UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE COMPUTACIÓN

ANÁLISIS DE DATOS Y HACKEO ÉTICO PARA LA DETECCIÓN DE VULNERABILIDADES EN LA RED WI-FI CON USUARIOS DEL LABORATORIO IOT DE LA UNIVERSIDAD POLITÉCNICA SALESIANA.

Trabajo de titulación previo a la obtención del

Título de Ingenieros en Ciencias de la Computación

AUTORES: ALEXIS SEBASTIÁN GUALLASAMÍN HARO

GABRIEL ALEXANDER SANTOS GUERRERO

TUTOR: MANUEL RAFAEL JAYA DUCHE

Quito – Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros Alexis Sebastián Guallasamín Haro con documento de identificación N° 1719995829 y Gabriel Alexander Santos Guerrero con documento de identificación N° 1724924160; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 13 de Septiembre del año 2022

Atentamente,

Alexis Sebastián Guallasamín Haro

1719995829

Gabriel Alexander Santos Guerrero

1724924160

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Alexis Sebastián Guallasamín Haro documento de identificación N° 1719995829 y Gabriel Alexander Santos Guerrero con documento de identificación N° 1724924160, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: "Análisis de datos y hackeo ético para la detección de vulnerabilidades en la red Wi-fi con usuarios del laboratorio IoT de la universidad politécnica salesiana.", el cual ha sido desarrollado para optar por el título de: Ingenieros en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 13 de Septiembre del año 2022

Atentamente,

Alexis Sebastián Guallasamín Haro

1719995829

Gabriel Alexander Santos Guerrero

1724924160

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Manuel Rafael Jaya Duche con documento de identificación N° 1710631035, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: ANÁLISIS DE DATOS Y HACKEO ÉTICO PARA LA DETECCIÓN DE VULNERABILIDADES EN LA RED WI-FI CON USUARIOS DEL LABORATORIO IOT DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, realizado por Alexis Sebastián Guallasamín Haro con documento de identificación N° 1719995829 y por Gabriel Alexander Santos Guerrero con documento de identificación N° 1724924160, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 13 de Septiembre del año 2022

Atentamente,

Ing. Manuel Rafael Jaya Duche, MSc

1710631035

DEDICATORIA

La realización de este proyecto lo dedico en primer lugar a mis padres quienes me dieron la vida, mi padre quien me enseñó a no rendirme, esforzarme y seguir adelante a pesar de las circunstancias, me guio en este camino, darme mi primer computador gracias a él entre muchas cosas más a las que hoy día doy gracias por ser la base de mi inspiración para esta carrera en la que hoy día me estoy graduando, a mi madre por ser el pilar de mi vida quien estuvo para apoyarme y guiarme por el camino del bien inculcándome valores para no ser solo un buen profesional sino una excelente persona por llenarme de cariño y provisionarme de todos los recursos que necesite para culminar mis estudios, a mi abuelita por enseñarme persistir en alcanzar mis metas que todo camino tiene obstáculos pero que siempre los vea como retos para superarme y alcanzar el éxito.

También dedico este logro a toda mi familia que sin duda han sido fundamentales para alcanzar mi meta ya que cada uno de ellos han contribuido para ser de mí una mejor persona.

Gabriel Santos

Dedico este trabajo a mis padres Ximena Haro y Armando Guallasamín, por su apoyo en cada uno de los ámbitos de mi vida, más que todo en el arduo camino de mi carrera universitaria, a mis hermanos que han sido mi ejemplo a seguir, además de ser un motivo de esfuerzo para llegar a ser un buen profesional.

Alexis Guallasamín

AGRADECIMIENTO

Con la culminación de mi carrera quiero agradecer en especial a mis padres y mi abuelita por haberme dado la mejor herencia que es el estudio por formarme en el camino de la rectitud, por ser mi ejemplo de perseverancia, esfuerzo y dedicación a seguir, hoy me graduó gracias a ellos.

Quiero agradecer a mi novia, la mujer que siempre estuvo a mi lado apoyándome en las buenas y malas, no dudo de mí nunca y aun en mis momentos de debilidad me motivo a seguir adelante, a no rendirme, que siempre después de la tormenta viene la calma y sobre todo que jamás me dé por vencido que puedo llegar muy lejos, te agradezco por tu amor, cariño y paciencia.

A la Universidad Politécnica Salesiana que con sus docentes dejaron una huella no solo por su profesionalismo sino por su calidad humana, su pasión por enseñar y formar excelentes profesionales y seres humanos preparados para el mundo laboral gracias por su ardua labor.

Gabriel Santos

Agradezco a Dios por permitirme estudiar una carrera universitaria y poder culminarla sin ningún problema, y por brindarme todo lo necesario para poder salir adelante, a mi familia por apoyarme día a día, siendo ellos un pilar fundamental dentro de este logro, ya que sin su ayuda no podría haberlo logrado, sus palabras de aliento en cada problema me ayudaron a ser la persona que soy ahora, impartiéndome enseñanzas de vida que me ayudarán en cada una de las etapas que se me presenten. De igual manera agradezco la ayuda brindada por cada uno de mis profesores, por todo el conocimiento que adquirí gracias a la paciencia y buena actitud que mostraron, siempre velando por nuestra preparación para la vida laboral.

Alexis Guallasamín

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

RESUMEN

El presente proyecto técnico tiene como objetivo desarrollar una aplicación Sniffer (hardware y software) de bajo costo, de acuerdo al uso de herramientas para el análisis de datos, se busca capturar y analizar cada uno de los paquetes, los cuales poseen un tipo de información de la red Wi-Fi en el laboratorio IoT de la Universidad Politécnica Salesiana. Por medio de la creación de escenarios se prioriza ataques que puedan ser realizados con la información generada del Sniffer, todo esto bajo la Metodología para el Análisis de Vulnerabilidades, escogiendo las herramientas adecuadas para la implementación, es por esto que se utilizó el entorno de desarrollo de Arduino y la herramienta Wifislax para los ataques de hackeo ético. En la parte del análisis se detalla cada uno de los campos generados por el archivo de la herramienta Wireshark, de acuerdo a esto se evidencia las redes inalámbricas encontradas alrededor del prototipo y su respectiva información, en relación a los ataques se evidenció las vulnerabilidades existentes al trabajar en una red inalámbrica. El producto final es un prototipo desarrollado en un dispositivo que permite capturar datos y de acuerdo a su análisis, realizar distintos ataques de hacking ético, además cuenta con un sistema para guardar directamente los archivos generados en un almacenamiento externo, con el fin de agilizar el proceso de análisis, y de acuerdo a un desarrollo innovador en el campo de redes, generar conciencia que las conexiones inalámbricas están propensas a sufrir ataques de ciberseguridad en cualquier momento.

Palabras clave Análisis de datos, ESP32, Arduino, Wireshark

ABSTRACT

This technical project aims to develop a low-cost Sniffer application (hardware and software), according to the use of tools for data analysis, it seeks to capture and analyze each of the packets, which have a type of information from the Wi-Fi network in the IoT laboratory of the Salesian Polytechnic University. Through the creation of scenarios prioritizes attacks that can be performed with the information generated from the Sniffer, all this under the Methodology for the Analysis of Vulnerabilities, choosing the appropriate tools for implementation, which is why the Arduino development environment and the Wifislax tool were used for ethical hacking attacks. In the analysis part, each of the fields generated by the Wireshark tool file is detailed, according to this, the wireless networks found around the prototype and their respective information is evidenced, in relation to the attacks, the existing vulnerabilities when working in a wireless network were evidenced. The final product is a prototype developed in a device that allows capturing data and according to its analysis, perform different ethical hacking attacks, also has a system to directly save the files generated in an external storage, in order to streamline the analysis process, and according to an innovative development in the field of networks, generate awareness that wireless connections are prone to suffer cybersecurity attacks at any time.

Keywords Data analysis, ESP32, Arduino, Wireshark

INTRODUCCIÓN

Una red inalámbrica permite usar distintos dispositivos desde casi cualquier lugar, así como conectarse a otros ordenadores en la red o facilitar el acceso a Internet. Sin embargo, si la red inalámbrica no es totalmente segura, las personas se enfrentan a riesgos muy grandes en cuanto a seguridad se refiere.

ANTECEDENTES

A lo largo de los años, se han realizado investigaciones sobre seguridad inalámbrica en varios países con resultados alarmantes. Los sistemas de Internet de las Cosas *(*del inglés IoT*, Internet of Things)* que se producen profesionalmente, tienen un cierto nivel de seguridad de la información. En 2015 apareció en mercado el microcontrolador Sistema en un solo chip (del inglés SoC, *Sytems on Chip*) ESP32, se trata de un dispositivo de bajo costo que cuenta con un chip Wi-Fi y Bluetooth, (Barybin et al., 2019) las amenazas en cuanto a seguridad de información están relacionadas con la aparición de vulnerabilidades en la red inalámbrica Wi-Fi.

A nivel global fueron reportados 467,351 incidentes de ciberseguridad en 2019 según el estudio (Federal Bureau of Investigation, 2020), teniendo en cuenta que; 71% de los robos de información tuvieron como motivo obtener dinero; y 25% fueron con fines de espionaje (Verizon Enterprise Solutions, 2020).

Los ataques cibernéticos en Latinoamérica han aumentado un 24% en lo que va de año 2022, en comparación con los primeros 8 meses de 2021. México es el noveno país más afectado por el crimen cibernético, con 605 casos reportados en Latinoamérica. (Federal Bureau of Investigation, 2020)

Durante 2021, según Eset, en Ecuador hubo alrededor de 140 mil detecciones de exploits, cerca de 6 mil detecciones de ransomware y casi 8 mil detecciones de spyware, ocupando la séptima posición en detección phishing con el 5,1% en Latinoamérica. (DPL News, 2021)

PROBLEMA

El presente proyecto responderá a la siguiente problemática: ¿En qué medida ayudaría el análisis de hacking ético? La Universidad Politécnica Salesiana (del español UPS, Universidad Politécnica Salesiana) integró un laboratorio de IoT con sus respectivos servicios, el cual no ha sido sometido a ninguna técnica de "hacking ético" para detectar vulnerabilidades en cuanto a seguridad, pues este espacio presenta un nivel alto de riesgo en la información, la misma que puede ser alterada o en el peor de los casos sustraída en su totalidad, debido a que nunca se emplearon herramientas para la detección de vulnerabilidades en su desarrollo. Por tal motivo la UPS requiere llevar a cabo la aplicación de hacking ético para detectar fallos en la red inalámbrica del laboratorio IoT, con el objetivo de mejorar la seguridad y así disponer de una red de datos eficiente y segura ante distintos ataques.

Entre las diferentes causas que pueden estar originando este problema, se ha detectado las siguientes: i) Contraseñas predecibles, ii) Problemas de caídas por ataques externos en la red inalámbrica, iii) Desconocimiento de técnicas de seguridad informática iv) Bajo control de acceso a la red inalámbrica, v) Accesos no autorizados, vi) Perdida de información, vii) Intercepción de los datos por terceras personas, de acuerdo a esto se va abordar las tres primeras causas.

Los efectos que pueden traer este problema son diversos, entre los que se ha detectado están: i) Suplantación de identidad, el cual es un acto maligno que realizan los delincuentes utilizando sus cargos o nombres, en la identidad de otros, de forma oral o utilizando documentos

falsos. ii) Utilización de la red para realizar acciones ilegales. Por lo tanto, el proyecto contempla resolver los efectos descritos previamente, los cuales están relacionados con todas las causas definidas.

El presente proyecto tiene como finalidad teórica definir características primordiales del dispositivo a usar, de tal forma que se detallen tanto las funcionalidades como los periféricos que existen, teniendo en cuenta los diferentes entornos y lenguajes de programación en los que se puede desarrollar la aplicación. En cuanto a la parte práctica, la idea es implementar una aplicación en el SoC ESP32 mediante la cual pueda facilitar la aplicación del hacking ético y monitorización del tráfico de red.

JUSTIFICACIÓN

En los últimos años las redes inalámbricas se han convertido en una tendencia tecnológica usualmente utilizada para proveer conectividad en diferentes lugares como entidades, establecimientos, institutos, entre otros, lo cual genera una gran cantidad de tráfico de información que queda expuesta a las vulnerabilidades propias de la red.

Por este motivo se busca evidenciar las falencias(vulnerabilidades) que tienen las redes Wi-Fi del laboratorio IoT de la UPS mediante el uso de aplicaciones para el hacking ético, contiguo con el módulo SoC Esp32 simulando ataques para la extracción de información, ya que se demostró en (Barybin et al., 2019) que ESP32 puede ser utilizado con éxito en la adquisición de datos y el control de diversos dispositivos a través de redes inalámbricas.

Es por eso, que se busca implementar la manera de abaratar costos en la evaluación de seguridad de la red Wi-Fi, debido a que en el laboratorio se tuvo como prioridad la comunicación de los datos y no se puso igual énfasis en lo que conlleva la seguridad de la información.

El proyecto servirá de apoyo a diversos grupos beneficiarios como: i) *directos*, proporcionando información a los administradores de red acerca de las vulnerabilidades existentes; ii) *indirectos*, a los usuarios del laboratorio IoT de la UPS protegiendo su información y accesos no autorizados, y así evitar el robo de datos.

OBJETIVOS GENERALES Y ESPECIFICOS

Objetivo General:

Identificar las vulnerabilidades en la red Wi-Fi debido a la poca seguridad a los que se encuentran expuestos los usuarios mediante el SoC ESP32 en el Laboratorio IoT de la Universidad Politécnica Salesiana.

Objetivos Específicos:

- Realizar un estudio del estado del arte sobre las vulnerabilidades de red Wi-Fi.
- Crear escenarios de ataques de irrupciones controladas (hacking ético) en ESP32 para comprometer datos transmitidos en la red Wi-Fi.
- Analizar paquetes de datos con la herramienta Wireshark en cada uno de los escenarios.
- Interpretar los resultados de pruebas de funcionamiento de los escenarios creados.

METODOLOGÍA

Para la elaboración de este proyecto se busca seguir una metodología que vaya de acuerdo a los objetivos planteados, es por esto que según (Ferrer, R.) los procesos que tienen mayor peso se basan en la Metodología para el Análisis de Vulnerabilidades, esta comprende que el funcionamiento de la red inalámbrica debe ser totalmente independiente teniendo en cuenta que el análisis de las vulnerabilidades no debería influenciar en la red, más bien solo reconocer ciertas

grietas de seguridad existentes, un medio para lograr este objetivo es el hacking ético, por esto se realizarán pruebas en distintos días y a diferentes horarios donde el laboratorio IoT tengo un menor uso para evitar contratiempos con los usuarios.

- **Análisis de la Infraestructura**

Esta fase busca identificar cada componente de hardware y de software que resida en la infraestructura de la red inalámbrica.

- **Selección de Herramientas**

Esta fase busca herramientas que puedan detectar todo tipo de vulnerabilidades, teniendo en cuenta sus pro y contras, también cada requerimiento de hardware para su uso.

- **Generación de Ataques**

Esta fase establece los ataques posibles, de acuerdo al estado de la infraestructura de la red inalámbrica, y a su vez a cada una de las vulnerabilidades halladas. Es por esto que se define cada uno de los objetivos, y el proceso de cada una de las herramientas para llegar a conclusiones de cada caso.

- **Búsqueda de Vulnerabilidades en Infraestructura**

Esta fase rastrea vulnerabilidades de todas las versiones existentes en base a cada servicio ya instalado, como las de firmware de cada punto de acceso, todo esto se puede realizar con la ayuda de software para localizar las vulnerabilidades en la red Wi-Fi.

CAPÍTULO 1

En el primer capítulo, se realiza un análisis de las vulnerabilidades de seguridad encontradas en las redes Wi-Fi, que define completamente los tipos de ataques que enfrentan y cómo les afectan principalmente en las comunicaciones de una organización. El capítulo también desarrolla fundamentalmente los principales conceptos involucrados en el resto del trabajo que facilitan esta contextualización.

1.MARCO TEORICO / ESTADO DEL ARTE

En el trabajo "Análisis de vulnerabilidades de internet de las cosas en casos de estudio de intrusión y detección en el laboratorio de sistemas embebidos, UPS Q-SUR" realizado por Jácome María y Báez Henry en el año 2022, las vulnerabilidades en las redes crecen a medida que el mercado va en auge, es por esto que las instituciones o entidades se enfrentan a un mayor riesgo en cuanto a los ataques de seguridad informática.

El análisis de datos en una red ayuda a tener una mejor compresión de cómo las personas están interactuando con el lugar, es por esto que hay diferentes maneras de captar la transmisión de los datos debido a las constantes pruebas en cuanto a seguridad de cada uno de los dispositivos se refiere.

El objetivo del estudio realizado por (Babiuch et al., 2019) trata de las experiencias con el desarrollo de aplicaciones de los microcontroladores ESP32, la metodología utilizada es descriptiva y explicativa, ya que proporciona una revisión exhaustiva de las posibilidades de desarrollo de aplicaciones en esta plataforma en el ámbito de la medición y el procesamiento de datos. Los microcontroladores suelen conectarse con módulos IoT y otros sensores inteligentes y proporcionan datos al sistema superior. Este artículo también describe la implementación de la

aplicación con la versión de pantalla OLED conectada y con la placa de desarrollo ESP32 Wrover con pantalla integrada. Concluyendo que, para cada plataforma se ha descrito las ventajas, las recomendaciones para qué tipo de aplicación es adecuada la plataforma, y al mismo tiempo se fue comparando el nivel de requisitos previos para esa plataforma en términos de habilidades de software y experiencia del desarrollador.

En cuanto a (Wang & Yang, 2017) se revisa el estado del arte de las actuales herramientas de escaneo de vulnerabilidades de código abierto, por lo que la metodología utilizada fue la revisión bibliográfica de las principales investigaciones en artículos científicos relacionados con el tema. Se introduce un entorno de laboratorio virtual como parte del diseño de laboratorio. Se presenta en los laboratorios prácticos diseñados en detalle utilizando la herramienta de escaneo de vulnerabilidades OpenVAS. Concluyendo que, se revisa los resultados después de realizar los laboratorios prácticos en los cursos de ciberseguridad e identifica el trabajo futuro para áreas de investigación abiertas en estos campos.

El objetivo de estudio de (Yevdokymenko et al., 2017) es la defensa de la red y las info-comunicaciones a lo largo de la plataforma de ciberseguridad para profundizar en lo que ahora se conoce como "Hackeo Ético", la metodología usada tiene como fin diferentes fases: definir, analizar, discutir y resolver algunas de las amenazas más comunes y ampliamente difundidas y sus funcionalidades. Concluyendo que, según las vulnerabilidades que están actualmente inmersas en la mayoría de los incidentes de amenazas, se busca la manera de llegar al desarrollo de nuevas técnicas para tener más control en la eficacia de tales problemas.

Para (Patil et al., 2017) el hacking ético debe practicarse. Requiere conocimientos básicos de redes y ciberseguridad. Es por esto que se utilizó la metodología cuantitativa, de acuerdo al estudio descriptivo. Este documento colabora con la mayoría de las terminologías básicas

relacionadas con el hacking ético. Ofrece una breve información sobre quién es un hacker ético y por qué es necesario que el mundo lo aprenda. También describe cómo se lleva a cabo el hacking y cuáles son las diferentes herramientas y tecnologías utilizadas. Concluyendo que, según la recopilación de la información relevante se puede ofrecer una comprensión básica en el contexto del hacking ético.

Mientras que, para (Barybin et al., 2019) se propuso el modelo físico de un sistema IoT hecho a mano que incluye un dispositivo para medir la temperatura basado en ESP32, una red doméstica WiFi y una interfaz web, y se implementó a escala de laboratorio. La metodología utilizada fue descriptiva y explicativa, ya que se logra identificar diferentes escenarios y así conseguir que el resultado del experimento basado en este modelo para intentar obtener acceso no autorizado a los datos transmitidos fuese exitoso. El escenario de ataque se formuló y consta de cuatro etapas: obtener acceso no autorizado a una red, interceptar y analizar el tráfico de red, crear un cliente ESP32 falso y desconectar el ESP32 original de un servidor. Concluyendo que, el atacante, que tiene los conocimientos y habilidades básicas para trabajar con herramientas comunes de hacking de redes inalámbricas y un conocimiento básico de ESP32 de acuerdo a habilidades de programación de ESP32 pueda acceder al sistema y enviar información falsa a la interfaz web, es por esto que se busca reducir la probabilidad del escenario propuesto utilizando TCP en lugar de UDP.

En cuanto a (Trabelsi & Ibrahim, 2013) los ataques de denegación de servicio distribuido (del inglés DoS, Denial of Service) son temas importantes para los cursos de seguridad que enseñan técnicas de hacking ético y detección de intrusiones. Este artículo presenta un caso de estudio de la implementación de ejercicios prácticos ofensivos completos de laboratorio sobre tres ataques DoS comunes. La metodología utilizada fue explorativa y descriptiva, ya que los ejercicios enseñan

a los estudiantes a realizar prácticamente los ataques DoS en un entorno de laboratorio de red aislado. Concluyendo que, de acuerdo al estudio y discusión de algunas cuestiones éticas y legales relacionadas con la enseñanza del hacking ético, se enumera los pasos que las escuelas y los educadores deben tomar para mejorar las posibilidades de tener un programa de seguridad de la información exitoso y libre de problemas.

