * HTTP/1.1
224759 18992.760… 192.168.1.3    239.255.255.250 SSDP                 143 M-SEARCH * HTTP/1.1
224763 18993.239… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
224766 18993.876… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
224785 18995.951… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
224789 18996.314… 192.168.1.3    239.255.255.250 SSDP                 136 M-SEARCH * HTTP/1.1
224790 18996.314… 192.168.1.3    239.255.255.250 SSDP                 143 M-SEARCH * HTTP/1.1
224835 18999.226… 192.168.1.3    239.255.255.250 SSDP                 136 M-SEARCH * HTTP/1.1
224836 18999.226… 192.168.1.3    239.255.255.250 SSDP                 143 M-SEARCH * HTTP/1.1
224837 18999.956… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
224867 19007.885… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
225055 19024.027… 192.168.1.3    224.0.0.251     MDNS                 144 Standard query 0x0000 PTR _viziocast._tcp.local, "QM" question PTR _amzn-alexa._tcp.local, "QM" question PTR _matter._tcp.lo…
258710 22538.758… 192.168.1.3    255.255.255.255 UDP                   78 50026 → 56700 Len=36
258711 22538.758… 192.168.1.3    255.255.255.255 UDP                   78 50026 → 56700 Len=36
258712 22538.758… 192.168.1.3    255.255.255.255 TPLINK-SMARTHOME/JSON  71 UDP Cmd: {"system":{"get_sysinfo":{}}}
258713 22538.758… 192.168.1.3    255.255.255.255 TPLINK-SMARTHOME/JSON  71 UDP Cmd: {"system":{"get_sysinfo":{}}}
258714 22538.862… 192.168.1.3    255.255.255.255 TPLINK-SMARTHOME/JSON  71 UDP Cmd: {"system":{"get_sysinfo":{}}}
258715 22538.862… 192.168.1.3    255.255.255.255 UDP                   78 50026 → 56700 Len=36
258716 22538.862… 192.168.1.3    239.255.255.250 UDP                  137 50028 → 1982 Len=95
258717 22538.862… 192.168.1.3    239.255.255.250 UDP                  137 50028 → 1982 Len=95
258718 22538.862… 192.168.1.3    239.255.255.250 UDP                  137 50028 → 1982 Len=95
259014 22579.072… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
259017 22579.923… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
259080 22581.881… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
259110 22586.037… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
259159 22594.054… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
259263 22609.953… 192.168.1.3    224.0.0.251     MDNS                  79 Standard query 0x0000 PTR _sengled._udp.local, "QM" question
```

Apéndice 11. Tráfico de red Alexa 3ra generación. Fuente: Elaboración propia



```
ip.addr == 192.168.1.9
No.  Time        Source          Destination     Protocol  Length  Info
     1 0.000000  192.168.1.13    192.168.1.9     TCP          164  60691 → 8009 [PSH, ACK] Seq=1 Ack=1 Win=510 Len=110 [TCP segment of a reassembled PDU]
     2 0.004985  192.168.1.9     192.168.1.13    TCP          164  8009 → 60691 [PSH, ACK] Seq=1 Ack=111 Win=607 Len=110 [TCP segment of a reassembled PDU]
     3 0.046835  192.168.1.13    192.168.1.9     TCP           54  60691 → 8009 [ACK] Seq=111 Ack=111 Win=509 Len=0
    11 3.428998  192.168.1.13    192.168.1.9     TCP          164  60690 → 8009 [PSH, ACK] Seq=1 Ack=1 Win=507 Len=110 [TCP segment of a reassembled PDU]
