BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

Análisis de las Capacidades Ofensivas y Defensivas en Dispositivos IoT de Bajo Costo en Arquitecturas tipo Edge Computing

Diciembre 2022

Tesis para obtener el grado académico de Licenciatura en Ingeniería en Ciencias de la Computación

Presenta:
Jorge García Adauta

Asesora de Tesis:
M. C. Ana Claudia Zenteno Vázquez

Contenido

Lista de abreviaturas y definiciones

IP: Internet Protocol

ICMP: Internet Control Message Protocol

Ping: Comando utilizado para comprobar conectividad entre dispositivos

DHCP: Dynamic Host

VLAN: Redes de área local virtuales

Smartphones: Termino popular para referirse a un teléfono celular actual

Smartwaches: Dispositivo electrónico tipo pulsera con varias funciones, incluida la conexión con otros dispositivos

SmartTVs: Televisor con múltiples funciones, incluida la compatibilidad con varias aplicaciones y conectividad a internet

IoT: Internet de las cosas

AP: Punto de acceso

La nube: Termino popular para referirse a internet

DNS: Domain Name Service

Handshake de tres vías: Proceso para establecer comunicación

SSID: Service Set Identifier o identificador de red, es decir, el nombre público que recibe una red.

API: Application Programming Interface o Interfaz de programación de aplicaciones

Dedicatoria

A mis padres por brindarme la vida y la oportunidad de estudiar en una universidad.

A aquellas maestras que me impulsaron en el área de redes y ciberseguridad; su dedicación y empeño a su labor motivaron a un estudiante para escribir una tesis.

A aquello que se encuentra en algún plano existencial, por brindarme la oportunidad de cumplir mis sueños, aún después de tantos contratiempos en la vida.

1. Resumen

Actualmente la información viaja a través de sistemas de red que incorporan una infinidad de dispositivos, como es el caso de videocámaras, teléfonos inteligentes, sensores, sistemas inteligentes, tabletas, entre otros. Gracias al IoT (Internet de las cosas) los sistemas de redes ya no están limitados a ser conformados únicamente por computadoras, sino que ahora, pueden estar conformados por múltiples aparatos que poseen la capacidad de acceder a una red. De acuerdo con muchos autores como Barrio (2018), se define como IoT (Internet de las cosas) a la intercomunicación entre dispositivos a través del internet, se incluyen sensores y actuadores con módulos de conexión Wi-fi. En este sentido, estos dispositivos poseen la capacidad de conectarse a las redes locales para obtener acceso a servidores y transmitir datos en tiempo real.

Dado que estos dispositivos pueden recibir y transmitir datos son susceptibles a ataques informáticos, o, en otro tenor a ser utilizados como herramientas de ataque. Por otra parte, debido a la reciente contingencia global suscitada por el virus COVID-19 una gran ola de nuevos usuarios y dispositivos se han incorporado a la internet, y, en consecuencia, una extensa superficie de ataque llena de usuarios y administradores que no son conscientes del potencial de ataque que representan los dispositivos IoT.

El presente trabajo tiene como objetivo realizar un análisis de las capacidades ofensivas que representan estos dispositivos IoT, desde el ámbito de la ciberseguridad. También se desarrolla una implementación demostrativa de este potencial, a través del módulo programable ESP32, el cual, posee la capacidad de conectar a internet. Sí bien, existen otros módulos programables adecuados para realizar actividades de hacking y pentesting, el objetivo principal es ofrecer una reflexión sobre el peligro que representan estos dispositivos de bajo costo.

Por último, se considera el estudio de escenarios en arquitecturas donde los datos recogidos por los dispositivos IoT se procesan y transmitan en ambiente que asemeja a un ambiente local, con la finalidad de observar el nivel de penetración de estos dispositivos pueden lograr a través de la técnica de ingeniería social: ROGUE AP. Para este caso, la arquitectura emergente *Edge computing* o computo de borde contemplan en gran medida el uso de dispositivos IoT, por lo que es idónea para desarrollar el presente trabajo, ya que, ejemplifica el potencial que representan los dispositivos IoT como vectores de ataque.

2. Introducción

2.1 El internet y las redes

El mundo en el que vivimos está conectado a través de la **internet**; en su concepto más general, comprendemos que la internet es lo que nos permite realizar búsquedas en navegadores, usar redes sociales, enviar correos, ver videos y usar plataformas de streaming. En otras aplicaciones más específicas nos permite tomar clases en línea desde nuestros hogares, asistir a juntas de trabajo con proveedores que se encuentran en otros estados desde nuestras oficinas y en un orden reciente, tener la capacidad usar los procesadores y los recursos de computadoras que se encuentran en otros países desde la comodidad de nuestros dispositivos. Sin embargo, a pesar de conocer todas sus bondades aún queda la incógnita más importante ¿qué es el internet?

En un primer momento asociamos el internet con nuestro Proveedor de Servicios de Internet (ISP), es decir, la empresa con la que contratamos un servicio de internet, generalmente esta empresa entrega un equipo tipo "modem" para realizar dicha conexión a "internet", sin embargo, este concepto no es del todo correcto. Una de las definiciones más universales y que corresponden a la cultura popular respecto al internet consiste en el siguiente enunciado, *el internet es la red de redes*. Dicha definición suena impresionante a primer oído, sin embargo, no es del todo clara, específicamente cuando no se tiene claro la definición de "red".

Para explicar qué es una red, primero debemos repasar un poco de historia, el internet como lo conocemos se remonta hacia 1969, con el famoso proyecto ARPANET creado por el departamento de Defensa de los Estados Unidos. ARPANET consistía únicamente de cuatro computadoras de tipo host conectadas entre sí, cada una en una ubicación diferente: en la UCLA-Universidad de California, en el Instituto de Investigaciones de Standford, en la Universidad Santa Bárbara California y en la Universidad de Utah. A través de estas cuatro computadoras se estableció una comunicación mediante una red descentralizada, de tal forma que cada una de las computadoras podía enviar y recibir información (Fresno, 2018).

De esta forma, si entendemos como *"red*" a la comunicación entre computadoras, podemos entender que el famoso proyecto ARPANET es una de las primeras redes de computadoras. Como lo mencionan los autores Molina & Polo, *"Internet es una Red que está enfocada al intercambio de información entre usuarios y equipos*". Bajo estos términos, la anterior definición, *"el internet es una red de redes*" parece un poco más clara, pues a través de diferentes computadoras, servidores, conmutadores, routers y de más equipos enfocados al envío y recepción de información se constituye una inmensa red, que a su vez se compone de redes más pequeñas.

El objetivo más importante de una red es ofrecer un intercambio de información, este intercambio se realiza a través de diferentes protocolos, modelos, estructuras y jerarquías; gracias a estas estructuras es posible acceder a servicios como es el caso de transferencia de archivos, correos electrónicos, mensajería, conexión remota a otros dispositivos, acceso a páginas web, reproducción de medio multimedia como videos, audios, etc. (Molina & Polo, 2015). A través de estos servicios se compone nuestra interacción conceptual de lo que normalmente conocemos como *"internet"*.

Precisamente, es esta falta de conocimiento conceptual la que dificulta nuestras primeras interacciones con el internet y con la tecnología en general. Si bien es cierto que no todos los usuarios podemos ser expertos capacitados en el tema, se debe crear conciencia sobre como el internet actualmente ha logrado un proceso de digitalización de nuestra vida, abarcando influencia en nuestros trabajos, escuelas y vida personal. Actualmente subimos nuestros documentos personales a internet en páginas web de gobierno, nos inscribimos en universidades a través de internet e incluso pagamos nuestros servicios tecleando los números de nuestras tarjetas de crédito en un portal web.

Retomando el concepto de que internet es una red compuesta de varias redes, entonces es relevante poner preguntarse ¿a dónde estamos enviando nuestra información? Cuando recibimos un correo que nos pide nuestros datos, ¿a qué computadora, servidor, host estamos enviando nuestra información? Cuando descargamos algún archivo, ¿de qué lugar estamos descargando dicha información? Una vez que se toma conciencia sobre acciones cotidianas como descargar un archivo, surgen más cuestionamientos. En párrafos anteriores se mencionó que el internet funciona a través de protocolos, jerarquías y procesos; con ello en mente, es natural preguntarse acerca de los procesos que llevan a cabo en internet, es decir, sobre cómo se realiza dicha comunicación entre computadoras, dentro de una red o hacia otras redes.