1.1 Esp32

ESP32 es considerado un microcontrolador, el cual se utiliza para múltiples aplicaciones en base a su nivel alto de integración en cuanto a Wi-Fi, Bluetooth y Bluetooth Low Energy, es por esto que se permite una potencia alta en el desarrollo del módulo genérico.

En cuanto al consumo de energía, el nivel es muy bajo ya que permite la inclusión de funciones de ahorro, incluyendo diferentes modos de operación y la sincronización del reloj. Con ello, se busca actuar de forma directa desde una trama de sensores de potencia mínima hasta 8 adaptaciones “más exigentes como transmisión y decodificación MP3, y codificación para la voz” (Espressif Inc, 2019c, pág. 1).

1.2 Arduino

Es una plataforma de electrónica que se basa en software libre, cuyos principios se basan en el fácil uso del software y hardware. Esta herramienta permite generar diferentes tipos de microordenadores de una sola placa, por lo que se puede utilizar para el desarrollo en base a elementos independientes, sino a su vez conectarse a otros dispositivos teniendo interacción con diferentes programas, es decir, una manera fácil de crear cualquier proyecto interactivo.

Es por esto que, la comunidad que se tiene de esta plataforma tiene un constante desarrollo en base a ideas innovadoras de utilizar los dispositivos, desde prototipos a gran escala hasta

productos finales relacionados con el diseño, como la domótica, que son un tema de gran importancia en la actualidad y sobre todo para este proyecto técnico. (Álvarez Carulla, 2021)

1.3 Wireshark

Es un programa el cual sirve como analizador de tráfico para redes informáticas, su función es capturar paquetes de la red, registrar cada uno de los datos en línea y así analizar los mismos para presentar toda la información que se pudo generar con sus respectivos detalles, es por ello que tiene compatibilidad con variedad de protocolos.

Un analizador de paquetes de red es como un dispositivo de medición utilizado para examinar lo que está pasando en el interior de un cable de red, permitiendo así dar una respuesta rápida a intrusiones no deseadas. (Ndatinya et al., 2015)

1.4 Paquete de Datos

Un paquete de datos es considerado como un enlace de información, es por esto que se encuentran inmersas en cada una de las redes que existen en la modernidad, todo esto con el fin de realizar una comunicación adecuada para cada entorno.

Posee entre sus elementos: cabecera, es la encargada de reunir la información general, para que a su vez pueda ser manejada y transmitida mediante un paquete desde el emisor hasta un receptor destinado; área de datos, es la encargada de almacenar todos los datos necesarios para su transmisión; y la cola, es la encargada de detectar cualquier tipo de error que se presente mediante el uso de códigos para este fin. (Ndatinya et al., 2015)

1.5 Wifislax

Es un programa, utilizado como una herramienta de auditoría de la red inalámbrica y, permite descartar deficiencias y mejorar el nivel de seguridad, post resultados, con el objetivo de prevenir para no ser víctimas de cualquier vulneración en alguna red, ya que está muy de moda este proceso debido a su gran facilidad de manejo. Distribuido por Linux y considerado como un sistema operativo o detector de amenazas de red. (Vargas, Guarda et al. 2019)

1.6 Análisis de Datos

El análisis de datos es comprendido como la detección de paquetes, y a su vez el respectivo análisis de cada uno de los protocolos encontrados, el proceso que sigue es la recolección e interpretación de todos los datos en tiempo real, por lo que esto ayuda a comprender de mejor manera lo que está ocurriendo en la red. Para realizar el análisis de paquetes generalmente se utiliza Wireshark, la herramienta que captura datos en demasía a través de cable.

Según el contexto de (Martínez 2011) el analizar datos sirve para comprender cada una de las características de la red, captando que es lo que ocurre y quién está utilizando la disponibilidad de los recursos, para así identificar las horas en la que la red se encuentra en su máximo pico, evidenciando algún posible ataque malicioso. Es por esto que existe la necesidad de solucionar eficazmente el tráfico de la red en base a su funcionamiento normal.

1.7 Hacking Ético

En 1961 fue cuando se dio comienzo a la cultura hacker, desde ese entonces se adoptó la computadora como el artefacto tecnológico de preferencia, y así es como surgen los distintos indicios de lo que tiene que ver con programación, estos primeros años fueron descritos en la primera parte del libro de (Steven Levy, Hackers).

Según (Users, 2011, p. 48) el hacking ético se basa en todos los conocimientos sobre informática y ciberseguridad, para que, en base a toda la información receptada de la red de un sistema, se pueda hallar vulnerabilidades de todo tipo o cualquier falla en la seguridad, con el objetivo de proteger el sistema y la información respectiva de los intrusos.

El autor (Agé, Baudru y Crocfer, 2015, p. 134), lo describe como un pentester, con el objetivo de medir el nivel de seguridad de un sistema de información, por medio de ataques o pruebas de penetración de seguridad, teniendo en cuenta que se busca la corrección de las vulnerabilidades o fallos encontrados en el sistema al momento de realizar cada una de las pruebas.

Es importante conocer que un hacker ético sigue un código estricto de comportamiento, esto le permite utilizar sus habilidades para el beneficio de una institución u organización que haya solicitado sus servicios, cabe destacar que frecuentemente deben renovar sus conocimientos, invertir el suficiente tiempo y paciencia para investigar como también dominar herramientas de pruebas. (Users, 2011, p. 48).

1.8 Ataque Sniffer

Es un software que está diseñado específicamente para redes informáticas, es una de las maneras más populares que utilizan los atacantes, debido a que logra capturar y analizar los paquetes o más bien todo el tipo de comunicación que se realiza, los atacantes denominados rastreadores tienen la facilidad de espiar cada uno de los datos en cuestiones de nodos o enlaces, a su vez pueden monitorear el estado de la red, con ello su objetivo es sustraer datos confidenciales en basa a cada usuario y su respectiva contraseña. Este tipo de proyectos son útiles para evidenciar problemas de seguridad de las comunicaciones y tener un mayor control de la red. (Zhao et al., 2016)

1.9 Ataque DoS

Los ataques de denegación de servicio son ataques informáticos con el objetivo de inhabilitar un sistema, a través de la saturación de peticiones en un corto tiempo, con el fin de colapsar el servidor, tomando en cuenta que las aplicaciones en línea tienen un límite de capacidad, las vuelve un claro objetivo de ataque en un tiempo determinado. (Palo Alto Networks, 2019).

Un sitio web es como una puerta de acceso, solo puede atender a un número limitado de personas a la vez, por lo que si recibe más solicitudes de las que puede manejar, el servicio se bloquea y no permite entrada ni salida de ningún tipo. Estos ataques aparecieron alrededor de 1995, se caracterizaron por el uso de múltiples dispositivos en lugar de uno solo, con el fin de crear un ataque más poderoso en conjunto. (Espinosa, 2017).

1.10 Protocolo EAPOL

Es un protocolo de autenticación extendido basado en LAN, el cual es un mecanismo muy común para el proceso de autenticación, en gran mayoría su uso es para las redes inalámbricas o dichas conexiones que se realizan de punto a punto. Su punto de partida es el mismo protocolo de autenticación extendido (del inglés EAP, Extensible Authentication Protocol), que se puede implementar tanto para redes locales inalámbricas como para las redes locales cableadas, por lo que se le considera un marco de autenticación muy favorable. Gracias a esto, mediante funciones previstas se puede mejorar el mecanismo para que sea el deseado por todos, se los conocen como métodos EAP y hoy en día existen varios métodos diferentes según las necesidades.

CAPÍTULO 2

En el segundo capítulo, se realiza el diseño y desarrollo del prototipo, por ende, se especifica cada uno de las componentes en base a software y hardware, todas las conexiones y las funciones que ayudan a su funcionamiento. El capítulo también desarrolla fundamentalmente las principales herramientas involucradas en el resto del trabajo que facilitan su implementación.

2. DISEÑO Y DESARROLLO

2.1 DISEÑO

2.1.1 Hardware

2.1.1.1 Esp32 Heltec Wi-Fi Kit 32

WiFi Kit 32 fue fabricada por Heltec Automation, es una placa diseñada para todo lo que tiene que ver con desarrollo IoT, se integra fácilmente en ESP32, posee un sistema que gestiona la batería como es Litio y Polímero (del español Li-Po, Litio y Polímero) y Diodo orgánico emisor de luz (del inglés OLED, Organic Light-Emitting Diode). Sus principales funciones se desarrollan en el campo de fabricantes IoT.

2.1.1.2 Componentes

Heltec ESP Arduino viene conformado por varios kits que ayudan al desarrollo de aplicaciones, en este caso se utiliza WiFi Kit 32, el cual cuenta con determinadas características como:

- SoC: ESP32.
- Interfaz micro USB.
- Interfaz de batería SH1.25-2.
- Antena PCB Wi-Fi e integración de Bluetooth 2,4 GHz.

- Pantalla OLED.
- Chip integrado CP2102 USB a puerto serie.
- Es compatible con Arduino, entorno de desarrollo.

Figura 1

Parámetros Técnicos

Resource	Parameter	
Master Chip	ESP32(240MHz Tensilica LX6 dual-core + 1 ULP, 600 DMIPS)	
Wireless Communication	Wi-Fi	Bluetooth
	802.11 b/g/n (802.11n up to 150 Mbps)	Bluetooth V4.2 BR/EDR and Bluetooth LE specification
Hardware Resource	UART x 3; SPI x 2; I2C x 2; I2S x 1; 12-bits ADC input x 18; 8-bits DAC output x 2; GPIO x 22，GPI x 6	
FLASH	4MB(64M-bits) SPI FLASH	
RAM	520KB internal SRAM	
Interface	Micro USB x 1; 18 x 2.54 pin x 2	
Maximum Size (Including protruding parts such as switch and battery compartment)	51 x 25.5 x 10.6 mm	
USB to Serial Chip	CP2102	
Battery	3.7V Lithium(SH1.25 x 2 socket)	
Solar Energy	x	
Battery Detection Circuit	√	
External Device Power Control (Vext)	√	
Display Size	0.96-inch OLED	
Working Temperature	-40~80℃	

Nota. *Parámetros técnicos detallados de ESP32. Fuente: (Heltec Automation, 2019)*

Figura 2

Características eléctricas

Electrical Features	Condition	Minimum	Typica	Maximum
Power Supply	USB powered (≥500mA)	4.7V	5V	6V
	Lithium powered (≥250mA)	3.3V	3.7V	4.2V
	3.3V (pin) powered (≥150mA)	2.7V	3.3V	3.5V
	5V (pin) powered (≥500mA)	4.7V	5V	6V
Power Consumption(mA)	WIFI Scan		115mA	
	WIFI AP		135mA	
Output	3.3V pin output			500mA
	5V pin output (USB powered only)		Equal to the input current	
	External device power control (Vext 3.3V)			350mA

Nota. *Especificaciones eléctricas detalladas de ESP32. Fuente: (Heltec Automation, 2019)*

2.1.1.4 Pines

Figura 3

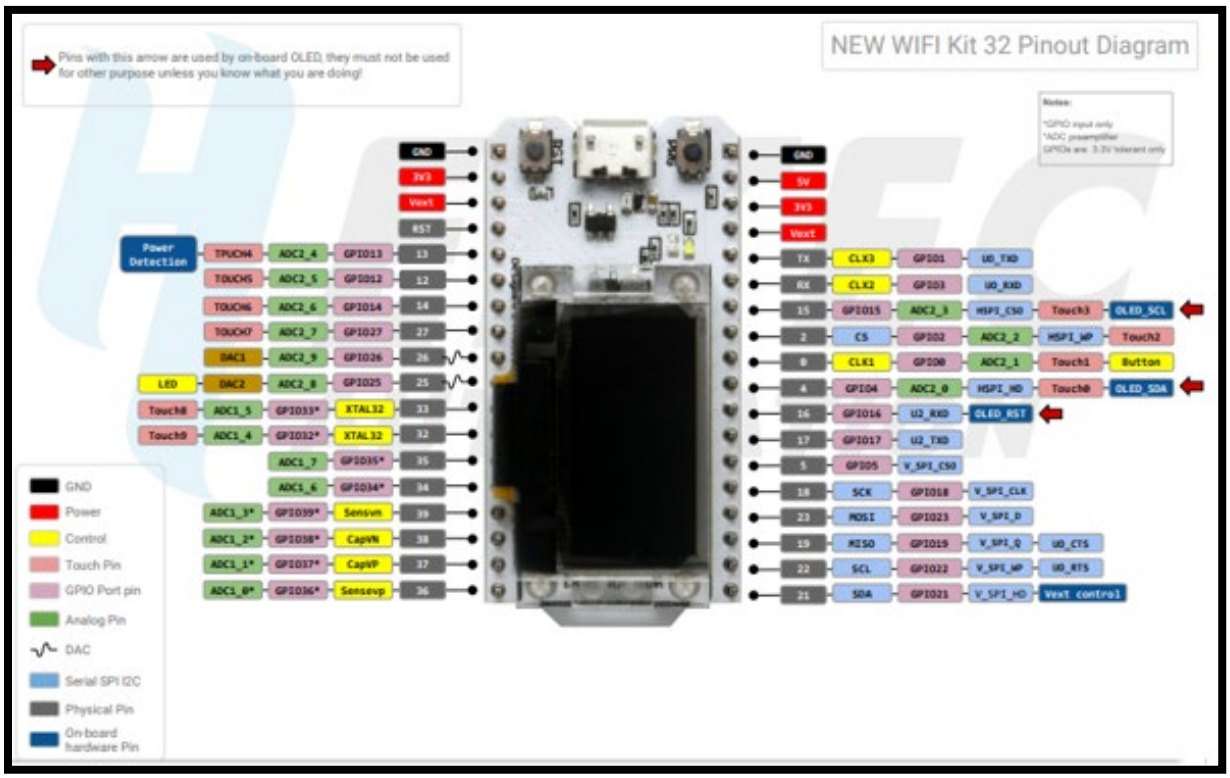
Conexiones de pines en arduino

COMPONENTE	PIN	PUERTO
Botón	Pin 5	GPIO5
Tarjeta SD	• Pin 1 • Pin 2 • Pin 8 • Pin 14 • Pin 15 • Pin 16	• GND (Tierra) • VCC 5V (Voltaje de corriente continua) • CS (Selección de chip) • SCK (Señal de reloj del bus • MISO (Entrada maestra Salida esclava) • MOSI (Salida maestra Entrada esclava)

Nota. *Especificaciones eléctricas detalladas de ESP32. Elaborado por: Guallasamín Alexis y Santos Gabriel*

Figura 4

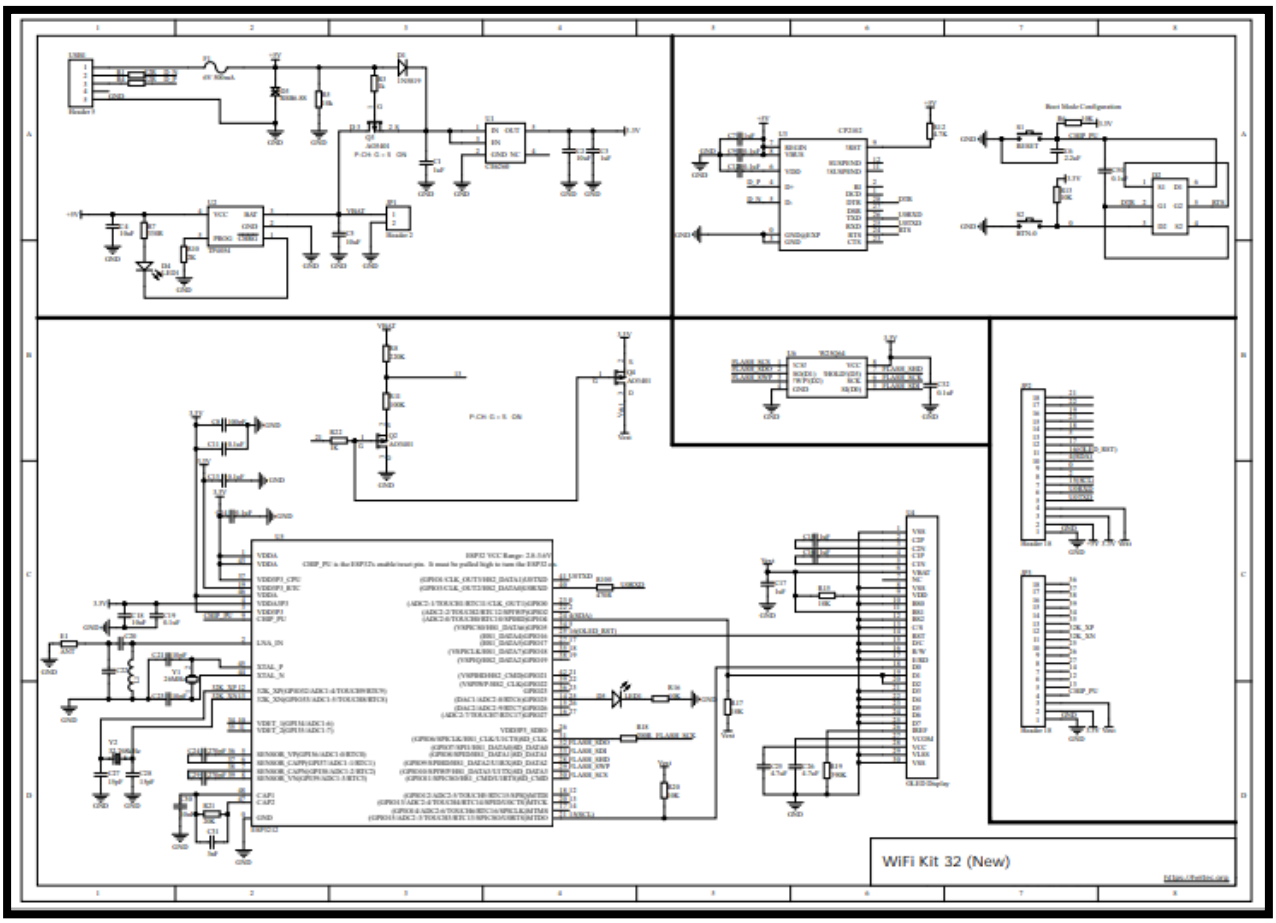
Diagrama de salida de 32 pines del kit Wi-Fi



Nota. *Diagrama de Salida WIFI Kit 32. Fuente: (Heltec Automation, 2019a)*

Figura 5

Diagrama esquemático del kit Wi-Fi 32



Nota. *Diagrama esquemático WIFI KIT 32. Fuente: (Heltec Automation, 2019b)*

2.1.1.3 Costo

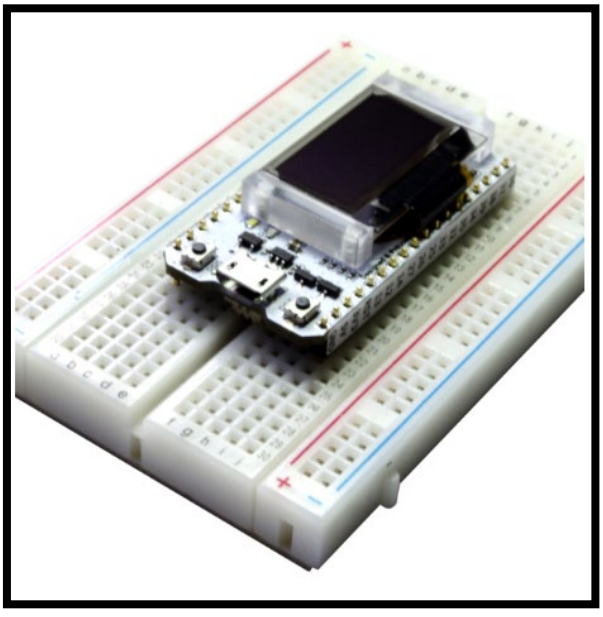
Entre muchas opciones de microcontroladores se encuentra el ESP32, el motivo de ser el dispositivo escogido para la realización del proyecto es porque se la considerada una plataforma, es decir, es una placa de desarrollo de bajo costo, la misma que tiene características esenciales que la hacen muy competitiva en el mercado como capacidades Wi-Fi, Bluetooth y Bluetooth de Baja Energía (del inglés BLE, Bluetooth Low Energy).

2.1.2 Software

2.1.2.1 Instalación Heltec ESP32 entorno de Desarrollo Arduino. Se toma como entorno de desarrollo Arduino, se realiza las configuraciones correspondientes para empezar con el diseño de la aplicación en base al SoC ESP32.