    13 3.463724  192.168.1.9     192.168.1.13    TCP          164  8009 → 60690 [PSH, ACK] Seq=1 Ack=111 Win=507 Len=110 [TCP segment of a reassembled PDU]
    15 3.507472  192.168.1.13    192.168.1.9     TCP           54  60690 → 8009 [ACK] Seq=111 Ack=111 Win=513 Len=0
    56 5.012445  192.168.1.13    192.168.1.9     TCP          164  60691 → 8009 [PSH, ACK] Seq=111 Ack=111 Win=509 Len=110 [TCP segment of a reassembled PDU]
    57 5.017293  192.168.1.9     192.168.1.13    TCP          164  8009 → 60691 [PSH, ACK] Seq=111 Ack=221 Win=607 Len=110 [TCP segment of a reassembled PDU]
    58 5.059799  192.168.1.13    192.168.1.9     TCP           54  60691 → 8009 [ACK] Seq=221 Ack=221 Win=509 Len=0
    59 5.152385  192.168.1.9     255.255.255.255 UDP           82  9487 → 9478 Len=40
    89 8.483060  192.168.1.13    192.168.1.9     TCP          164  60690 → 8009 [PSH, ACK] Seq=111 Ack=111 Win=513 Len=110 [TCP segment of a reassembled PDU]
    90 8.490567  192.168.1.9     192.168.1.13    TCP          164  8009 → 60690 [PSH, ACK] Seq=111 Ack=221 Win=507 Len=110 [TCP segment of a reassembled PDU]
    91 8.545482  192.168.1.13    192.168.1.9     TCP           54  60690 → 8009 [ACK] Seq=221 Ack=221 Win=512 Len=0
   132 10.027290 192.168.1.13    192.168.1.9     TCP          164  60691 → 8009 [PSH, ACK] Seq=221 Ack=221 Win=509 Len=110 [TCP segment of a reassembled PDU]
   134 10.035726 192.168.1.9     192.168.1.13    TCP          164  8009 → 60691 [PSH, ACK] Seq=221 Ack=331 Win=607 Len=110 [TCP segment of a reassembled PDU]
   138 10.084424 192.168.1.13    192.168.1.9     TCP           54  60691 → 8009 [ACK] Seq=331 Ack=331 Win=508 Len=0
   146 10.274210 192.168.1.9     255.255.255.255 TPLINK-SMARTHOME/JSON  71 UDP Cmd: {"system":{"get_sysinfo":{}}}
   170 13.500554 192.168.1.13    192.168.1.9     TCP          164  60690 → 8009 [PSH, ACK] Seq=221 Ack=221 Win=512 Len=110 [TCP segment of a reassembled PDU]
   171 13.509250 192.168.1.9     192.168.1.13    TCP          164  8009 → 60690 [PSH, ACK] Seq=221 Ack=331 Win=507 Len=110 [TCP segment of a reassembled PDU]
   172 13.562660 192.168.1.13    192.168.1.9     TCP           54  60690 → 8009 [ACK] Seq=331 Ack=331 Win=512 Len=0
   228 15.047117 192.168.1.13    192.168.1.9     TCP          164  60691 → 8009 [PSH, ACK] Seq=331 Ack=331 Win=508 Len=110 [TCP segment of a reassembled PDU]
   229 15.051948 192.168.1.9     192.168.1.13    TCP          164  8009 → 60691 [PSH, ACK] Seq=331 Ack=441 Win=607 Len=110 [TCP segment of a reassembled PDU]
   230 15.094560 192.168.1.13    192.168.1.9     TCP           54  60691 → 8009 [ACK] Seq=441 Ack=441 Win=508 Len=0
   257 18.518937 192.168.1.13    192.168.1.9     TCP          164  60690 → 8009 [PSH, ACK] Seq=331 Ack=331 Win=512 Len=110 [TCP segment of a reassembled PDU]
   258 18.637617 192.168.1.9     192.168.1.13    TCP          164  8009 → 60690 [PSH, ACK] Seq=331 Ack=441 Win=507 Len=110 [TCP segment of a reassembled PDU]
   259 18.691441 192.168.1.13    192.168.1.9     TCP           54  60690 → 8009 [ACK] Seq=441 Ack=441 Win=511 Len=0