Sin embargo, antes de continuar es necesario formalizar el concepto que tenemos respecto a **qué es un a red**, Molina Robles lo describe como *"el conjunto de computadoras autónomas que se encuentran interconectadas mediante algún medio de transmisión".* Otro concepto útil corresponde a **red de transmisión de datos**, el cual describe a *"la estructura formada por medios físicos y lógicos para satisfacer las necesidades de comunicación de alguna zona"* (Molina, 2015). Estas definiciones de red son importantes, ya que el concepto de red puede ser muy amplio dependiendo del contexto. Por ejemplo, haciendo una comparación de infraestructuras de red, la red de una casa conformada por un "modem" y una laptop, es extremadamente simple en comparación con la red de una universidad grande, que puede estar conformada por antenas de teléfono, switches, routers, servidores, firewalls, enlaces, cables de cobre, fibra óptica, etc. Mientras la red de una casa solo se encarga de dar salida a internet a una computadora, la red de una universidad está diseñada para comunicar campus o sedes que se encuentren en diferentes ciudades.

De la misma forma, el acceso a internet no es lo mismo que la *red*. Si bien, la misma infraestructura de red es la que permite que cada una de las computadoras conectadas puedan a salir a internet, al mismo tiempo, la conexión a internet es un tema que puede ser independiente a la red. Por ejemplo, en el caso de que el proveedor de internet tenga una falla o presente problemas en sus servicios, sus usuarios y por defecto las redes que dependan de este servicio se verán afectadas a pesar de que la misma infraestructura de red permita una perfecta comunicación a nivel local entre las computadoras conectadas; en este sentido, la red estaría funcionando perfectamente, puesto que hay conexión entre cada uno de sus equipos. Los equipos que conforman una red son importantes; permiten enviar la información y conocerlos, nos permite entender como viaja la información hacia el internet o en su defecto hacia otros equipos.

2.2. Las redes y sus elementos

Una vez comprendida la importancia de las redes y su relación con el internet podemos comenzar a identificar los elementos que las componen, así como sus características. De acuerdo con Castaño (2015), en su libro *"Redes locales"* podemos identificar algunos de los componentes principales de una red:

- **Hosts:** Conocidos comúnmente como equipos terminales o finales, se trata de dispositivos que inician o terminan con un proceso de comunicación en una red, pueden ser emisores o receptores. Estos equipos pueden ser: computadoras (clientes o servidores), periféricos (escáneres, impresoras, etc.), u otros dispositivos (celulares, tablets, IoT, etc).
- **Medios de transmisión:** Normalmente llamado simplemente "medio" se refiere al ambiente, material o condiciones por las cuales viajan los paquetes de datos. Por ejemplo, mediante cables, como son: cables telefónicos, cables coaxiales, cable UTP, Cable STP y fibra óptica. Recientemente, gracias a la tecnología inalámbrica, también es posible utilizar la codificación de ondas electromagnéticas en el vacío como medio de transmisión (mejor conocido como WiFi).
- **Equipos intermedios:** Se trata de los elementos más comunes dentro de una red local, son nodos (o equipos) que se colocan entre los equipos terminales y la conexión a internet. Se encargan de repetir señales de red, enrutar los paquetes y de la administración de la red. Entre estos equipos encontramos:

    - **Switches (conmutadores):** Se trata de dispositivos que funcionan como puentes multipuertos que poseen características para la administración de red, como es el caso de funciones para dividir la red en segmentos, logrando así que no haya colisiones de red. También existen dispositivos como los *hubs* los cuales son similares en funcionamiento, la diferencia entre estos es que los switches pueden tener múltiples dominios de broadcast y de colisión, es decir, posen la capacidad de segmentar la red.
    - **Enrutadores (routers):** Se encargan de distribuir el tráfico de la red, son útiles cuando se tienen múltiples redes LAN, y se necesita que los paquetes de datos sigan una ruta definida. También se utilizan para comunicar diferentes redes entre sí, y, pueden ser utilizados para comunicar una red a internet.
    - **Cortafuegos (firewall):** Dispositivo dedicado a la seguridad de la red, actúa como un router, pero con la capacidad de analizar el tráfico de la red y aceptar o rechazar los paquetes que contiene, es decir, se encarga de filtrar la información que entra hacia la red o que sale de la red.
    - **Modem:** Se trata del equipo que nos conecta a internet o con el ISP (Internet Service Provider). Su nombre se debe a que envía señales de datos a través de los cables de teléfono mediante la modulación-demodulación de la señal. Actualmente ya no es común su uso, debido a que ahora es posible usar routers inalámbricos.

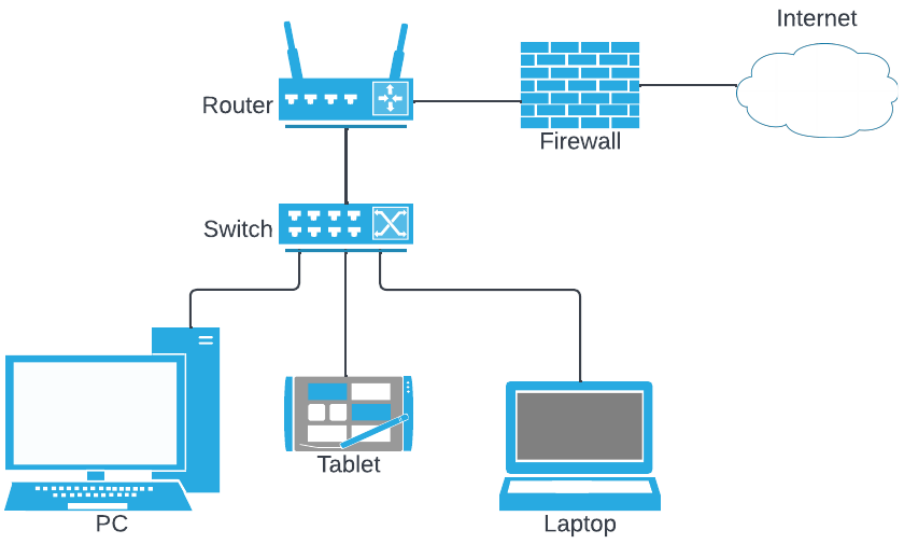

Figura 1. Representación de los elementos de una red local (Fuente: de elaboración propia)

De acuerdo con los elementos anteriormente descritos y con ayuda de la *figura 1* se construye una representación de los elementos que conforman una red. De manera adicional, en *Figura 1* se aprecian representaciones conceptuales de algunos dispositivos de red; estas interpretaciones se usan comúnmente en la industria para diferenciarlos de otros dispositivos, por ejemplo para el caso de un Firewall, en la vida real su apariencia puede asemejarse a un Router (dependiendo de la marca y modelo), mientras que para el caso de "el internet", un símil más apropiado sería la representación de múltiples redes conectadas entre sí, pero con fines conceptuales se representa mediante una nube. Si bien, existen aún más elementos que conforman una red, los descritos anteriormente son los más importantes para comprender como se conforma una red. Otros conceptos importantes para entender se relacionan con el tipo de red y su función:

- **Redes de Área Local (LAN):** Se trata de redes usadas en ámbitos privados como edificios, casas e instalaciones pequeñas de pocos kilómetros. Consiste en la interconexión de distintas máquinas para compartir recursos y datos.
- **Redes de Área Extendida (WAN):** Como su nombre lo indica, se trata de una red que abarca grandes distancias como es caso, de países, continentes y grandes porciones del planeta, como es el caso del internet.
- **Redes de Área Metropolitana (MAN):** Similares a las redes LAN, pero con una extensión aun mayor, capaces de abarcar ciudades o grupos de edificios. Se especializa por el estándar DQDB (Bus Dual de Cola Distribuida) creado por la IEEE para conectar redes LAN y WAN (Sánchez et. al, 2020).