Figura 6

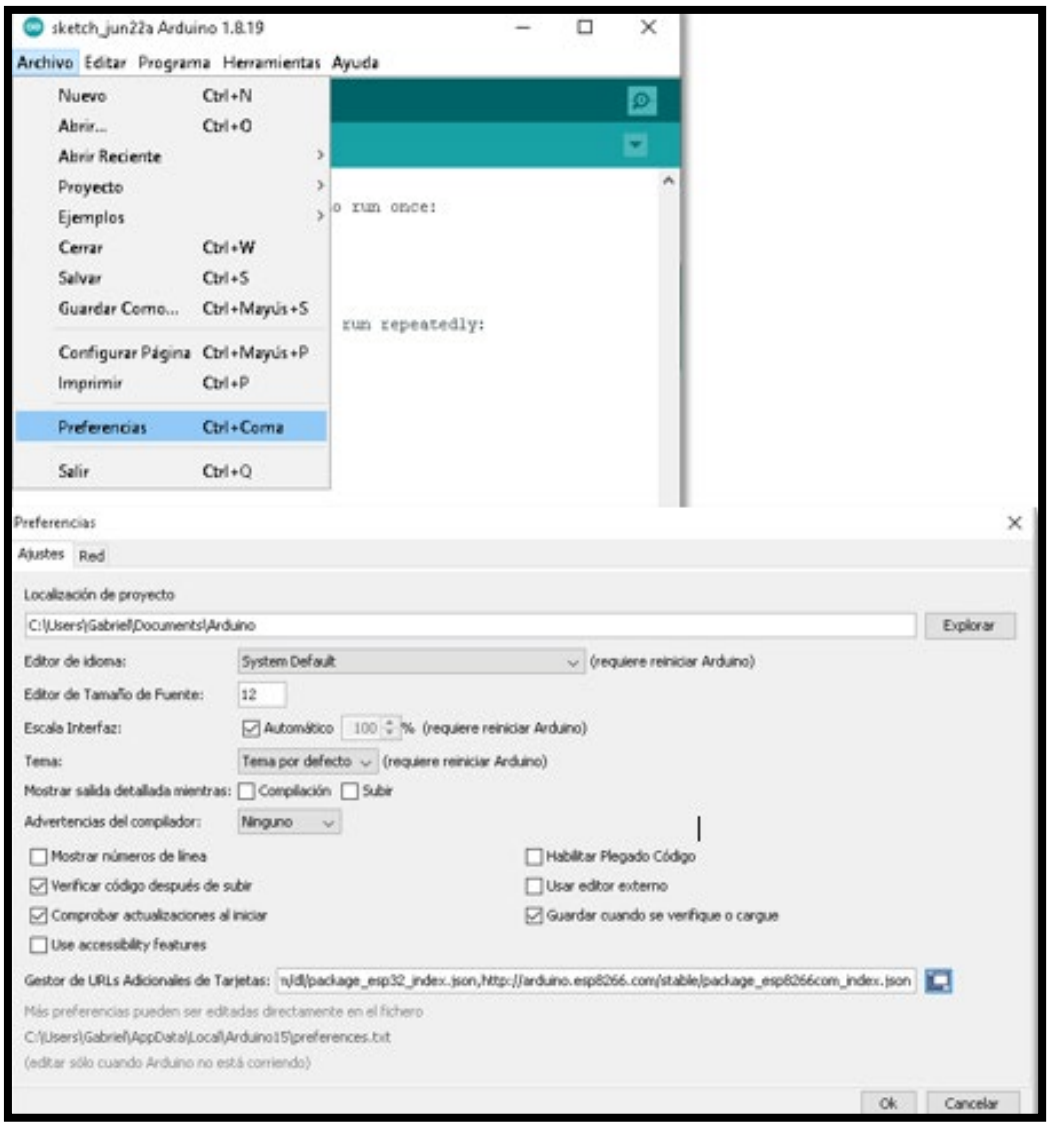
Dispositivo Heltec ESP32



Nota. Dispositivo Heltec ESP32. Elaborado por: Guallasamín Alexis y Santos Gabriel

Figura 7

IDE de Arduino



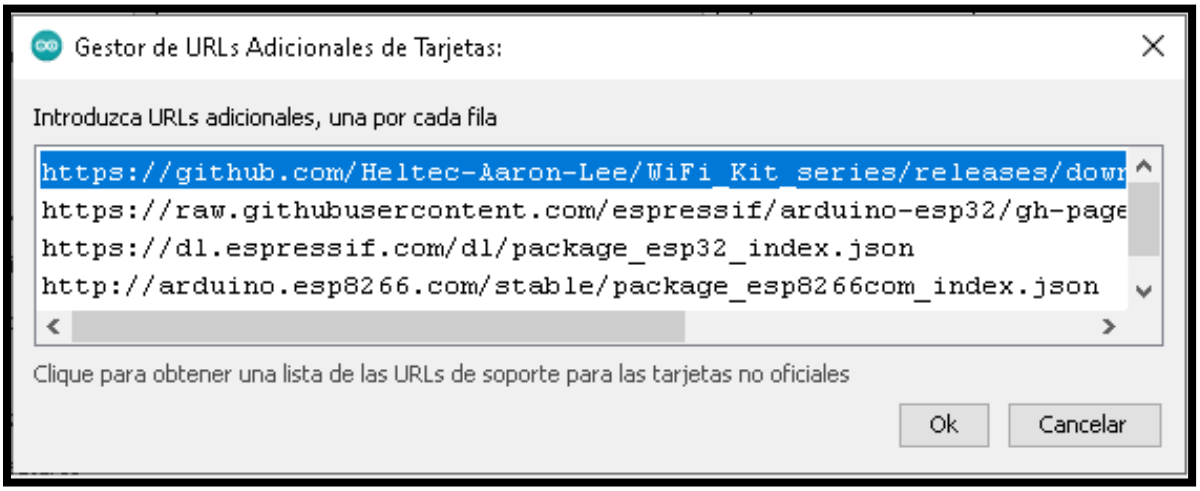
Nota. *Preferencias de IDE Arduino. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.1.2.2. Paquetes. Para que se pueda descargar el paquete de la tarjeta Heltec ESP32 con la que se va a trabajar, se coloca la siguiente URL, la cual contiene todos los archivos que se necesita para su funcionamiento.

Repositorio: https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series/releases/download/0.0.5/package_heltec_esp32_index.json

Figura 8

Paquetes

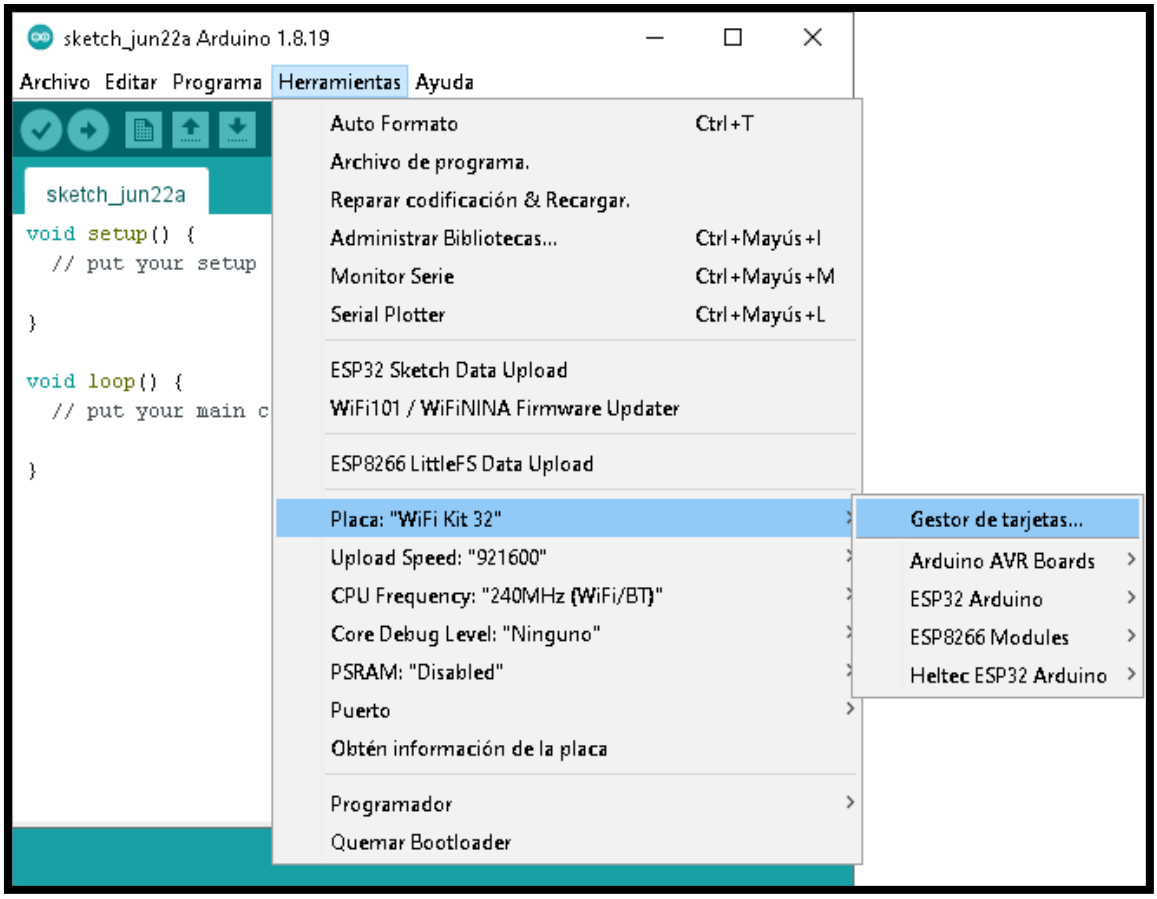


Nota. *Apartado de Gestor de URLs Adicionales de Tarjetas. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.1.2.3. Placas. Tiene como objetivo mostrar el soporte del tipo de placas que se tiene.

Figura 9

Placas

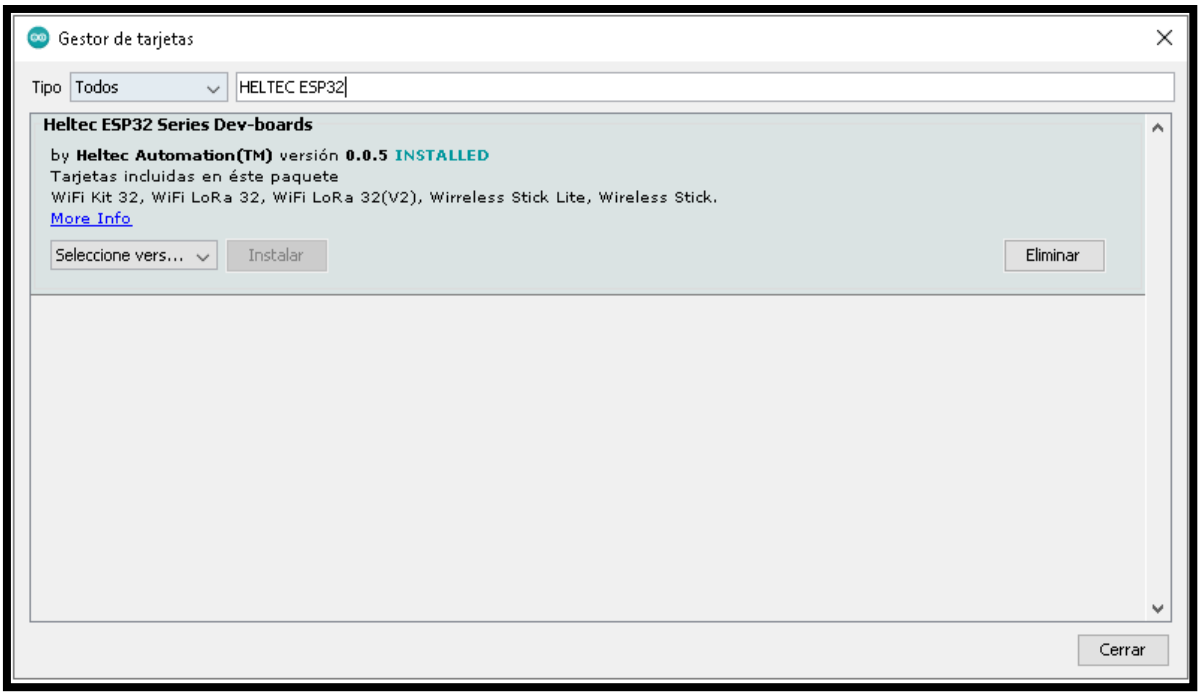


Nota. *Lectura de placa conectada WiFi Kit 32. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.1.2.4. Gestor de tarjetas. El gestor de tarjetas se encuentra disponible en el menú de herramientas, y su vez tiene la facilidad de instalar distintos tipos de placas que se puede llegar a necesitar, de acuerdo a lo requerido.

Figura 10

Instalación Heltec ESP32

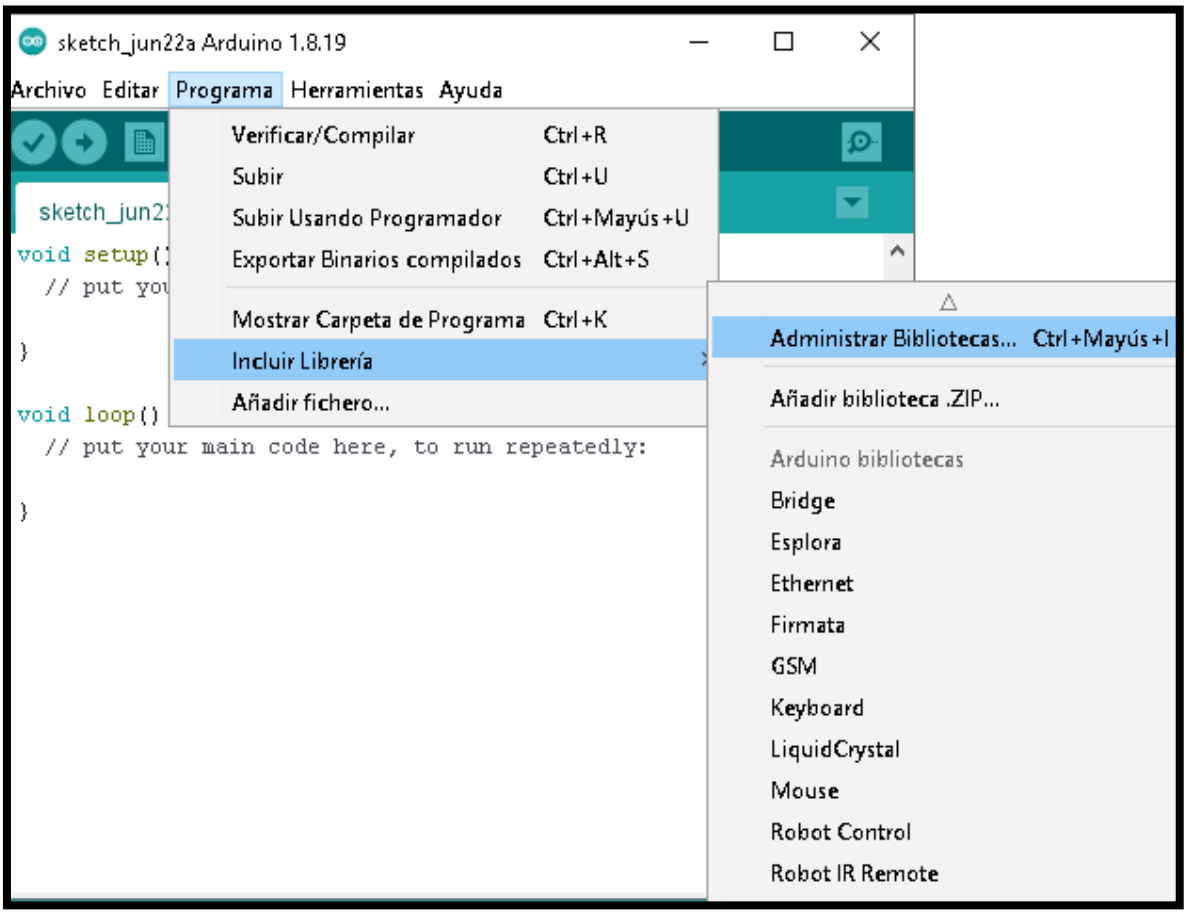


Nota. *Instalación de una nueva tarjeta de acuerdo a Heltec ESP32. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.1.2.4 Gestor de Librerías. Este gestor permite incluir, borrar y actualizar cualquiera de las librerías que se encuentran disponibles en el IDE.

Figura 11

Gestor de Librerías



Nota. *Administrar las librerías necesarias. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.1.2.5 Instalación Librería Heltec ESP32. Buscar Heltec ESP32 Dev para el inicio de su instalación, con ellos se obtiene cada una de las librerías necesarias para su correcto funcionamiento.

Figura 12

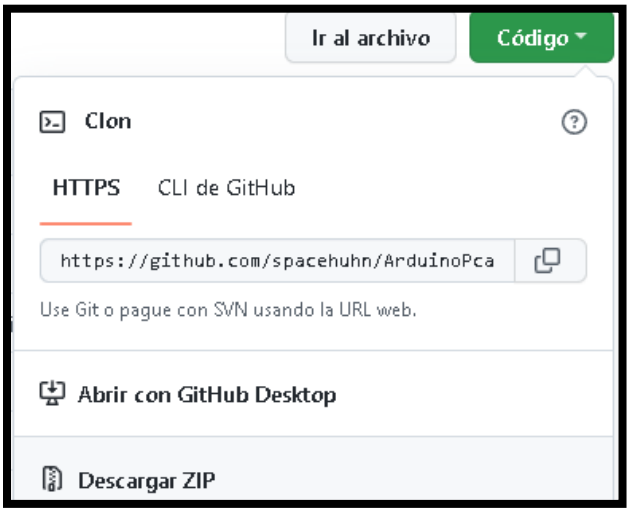
Librerías

Nota. *Instalación de librería Heltec ESP32 para su correcto funcionamiento. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.1.2.6 Repositorio Librería PCAP. Identificar librería requerida PCAP del repositorio de GitHub, dada por la siguiente URL: https://github.com/spacehuhn/ArduinoPcap

Figura 13

Repositorio

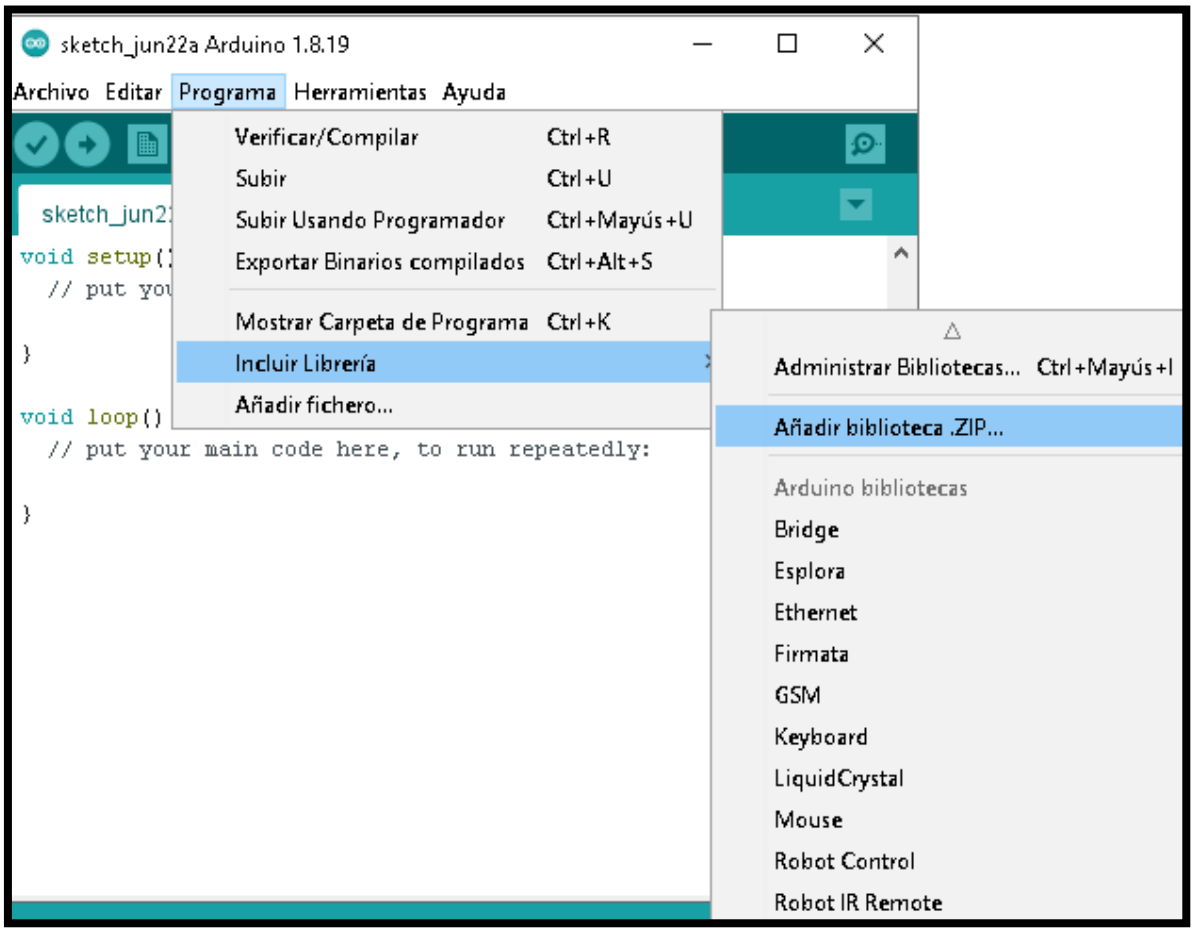


Nota. *Repositorio de librería PCAP. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.1.2.7 Descargar librería externa. Para utilizar una librería externa, se debe descargar toda la carpeta comprimida para su funcionamiento.