   262 20.060463 192.168.1.13    192.168.1.9     TCP          164  60691 → 8009 [PSH, ACK] Seq=441 Ack=441 Win=508 Len=110 [TCP segment of a reassembled PDU]
   263 20.099187 192.168.1.9     192.168.1.13    TCP          164  8009 → 60691 [PSH, ACK] Seq=441 Ack=551 Win=607 Len=110 [TCP segment of a reassembled PDU]
   264 20.154617 192.168.1.13    192.168.1.9     TCP           54  60691 → 8009 [ACK] Seq=551 Ack=551 Win=508 Len=0
   268 21.037959 192.168.1.9     224.0.0.251     MDNS         132  Standard query 0x0000 SRV google-nest-mini-d7d35654ac7094823476e23549cb4188._googlecast._tcp.local, "QM" question
   270 21.037959 192.168.1.9     224.0.0.251     MDNS         199  Standard query response 0x0000 SRV, cache flush 0 0 8009 d7d35654-ac70-9482-3476-e23549cb4188.local A, cache flush 192.168.1…
   277 21.866141 192.168.1.9     224.0.0.251     MDNS          82  Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
   279 21.866141 192.168.1.9     224.0.0.251     MDNS         400  Standard query response 0x0000 PTR Google-Nest-Mini-d7d35654ac7094823476e23549cb4188._googlecast._tcp.local TXT, cache flush…
   299 23.277009 192.168.1.9     224.0.0.251     MDNS          82  Standard query 0x0000 PTR _googlezone._tcp.local, "QM" question
   301 23.277009 192.168.1.9     224.0.0.251     MDNS         256  Standard query response 0x0000 PTR d7d35654-ac70-9482-3476-e23549cb4188._googlezone._tcp.local TXT, cache flush SRV, cache f…
   320 23.660258 192.168.1.13    192.168.1.9     TCP          164  60690 → 8009 [PSH, ACK] Seq=441 Ack=441 Win=511 Len=110 [TCP segment of a reassembled PDU]
   321 23.666932 192.168.1.9     192.168.1.13    TCP          164  8009 → 60690 [PSH, ACK] Seq=441 Ack=551 Win=507 Len=110 [TCP segment of a reassembled PDU]
   328 23.733884 192.168.1.13    192.168.1.9     TCP           54  60690 → 8009 [ACK] Seq=551 Ack=551 Win=511 Len=0
```



Apéndice 12. Tráfico de red Google Nest mini (parcial). Fuente: Elaboración propia

Apéndice E: Permisos de la aplicación Amazon Alexa y Google Home

ACCESS_NETWORK_STATE normal

Allows an application to view the status of all networks.

ACCESS_WIFI_STATE normal

Allows an application to view the information about the status of Wi-Fi.

BLUETOOTH normal

Allows applications to connect to paired bluetooth devices.

BLUETOOTH_ADMIN normal

Allows applications to discover and pair bluetooth devices.

CHANGE_NETWORK_STATE normal

Allows applications to change network connectivity state.

CHANGE_WIFI_MULTICAST_STATE normal

Allows an application to receive packets not directly addressed to your device. This can be useful when discovering services offered nearby. It uses more power than the non-multicast mode.

CHANGE_WIFI_STATE normal

Allows an application to connect to and disconnect from Wi-Fi access points and to make changes to configured Wi-Fi networks.

FOREGROUND_SERVICE normal

Allows a regular application to use Service.startForeground.

INTERNET normal

Allows an application to create network sockets.

MANAGE_OWN_CALLS normal

Allows a calling application which manages it own calls through the self-managed ConnectionService APIs.

MODIFY_AUDIO_SETTINGS normal

Allows application to modify global audio settings, such as volume and routing.