Ahora que conocemos ciertos elementos y características sobre una red es momento de agregar cierta complejidad a la definición de internet que visitamos en los primeros párrafos, por lo tanto, podemos hablar de conceptos como el que presenta Sánchez et. al (2020), *"Internet es la unión de redes interconectadas que utilizan la familia de protocolos TCP/IP"* (2020). Otra definición, corresponde a la Internet Society (2022), el cual es un organismo enfocado a la preservación del

internet; *"*sistema de información global que se encuentra enlazado globalmente mediante direcciones únicas basadas en el protocolo IP, compatible con el protocolo TCP/IP, capaz de proveer accesibilidad a servicios de alto nivel superpuestos en las comunicaciones…".

El internet y la redes sin duda alguna ofrecen un mecanismo de comunicación poderoso, con el alcance suficiente para cubrir al mundo, no obstante, para que un sistema como este sea capaz de funcionar es necesario que existan ciertas reglas o mecanismos que garanticen sus procesos, en este caso ese el papel que desempeñan los protocolos. Históricamente, el nacimiento de internet como lo conocemos está ligado a la implementación del protocolo TCP/IP. El uso de protocolos es algo necesario para que el internet cumpla con una de las características que menciona la Internet Society, la **disponibilidad**; la cual se puede traducir como la posibilidad de que cualquier computadora pueda de comunicarse con otra computadora al otro lado del mundo, es decir, el acceso a internet desde cualquier parte del mundo y en cualquier momento.

Se puede definir como protocolo de red a un conjunto de reglas establecidas que deben seguir los dispositivos de cómputo, desde hosts hasta dispositivos intermedios. Estas reglas definen el formato, forma de transmitir y recibir datos. Los protocolos se usan para que ocurra el envío y recepción de información de manera exitosa, sin importar el tipo de infraestructura, diseño o estándar. Es decir, si cada computadora tuviera un diferente protocolo la comunicación sería imposible lograr una comunicación universal, por tanto, los protocolos siguen un proceso de estandarización para que las computadoras de cualquier parte del mundo sean capaces de comunicarse entre sí (Chai & Irei, 2021).

Por ejemplo, retomándolo antes mencionado, la familia de protocolos TCP/IP es uno de los pilares de la internet. Dentro de este conjunto de protocolos, encontramos el protocolo IP; el cual, es el encargado de ofrecer direccionamiento lógico, lo cual nos permite identificar cada extremo (emisor/receptor) en una comunicación mediante direcciones IP. Por otra parte, TCP es un protocolo que ayuda a la comunicación fiable, enfocado a verificar que los datos lleguen a su destino y retransmitiendo automáticamente aquellos datos faltantes; esto es posible gracias a que TCP trabaja con números de secuencia, de esta forma etiqueta los datos para lleguen de forma ordenada y sin duplicarse.

2.3 Protocolos: TCP/IP vs OSI

Como bien menciona Fresno (2018), los protocolos definen la sintaxis a utilizar (formato), semántica (información) para coordinar el envío de información y temporización (útil para sincronizar). Gracias a ellos es posible realizar nuestras actividades cotidianas en internet. Existen protocolos enfocados al envío de correos electrónicos, otros enfocados a archivos, etc. Es importante no confundir los protocolos TCP e IP con la pila de protocolos TCP/IP, debido a que los primeros hacen referencias a los protocolos mencionados en el párrafo anterior, mientras que la pila de protocolos TCP/IP hace referencia a un conjunto de protocolos que conforman una arquitectura de cuatro capas, de acuerdo con Molina (2015) estas capas son:

- **Aplicación:** Esta capa es la más cercana al usuario, y en ella habitan los protocolos de alto nivel, como, por ejemplo, SMTP para el correo electrónico, FTP para la transferencia de

archivos, HTTP para la trasferencia de archivos de hipertexto (el cual se usa para que los navegadores web puedan visualizar las páginas web).

- **Transporte:** En esta capa se encuentran los protocolos encargados del transporte de los datos (paquetes), como es el caso de los protocolos TCP y UDP (User Datagram Protocol), el cual a diferencia de TCP se caracteriza por no estar orientado a la conexión y por ende ser menos confiable.
- **Internet:** Esta capa se encarga del funcionamiento de la red a un nivel lógico, es decir, se encarga principalmente de brindar el direccionamiento de los datos, por lo que encontramos el protocolo IP, ICMP, ARP, entre otros.
- **Interfaz de red:** También llamada capa de enlace, interfaz de red, etc. Se refiere a la capa que posee la interacción directa con los medios físicos, como es el caso de dispositivos intermedios como routers, switches, hosts, etc. Esta interacción ocurre a través de la conexión del medio (cables, WiFi, fibra, etc) a las interfaces de red que poseen estos dispositivos para así, dar un formato a los datos.

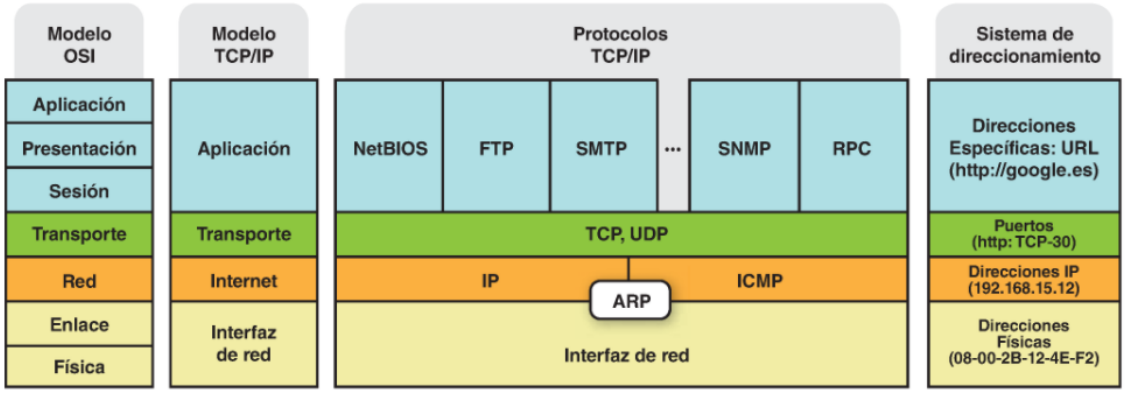

Figura 2. Pila de Protocolos TCP/IP en contraste con Modelo OSI (Fuente: ABAD DOMINGO, 2013)

Se observa en la figura 2 que existe una comparación directa de la Pila de protocolos TCP/IP con el Modelo OSI. Esto se debe a la estandarización de las redes; un estándar es un modelo que siguen los diferentes fabricantes de tecnología para que cada dispositivo fabricado sea compatible con cualquier otro. Existen varios organismos reguladores para estos estándares, como son la *Unión Internacional de Telecomunicaciones* (ITU), La O*rganización Internacional para la Estandarización* (ISO), la *Comisión Eléctrica Internacional* (IEC), el *Instituto de Ingenieros Eléctricos y Electrónicos* (IEEE). A través de estos organismos surgen varios estándares, como ejemplo, el **Open System Interconnection** (OSI). OSI es una propuesta por parte de la ISO para unificar las redes y servir como un modelo de referencia para todos los fabricantes de tecnología. El modelo OSI se publicó en 1984 pero en ese entonces se trataba de un modelo bastante complejo de implementar, sin embargo, gracias al surgimiento de ARPANET surgió otra propuesta más sencilla de implementar: la pila de protocolos TCP/IP.

Actualmente el modelo OSI es utilizado ampliamente como un modelo de referencia, ya que permite describir con detalle cada uno de los elementos que participan en un proceso de comunicación. Se divide en siete capas, cada una abarca un aspecto en específico de la comunicación, desde aspectos que involucran al usuario, mientras que otras capas se enfocan en las operaciones que realizan los dispositivos de red. Por otra parte, un aspecto que sale a relucir

cuando se comparan ambos modelos (TCP/IP vs OSI), es que existen algunas similitudes entre sus capas. Las similitudes entre el modelo OSI y TCP/IP se deben a la simplificación del modelo OSI, reduciéndolo a cuatro y tomando como base TCP/IP (Castaño, 2013). Actualmente OSI se trata de modelo utilizado en la industria; su modelo consiste en siete capas, cada capa se especializa en tareas en concreto, lo cual ayuda a reducir la complejidad. Las capas superiores se relacionan con la aplicación y tienen una interacción más directa con el usuario, mientras que las inferiores se encargan de la transferencia de los datos, e interactúan directamente con el software y hardware.