Figura 14

Librería Externa

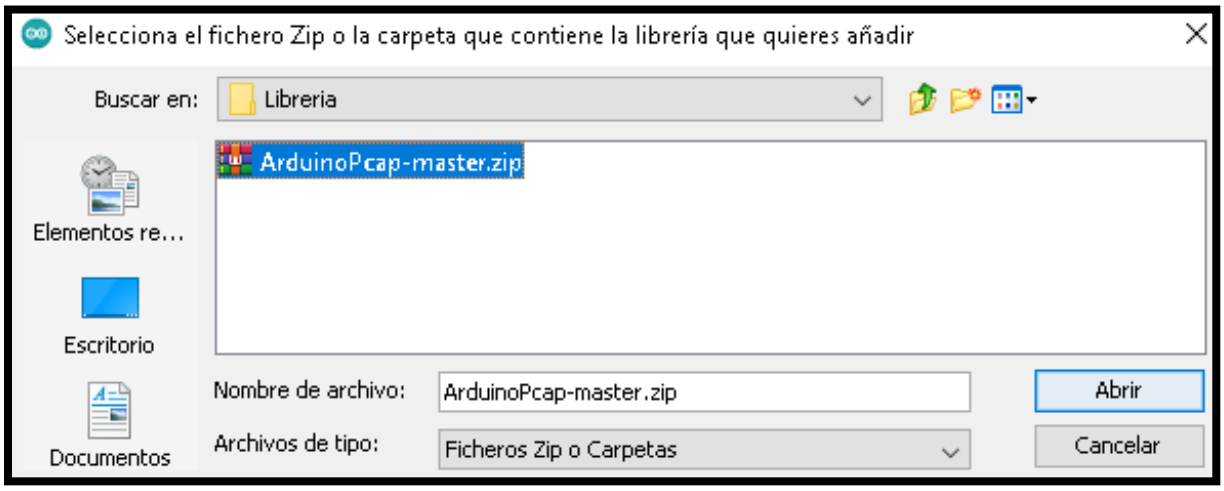


Nota. *Añadir carpeta comprimida de la librería descargada. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.1.2.8 Añadir fichero de librería requerida. Se agrega la librería ArduinoPcap-master.Zip por medio de la biblioteca del IDE para su uso.

Figura 15

Ingreso de librería externa a Biblioteca



Nota. *Selección de la librería externa que se encuentra en una carpeta comprimida. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.2 DESARROLLO

En relación con el desarrollo de la aplicación, se tiene como base esencial los monitores de paquetes Wi-Fi, los cuales son considerados aquellos programas informáticos o muchas veces una pieza de hardware aparte, de tal forma que su objetivo es obtener un registro del tráfico de red Wi-Fi mediante la intercepción de cualquier tipo de señal. Existe la posibilidad de que mediante herramientas de monitorización de paquetes se puede reconocer que tipo de datos se envían de un dispositivo a otro, todo esto teniendo en cuenta que es un medio inalámbrico, y así usar los datos que se obtienen para un propósito determinado.

Es por esto que, la aplicación que se genera permite observar cada uno de los paquetes de datos que están volando por medio del aire, teniendo en cuenta que, para la captación de los mismos y un mejor resultado, se lo hace por medio de tiempo real, dando a entender que se genera resultados de acuerdo a los paquetes de Wi-Fi que se envían por segundo y por ende diferenciar el canal en que se encuentran cada uno de los datos encontrados. De acuerdo a toda la información captada se puede visualizar el resultado en una pequeña pantalla OLED, la cual genera gráficas de todos canales captados y sus correspondientes paquetes de datos, mediante una microSD se puede guardar un archivo con extensión pcap, el cual contiene todo el flujo de paquetes interceptados de la red inalámbrica, y así analizar todos estos utilizando diferentes herramientas como Wireshark.

2.2.1 Librerías

- **FreeRTOS.h:** Es un sistema operativo que tiene como función administrar cada uno de los recursos de hardware, y a su vez controlar el tiempo de ejecución de todas las diferentes tareas que son implementadas en un determinado microcontrolador, teniendo en cuenta que todo este proceso lo realiza en tiempo real en base a las multitareas que puede integrar. Se encuentra desarrollado en lenguaje ANSI-C, y está distribuido de forma gratuita bajo licencia de "Código Abierto", incluyendo de una u otra forma aplicaciones comerciales para distintos casos. En la actualidad FreeRTOS posee puertos para diferentes arquitecturas y marcas, de acuerdo a cada una de las necesidades. (Trabelsi & Ibrahim, 2013)

- **Wifi Library (esp_wifi.h, esp_wifi_types.h, esp_system.h, esp_event.h, esp_event_loop.h):** Esta librería tiene como finalidad dar un soporte específico al momento de la configuración y monitorización de cada una de las funcionalidades de la red Wi-Fi ESP32. Según (Trabelsi & Ibrahim, 2013) existen diferentes modos en los que pueden actuar como son:

- Modo Estación (Modo Cliente Wi-Fi). El SoC ESP32 está conectado al punto de acceso.

- Modo AP (Modo Punto de Acceso), estas estaciones están conectadas al ESP32.
- Modo de coexistencia estación/AP (ESP32 es tanto un punto de acceso como una estación conectada a otro punto de acceso).
- Varios modos de seguridad anteriores (WPA, WPA2, WEP, etc.)
- Escaneo de puntos de acceso (escaneo activo y pasivo).
- Modo Promiscuo, supervisa los paquetes Wi-Fi IEEE802.11.

- **nvs_flash.h:** El almacenamiento no volátil (del inglés NVS, Non-volatile storage) es una biblioteca que se utiliza para almacenar valores de datos en la memoria flash del ESP32, razón por la cual los datos se almacenan en la memoria de tal manera que no haya problema con que se apague o se reinicie el ESP32. Los elementos representados por pares de datos de identidad se validan antes de escribirse para que puedan almacenarse en flash utilizando un búfer circular administrado por FIFO, es por esto que siempre comienza con una escritura de datos, seguida de una escritura de metadatos. No ocurre ninguna escritura en la memoria flash si es que el par id-datos no ha cambiado de alguna forma. NVS tiene un mecanismo de protección para evitar quedar atrapado en un bucle infinito eliminando páginas flash cuando el espacio libre es limitado. (Trabelsi & Ibrahim, 2013)

- **Stdio.h:** El encabezado estándar de entrada y salida (del inglés STDIO.H, standard input-output header) es el archivo de cabecera que se basa en la biblioteca de C y que de igual forma es compatible con C++, en sus funciones contienen distintos apartados como definiciones de macros, constantes, declaraciones de funciones en base a cada una de las operaciones de entrada/salida, y se encuentra su aplicación garantizada en cualquier tipo de plataforma mientras exista un compilador de C, en otro caso, las aplicaciones pueden usar solo los requerimientos de entrada/salida en base al sistema operativo correspondiente, estas funciones pueden tomar

diferentes conceptos como: manipulación de archivos, y manipulación de entrada/salida. Todas las funciones que se desarrollan en base al lenguaje C y sus derivaciones, se declaran esencialmente en cabeceras de fichero, para que se pueda hacer uso de cada una de las funciones declaradas que existen en su interior. (Avella, R., 2010)

- **Cstddef:** Este es un archivo perteneciente a la biblioteca estándar de C++, el cual debe incluirse en cualquier programa que lo necesite, omitiendo el archivo de implementación "*.h", su propósito es garantizar mediante enlaces externos que los nombres declarados en los archivos de cabecera de la biblioteca estándar de C, se declaren en el espacio de nombres estándar. (Microsoft, 2021)

- **Wire.h:** Esta librería permite la comunicación al Arduino mediante un módulo interno i2c, abarca dos formas de realizar este proceso, como maestro hacia otros dispositivos, el cual dentro de su descripción dicta que son los que inician y a su vez coordinan la comunicación, o pueden permanecer como esclavos, los mismos que esperan que el maestro tome el sentido de la comunicación, dado que reciben peticiones y responden con datos requeridos. Usa dos líneas: SDA pertenece a datos y SCL pertenece a reloj. La librería está considerada en el IDE Arduino, siendo esta el entorno de desarrollo, es por esto que está incluido por defecto y no se necesita su previa descarga. (Arduino, 2022)

- Preferences.h: Es una biblioteca exclusiva de Arduino en base a ESP32, la cual almacena distintos tipos de datos en forma de pares, como clave-valor, donde la clave es un elemento de los datos con su respectivo identificador y el valor son los datos respectivamente, todos estos forman parte del espacio de nombres. Los espacios de nombres están limitados por un máximo de quince caracteres, es por esto que se almacenan en un fragmento de flash principal, como tipo de datos y subtipo de NVS, lo cual puede retener datos después de un reinicio o un corte de energía, se

reservan 20 KB para las preferencias, lo que es mejor para almacenar muchos valores pequeños en lugar de unos pocos grandes. Cuando se instala por primera vez la placa ESP32 se obtiene de igual forma la biblioteca en el IDE de Arduino, la cual es utilizable por todas sus variantes en base a la placa. Esta biblioteca puede guardar los siguientes tipos de datos: char, short, int, long, float, double, bool, string, bytes, entre otras, algunas funciones de guardado de datos que posee son: credenciales de red, claves API, valores de umbral o a su vez visualizar estados de un GPIO, entre otras. (Trabelsi & Ibrahim, 2013)

- Heltec.h: Brinda toda la funcionalidad que necesita para poder utilizar la pantalla integrada en el Wifi kit 32, es por esto que en su uso también interviene el marco de desarrollo Heltec ESP32, ya que cuando se realiza la instalación automáticamente se puede obtener la biblioteca necesaria. (Hietala, H., 2020)

- **SD.h:** La librería permite que Arduino interactúe con tarjetas digitales seguras (de inglés SD, Secure Digital) a través del protocolo de Interfaz de Periféricos en Serie (del inglés SPI, Serial Peripheral Interface). Aunque hay tres tamaños diferentes, todos funcionan a cierta capacidad, su compatibilidad se presenta con sistemas de archivos FAT16, y por otro lado en tarjetas estándar SD y tarjetas SDHC se aplica con FAT32, se debe tener en cuenta que para verificar el correcto funcionamiento del dispositivo de prueba se debe aplicar las funciones necesarias de la librería. Después de conectar la tarjeta de memoria SD a la interfaz SPI de la placa Arduino, tiene la opción de crear y leer/escribir en los archivos requeridos, y a su vez interactuar con cada uno de sus directorios. De acuerdo al uso que tiene de SPI, los pines deben ser los que corresponden en el Arduino, en base a esto se puede conectar a cualquier pin digital que se encuentre libre, excepto SS/CS, existen limitantes que los archivos admitidos tienen en la biblioteca, es por eso que cada archivo tiene un tamaño máximo de 4 GB, por lo que la capacidad máxima admitida es de 32 GB.

Esta librería ya se localiza en conjunto con la descarga del IDE Arduino, por lo tanto, se podría hacer el uso en cada una de sus placas correspondientes. (Arduino, 2022)

Figura 16

Bibliotecas externas incluidas

```
#include "freertos/FreeRTOS.h"
#include "esp_wifi.h"
#include "esp_wifi_types.h"
#include "esp_system.h"
#include "esp_event.h"
#include "esp_event_loop.h"
#include "nvs_flash.h"
#include <stdio.h>
#include <string>
#include <cstddef>
#include <Wire.h>
#include <Preferences.h>
#include "heltec.h"
#include "FS.h"
#include "SD.h"
#include "Buffer.h"
```

Nota. *Declaración de cada una de las bibliotecas externas que se incluyen en la aplicación. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.2.2 Configuración Inicial

2.2.2.1 Definición de variables: Se define variables con nombre y un valor constante antes de compilar el programa.

Figura 17

Definición de variables

```
/* ===== configuración de compilación ===== */
#define MAX_CH 14       // 1 - 14 canales
#define SNAP_LEN 2324   // longitud máxima de cada paquete recibido
#define BUTTON_PIN 5    // botón para cambiar de canal
#define USE_DISPLAY
#define FLIP_DISPLAY
#define SDA_PIN 26
#define SCL_PIN 27
#define MAX_X 128
#define MAX_Y 51
#if CONFIG_FREERTOS_UNICORE
#define RUNNING_CORE 0
#else
#define RUNNING_CORE 1
#endif
```

Nota. *Definición de cada una de las variables que se va a utilizar. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.2.2.2 Declaración de variables: En este apartado se define valores enteros, booleanos, numéricos que garantiza 32 bits.

Figura 18

Declaración de variables

```
bool useSD = false;
bool buttonPressed = false;
bool buttonEnabled = true;
uint32_t lastDrawTime;
uint32_t lastButtonTime;
uint32_t tmpPacketCounter;
uint32_t pkts[MAX_X];       // aquí se guardarán los paquetes por segundo
uint32_t deauths = 0;       // cuadros de muerte por segundo
unsigned int ch = 1;        // canal 802.11 actual
int rssiSum;
```

Nota. *Variables declaradas por defecto. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.2.3 Funciones

Una función es una sección de un programa, o también se lo puede identificar como bloque de código, el mismo que actúa de forma independiente al resto, se encuentra conformado por tres componentes: parámetros, valores recibidos por la función como entrada, código, desarrollo de las operaciones que se va a realizar, resultado, es el valor que retorna de la ejecución del proceso, que tiene un nombre y un conjunto de instrucciones que son ejecutadas cuando se llama a la función, esto tiene el objetivo de reutilizar un solo bloque de todas las operaciones encapsuladas que se necesita, al momento de invocar a la función se lo puede hacer desde cualquier parte del programa, incluso si se necesita hacerlo varias veces, es por esto que un diseño funcional debe realizar una tarea bien definida evitando el uso de algoritmos complejos. (Corob-Msft, 2022)

2.2.3.1 Establecer canal: En este apartado se establece la función setChannel, realiza el proceso de cambio de canal para capturar paquetes que son enviados por ese canal.

Figura 19

Función establecer Canal

```
void setChannel(int newChannel) {
  ch = newChannel;
  if (ch > MAX_CH || ch < 1) ch = 1;

  preferences.begin("packetmonitor32", false);
  preferences.putUInt("channel", ch);
  preferences.end();

  esp_wifi_set_promiscuous(false);
  esp_wifi_set_channel(ch, WIFI_SECOND_CHAN_NONE);
  esp_wifi_set_promiscuous_rx_cb(&wifi_promiscuous);
  esp_wifi_set_promiscuous(true);
}
```

Nota. *Función para establecer Cana. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.2.3.2 Configuración SD: En este apartado se verifica el funcionamiento de la microSD para hacer uso de acuerdo a cada necesidad.

Figura 20

Configuración SD

```
bool setupSD() {
  if (!SD.begin()) {
    Serial.println("Card Mount Failed");
    return false;
  }

  uint8_t cardType = SD.cardType();

  if (cardType == CARD_NONE) {
    Serial.println("No SD_MMC card attached");
    return false;
  }

  Serial.print("SD_MMC Card Type: ");
  if (cardType == CARD_MMC) {
    Serial.println("MMC");
  } else if (cardType == CARD_SD) {
    Serial.println("SDSC");
  } else if (cardType == CARD_SDHC) {
    Serial.println("SDHC");
  } else {
    Serial.println("UNKNOWN");
  }

  uint64_t cardSize = SD.cardSize() / (1024 * 1024);
  Serial.printf("SD_MMC Card Size: %lluMB\n", cardSize);

  return true;
}
```

Nota. *Configuración SD. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.2.3.3 Dibujar: Se establece la función dibujar la cual será la encargada de graficar los paquetes en un canal especifico, intensidad de señal y tamaño de paquete.

Figura 21

Función Dibujar

```
void draw() {
#ifdef USE_DISPLAY
  double multiplicator = getMultiplicator();
  int len;
  int rssi;
  if (pkts[MAX_X - 1] > 0) rssi = rssiSum / (int)pkts[MAX_X - 1];
  else rssi = rssiSum;
  Heltec.display -> clear();
  Heltec.display->setTextAlignment(TEXT_ALIGN_RIGHT);
  Heltec.display->drawString( 10, 0, (String)ch);
  Heltec.display->drawString( 14, 0, ("|"));
  Heltec.display->drawString( 30, 0, (String)rssi);
  Heltec.display->drawString( 34, 0, ("|"));
  Heltec.display->drawString( 82, 0, (String)tmpPacketCounter);
  Heltec.display->drawString( 87, 0, ("["));
  Heltec.display->drawString(106, 0, (String)deauths);
  Heltec.display->drawString(110, 0, ("]"));
  Heltec.display->drawString(114, 0, ("|"));
  Heltec.display->drawString(128, 0, (useSD ? "SD" : ""));
  Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
  Heltec.display->drawString( 36,  0, ("Pkts:"));
  Heltec.display->drawLine(0, 63 - MAX_Y, MAX_X, 63 - MAX_Y);
  for (int i = 0; i < MAX_X; i++) {
    len = pkts[i] * multiplicator;
    Heltec.display->drawLine(i, 63, i, 63 - (len > MAX_Y ? MAX_Y : len));
    if (i < MAX_X - 1) pkts[i] = pkts[i + 1];
  }
  Heltec.display->display();
#endif
}
```

Nota. *Acciones para realizar el dibujo de gráficas. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.2.3.4 Void Setup: Contiene la configuración inicial del display, el inicio de rastreo de paquetes declaración del botón para el cambio de canal.

Figura 22

Función Void Setup

```
  /* mostrar pantalla de inicio */
  Heltec.display -> clear();
  Heltec.display->drawString(40, 6, "Esp32-Sniffer");
  Heltec.display->setFont(ArialMT_Plain_10);
  Heltec.display->drawString(6, 24, "@Autores");
  Heltec.display->drawString(6, 38, "Sebastian Guallasamin");
  Heltec.display->drawString(6, 48, "Gabriel Santos");
  Heltec.display->display();

  delay(10000);
//#endif

  // segundo núcleo
  xTaskCreatePinnedToCore(
    coreTask,               /* Función para implementar la tarea. */
    "coreTask",             /* Nombre de la tarea */
    2500,                   /* Tamaño de pila en palabras */
    NULL,                   /* Parámetro de entrada de tarea */
    0,                      /* prioridad de la tarea */
    NULL,                   /* Identificador de tareas. */
    RUNNING_CORE);          /*Identificador de tareas */

  // iniciar rastreador wifi
  esp_wifi_set_promiscuous_rx_cb(&wifi_promiscuous);
  esp_wifi_set_promiscuous(true);
}
```

Nota. *Configuración por defecto del visualizador. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.2.3.5 Void Loop: Se llama a todas las funciones creadas para el rastreo de paquetes, cambio de canal, verificar intensidad de la señal Wifi, graficar paquetes y guarda en la microSD la información de los paquetes capturados.

Figura 23

Función Void Loop

```
// botón de verificación
if (digitalRead(BUTTON_PIN) == LOW) {
  if (buttonEnabled) {
    if (!buttonPressed) {
      buttonPressed = true;
      lastButtonTime = currentTime;
    } else if (currentTime - lastButtonTime >= 2000) {
      if (useSD) {
        useSD = false;
        sdBuffer.close(&SD);
        draw();
      } else {
        if (setupSD())
          sdBuffer.open(&SD);
        draw();
      }
      buttonPressed = false;
      buttonEnabled = false;
    }
  }
} else {
  if (buttonPressed) {
    setChannel(ch + 1);
    draw();
  }
  buttonPressed = false;
  buttonEnabled = true;
}
```

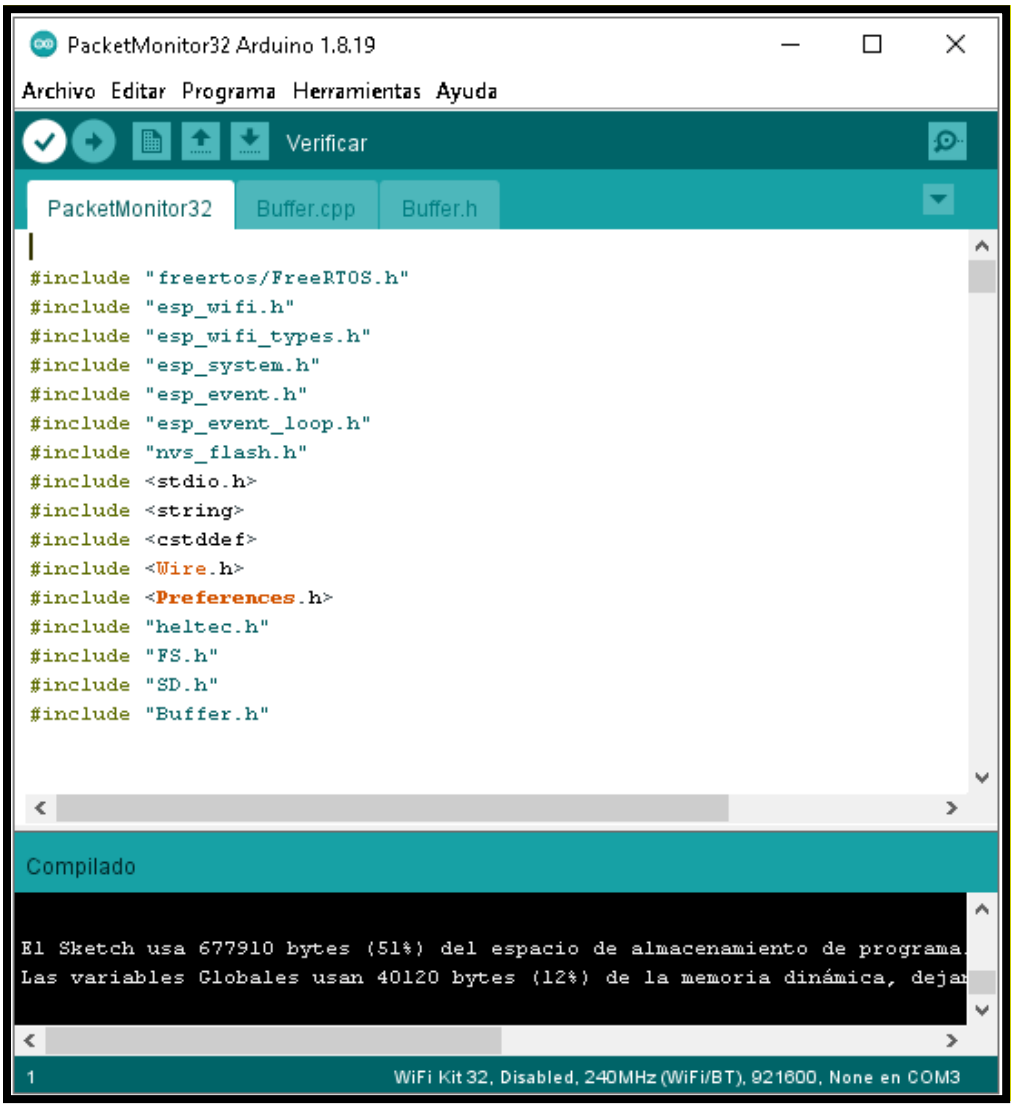
Nota. *Rastreo de paquetes mediante funciones. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.2.4 Ejecución del programa

2.2.4.1 Compilar Programa: En este apartado se realiza la compilación del programa para verificar si se obtiene algún error en la escritura del código para su posterior corrección.