RECEIVE_BOOT_COMPLETED normal

Allows an application to start itself as soon as the system has finished booting. This can make it take longer to start the phone and allow the application to slow down the overall phone by always running.

REQUEST_COMPANION_RUN_IN_BACKGROUND normal

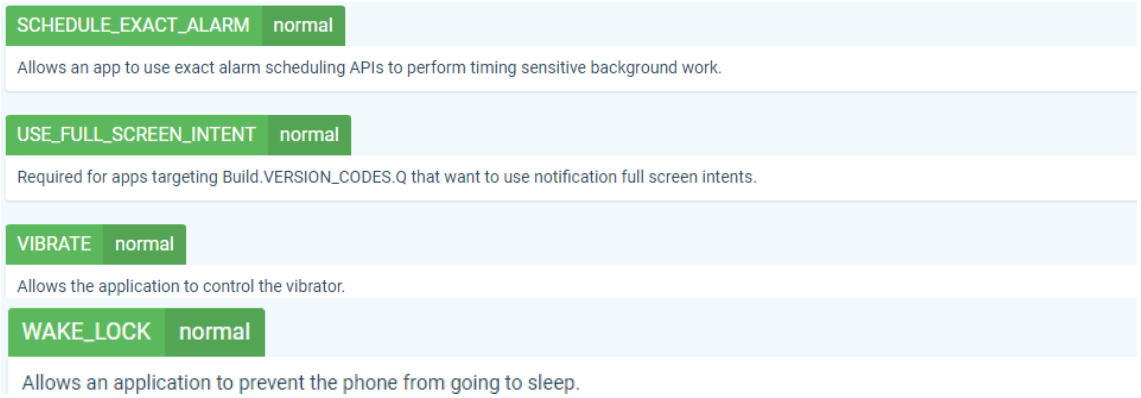
Allows a companion app to run in the background.

REQUEST_COMPANION_USE_DATA_IN_BACKGROUND normal

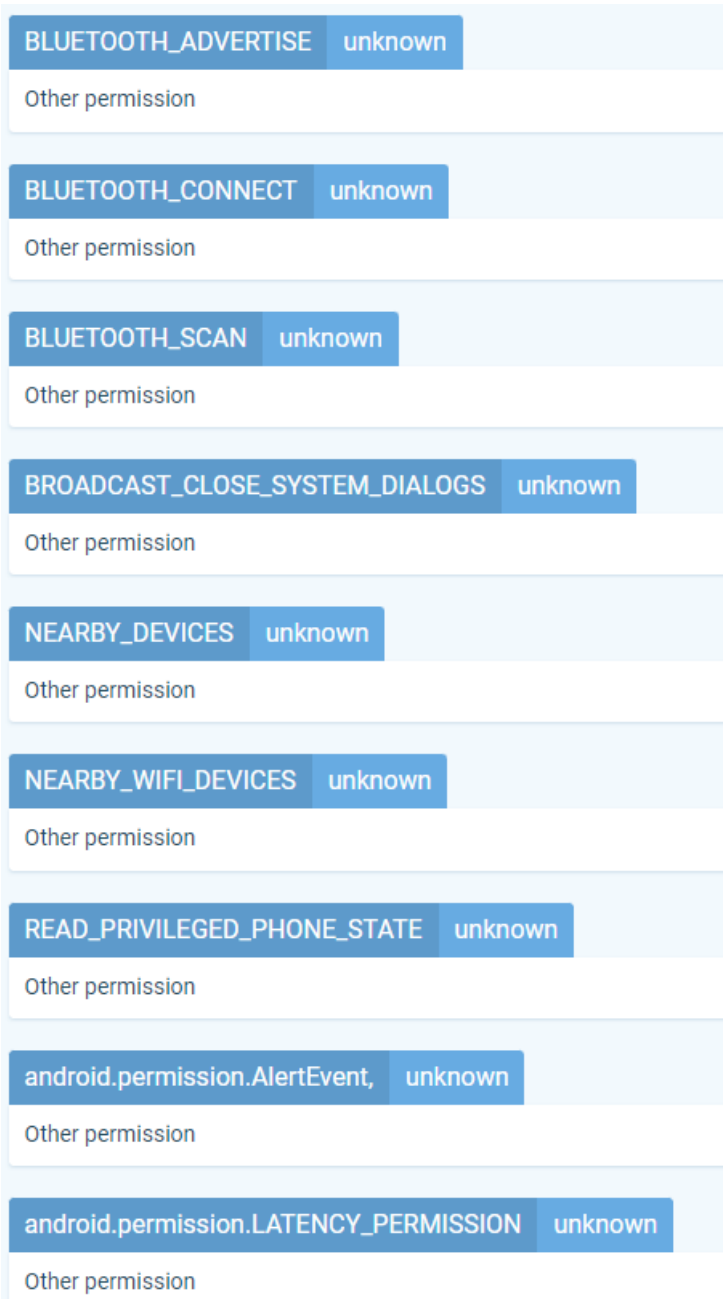
Allows a companion app to use data in the background.