Cada capa se comunica con sus equivalentes en otros sistemas mediante sistemas de control definidos a través de formatos, estos formatos se definen como *headers y finalizadores;* estos, se agregan respectivamente al principio y al final de los datos que agregan las capas superiores hacia las inferiores. Para ejemplificarlo se presenta la *figura 3*:

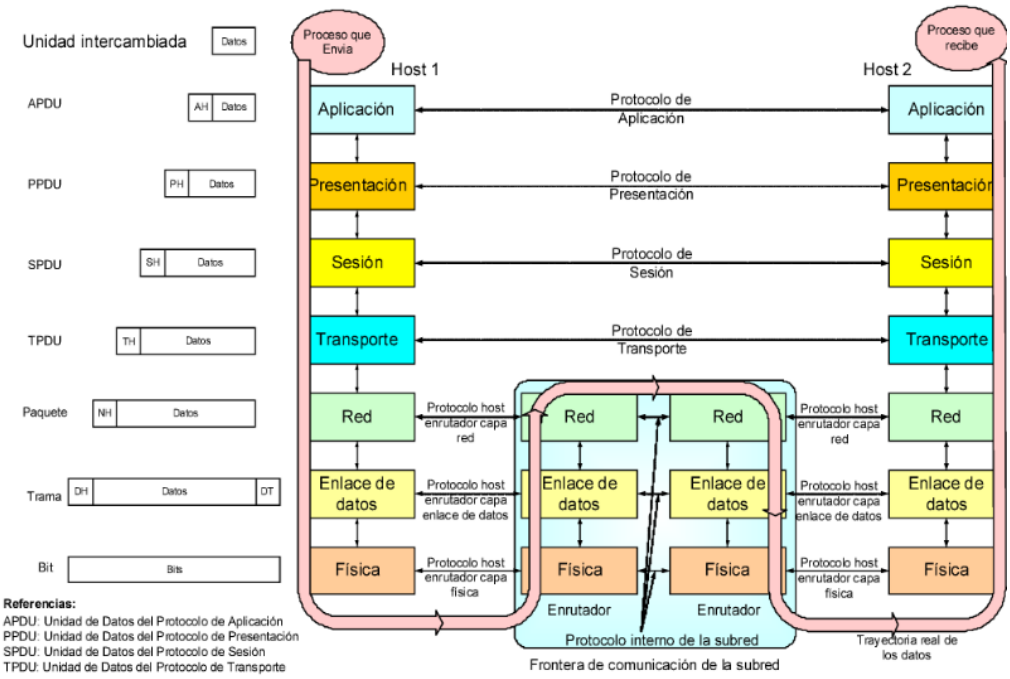

Figura 3. Esquema de comunicación de las capas del modelo OSI (Fuente: Riso & Saibene, 2020)

De manera inversa, conforme se envían los datos a las capas superiores la información se transforma, esto debido a la naturaleza de cada capa, por ejemplo, en la capa ***enlace de datos*** operan dispositivos como switches, y estos deben percibir la etiqueta de *VLAN* (la cual se utiliza para identificar el segmento de red) que posee cada trama, para así redirigirlo a su destino correspondiente, sin embargo, en el caso de un usuario que este contacto con la capa de ***aplicación*** esta información es irrelevante. Cada capa posee características especiales, de acuerdo con Riso & Saibene (2020) se describen:

- ***Capa física:*** Abarca las interfaces físicas de los dispositivos, así como la forma en que pasan los bits a través de estas. Contempla características y propiedades del medio de

transmisión, como tipos de conectores a usar, niveles de tensión de funcionamiento y funcionalidad para cada dispositivo.

- ***Capa de enlace de datos:*** Proporciona los medios para mantener, detener o comenzar la comunicación a través de mecanismos para la detección de errores y control del flujo de la comunicación. Esta capa principalmente interviene en la comunicación de las redes de área local (LAN), además garantiza el acceso a la red y el control del flujo del acceso al medio. La importancia de esta capa radica en los protocolos que interactúan en este nivel, como es el caso de Ethernet, MAC, las direcciones físicas, ARP, Spaning Tree Protocol, etc.

- ***Capa de red:*** Se encarga de proporcionar los medios para la transferencia de la información entre sistemas finales, liberando así a las capas superiores de cargas de información relacionadas a la transmisión de los datos y mecanismos para la conexión. Esta capa es crucial para la comunicación ya que define la dirección del destino y el tipo de comunicación. Ofrece ruteo mediante direccionamiento lógico, fragmentación y reensamblado de los paquetes que son transmitidos a través de diferentes redes. Por supuesto que el protocolo más importante asociado a esta capa es el IP, aunque también se le pueden añadir ICMP, NAT, BGP, OSFP, etc.

- **Capa de transporte:** Se encarga de definir los mecanismos para intercambiar los datos, es decir, los protocolos y normas para establecer una comunicación. Otras de sus funciones es la optimización de los servicios de red en función del uso, por ejemplo, garantizar el protocolo de transporte adecuado de acuerdo con el tipo de servicio en uso. La complejidad de la capa de transporte depende del tipo de protocolo de transporte usado, como es el caso de los protocolos vistos anteriormente (TCP y UDP).

- ***Capa de sesión:*** Esta capa permite establecer sesiones; se puede definir como sesión a la sincronización entre dos dispositivos que pretenden comunicarse, es decir, una conexión activa entre dos hosts. Las sesiones permiten el transporte de los datos, pero a diferencia de la capa de transporte las sesiones proporcionan servicios mejorados de acuerdo al uso de ciertas aplicaciones. Por ejemplo, es posible usar una sesión para que un usuario se conecte de manera remota a un sistema en tiempo real para compartir archivos. Otra función de esta capa es controlar el flujo de una comunicación, ya que se pueden definir que el tráfico vaya en ambos sentidos o en uno solo.

- ***Capa de presentación:*** Se encarga de las funciones de codificación y conversión que se aplican a los datos que recibe la capa de aplicación, estos mecanismos aseguran que la información en la capa de aplicación sea legible para otros sistemas. Algunos ejemplos de esto son esquemas de compresión de datos y procesos de encriptación de datos.

- ***Capa de aplicación:*** Por último, la capa de aplicación proporciona un medio para que los equipos y los usuarios puedan entender los datos; de la misma forma se encarga de ofrecer medios para que las aplicaciones puedan acceder al entorno OSI.

2.4 Después del internet: el Internet de las Cosas y otras arquitecturas

La comprensión de la internet y de las redes es algo complejo, abarca un amplio conjunto de protocolos, medios y estándares, no obstante, la tecnología avanza cada día, tan solo en las últimas décadas surgieron *smartphones, smartwaches, smartTVs, etc*. Antes, solo era posible acceder a internet mediante una computadora, pero ahora es posible hacerlo mediante relojes, teléfonos, refrigeradores, televisiones y etc. Por supuesto, estos dispositivos al contar con direccionamiento lógico (una dirección IP), también se consideran dispositivos hosts y por ende también forman parte de una red. El incremento de los dispositivos capaces de acceder a una red trae consigo múltiples consideraciones que de hecho se abordan en *the internet of things* (Internet de las cosas).

Como lo menciona Barrio (2018) en su libro "*Internet de las cosas"* el avance tecnológico en microprocesadores y sus comunicaciones ha permitido el desarrollo de sensores capaces de recopilar datos y transmitirlos, esto supone la base para definir que es el *IoT (internet of things)*, la cual se puede pensar como, *"la evolución de Internet desde una red de ordenadores interconectados hasta una red de objetos interconexionados"*. Actualmente, el IoT es una fuente para la recolección de datos, (y crece exponencialmente) gracias tecnologías como el *cloud computing.* Adicionalmente, no solo se trata de tabletas o *smartphones*, también se contemplan automóviles, videoconsolas, sensores de temperatura, cámaras de seguridad, controles de acceso; como su nombre lo indica, se trata de "cosas" con la capacidad de conectarse a internet.