Figura 24

Compilación del programa

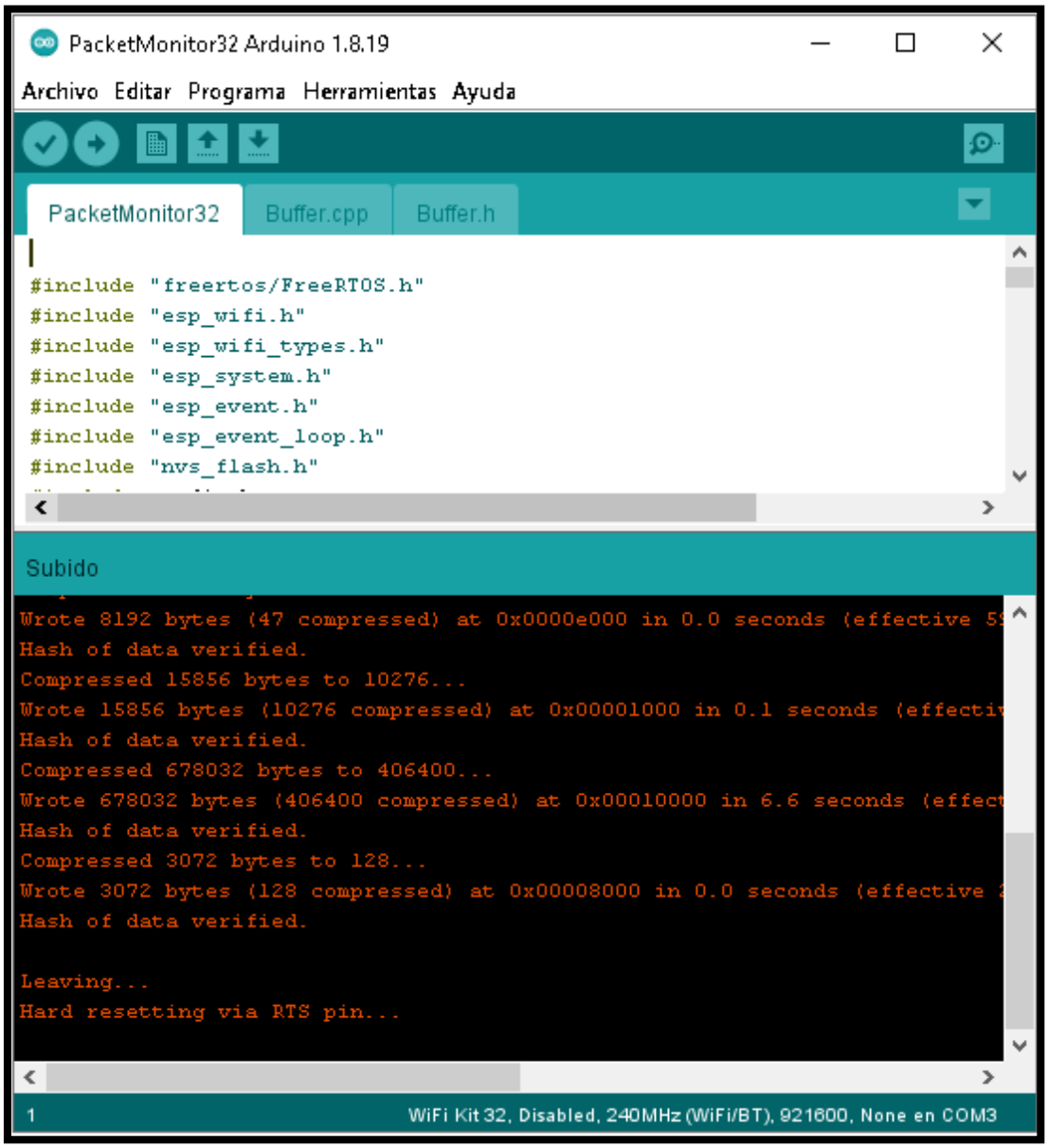


Nota. *Compilación del programa en Arduino. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.2.4.2 Cargar Programa ESP32: Una vez realizada la compilación y no presentar errores en el programa se procede a cargar el programa en el módulo ESP32.

Figura 25

Cargar el programa

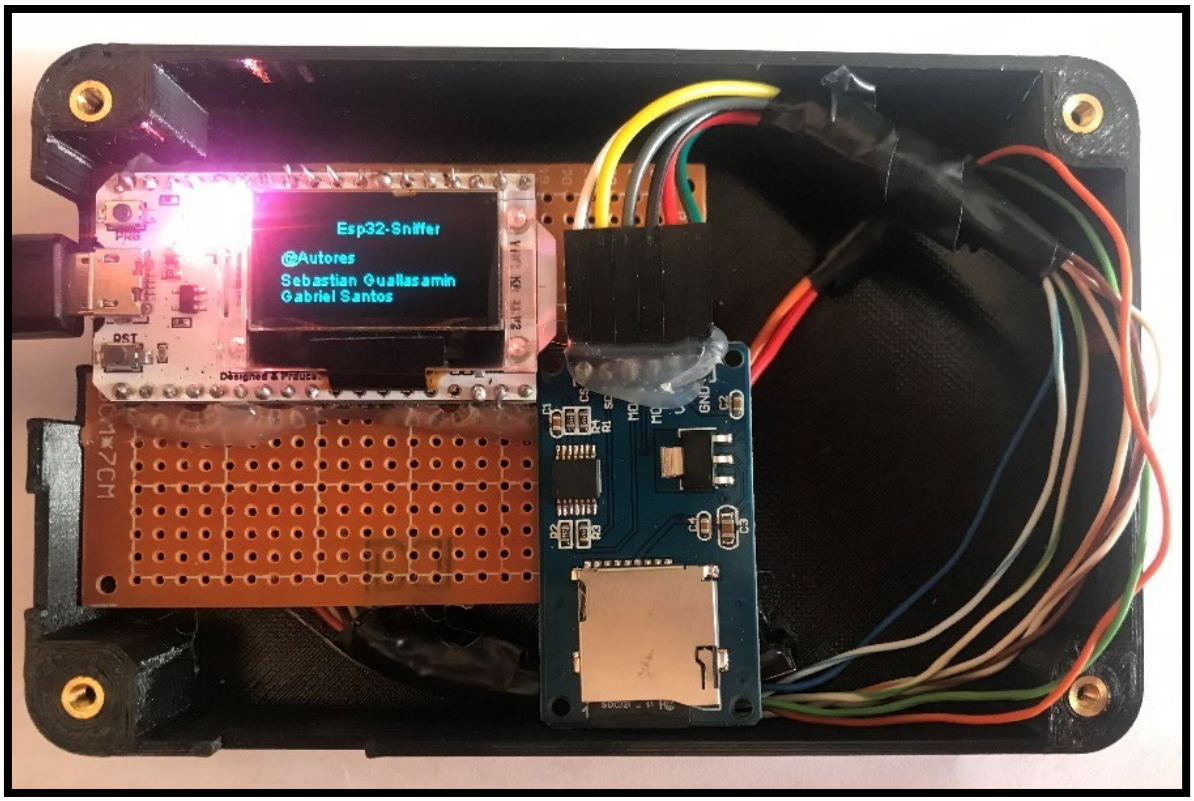


Nota. *Carga del programa en Arduino. Elaborado por: Guallasamín Alexis y Santos Gabriel*

2.2.4.3 Verificación de Funcionalidad de la aplicación: Para verificación del funcionamiento del aplicativo se utiliza una batería externa la cual proporciona 5v para el correcto funcionamiento.

Figura 26

Funcionalidad del programa



Nota. *Funcionalidad del programa en Arduino. Elaborado por: Guallasamín Alexis y Santos Gabriel*

CAPÍTULO 3

En el tercer capítulo, se realiza los escenarios de ataques designados al momento de realizar el análisis de los paquetes, es por esto que se detalla paso a paso la ejecución de los mismos con sus respectivos resultados.

3. PRUEBAS Y RESULTADOS

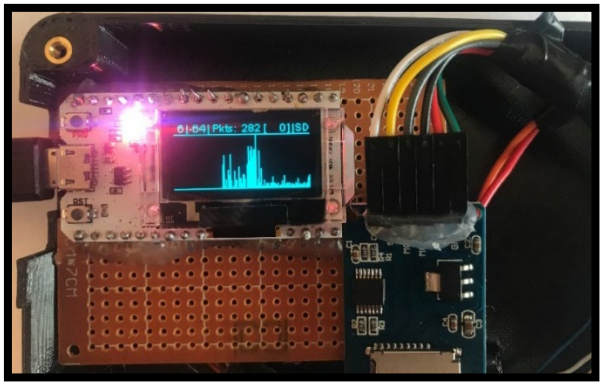
3.1 Captura Paquetes y ataque Handshake

Para la realización de esta prueba se procede con un ataque mediante diccionario en base a un archivo de extensión Captura de Paquetes (del inglés. pcap, *Packet Capture)* el cual se genera por la aplicación Wireshark. Para esta implementación se utiliza la herramienta Aircrack-ng, la misma que usa como dependencia la potencia de la Unidad Central de Procesamiento (del inglés CPU, *Central Processing Unit*) y la Unidad de Procesamiento Gráfico (del inglés GPU*, Graphics Processing Unit*).

3.1.1 Pruebas: se comienza realizando la captura los paquetes de las redes wifi cercanas a la aplicación durante 2 minutos.

Figura 27

Captura de Paquetes Handshake

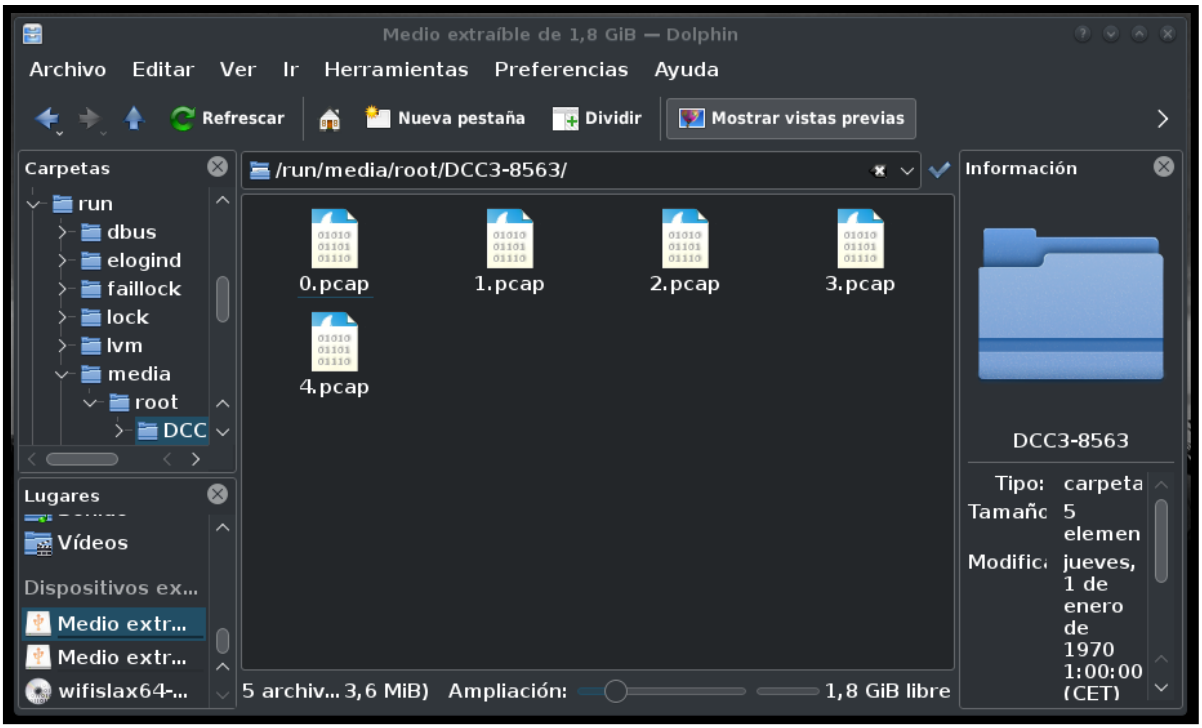


Nota. *Captura de paquetes Handshake por aplicación. Elaborado por: Guallasamín Alexis y Santos Gabriel*

Una vez realizado la captura de datos se extrae la microSD del prototipo y se procede a realizar la lectura de los archivos guardados en el sistema operativo Wifislax.

Figura 28

Extracción de archivos en microSD

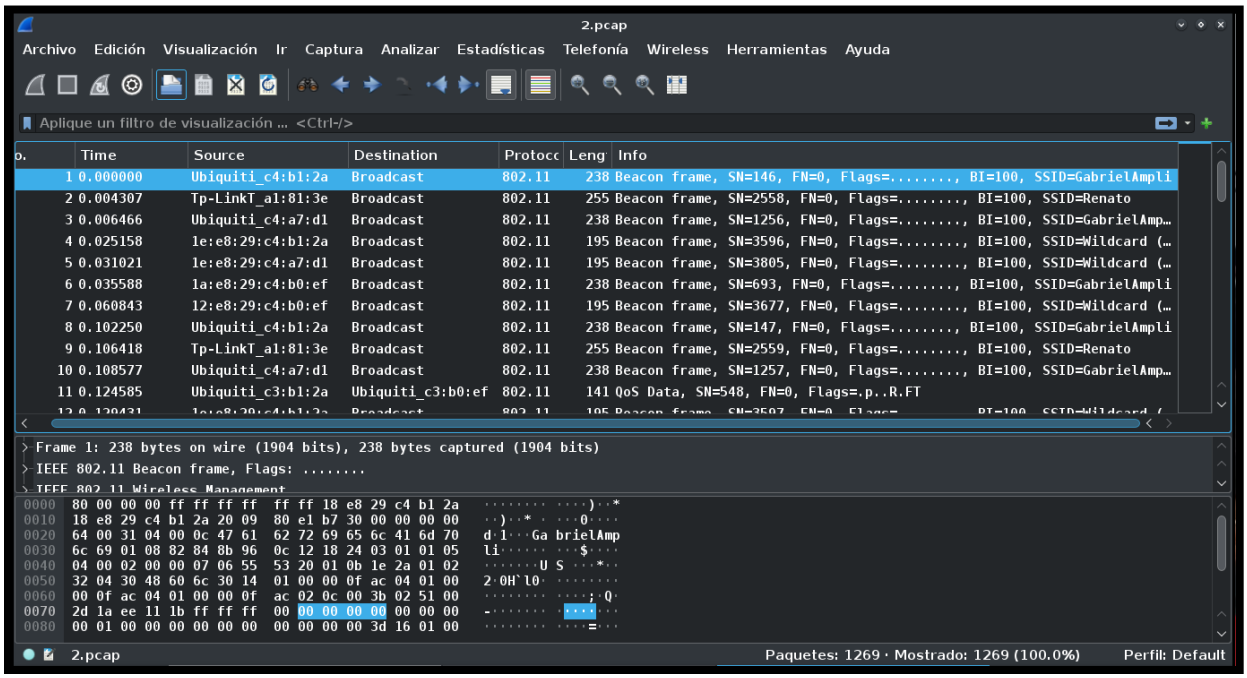


Nota. *Recolección de archivos en base a datos capturados en microSD. Elaborado por: Guallasamín Alexis y Santos Gabriel*

Se verifica los archivos .pcap en la aplicación Wireshark para evidenciar los paquetes capturados, en este caso se escoge los datos de 2 .pcap

Figura 29

Verificación de datos capturados

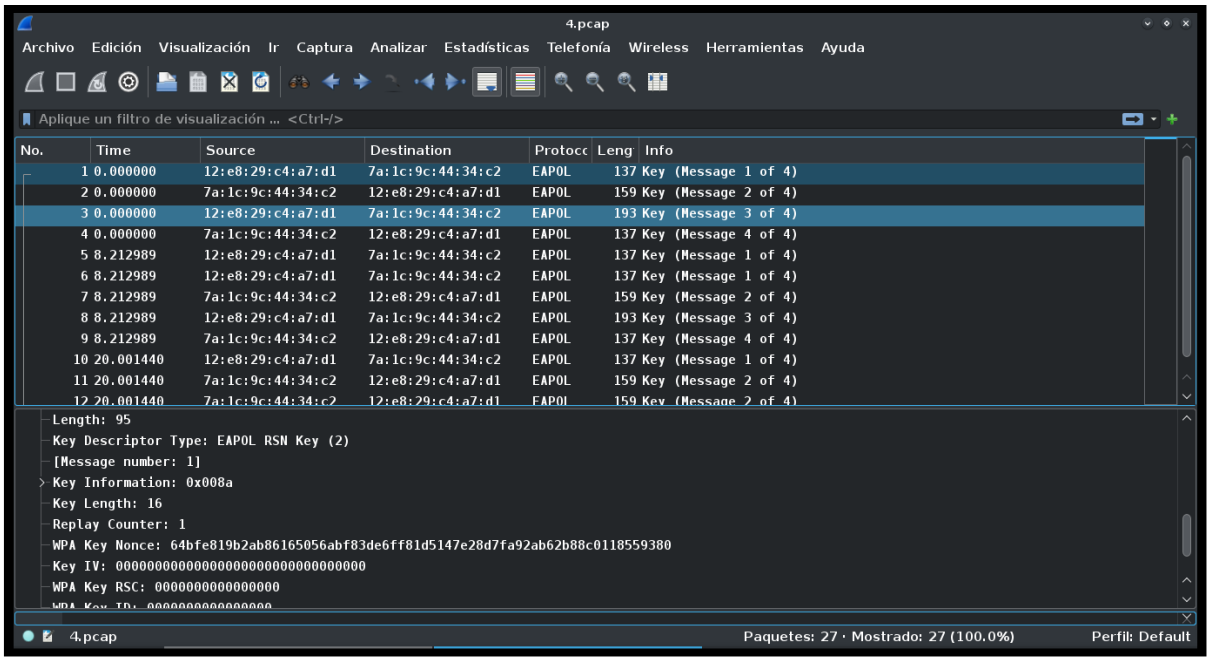


Nota. *Verificación de datos capturados en Wireshark. Elaborado por: Guallasamín Alexis y Santos Gabriel*

3.1.2 Análisis de Datos: como se puede evidenciar unos de los archivos generados por nuestro prototipo contiene un Protocolo de Autenticación extensible sobre LAN (del inglés LAN, Local Área Network).

Figura 30

Análisis de datos capturados



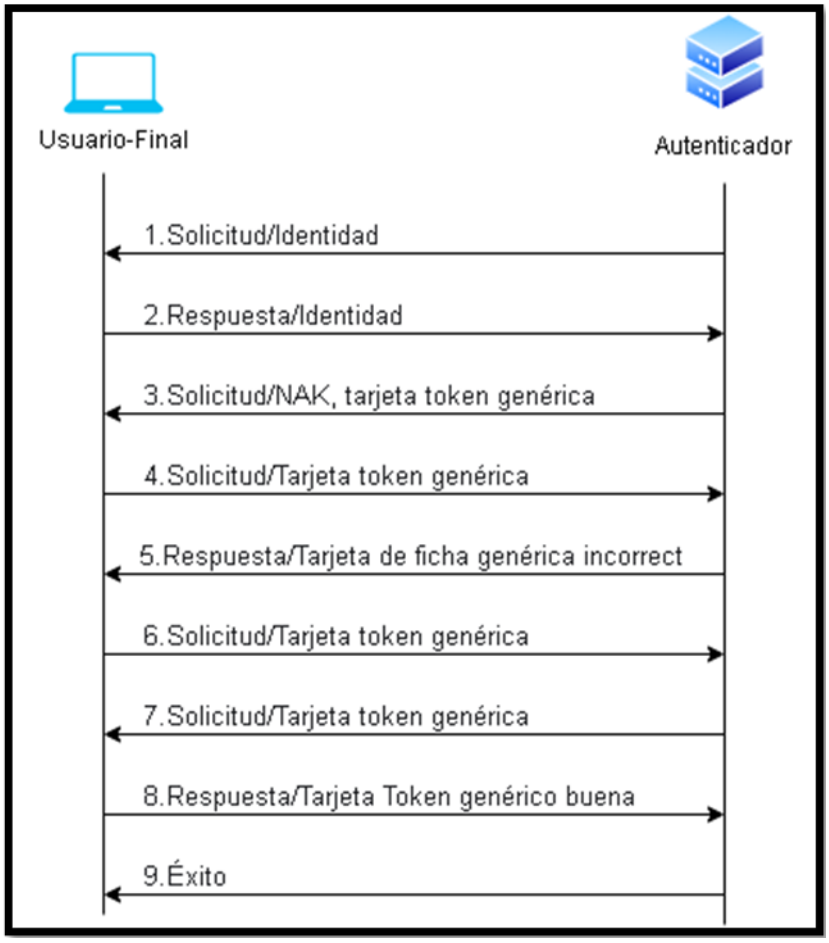
Nota. *Análisis de datos capturados en Wireshark. Elaborado por: Guallasamín Alexis y Santos Gabriel*

A continuación, se presenta el proceso de EAP, todo este se basa en el intercambio de solicitudes de autenticación, el cual finaliza cuando se presenta un mensaje de error o éxito, en base a cada uno de los requerimientos, mientras se realiza el intercambio se designa al paquete enviado por el autenticador mediante Solicitud / Método, de igual forma se designa a la respuesta del cliente mediante Respuesta / Método.