REQUEST_IGNORE_BATTERY_OPTIMIZATIONS normal

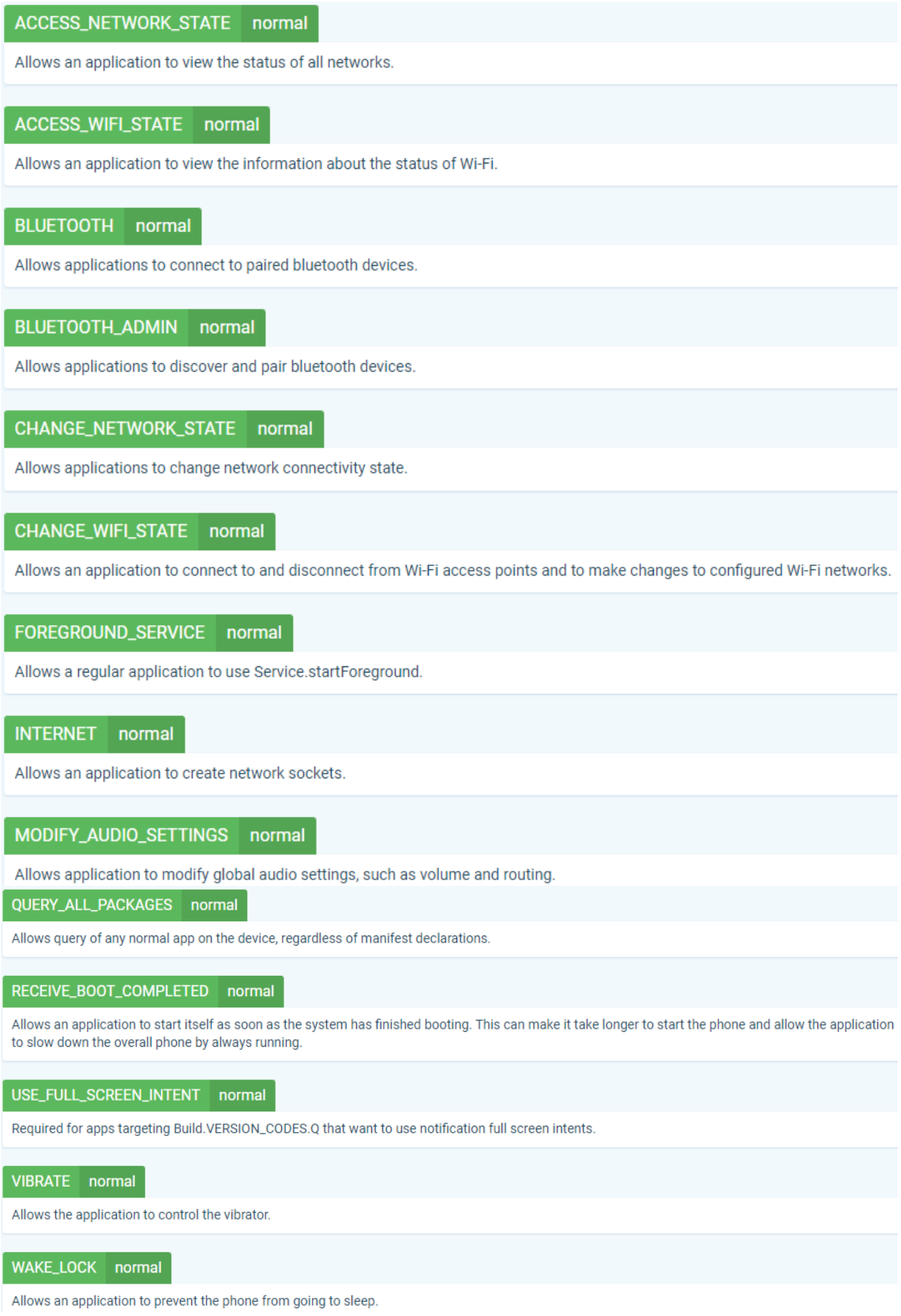
Permission an application must hold in order to use Settings.ACTION_REQUEST_IGNORE_BATTERY_OPTIMIZATIONS.



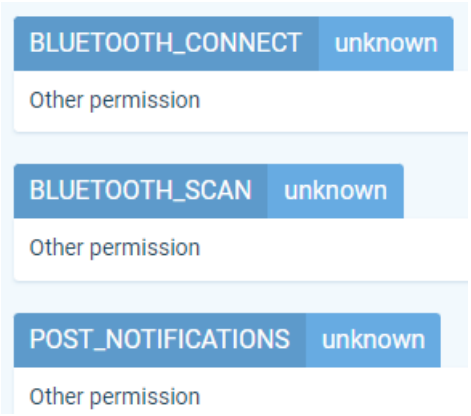
Apéndice 13. Permisos de nivel normal-Amazon Alexa. Fuente: Elaboración propia



Apéndice 14. Otros permisos-Amazon Alexa. Fuente: Elaboración propia