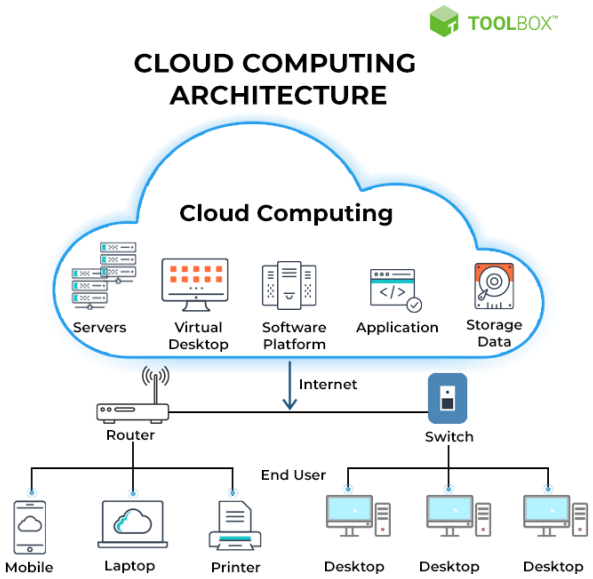

Figura 4. Diagrama que ejemplifica gráficamente el computo en la nube (Fuente: Prajakta Patil & Chiradeep BasuMallick, 2022).

Precisamente, el desarrollo de las comunicaciones a internet ha permitido que surjan arquitecturas en las cuales es posible acceder a recursos que se encuentran en la nube, es decir, en internet, como se muestra en la figura 4. *The cloud computing;* que se define como un paradigma basado en el uso de aplicaciones, servicios, cómputo y almacenamiento que no se encuentran en la computadora del

usuario, sino que se encuentran en un *Datacenter* que permite a los usuarios acceder a aplicaciones y servicios mediante internet (a través de la *nube)*. Pese a sonar novedoso, cotidianamente hemos tenido interacciones con la nube, por ejemplo, cuando abrimos una aplicación de *streaming* en nuestra televisión, en realidad estamos accediendo a un catálogo de archivos de películas ubicadas en varios servidores ubicados en otro país, o también, cuando usamos servicios como Google Drive y Dropbox en realidad estamos guardando archivos en otras computadora o más bien en un conjunto de computadoras alojadas en centros enormes, es decir, en *Datacenter*s (Villarino, 2018).

Por otra parte, existen otros paradigmas, como es el caso del *edge computing*, el cual resulta más sencillo de entenderlo a través de otro paradigma; mientras el *cloud computing* ofrece procesamiento, almacenamiento y servicios en la nube, el edge computing hace lo contrario, se centra en que estas acciones ocurran en un ambiente más local. Como menciona la empresa IBM (SAa), "Edge computing actúa con base en los datos en el origen", enfocándose en el origen de los datos, como es el caso de dispositivos IoT, servidores y periféricos locales. Sin embargo, este es un tema que se abordará con mayor profundidad en capítulos posteriores.

2.5 Acerca de la ciberseguridad

Hemos podido comprobar la complejidad que requiere hablar sobre internet, y la recompensa a ello, es comprender mejor las tecnologías con las que coexistimos todos los días, sin embargo, internet es tan basto como el mundo, y sus usos y aplicaciones son igualmente de inmensas, por ello, el presente trabajo enfoca su estudio al edge computing en un campo de acción muy específico: la ciberseguridad en dispositivos IoT de bajo costo. Es casi imposible como usuarios tecnológicos no haber leído alguna noticia sobre la ciberseguridad, ya sea en noticias, redes sociales o por conocimiento general, quizás nuestro primer acercamiento a este término haya sido al acercarnos por primera vez a una computadora y enterarnos que se necesita de un "antivirus".

Por otra parte, la ciberseguridad es un tema igual de extenso porque puede aplicarse a cualquier nivel de internet, como ya vimos en el modelo OSI que abarca cada aspecto de la comunicación, la ciberseguridad también puede ir a la misma profundidad e incluso salir hacia el exterior, buscando directamente al eslabón más factible para ser atacado: el usuario. La ciberseguridad es tan antigua que se remonta al nacimiento de la internet, e incluso antes, y al igual que la tecnología los riesgos y vulnerabilidades incrementan con el pasar del tiempo, tan solo hace 10 años, el concepto *IoT* se escuchaba como algo novedoso y hoy en día se trata de una realidad tecnológica.

Precisamente estos avances tecnológicos van de la mano con la ciberseguridad, puesto que las amenazas y los diferentes tipos de riesgos se desarrollan a la par de la tecnología. Por tanto, de manera puntual, debemos abordar la siguiente pregunta: ¿Qué es la ciberseguridad? Para contestarla debemos entender primero qué es una amenaza. En el contexto de la ciberseguridad, se trata de aquellos elementos (métodos, técnicas, programas, etc.) que ponen en riesgo o comprometen los elementos de un sistema de red. De acuerdo con los párrafos anteriores una red se compone de varios elementos como: hosts (que pueden ser computadoras, laptops, etc.), dispositivos intermedios como switches, routers, etc. Por lo tanto, a través de estos dispositivos y mecanismos (refiriéndonos a los protocolos) podemos entender de qué forma pueden viajar estas amenazas de una red a otra.

Cómo lo menciona IBM (SAb), "La ciberseguridad es la práctica de proteger los sistemas importantes y la información confidencial de los ataques digitales". Estos sistemas se componen de elementos enfocados a realizar un proceso comunicación de manera interna, y también, para lograr una comunicación hacia otros sistemas; en este sentido, los sistemas de información son creados con objetivos específicos, por ejemplo, proveer de servicios de internet a toda una empresa, ofrecer servicios a clientes, establecer servidores para que una aplicación acceda a datos o pueda almacenarlos, por lo tanto, la ciberseguridad es aplicable para proteger estos sistemas y sus elementos. Sin embargo, se trata de una tarea extremadamente amplia, debido a que se puede aplicar a cada elemento de un sistema de la información.
Debido a la variedad de elementos existentes en un sistema de información existen diferentes tipos de amenazas, entre ellas, IBM es su publicación "*¿Qué es la ciberseguridad?" (SAb),* nos menciona las más populares:

- **Malware:** Se le conoce como malware a los diferentes softwares usados para dañar, obtener acceso, espiar o hacer uso de los recursos de algún dispositivo de cómputo. Estos pueden ser: gusanos informáticos, virus, troyanos, programas espías, entre otros.
- **Ransomware;** Se trata de una categoría de malware especializado en bloquear, borrar, encriptar o secuestrar los datos del disco duro de un dispositivo de cómputo. Generalmente este tipo de malware son desplegados con la finalidad de obtener un pago a cambio del rescate de la información comprometida.
- **Estafas e ingeniería social:** Consiste en una seria de métodos o técnicas que tienen como objetivo lograr que el usuario proporcione de manera voluntaria (mediante engaños) información que puede ser utilizada para lograr estafas o en su defecto obtener credenciales para lograr el acceso a algún sistema. Generalmente estos datos se usan con algún objetivo lucrativo.
- **Ataques de DDoS (Denegeación de servicios**): Este tipo de ataques tienen como objetivo bloquear algún servicio o elemento encargado de proporcionarlo, por ejemplo, bloquear una página web, interrumpir el funcionamiento de un servidor, entre otros.
- **Ataques de intermediario:** Consiste en un ataque enfocado al espionaje, por lo tanto, su principal característica consiste en interceptar mensajes y reenviarlos entre ambas partes, sin que estas se den cuenta. Su principal diferenciador es que este tipo de ataques se ejecutan directamente sobre el proceso de comunicación entre los dispositivos.

Por supuesto, existen más tipos de ataques, algunos especializados que únicamente pueden ser ejecutados a través de procedimientos especiales. No obstante, abarcar cada uno de estos ataques daría suficiente material para crear otra tesis por completo. Por ello, durante el desarrollo del presente se abordan únicamente aquellos ataques que son posibles ejecutar mediante un dispositivo IoT.

Finalmente, en capítulos posteriores se aborda con detalle, los riesgos, amenazas y ataques que es posible efectuar mediante dispositivos IoT. Posteriormente se estudia la arquitectura tipo *edge computing* para obtener un panorama sobre cómo es que los dispositivos IoT de bajo costo pueden representar un riesgo en este paradigma. Finalmente se hacen pruebas para estudiar la factibilidad de estos ataques y así construir un reporte con las conclusiones pertinentes.