- El encargado del proceso de autenticación procede a enviar un paquete denominado solicitud / identidad, el cual sirve para identificar a un usuario determinado.
- El cliente necesita que cada uno de los usuarios ingresen con su respectivo identificador, para luego enviar el dato recogido en forma de mensaje de respuesta / identificación.
- Se envía un desafío de autenticación justo el momento que el autenticador haga el respectivo reconocimiento a un determinado usuario, todo esto mediante el Algoritmo de Resumen del Mensaje 5 (del inglés MD-5, Message-Digest Algorithm 5) para verificar la integridad de los datos recopilados.
- El cliente dispone de una tarjeta de token, la cual está configurada para dar permiso de autenticación, con esto se provee de un mecanismo de seguridad mediante reconocimiento.
- Se solicita el número de tarjeta token genérica para evidenciar mediante el autenticador que dicho usuario posee lo requerido para continuar con el proceso de autenticación.
- El usuario ingresa el número requerido de la tarjeta, y solicita una respuesta a esta petición.
- La respuesta del usuario es considerada incorrecta, por tal motivo su proceso de autenticación es incorrecto, en este caso el autenticador es capaz de recibir múltiples peticiones de autenticación mediante EAP, es así que se genera una nueva validación de la tarjeta token genérica.
- El usuario genera una nueva respuesta y pide colocar el número nuevamente de la tarjeta token para validación.
- En este caso la respuesta a la petición es validada de forma correcta, por tal motivo el autenticador procesa el envío de un mensaje de éxito.

Figura 31

Autenticación mediante EAP



Nota. *Proceso de Autentificación EAP. Elaborado por: Guallasamín Alexis y Santos Gabriel*

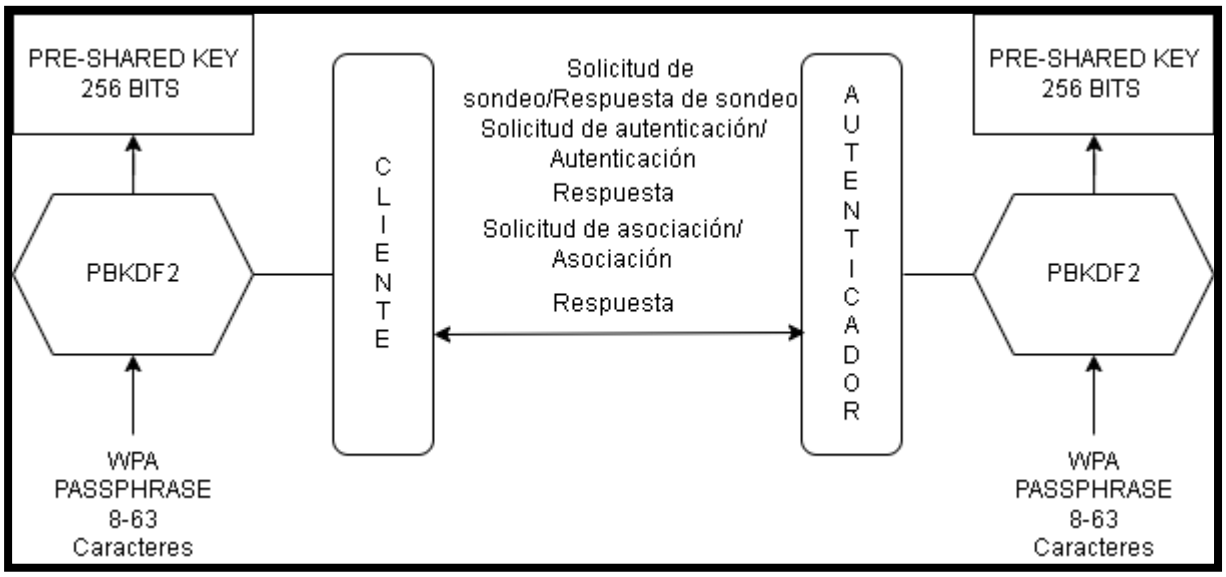
La implementación de la Función de Derivación de Clave Basada en Contraseña 2 (del inglés PBKDF2, Password Based Key Derivation Function 2) permite que se genere un patrón que sirve como medio de seguridad, más conocida como Clave Pre-compartida (del inglés PSK, Pre-Shared Key), y se lo entiende como un algoritmo basado en una clave que genera entre 8 y 63 caracteres, la misma que se toma como parámetro para que se pueda concebir con ese mismo valor

una nueva PSK. De acuerdo a la implementación interna del algoritmo PBKDF2 es importante tomar ciertos parámetros los cuales son:

1. La clave del Punto de Acceso seleccionada por el administrador de cada router.
2. El identificador de la red.
3. La longitud del identificador de red.
4. Número de veces que tendrá que ser codificada la clave del Punto de Acceso.
5. La longitud de la clave PSK.

Figura 32

Algoritmo PBKDF2



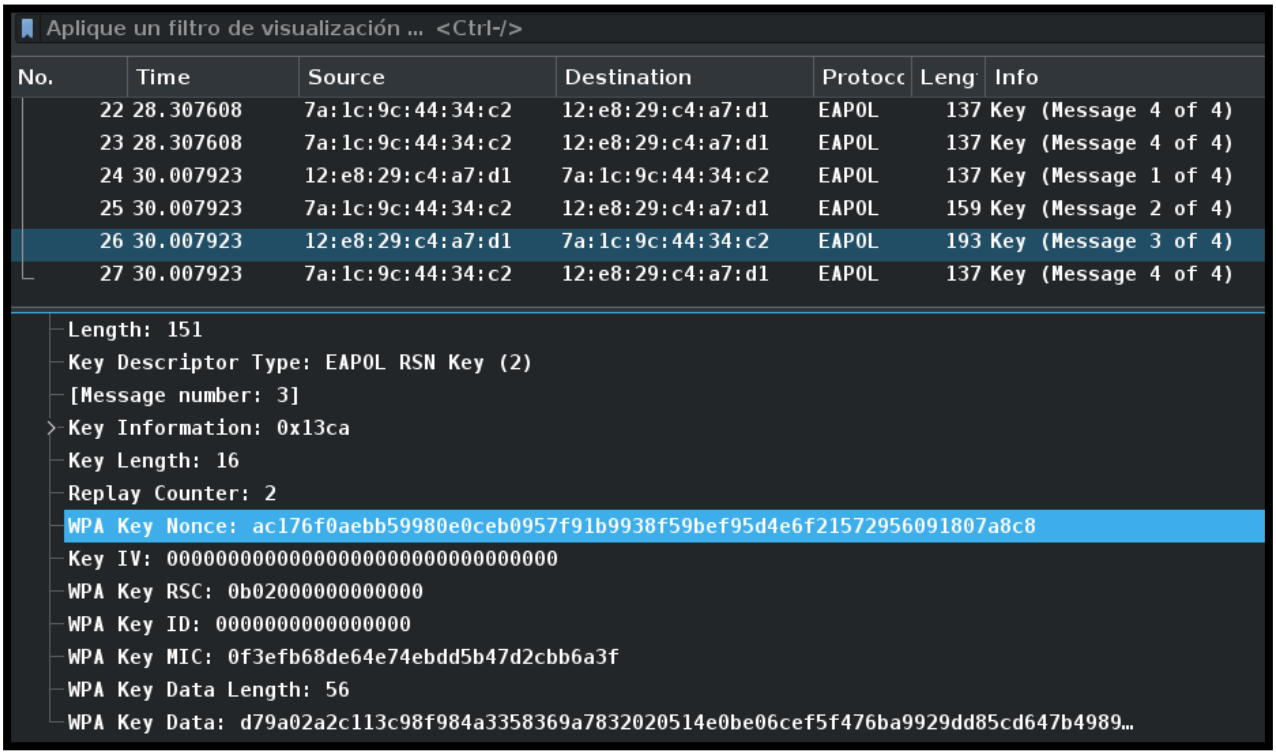
Nota. *Proceso de Autentificación EAP. Elaborado por: Guallasamín Alexis y Santos Gabriel*

Los paquetes de datos obtenidos tienen que ser intercambiados entre las entidades que corresponden haciendo uso de claves PSK, dicho proceso es conocido como 4-Way Handshake, que contempla los siguientes apartados:

El Autenticador envía un mensaje, con el valor que fue generado de forma aleatoria mediante el uso de la clave PSK, hacia el solicitante, el mensaje se denomina como Autenticado, siendo el valor de este el texto aleatorio con la clave PSK del Autenticador, tal y como se muestra en la figura 30, que corresponde a la captura que se realiza con la herramienta Wireshark.

Figura 33

WPA Key Nonce

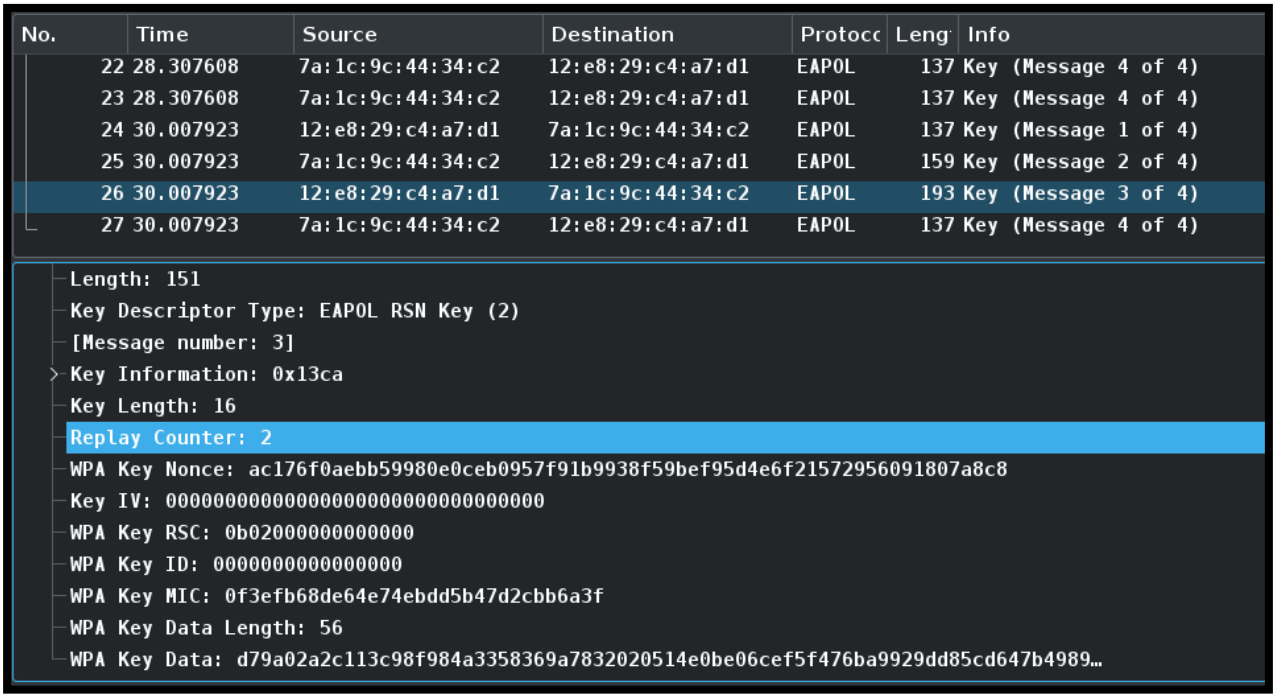


Nota. *WPA Key Nonce. Elaborado por: Guallasamín Alexis y Santos Gabriel*

El campo Replay Counter es denominado como un indicador, el cual permite tener conocimiento de la cantidad de paquetes que se han enviado en determinado tiempo, dando esta información al Autenticador y al Solicitante.

Figura 34

Replay Counter



Nota. *Replay Counter. Elaborado por: Guallasamín Alexis y Santos Gabriel*

La generación de la Clave Transitoria por Parejas (del inglés PTK, Pairwise Transient Key) se la formula mediante la clave maestra por pares (del inglés PMK, Pairwise Master Key), la misma que utiliza la función aleatoria de claves y designa como parámetros:

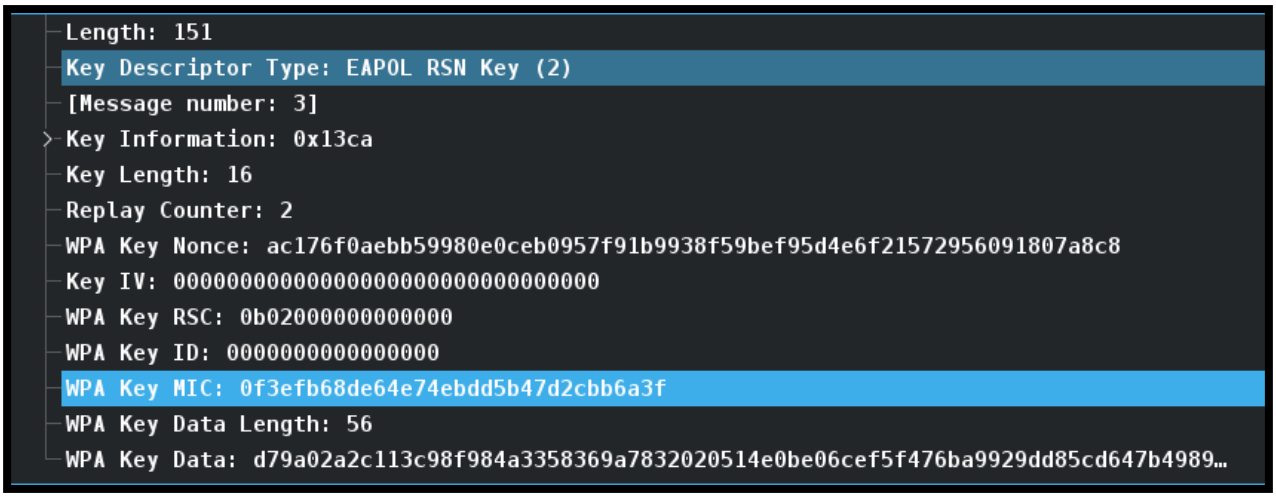
- **PMK**, denominada PSK generada por las entidades involucradas, es decir una clave mediante Hash criptográfico, en este caso la de Solicitante y Autenticador, mediante el uso del algoritmo de PBKDF2.
- **Anonce**, denominado paquete concebido por el Autenticador, el mismo que almacena cifrado un determinado texto aleatorio con la clave PSK.

- **Snonce**, denominado paquete concebido por el Solicitante, el mismo que almacena cifrado un determinado texto aleatorio con la clave PSK.
- **MAC del Autenticador,** denominado identificador único que cada fabricante establece a la tarjeta de red, en este caso del Autenticador.
- **MAC del Solicitante,** denominado identificador único que cada fabricante establece a la tarjeta de red, en este caso del Solicitante.

Con este proceso, el Solicitante puede enviar un paquete al Autenticador, el cual contenga un mensaje Snonce y un determinado Campo de Modulación por Impulsos Codificados (del inglés MIC, Pulse Code Modulation), mediante esto se puede verificar la integridad y consistencia del paquete obtenido, dicho campo se lo establece por medio del Solicitante, haciendo uso de PTK y PMK.

Figura 35

Campo WPA Key MIC



Nota. *WPA Key MIC. Elaborado por: Guallasamín Alexis y Santos Gabriel*

Por último, el Solicitante puede enviar un mensaje basado en el Reconocimiento de Instalación de Llaves, por medio del mismo se confirma al Autenticador que debe utilizar la misma PTK y Punto de Acceso generada en base al cliente, en el proceso de intercambio de cada uno de los paquetes, en este paquete se encuentra un campo Key ACK, el cual se utiliza para reconocer distintos mensajes de error a través de un determinado controlador, el valor en este caso es 0, por lo que indica ser el último mensaje enviado en todo el proceso de autenticación entre el Solicitante y el Autenticador.

3.1.3 Ataque a la red Wifi por diccionario: se toma como punto de partida la herramienta Aircrack-ng, se procede al análisis de los archivos .pcap generados por la aplicación Wireshark, con el fin de visualizar la captura del Handshake de una red específica, en este caso la red 'Ups_Laboratorio'.

Figura 36

Herramienta Aircrack-ng

Nota. *Análisis mediante Aircrack-ng. Elaborado por: Guallasamín Alexis y Santos Gabriel*

La herramienta Aircrack-ng tiene como finalidad de uso para para ataques de diccionario, los mismos que se ejecutan en la CPU. Como primer paso se debe generar o descargar diccionarios que contengan determinadas contraseñas de red con las que puedan estar configuradas, para así proceder con el ataque.

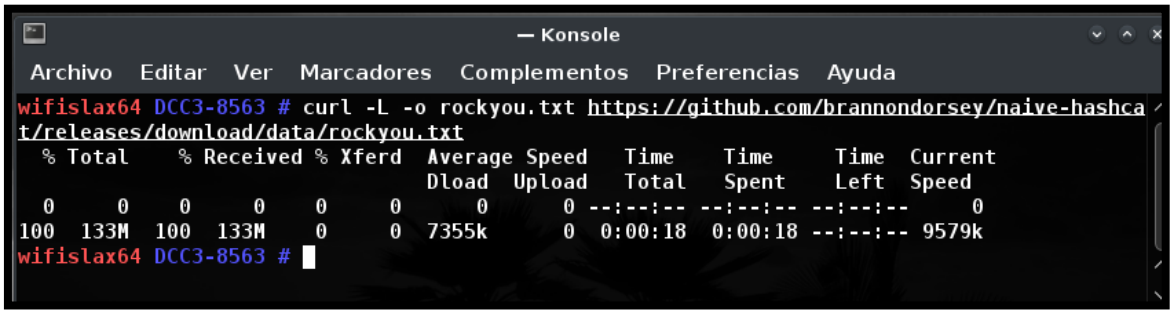
El siguiente comando lo que permite realizar es lectura de una página y la descarga del diccionario rockyou.txt, en base a una ruta seleccionada.

curl -L -o rockyou.txt https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt

- curl: muestra el contenido de una página.
- -o: permite especificar un nombre de archivo o ubicación diferente.