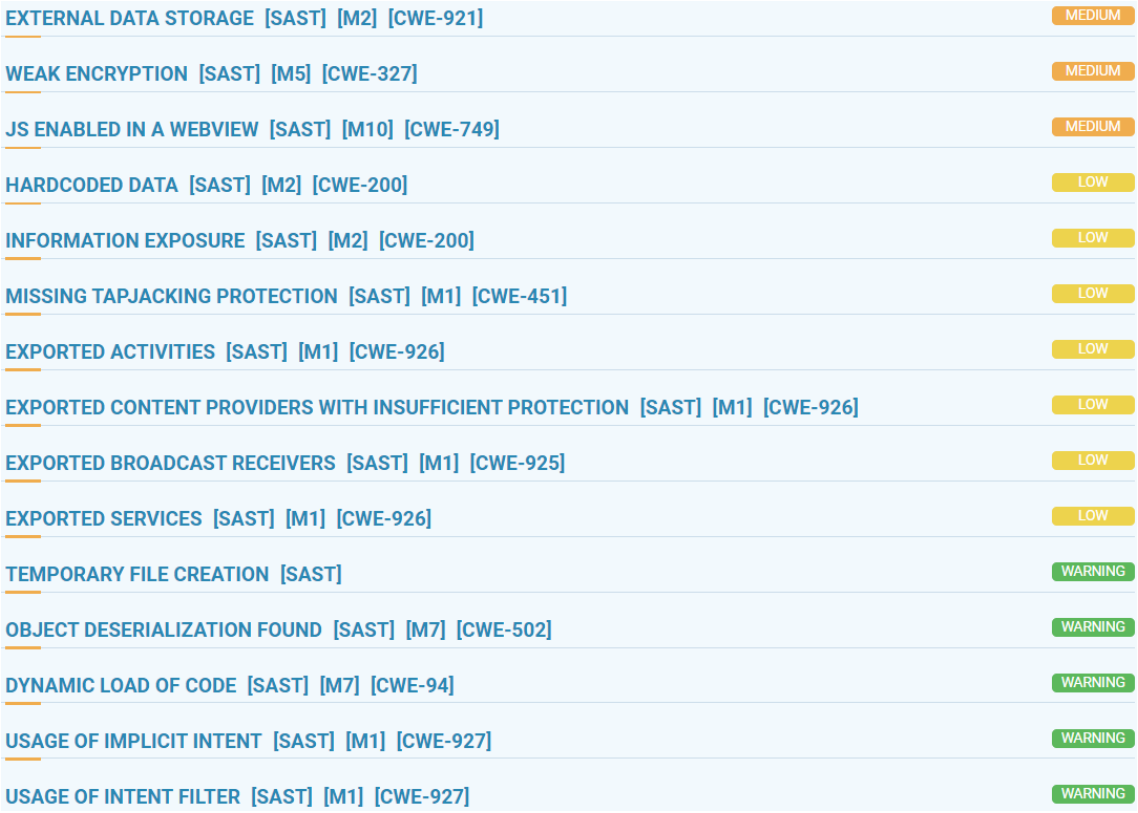


Apéndice 15. Permisos de nivel normal-Google Home. Fuente: Elaboración propia

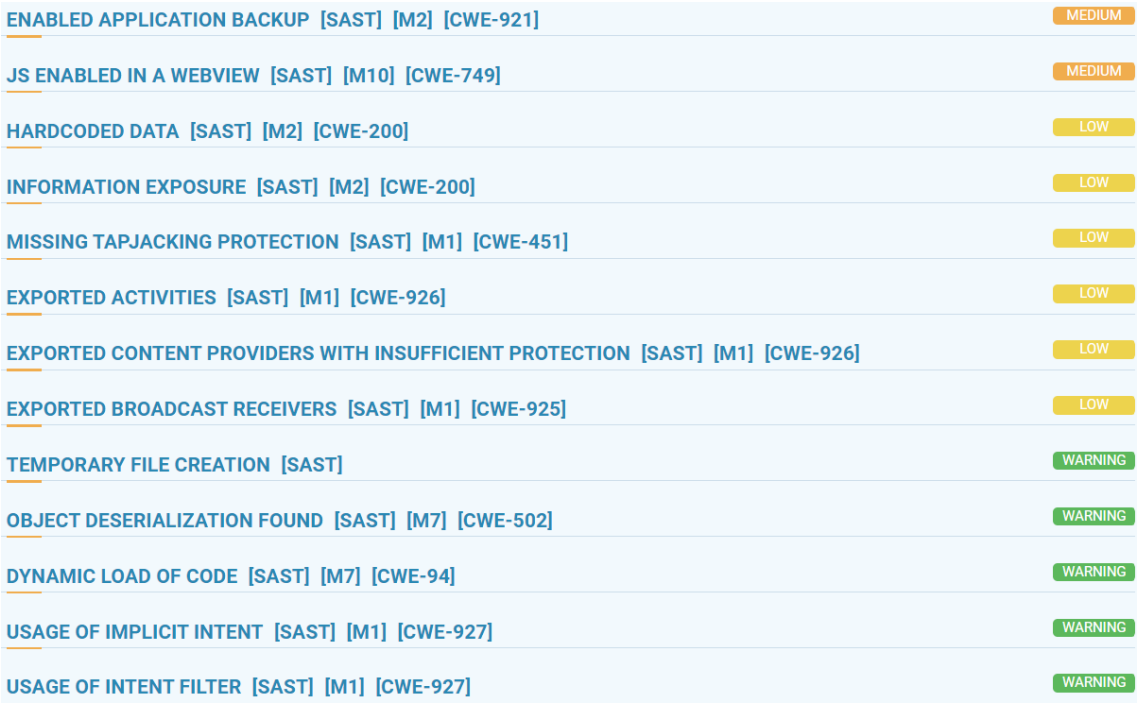


Apéndice 16. Otros permisos-Google Home. Fuente: Elaboración propia

Apéndice F: Prueba de seguridad OWASP Mobile Top 10 Amazon Alexa y Google Home



Apéndice 17. Prueba de seguridad-Amazon Alexa. Fuente: Elaboración propia



Apéndice 18. Prueba de seguridad-Google Home. Fuente: Elaboración propia

Anexos

Anexo 1: Resumen de las áreas de investigación recientes de IoT

Research area	Techniques	Solutions	Open area
Networking [11]	MANET	The authors have used the MANET as a way to maintain connection between things.	• Improving Ad-Hoc network. • Network technologies. • RFID. • Communication protocols.