3. Planteamiento del problema

La importancia del IoT radica en el impacto tecnológico que representó, ya que cambio la concepción de los paradigmas que se tenían respecto a internet, por ejemplo, como lo mencionó Cisco en su momento, *"El internet de las cosas (IoT) se puede considerar como un paso natural en el desarrollo de la internet*" (Evans, 2011). Otro ejemplo, tiene que ver con la evolución de la web como tal (no confundir con internet), en donde, la web (*World wide web o WWW)* se define como *"un sistema de hipermedia distribuido, accesible y enfocado a permitir la navegación y visualización de información de internet"* Adell (1995). En este contexto, la web tuvo varias etapas de desarrollo, de acuerdo con su contenido e interacción con el usuario; sea el caso de las primeras etapas de la web, en las cuales se conformaba por páginas web de texto plano y usuarios con necesidades meramente de consulta, posteriormente, en etapas más recientes, la web y sus usuarios se enfocaron a las interacciones sociales y contenido multimedia, gracias a la aparición de las *"redes sociales"* Evans (2011). Entonces, dada esta información ¿qué relación existe entre el IoT y la web? La respuesta es sencilla, actualmente ya no se trata de la interacción de los usuarios hacia la web, sino de las cosas hacia el internet; puesto que ahora existen sensores y dispositivos electrónicos capaces de conectarse a internet, acceder a la web e interactuar con ella.

Por otra parte, la importancia del IoT no se reduce a una contribución tecnológica, también se debe considerar su impacto sobre el mundo. Como lo mencionan Salinas, Galván, Guzmán & Orrante en su publicación "*El impacto del internet de las cosas (IoT) en la vida cotidiana";* la capacidad que poseen los dispositivos IoT de conectarse a una red representa una oportunidad de innovación, ya que se incluyen teléfonos, televisores, relojes, cámaras de vigilancia e incluso ciudades enteras. Precisamente uno de los temas más atractivos del IoT, son las ciudades inteligentes, las cuales en algunas partes del mundo son proyectos en desarrollo.

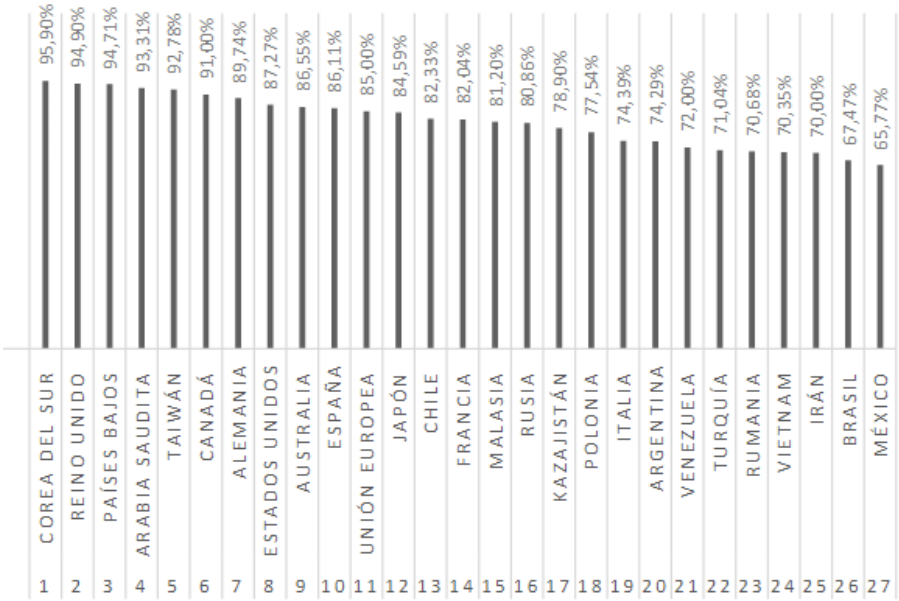

Figura 5. Ranking de países de acuerdo con el porcentaje de usuarios conectados a internet respecto a su población total (Fuente: Salinas, Galván, Guzmán, Orrante & Sakannassi, 2022).

Se muestra en la Figura 4, cada vez hay más usuarios conectados a internet, sumado a ello existen otros factores que se deben considerar, por ejemplo, el número de dispositivos conectados por usuarios; décadas atrás tener una computadora se consideraba un lujo, actualmente en gran parte de la población de los hogares mexicanos es relativamente sencillo encontrar una computadora con conexión a internet, sin embargo, no solo las computadoras son capaces de tener una conexión a internet. Entonces, se estaría hablando de que un usuario podría poseer varios dispositivos con acceso a internet, sin considerar una PC.

Por último, hablando de la importancia del *IoT*, existe un ferviente interés sobre sus aplicaciones y aplicaciones, sin embargo, esta euforia no se refleja con la misma intensidad en el tema de la *ciberseguridad*. Considerando la relevancia que posee el *IoT,* es prudente realizar un ejercicio mental pensar en todas las vulnerabilidades posibles que representan estos dispositivos a la seguridad en general. Si bien es acertado no magnificar los riesgos potenciales a la ciberseguridad que posee cada herramienta o tendencia tecnológica, para el caso del *IoT* es necesario, pero ¿por qué debería preocuparnos la ciberseguridad en el IoT?

La respuesta más directa es que las características del IoT se predisponen a ser un riesgo a la ciberseguridad. Como lo mencionan *Andrade & Gualli* en su publicación "*Retos de la ciberseguridad en ciudades inteligentes",* una característica de los dispositivos IoT es que fueron diseñados para ser discretos y sin interrupciones, por lo tanto, no cuentan con procesos de autorización de datos, sumado a ello, son dispositivos con recursos de memoria y procesamiento limitados, por lo que no es posible implementar la mayoría de las soluciones conocidas en ciberseguridad (Andrade & Gualli, 2019).

Otra respuesta, se relaciona con sus características propias, por ejemplo, algunos dispositivos IoT como las placas de desarrollo (Arduino, raspberry, etc.) poseen las herramientas para ser reprogramadas, en este sentido estos dispositivos no se limitan a su función, pues esta puede ser definida por el usuario. Por ejemplo, de manera anecdótica, en un laboratorio de pentesting que fue desarrollado durante mis pasantías profesionales se realizó un ejercicio para programar una placa *Arduino* como un *rubber ducky,* el cual es un dispositivo que emula el funcionamiento de un teclado USB, con la finalidad de engañar al sistema final y cargar un código malicioso o ejecutar una serie de tareas definidas; la conclusión de esa prueba de laboratorio fue exitosa, se pudo vulnerar el sistema y obtener permisos de administrador para hacer descargas en internet, y para ello, solo fue necesario conseguir una placa Arduino y hacer un par de búsquedas en internet para hallar el código necesario. El asunto es precisamente este, es relativamente sencillo trabajar con estos dispositivos IoT y existen comunidades activas enfocadas a desarrollar estos ejercicios de pentesting con ellos. Entonces, sí existe información al respecto y es conocido el uso de estos dispositivos como herramientas de ataque. ¿cuál es el problema?

El problema radica en la consciencia colectiva, el tema con esta clase de dispositivos es que no son percibidos como una vulnerabilidad y vector de ataque. Cuando se habla de internet y computadoras la gente por lo general sabe acerca de sus peligros, la descarga de virus, archivos maliciosos, robo de información, etc. La sociedad en general esta advertida de la existencia de

virus, troyanos, gusanos, entre otros. Pero en el caso de los dispositivos IoT no existe esta conciencia colectiva, a pesar de que esta tecnología continúa avanzando e incorporándose en arquitecturas modernas como es el caso del *cloud computing* y en tendencias como el *Edge computing*.

Por otra parte, debido a los recursos limitados de procesamiento que posee el IoT, no es posible ejecutar ataques que requieran poder de procesamiento (al menos de manera individual), sin embargo, si es posible utilizarlos como parte del ataque, o en su defecto en alguna de las fases de este. Por ejemplo, retomando la anécdota de mi pasantía, el rubber ducky únicamente se encargó de obtener los permisos de administrador, pero con ello fue posible hacer descargar de internet, lo cual nos podría llevar a otro ataque más enfocado, en este caso el rubber ducky se desempeñó únicamente en la fase de elevación de permisos, y fue útil para para abrir paso a otro medio de ataque.