Figura 37

Descarga de diccionario



Nota. *Descarga de diccionario rockyou.txt. Elaborado por: Guallasamín Alexis y Santos Gabriel*

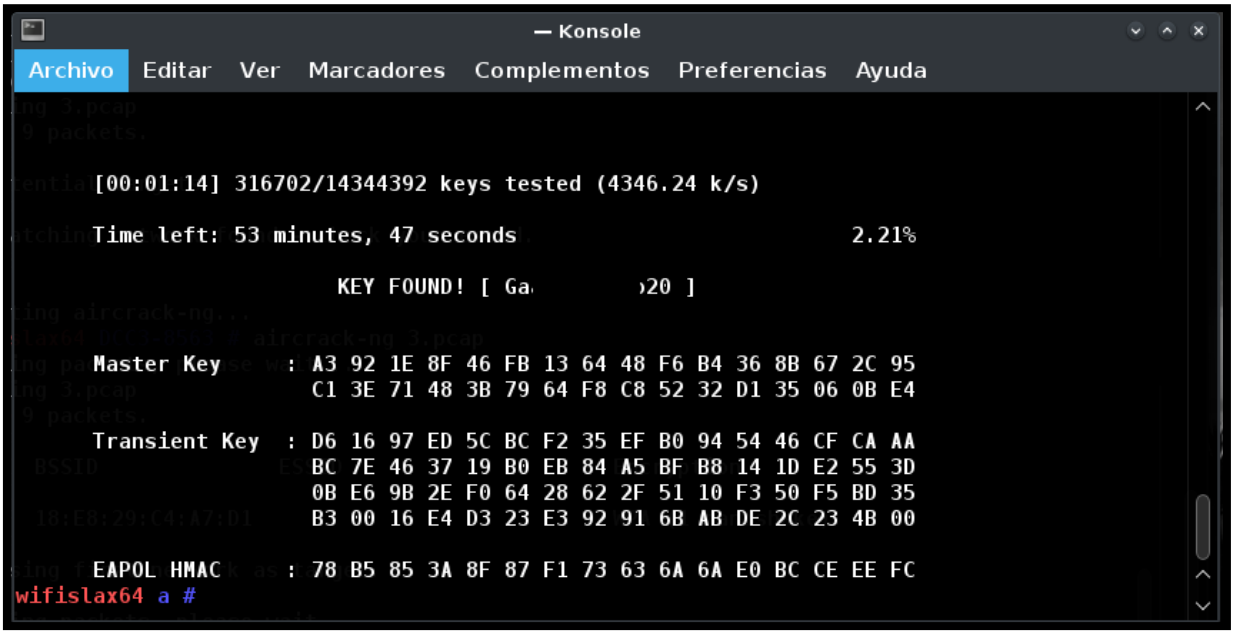
Una vez obtenido la dirección MAC de la víctima ya antes mencionado mediante el Sniffer, se procede a realizar de acuerdo al archivo de captura, el ataque mediante el siguiente comando:

aircrack-ng -a2 18:E8:29:C4:A7:D1 -e Ups_Laboratorio -w rockyou.txt 3.pcap

- aircrack-ng: solo se puede intentar la obtención de claves pre-compartidas.
- -a2: proporcionar la MAC obtenida y los números de paquetes.
- -e: especifica el identificador de red del Punto de Acceso.
- -w: especificar el diccionario que se va a utilizar

Figura 38

Prueba Aircrack

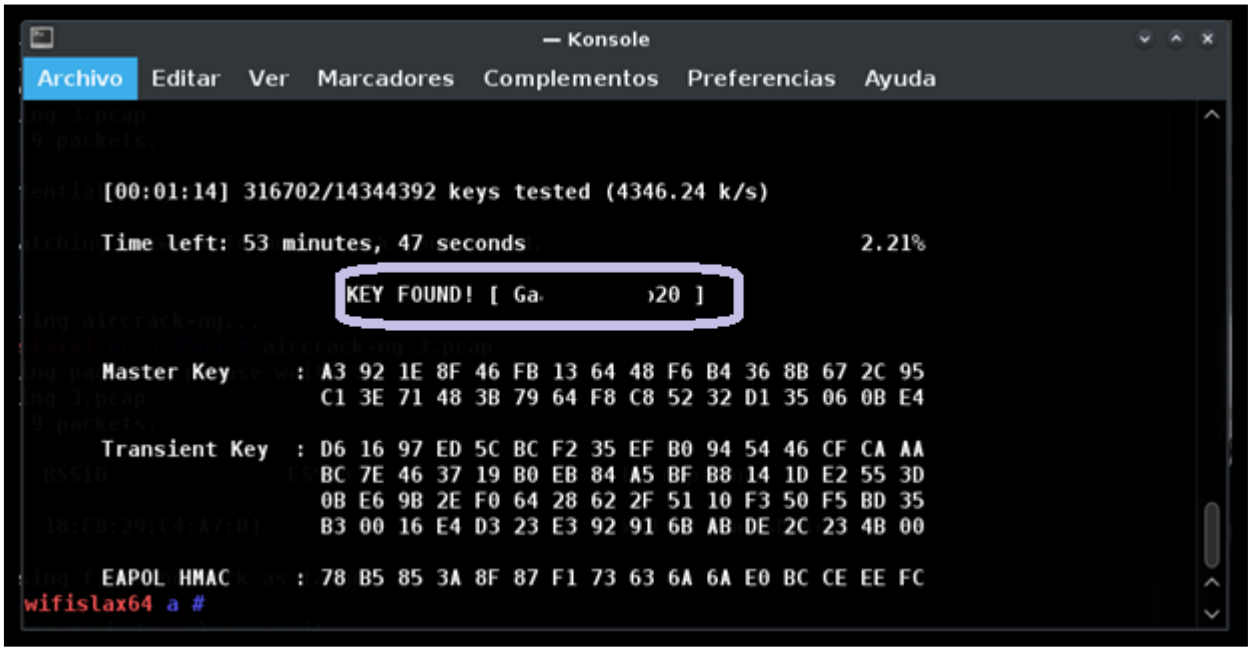


Nota. *Prueba generada con herramienta aircrack-ng junto con el diccionario rockyou.txt antes descargado. Elaborado por: Guallasamín Alexis y Santos Gabriel*

3.1.4 Resultado: De acuerdo al desarrollo de la prueba se pudo evidenciar lo fácil que es obtener las credenciales de alguna red Wi-Fi, el tiempo que se tomó el ataque en obtener la clave y en ser completado fue de 1 minuto 14 segundos, esto sucedió porque la contraseña no poseía la longitud adecuada de cada uno de sus caracteres, ya que se considera una clave muy segura a la que emplea una combinación de letras entre mayúsculas y minúsculas, números, y símbolos con el objetivo de establecer una cadena impredecible de caracteres y así generar una contraseña única, la cual cuente con todos los parámetros necesarios para dificultar al máximo que un intruso ingrese a la red por cualquier motivo, lo que puede ocasionar graves problemas al comprometer la seguridad de la red.

Figura 39

Clave encontrada

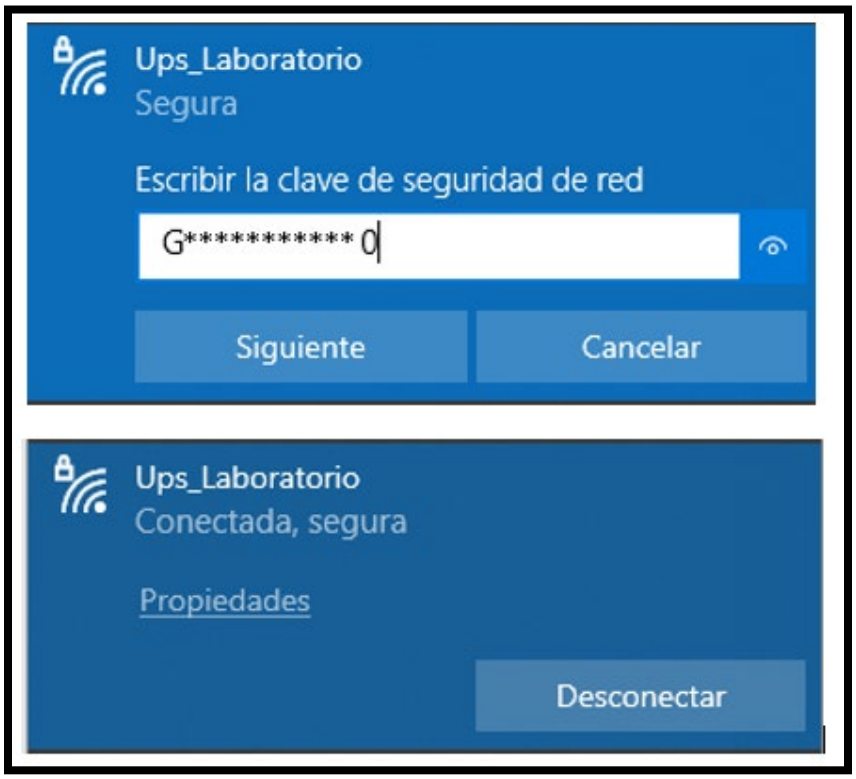


Nota. *Obtención de Clave Wi-Fi. Elaborado por: Guallasamín Alexis y Santos Gabriel*

De acuerdo a la figura 40, se puede observar que la credencial obtenida por aicrack-ng funciona y ahora se puede acceder dentro de la red “Ups_Laboratorio”.

Figura 40

Clave validada



Nota. *Validación de Clave Wi-Fi. Elaborado por: Guallasamín Alexis y Santos Gabriel*

3.2 Captura de Paquetes y ataque DoS

Para la realización de esta prueba se procede con un ataque DoS junto un archivo .pcap generado por la aplicación. Para esta implementación se utilizará la herramienta Aireplay-ng, que depende respectivamente de la potencia de la CPU, este ataque pretende inhabilitar el uso de una máquina, con el objetivo de bloquear un servicio que se encuentra en constante uso y se distribuye a varios destinos. Por medio del mismo se puede afectar considerablemente a la fuente de la

información como al sistema en sí, debido a esto se vulnera la seguridad y quedan desprotegidos los canales de transmisión por donde pasan los paquetes que son necesarios para un correcto funcionamiento.

3.2.1 Prueba: se da inicio al proceso mediante la captura de los paquetes de redes Wi-Fi cercanas a la aplicación durante 2 minutos.

Figura 41

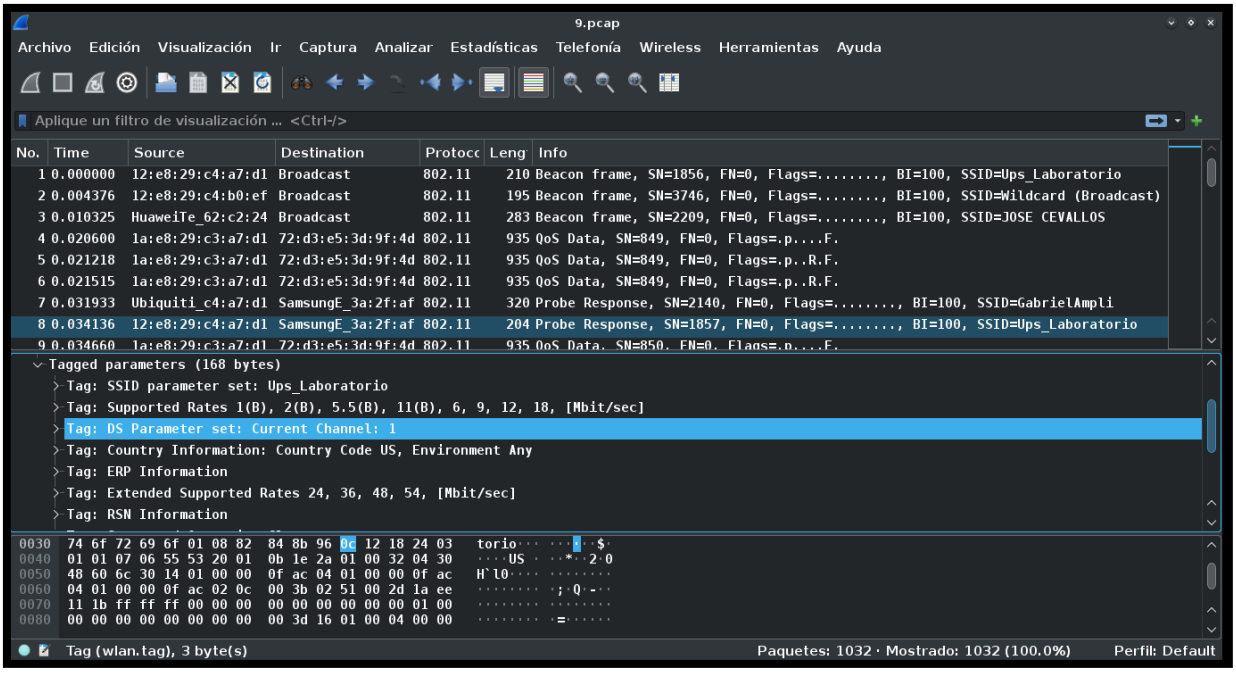
Captura de Paquetes DoS

Nota. *Captura de paquetes DoS por aplicación. Elaborado por: Guallasamín Alexis y Santos Gabriel*

Se verifica los archivos .pcap generados en la aplicación Wireshark, con el fin de obtener la información de cada uno de los paquetes que se capturaron en el instante por el prototipo, en este caso el archivo tomado es 9.pcap.

Figura 42

Captura de paquetes DoS



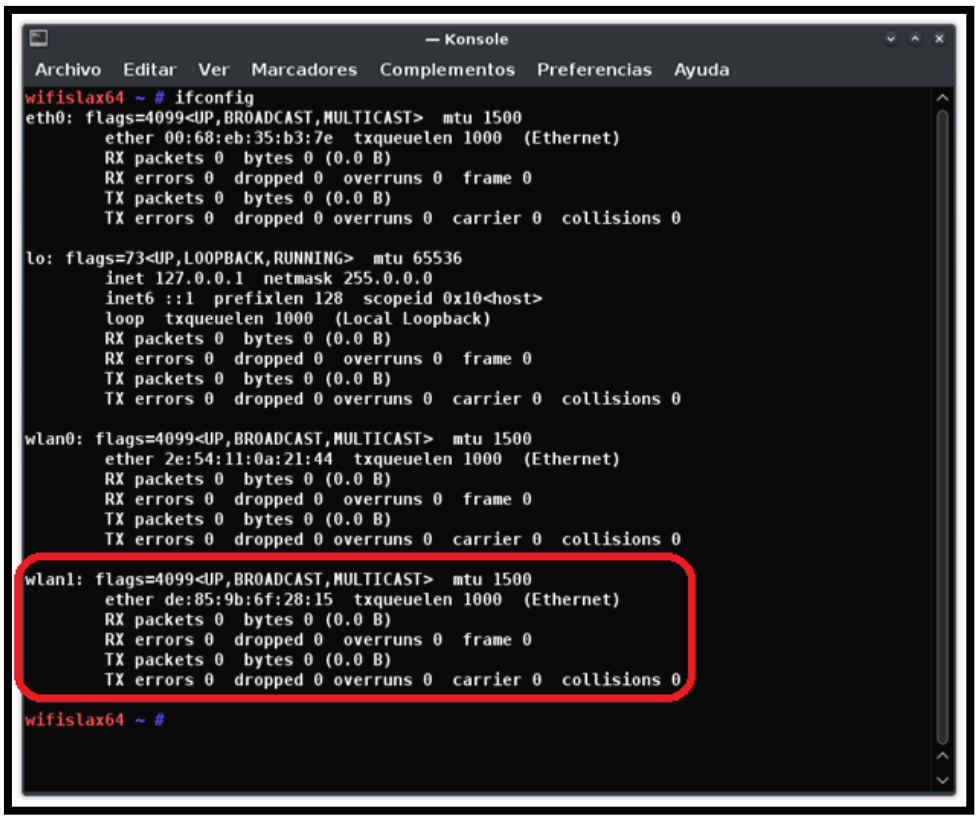
Nota. *Paquetes capturados para DoS. Elaborado por: Guallasamín Alexis y Santos Gabriel*

Como se puede evidenciar en la figura 33, el Sniffer logra capturar el identificador de la red Wifi “Ups_Laboratorio”, por lo que se presenta la dirección MAC 12:E8:29:C4:A7:D1 , mientras el canal de transmisión en el que se encuentran los paquetes capturados se identifique con claridad, se podrá utilizar estos medios para realizar el ataque DoS.

3.2.2 Ataque a red Wifi: Como primer paso se verifica cada uno de los adaptadores Wi-Fi disponibles que se encuentren dentro del sistema, es por esto que se utiliza el comando **ifconfig**, en este se debe seleccionar el adaptador wlan1.

Figura 43

Selección de Adaptador de Red



Nota. *Selección del adaptador wlan1. Elaborado por: Guallasamín Alexis y Santos Gabriel*

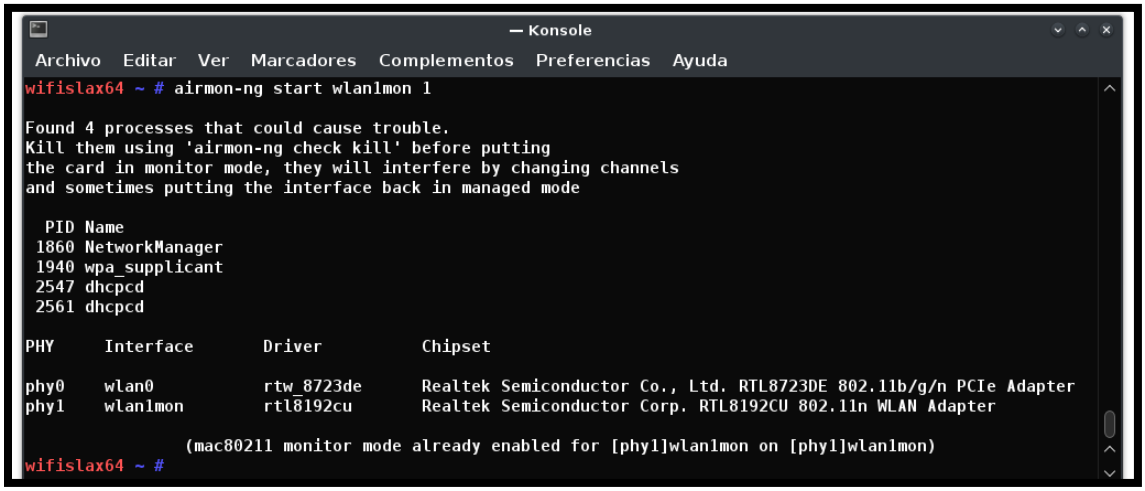
Mediante la herramienta Airmon-ng se tomará ahora el adaptador y se lo colocará en modo de monitorización, por medio del canal 1 con el comando:

airmon-ng start wlan1mon 1

- airmon-ng: habilitar modo monitor en las interfaces inalámbricas.
- start: indica el proceso de inicio de la interfaz.
- wlan1mon 1: nombre que toma el modo monitor de la wlan1.

Figura 44

Configuración modo monitor de Tarjeta Wi-Fi

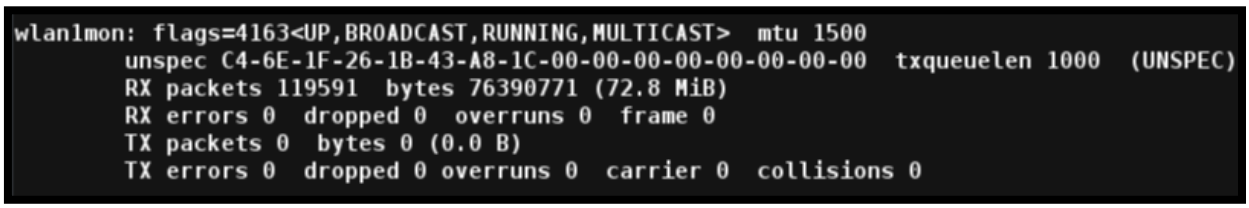


Nota. *Configuración de la tarjeta red Wi-Fi en modo monitor. Elaborado por: Guallasamín Alexis y Santos Gabriel*

Ahora el adaptador de red wlan1 se encuentra en modo monitor con el nombre de wlan1mon, para verificar que el proceso se cumplió correctamente se vuelve a listar los adaptadores de red con el comando **ifconfig.**

Figura 45

Creación de Tarjeta de red Wi-Fi en modo monitor



Nota. *Tarjeta de red Wifi en modo monitor. Elaborado por: Guallasamín Alexis y Santos Gabriel*

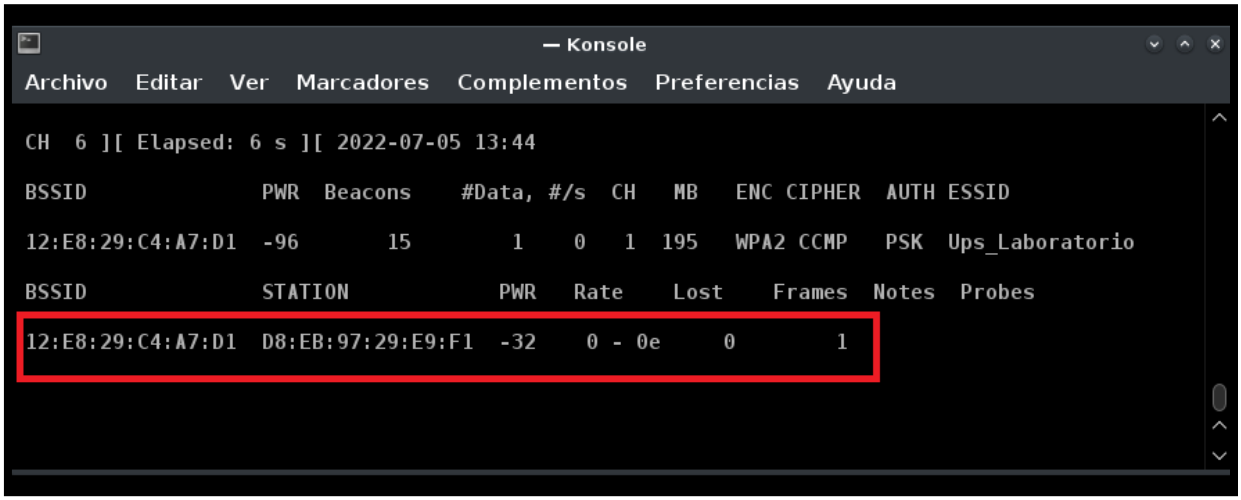
Para empezar con la verificación de la conexión WiFi se procede con el inicio de la inspección de los clientes que se encuentran conectados a la red “Ups_Laboratorio”, mediante la dirección MAC, misma que fue obtenida gracias al uso del Sniffer.

```
airodump-ng --bssid 12:E8:29:C4:A7:D1 wlan1mon
```

- airodump-ng: se utiliza para la captura de paquetes, en base a normas inalámbricas que pueden estar sin formato.
- --bssid: limitar la captura de datos a un solo punto de acceso.
- wlan1mon: nombre que toma el modo monitor de la wlan1.