Routing [12]	Learning automates (LA) & Cross-layer.	The authors proposed a protocol called "fault-tolerant routing" protocol to serve the IoT environment.	• Improving the proposed algorithm to cover the wide range of the application domains
Heterogeneity [13]	Midgar Software	The authors have been sought to improve the interaction between objects or things via the graphic editor that generates model defined by a Domain Specific Language.	• Domain Specific language and graphic editor to generate smart objects. • Insertion of boxes in the graphic editor with support for data analysis and fuzzy logic. • Scalability of IoT platform. • Security and privacy.
Interoperability [16]	Semantic Interoperability Architecture	The authors have been sought to build the semantic interoperability architecture to access the information easily, also the paper used the monitoring and updating events of physical world in real time.	• The architecture needs tools that support development and deployment of devices and applications into the future IoT systems.
QoS [19]	BT, IP & GA	The authors have been reviewed comparison between three algorithms to determine Qos metrics for composes service. BT algorithm is the most appropriate to IoT environment more than ILP & GA, because it suitable to serve both high scale of service and real-time application.	• Decreasing steps of BT algorithm for calculating QoS.
Scalability [5]	Aneka Hybrid cloud computing (private cloud + public cloud)	The authors have used cloud computing technology with IoT environment to improve scalability and provide storage resources.	• Cloud computing. • Energy efficient sensing. • Architecture of IoT. • Data mining. • Secure reprogrammable networks and privacy.
Virtualization [3]	IoT Virtualization Framework based on SenaaS technology.	The authors have created a new framework based on SenaaS notion named it "IoT Virtualization Framework", the main objective of this frame is reusability of the sensor information via web browser.	• The development of IoT framework services micro-formats for advertising on social networks. • Improving the performance of an IoT framework at real time.
Big Data [22]	SMARTCAMPUS.	The authors have sought to improve the software architecture named "real-life" based on extracted from the SMARTCAMPUS project to handle Big	• Cloud Computing. • Scalability/elasticity. • Computation time. • Security.
		Data in the IoT environment.	
Cloud Computing [24]	CloudIoT Paradigm.	The authors have highlighted the integration between the cloud computing and IoT, also reviewed the previous literatures about them.	• Standardization of framework. • Power consuming. • Fog Cloud. • Complex data mining.
Power Consumption [26]	Self-Organized Power Consumption Approximation (SOPCA) Algorithm	The authors have used a self-organizing for dynamic approximation of power consumption to create (SOPCA) algorithm. Also the authors have used Agent-based model to test this algorithm.	• The SOPCA can be further explored by the evaluation of flooding as well as by using artificial intelligence algorithms.
Security [29]	SCH	The authors have used SCH to improve and ensure security of RFID system.	• Improving and addressing the integration of a tag tamper resistance mechanism.

Anexo 1. Tabla de resumen de las áreas de investigación. Fuente: Tabla de resumen tomada de [13]