Por último, a manera de consciencia social, la ciberseguridad es un tema que debemos tener presente como sociedad en todo momento, porque no atender a la seguridad lógica representa pérdidas. De acuerdo con un artículo publicado por la capitana de Navío de la Armada de República Dominicana e Ingeniera en Sistemas Rocío Santana, México y Colombia son los países que reciben más ciberataques; tan solo en América Latina se estiman perdidas por noventa mil millones de dólares (USD) a causa de ciberataques. Por otra parte, en los últimos años las empresas y la industria generan mayores cantidades de información en la que se incluyen datos sensibles, direcciones, correos y otros datos. (Santana, 2021). El valor monetario no es lo único que se ve afectado por los ataques cibernéticos, también se debe considerar la inversión de tiempo que se requiere para recuperar un sistema atacado. Evitar ataques y vulnerabilidades comunes permite proteger la información e integridad de los dispositivos de los usuarios, lo que se representa en el ahorro de tiempo de recuperación y restauración de equipos y sistemas afectados. Establecer una guía de las principales vulnerabilidades permite implementar estrategias de seguridad para minimizar los posibles daños.

Por ello, mediante el análisis de las vulnerabilidades en dispositivos IoT de bajo costo que ofrece el presente trabajo se plantea proteger la información y la integridad de los usuarios mediante la prevención, ya que una de las principales vulnerabilidades para la ciberseguridad es el factor humano. Un usuario informado es un usuario protegido y con ello el ahorro de dinero, tiempo, esfuerzo y protección de la línea de producción es una consecuencia de la prevención.

3.1 Justificación

"La única defensa contra el mundo es un conocimiento perfecto de él"

-John Locke

El presente proyecto parte de la idea de explorar las capacidades ofensivas y defensivas de los dispositivos IoT de bajos recursos dado el potencial que estos ofrecen en términos de versatilidad;

pues, son dispositivos de fácil acceso y transporte, con la capacidad de ser reprogramados, establecer comunicación con otros dispositivos, emular el comportamiento de otros dispositivos e incluso, expandir su funcionalidad mediante el equipamiento de piezas electrónicas y módulos compatibles, entre otras características. Sin embargo, las motivaciones del presente se encuentran en las raíces del *hacking ético;* en palabras de la profesora en ciberseguridad, hacker ético y abogada Alana Maurushat, el *hacking ético* es "el uso no violento de la tecnología en pos de un objetivo, una política u otras causas dentro de un marco legal o moralmente ambiguo" (Maurushat, 2019). No obstante, Maurushat nos advierte que existen varias definiciones y categorías para el *hacking ético*, como es caso de desobediencia civil en línea, hacktivismo, pruebas de intrusión, vulnerabilidad y penetración, técnicas de contra ataque y activismo de seguridad.

Existen varias definiciones para cada una de las categorías mencionadas anteriormente, sin embargo, la que más coincide con los intereses de la presente es la siguiente: *Pruebas de penetración, vulnerabilidad e intrusión (pentesting)*. El *National Cyber Security Centre* (Centro para la seguridad nacional, en el Reino Unido) define al *pentesting* como "un método para obtener seguridad en un sistema de tecnologías de la información (IT), mediante el intento de violar parcial o totalmente los mecanismos de seguridad de este; se incluyen herramientas y técnicas que un atacante con malas intenciones usaría." (2017). Por lo tanto, el pentesting se puede comprender como un conjunto de técnicas enfocadas al ataque intencionado (y bajo consentimiento explícito) de un sistema para evaluar su seguridad y mejorarla, sin embargo ¿de qué forma se demuestra que este conjunto de técnicas de *pentesting* tienen una finalidad positiva?

Como bien menciona The *Open Source Security Testing Methodology Manual* (OSSTMM), "… cuando el resultado de una prueba de seguridad no es verificable socava el valor de la misma, por ello, se necesita comprobar que dicha prueba ha sido correctamente realizada mediante una metodología bien establecida..." (Herzog, 2010). El uso de metodologías y *frameworks* para ejecutar correctamente un ejercicio de pentesting es necesario, tanto a un nivel ético como metodológico ya que definen las métricas, alcances y objetivos para realizar pruebas de esta naturaleza. Existen varias organizaciones encargadas de crear frameworks enfocados a mantener la seguridad en ámbitos empresariales, como es el caso de *NIST (National Institute of Standard and Technology)* que ofrece su propio framework de ciberseguridad (SA). De la misma forma existen diferentes frameworks, enfocados a diferentes áreas, por ejemplo, el caso de the *Open Web Application Security Project (*OWASP, 2022*),* el cual, cómo su nombre lo indica es un framework especializado a la seguridad de aplicaciones web.

Existen diferentes frameworks para la ejecución de pruebas de seguridad, por lo tanto, existen diferentes formas de ejecutar una prueba de seguridad o en su defecto variaciones con respecto a otros frameworks. Por ejemplo, Herzog (2010) define 7 pasos para ejecutar una prueba de seguridad apropiadamente:

1. ***Definir limitaciones:*** Refiriéndose a la formalización de los elementos que se busca proteger, de los cuales existen *Controles* encargados de su protección (los cuales se pondrán a prueba).
2. ***Zona de compromiso:*** Consiste en la identificación de los elementos, mecanismos y servicios que interactúan con los elementos a proteger.
3. ***Alcance:*** Se trata de la definición de todos los elementos que se encuentran afuera de la zona de compromiso y que son necesarios para que el elemento a proteger se mantenga

funcional. Se contemplan elementos que no influyen directamente, aquellos con los que es posible trabajar y elementos que mantienen la infraestructura funcional.

4. ***Vectores:*** Consiste en el entendimiento y definición de como interactúa el alcance dentro de sí mismo y hacia afuera. Se definen en concreto la dirección de las interacciones tanto de adentro hacia afuera como viceversa; cada interacción es un vector y estos vectores representan pruebas separadas cortas.
5. ***Canales:*** Es la clasificación de los niveles con los que puede interactuar cada vector, estos son: Humano, Físico, inalámbrico, Telecomunicaciones y redes de datos.
6. ***Tipo de prueba:*** Es el establecimiento de qué tipo de información que desea obtener de la prueba
7. ***Reglas de compromiso:*** Se trata de una guía que define de manera clara y bien delimitada todo el procedimiento a seguir durante la prueba, de tal forma que no existan dudas, falsas expectativas y errores conceptuales.

A partir de esta serie de pasos es posible ejecutar pruebas de penetración bajo una metodología apropiada y conceptualmente bien definida, capaz de ofrecer una delimitación apropiada a todos los elementos de la infraestructura en la que se ejecute la prueba. Por ejemplo, para delimitar la ***superficie de ataque***, que contempla todos los elementos y activos de cualquier tipo en los que una infraestructura es vulnerable (Muscaster, 2021), es necesario conocer otros elementos; a través de la metodología anteriormente descrita se obtiene una medida de la superficie de ataque, ya que se define un ***alcance***, que delimita el funcionamiento del objeto a proteger y se definen ***vectores***, que representan los flujos de las interacciones. Por lo tanto, siguiendo el orden de ideas establecido en los pasos del *framework* de la OSST, se puede afirmar lo siguiente: "El resultado final de la aplicación de estos pasos es la obtención de una medida de la superficie de ataque, en donde la superficie de ataque representa la parte desprotegida del alcance, partiendo desde un vector definido" (Herzog, 2010).

A través de la definición de los diferentes elementos que conformar la prueba es posible obtener certeza acerca del procedimiento a seguir, los límites y flujos de interacciones que se deben tener en cuenta. Precisamente, la premisa del presente trabajo surgió a partir de la comprensión de los pasos anteriormente explicados, específicamente, en los pasos ***4. Vectores*** y **5. *Canales***; se realizó un ejercicio mental que contempló la interacción que poseían los dispositivos IoT en una infraestructura industrial, en donde el canal en el que estos dispositivos operan es el inalámbrico. A partir de esta premisa, surgieron algunas preguntas: ¿cuál es el flujo de ataque que contemplara un atacante en este canal?, ¿qué alcance tendría unos de estos dispositivos dentro de una infraestructura?, ¿qué tan versátiles pueden ser estos vectores tomando en cuenta que estos dispositivos se pueden reprogramar?