Figura 46

Inspección de clientes



Nota. *Inspección de clientes en base a MAC. Elaborado por: Guallasamín Alexis y Santos Gabriel.*

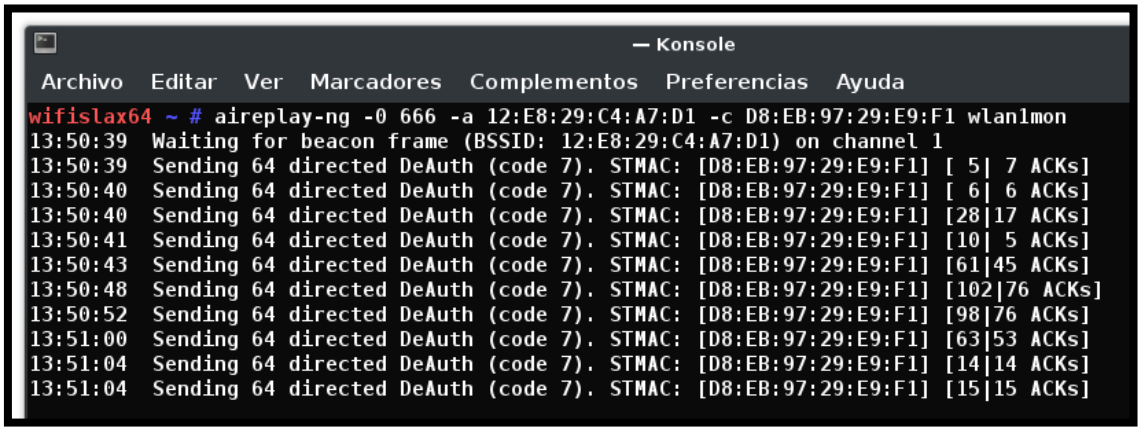
Después de haber visualizado el objetivo en base a los clientes, se procede con la utilización de la herramienta Aireplay-ng, la cual se utiliza para inyectar fotogramas, es decir genera tráfico en la red para su uso posterior. Para esto se ejecuta el siguiente comando:

aireplay-ng -0 666 -a 12:E8:29:C4:A7:D1 -c D8:EB:97:29:E9:F1 wlan1mon

- aireplay-ng: genera tráfico en la red.
- -0: especifica el ataque deauth.
- 666: número de paquetes deauth para ser enviados.
- -a: Establece la dirección MAC del Punto de Acceso.
- -c: Establece la dirección MAC del objetivo.
- wlan1mon: Nombre de la interfaz inalámbrica en modo monitor.

Figura 47

Ataque Aireplay-ng

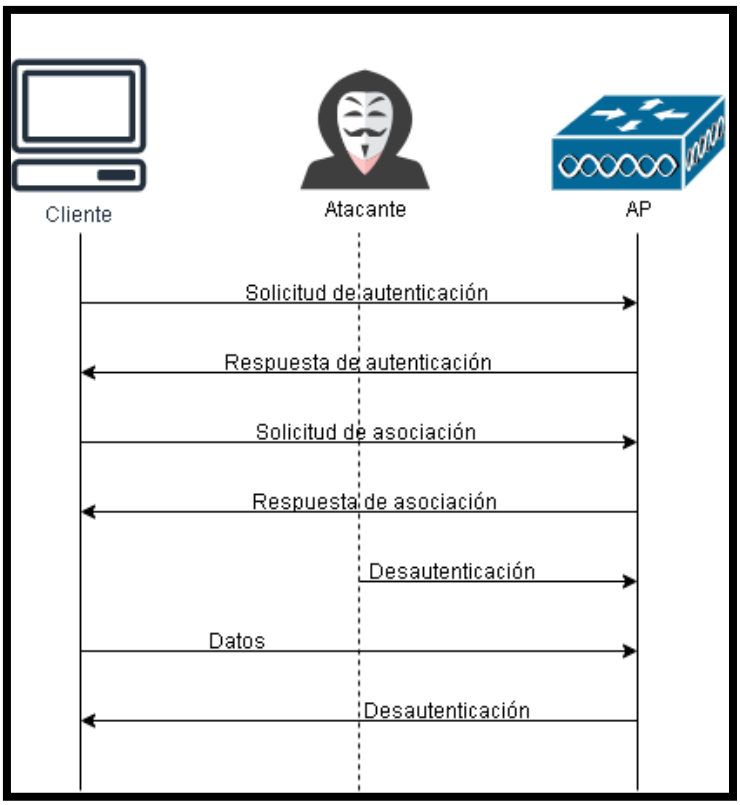


Nota. *Ejecución del ataque Aireplay-ng. Elaborado por: Guallasamín Alexis y Santos Gabriel.*

3.2.3 Análisis de Datos: La red inalámbrica es susceptible a un ataque de denegación de servicio "ataque DoS" mediante el cual el atacante puede usar un comando para revocar la autenticación siendo esto suplantado, para obligar al Punto de Acceso a que genere de nuevo un proceso para autenticar a los clientes conectados. Las siguientes figuras ilustran el mecanismo ataque de desautenticación.

Figura 48

Ataque Aireplay-ng



Nota. *Ataque de desautenticación. Elaborado por: Guallasamín Alexis y Santos Gabriel.*

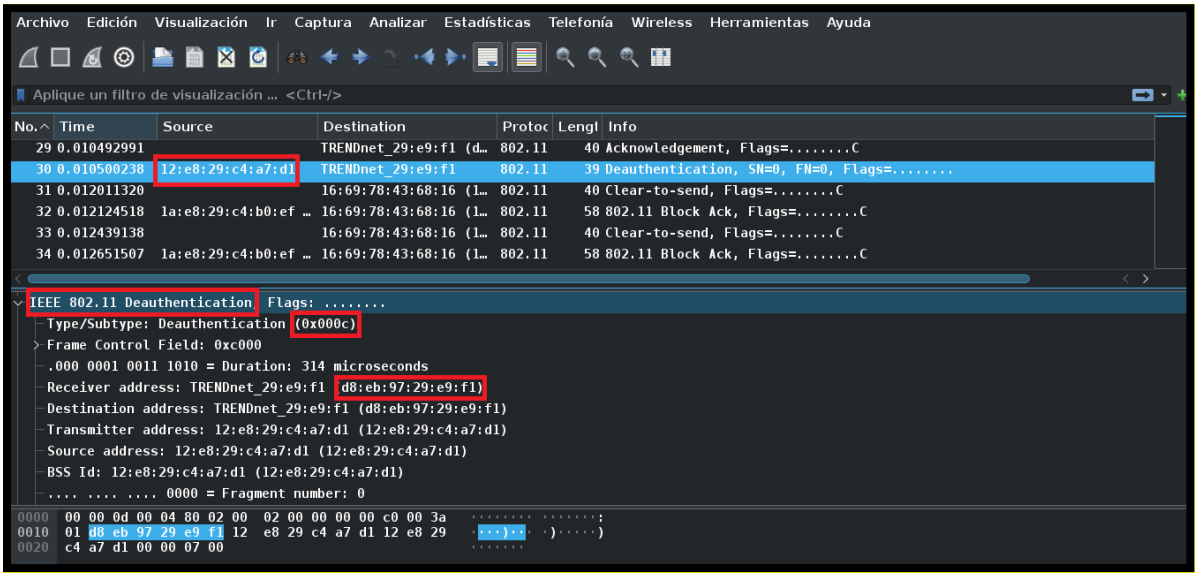
La herramienta de Aircrack-ng en Wifislax se usa para escanear y descifrar el respectivo cifrado que poseen las redes inalámbricas, junto con esto existe otro tipo de herramientas como aireplay-ng, la misma que actúa como agente falsificador, de acuerdo a ello envía paquetes en los cuales existe la manera de quitar la autenticación de uno o más clientes asociados a un Punto de Acceso ubicado en cualquier sitio.

El mismo ataque tiene su accionar en base a los numerosos paquetes que envía, todo esto en busca de lograr la desautenticación de forma periódica del cliente conecto a dicho punto, se lo realiza mediante la dirección MAC. De acuerdo a su método para inmiscuirse en los paquetes el

cliente se desautenticará de forme inmediata, por lo que permite que aparezca un nuevo punto de acceso, mismo que no posee ninguna autorización, con el fin de que el cliente sea atacado ya que se conectaran a un punto suplantado. Mientras se continuaba con el ataque, todos los paquetes que fueron transmitidos en tiempo real se encontraban en constante monitorización de la red mediante Wireshark.

Figura 49

Análisis de datos en Wireshark – Desautenticación



Nota. *Análisis de datos en base a desautenticación. Elaborado por: Guallasamín Alexis y Santos Gabriel.*

Con la información resultante del ataque, se puede concluir los siguiente:

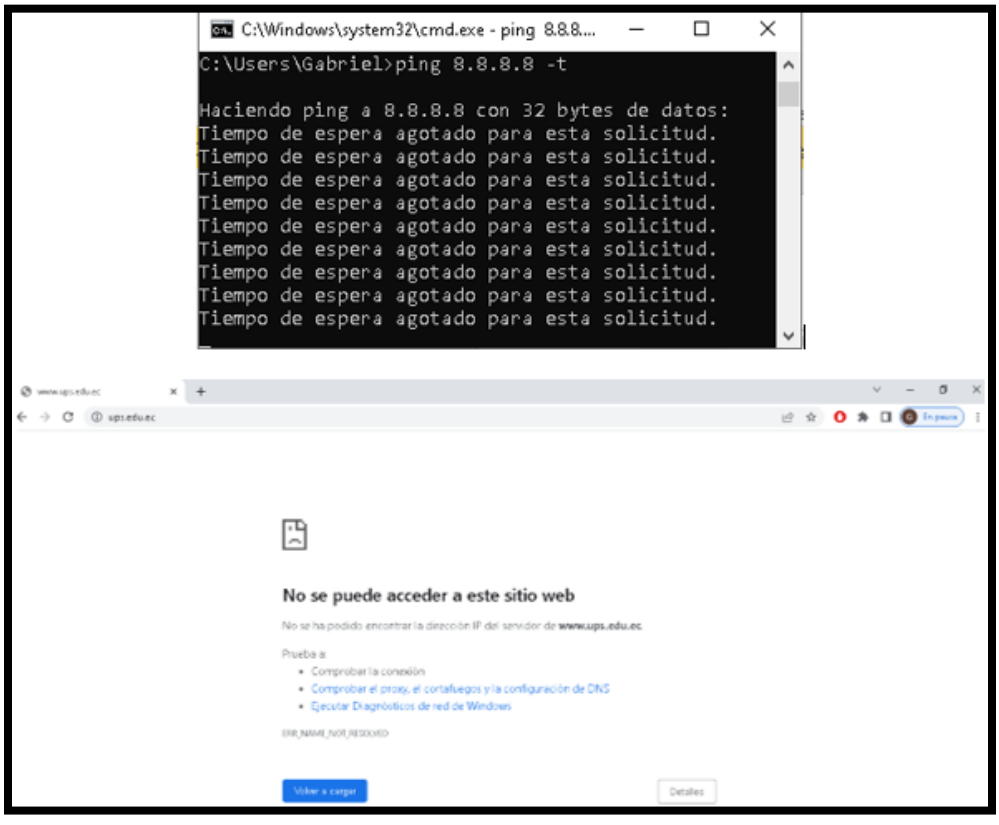
- El tipo de marco de gestión Deauthentication fue falsificado y el atacante lo representa por dos bits (00).
- La desautenticación tiene un valor de subtipo fijo representado por cuatro bits (1100)
- El atacante está enviando un mensaje de difusión a desautenticación a todos los clientes.

- La dirección física del atacante (Dirección MAC) que es idéntica al punto de acceso.

3.2.4 Resultados: De acuerdo al ataque ya aplicado, se verifica el funcionamiento del mismo, es por esto que se realiza un ping al DNS público de Google 8.8.8.8, para saber si se llega al destino determinado de forma correcta, o si, por el contrario, no se puede llegar a él, en este caso se produce el ataque de Denegación de servicio, y por ello solo se encuentra saturado de peticiones al servicio y no permite la conexión al mismo, a su vez mediante el navegador se prueba el ingreso a una dirección web, en este caso **www.ups.edu.ec** y se observa que tampoco existe algún tipo de llegada o comunicación con dicha página.

Figura 50

Prueba de ataque DoS



Nota. *Prueba del ataque DoS. Elaborado por: Guallasamín Alexis y Santos Gabriel.*

CONCLUSIONES

El proyecto técnico realizado culminó con la creación de dos escenarios de prueba, el primero se implementó un ataque por diccionario y el segundo se basó en un ataque DoS, de acuerdo a su desarrollo se pudo evidenciar las vulnerabilidades en la red inalámbrica, esto se debe a que el protocolo 802.11, mejor conocido como Wi-Fi, tienen fallos como la forma de introducir tramas de dudosa procedencia con el fin de interceptar el tráfico de red de la víctima, es por esto que mediante la realización de las pruebas en la red Wi-Fi de la UPS se torna un punto crítico todo lo que tiene que ver con la seguridad de la red inalámbrica.

Con respecto al análisis de los datos, se pudo capturar cada uno de los paquetes que viajan por medio del aire de las distintas redes Wi-Fi, la información que proporciona y su significado, mediante la herramienta Wireshark, con esto se pudo identificar la captura del protocolo EAPOL, el cual es esencial para la captura del Handshake, de igual manera los nombres de los identificadores de red, el canal, la dirección MAC del router y con esta información se pudo realizar los ataques a la red inalámbrica.

Para finalizar, mediante la implementación del Sniffer fue posible capturar el proceso de transmisión de datos, identificando información sensible para cualquier organización, lo que permitió realizar el análisis de datos y así se pudo crear diferentes escenarios, en el primero se logró obtener las credenciales de acceso mediante un ataque por diccionario, el mismo que permite generar nuevos ataques dentro de la red, en el segundo se puede evidenciar la denegación del servicio a un usuario en específico, mediante la suplantación de identidad, denegando el acceso total a la red, con esto se logró evidenciar que el SoC ESP32, al ser ligero tanto en términos de precio y tamaño, es una excelente herramienta para la captura de paquetes en lugares poco accesibles.

RECOMENDACIONES

Todo lo que tiene que ver con seguridad en una red inalámbrica se basa en principios activos, es decir, para analizar el tráfico de una red siempre deben existir tiempos en los cuales se hagan chequeos, más que todo con el fin de observar en que condición se encuentra toda la red, y así puedan generar toma de decisiones para mitigar cualquier tipo de falla en la seguridad. No se debe esperar que suceda algún contratiempo fatal, más bien se tiene que tener la debida precaución mediante la constante monitorización, con eso se maximiza la eficiencia de este proceso, evitando de tal forma la intervención del campo económico, e involucre pérdidas significantes para la organización.

Es por esto que se recomienda tener una línea estable, la cual ejecute todas sus actividades teniendo en cuenta que las circunstancias sean totalmente normales, con lo que ayudaría a tener medios comparativos de acuerdo a cada análisis que se vaya realizando durante el tiempo de servicio, lo que ayudaría a descubrir cualquier tipo de grieta en la seguridad de forma rápida, y mediante soluciones rápidas, se pueda actuar para que no se convierta en una amenaza a gran escala para todos los usuarios.

REFERENCIAS BIBLIOGRÁFICAS:

ACISSI. Agé, M. Baudru, S. Crocfer, N. (2015). *Seguridad Informática Hacking Ético.*

Álvarez Carulla, A. (2021). *Comunicación de un módulo ESP32 con Ubidots mediante MQTT.*

Arduino. (2022, 20 junio). *Librería Wire*. Arduino Reference. https://www.arduino.cc/reference/en/language/functions/communication/wire/

Arduino. (2022). *Librería SD*. Arduino Reference. https://www.arduino.cc/reference/en/libraries/sd/

A. Maier, A. Sharp, Y. Vagapov *"Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things"*, in 2017 Internet Technologies and Applications (ITA), Wales, UK, 2017, pp. 1-6. DOI: https://doi.org/10.1109/ITECHA.2017.8101926

Avella, R. (2010, 25 agosto). *Stdio.h y sus funciones*. El blog de rikrdoavella.over-blog.es. http://rikrdoavella.over-blog.es/article-stdio-h-y-sus-funciones-55958648.html

Babiuch, M., Foltýnek, P., & Smutný, P. (2019, 26-29 May 2019). *Using the ESP32 Microcontroller for Data Processing.* 2019 20th International Carpathian Control Conference (ICCC),

Barybin, O., Zaitseva, E., & Brazhnyi, V. (2019, 8-11 Oct. 2019). *Testing the Security ESP32 Internet of Things Devices.* 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T).

Corob-Msft. (2022, 28 mayo). *Funciones (C++)*. Microsoft Docs. https://docs.microsoft.com/es-es/cpp/cpp/functions-cpp?view=msvc-170

DPL News. (5 de 8 de 2021). *Ecuador | Consejo de Comunicación recibió ataque informático a sus sistemas.* Obtenido de https://dplnews.com/ecuador-consejo-de-comunicacion-recibio-

ataque-informatico-a-sus-sistemas/#:%7E:text=Solo%20durante%202020%2C%20seg%C3%BAn%20ESET,mil%20detecciones%20de%20ransomware%20(malware

Espinosa, C. (2019, 30 agosto). *¿El primer ataque DoS de la historia?* SecurityInside.info. https://securityinside.info/el-primer-ataque-dos-de-la-historia/

Espressif Systems. (2016). *Non-volatile storage library - ESP32 Programming Guide latest documentation*. ESP-IDF Programming Guide. https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/storage/nvs_flash.html

Espressif Systems. (2016). *Preferences - Arduino ESP32 2.0.2 Documentation*. Arduino-ESP32 Espressif. https://docs.espressif.com/projects/arduino-esp32/en/latest/tutorials/preferences.html

Espressif Systems. (2016). *Wi-Fi - ESP32 Programming Guide latest documentation*. ESP-IDF Programming Guide. https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_wifi.html?highlight=esp_wifi#_CPPv417esp_wifi_80211_tx16wifi_interface_tPKvib

Federal Bureau of Investigation. (2020, 29 septiembre). *2019 Internet Crime Report Released.* https://www.fbi.gov/news/stories/2019-internet-crime-report-released-021120

Ferrer, R. (2017). *Metodología de Análisis de Riesgo.* SISTESEG.

Heltec Automation. (2019a). *NEW WIFI Kit 32 Pinout Diagram.* Docs & Resource. https://resource.heltec.cn/download/WiFi_Kit_32/WIFI_Kit_32_pinoutDiagram_V2.pdf

Heltec Automation. (2019b). WIFI Kit 32 Schematic Diagram. Docs & Resource. https://resource.heltec.cn/download/WiFi_Kit_32/WIFI_Kit_32_Schematic_diagram_V2.pdf

Hietala, H. (2020, 16 abril). How to use the Heltec OLED display on the ESP32. Sabulo, Inc. https://www.sabulo.com/sb/esp32-development-board/how-to-use-the-heltec-oled-display-on-the-esp32/

Holloway, C. (2020, 8 octubre). *El estado de la seguridad IT en 2020: Las superficies de ataque se amplían y las personas nunca fueron tan importantes para la defensa.* IT Masters Mag. https://www.itmastersmag.com/informes-whitepapers/el-estado-de-la-seguridad-it-en-2020-las-superficies-de-ataque-se-amplian-y-las-personas-nunca-fueron-tan-importantes-para-la-defensa/

Levy, Steven; *Hackers*, Anchor/Doubleday 1984, ISBN 0-385-19195-2.

Martínez, R. C. Z. (2011). *"ANÁLISIS Y CAPTURA DE PAQUETES DE DATOS EN UNA RED MEDIANTE LA HERRAMIENTA WIRESHARK."* 139.

Microsoft. (2021, 7 diciembre). *Librería cstddef*. Microsoft Docs. https://docs.microsoft.com/en-us/cpp/standard-library/cstddef?view=msvc-170

Ndatinya, V., Xiao, Z., Manepalli, V. R., Meng, K., Xiao, Y. J. I. J. o. S., & Networks. (2015). *Network forensics analysis using Wireshark. 10*(2), 91-106.

Palo Alto Networks. (2019). *What is a denial of service attack (DoS)?* https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos

Patil, S., Jangra, A., Bhale, M., Raina, A., & Kulkarni, P. (2017, 21-22 Sept. 2017). *Ethical hacking: The need for cyber security.* 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI).

Rincón Cruz, E. A. (2017, 5 agosto). *FreeRTOS.* CoffeeBrain-Wiki. http://www.coffeebrain.org/wiki/index.php?title=FreeRTOS#:%7E:text=Descripci%C3%B3n%3A,en%20microcontroladores%20o%20microprocesadores%20peque%C3%B1os.

Trabelsi, Z., & Ibrahim, W. (2013, 13-15 March 2013). *Teaching ethical hacking in information security curriculum: A case study.* 2013 IEEE Global Engineering Education Conference (EDUCON).

USERS-pdf Long, J., Gardner, B., Brown, J. (2016). *Google Hacking For Penetration Testers.*

Vargas, G., et al. (2019). *Obtención de claves en redes WLAN/WPS usando Wifislax y Denegación de Servicios con Kali Linux.* (E18): 318-331.

Verizon Enterprise Solutions. (2020, 19 mayo). *Cyber Security Basics.* Verizon Enterprise. https://www.verizon.com/business/resources/reports/dbir/2020/introduction/

Wang, Y., & Yang, J. (2017, 27-29 March 2017). *Ethical Hacking and Network Defense: Choose Your Best Network Vulnerability Scanning Tool.* 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA).

Yevdokymenko, M., Mohamed, E., & Onwuakpa, P. (2017, 10-13 Oct. 2017). *Ethical hacking and penetration testing using raspberry PI.* 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T).

Zhao, Z., Gong, D., Lu, B., Liu, F., & Zhang, C. (2016). *SDN-Based Double Hopping Communication against Sniffer Attack.* Mathematical Problems in Engineering, 2016, 8927169. https://doi.org/10.1155/2016/8927169