Por ello, (y reiterando la idea planteada al principio de esta sección) las razones que me conducen a escribir esta tesis nacen a partir de las raíces del hacking ético y de la motivación de desarrollar el ejercicio mental que anteriormente fue explicado. Por lo tanto, este trabajo se redacta con la finalidad de aportar a la comunidad en ciberseguridad sobre la versatilidad que poseen los dispositivos IoT como vector de ataque, desde el punto de vista del hacking ético.

3.2 Hipótesis

De acuerdo con la información recopilada en estos capítulos es posible construir un panorama general de los elementos que conforman una red, así como el rol que desempeñan en la transmisión de datos. En este sentido, se contemplan los dispositivos IoT como parte del ecosistema de una red y, por lo tanto, se consideran como elementos que pueden representar un riesgo a la ciberseguridad. Se debe considerar que tipo de rol pueden desempeñar como vector de ataque y qué tipo de ataque es más efectivo a través de estos dispositivos.

Es entonces que se concibe la siguiente premisa a desarrollar:

Dadas las características de los dispositivos de tipo IoT de bajo costo, es posible utilizarlos en algunas de las diferentes fases de un ataque, a través de diferentes técnicas de ataque, y, por lo tanto, estas mismas técnicas pueden ser replicadas en ataques reales, en entornos tipo edge computing.

3.3 Objetivos

Como principal objetivo del presente trabajo se tiene la creación de un reporte comparativo de las capacidades ofensivas y defensivas que poseen dispositivos IoT de bajo costo, tomando como base las especificaciones técnicas que posee el módulo ESP32.

En el reporte se considera:

- La cantidad de dispositivos IoT usados comúnmente en las infraestructuras de red, sus usos y aplicaciones.
- La cantidad de software/código malicioso disponible por la red.
- Los tipos de ataques más comunes que se pueden realizar a través de estos dispositivos.
- El nivel de eficacia de los ataques realizados mediante dispositivos IoT en arquitecturas edge computing; en donde la eficacia de los ataques programados a través del módulo ESP32 será evaluada.

Por lo tanto, teniendo en cuenta estas características se pueden consideran los siguientes objetivos del presente trabajo:

Objetivo general: Analizar las capacidades ofensivas y defensivas de dispositivos IoT de bajo costo en arquitecturas tipo edge computing.

Objetivos específicos

1. Analizar el nivel de penetración que puede lograr un dispositivo IoT de bajo costo programado para realizar ataques en arquitecturas hogareñas y laborales.
2. Analizar el nivel de efectividad de las técnicas más populares de ataque disponibles para implementar con un módulo programable ESP32.

3. Analizar el nivel de resistencia que ofrecen estos módulos programables al momento de ser atacados.

La construcción de estos objetivos permite que el presente trabajo pueda ofrecer análisis de utilidad hacia la comunidad universitaria y en general, como es el caso de:

1. El análisis del nivel de penetración que puede lograr un dispositivo IoT de bajo costo en arquitecturas hogareñas, laborales y enfocadas al edge computing; basado en el nivel de efectividad que tuvo la programación del ESP32 para la ejecución de ataques.
2. Análisis del nivel de efectividad de las técnicas más populares de ataque disponibles para implementar con un módulo programable ESP32; basado en el grado de compatibilidad y efectividad que poseen con el módulo ESP32 en una configuración de bajo costo y de fácil acceso.
3. Por último, un análisis del nivel de resistencia que ofrecen estos módulos programables al momento de ser atacados; basado en el nivel de resistencia que ofrece un módulo ESP32 a los ataques, considerando que esta clase de dispositivos se pueden incorporar a diferentes proyectos de domótica y electrónica.

4. Marco Teórico

4.1 Antecedentes

Desde principios del siglo XXI ha existido una discusión respecto a cómo definir las tendencias y tecnologías emergentes que tienen el potencial de marcar la vida cotidiana y el mundo; desde el famoso Kevin Ashton que en 1999 propuso uno de los términos más usados en la actualidad: "Internet of Things" (Cueva et. al, 2015), hasta la famosa "*Industria 4.0*", acuñada el 2011, en Alemania (Tapia, 2017). Actualmente estos términos se consideran sinónimos por muchos autores ya que es casi imposible hablar de alguno de los dos sin mencionar al otro. Respecto al Internet de las Cosas, Cueva et. al, nos ofrece una de las concepciones más aceptadas para definirla, cómo la comunicación entre sensores, actuadores, smartphones, y demás dispositivos móviles a través de internet. (2015). Mientras que, para autores como Tapia, la industria 4.0 se basa en la intercomunicación de las máquinas y dispositivos en nuevos modelos de negocio y sistemas de fabricación (2017).

Lo cierto es que dichas discusiones acerca de las tecnologías emergentes se convirtieron en una realidad, y sumado a ello, hasta el día de hoy, los autores continúan poniendo sobre la mesa nuevas definiciones y paradigmas como es el caso de la Computación de niebla (*Fog computing*) & Computación de borde (*Edge computing*) que surgieron a través de la proliferación de la nube (*Cloud computing*). Ciertamente, cada uno de los conceptos anteriormente mencionados poseen sus propias características y enfoques de operación, no obstante, cada uno de ellos contemplan un factor clave en su propuesta: la incorporación de dispositivos IoT como sensores y actuadores con acceso a internet.

En el 2018 se estimaba que para el año 2020 habría entre 10 y 12 millones de dispositivos conectados gracias al IoT (Valle & Ronaldo, 2019). Actualmente, sabemos con certeza que esa estimación ha sido superada con creces debido a situación mundial originada por la pandemia COVID19 que trajo una la cantidad de nuevos dispositivos que fueron incorporados a la red gracias al confinamiento y debido a que surgieron nuevas necesidades IoT en instituciones públicas y privadas como es el caso de la instalación de módulos con oxímetros, sensores de temperatura, cámaras infrarrojas, entre otros.

Como resultado, y en suma a la tendencia que ya existía sobre dispositivos IoT, la cantidad de nuevos dispositivos conectados resulta en una superficie de ataque más extensa, en donde las amenazas se pueden entender como la presencia de factores dentro de un sistema de información o procesos que pueden resultar en vulnerabilidades que pueden ser explotadas en forma de: malware, kits de explotación, ataques dirigidos, Denegación de Servicios, ataques a la privacidad (MITM), ataques físicos y otros (Cartuche et. al, 2020).

En este sentido, es lógico hablar de la arquitectura o estructura de comunicación sobre la que operan los dispositivos IoT. Actualmente es muy difícil encontrar alguna arquitectura que no incluya IoT devices, los tenemos en nuestras salas mediante televisores inteligentes, en nuestras muñecas y bolsillos que portan algún smartwatch o smartphone, en entornos más complejos como en universidades que poseen sensores o módulos que recopilen alguna información y tengan acceso a

la red local; se encuentran presentes en hogares, oficinas e inclusive en estructuras más complejas, como es el caso de la computación de nube.

De manera general, se define como Cloud Computing (computación de nube) a la tecnología que permite hacer uso de recursos informáticos ajenos a petición desde nuestro dispositivo final, contando con servicios tan flexibles como de Infraestructura, Software, Plataforma, etc. (IBM, SAc). Siendo la computación de la nube una tecnología con gran variedad de servicios es natural que hoy en día esté presente en casi todas nuestras actividades, desde las películas que vemos en plataformas streaming, los documentos que guardamos en línea y por supuesto las topologías de las que somos usuarios. Sin embargo, a pesar de ello, es prudente cuestionarnos acerca de nuestra interacción con la famosa nube, ¿realmente estamos siempre en contacto con ella?

En los últimos años, la multinacional y famosa empresa Cisco (2022a) lanzó una propuesta respecto al cuestionamiento anterior, en donde existen pequeñas nubes que conforman niebla y que están más cerca del suelo (en vez, de las alturas, en donde se encuentra la nube). A este paradigma se le conoce como Fog Computing (computación de niebla), donde existen Fog Nodes, o pequeñas nubes que se sitúan en la frontera como extensión de la nube, y en estas se realiza el procesamiento de los datos de manera más rápida, cercana al origen y en tiempo real; de esta forma no sobresaturaría la nube o en su defecto no se dependería en demasía de sus recursos.

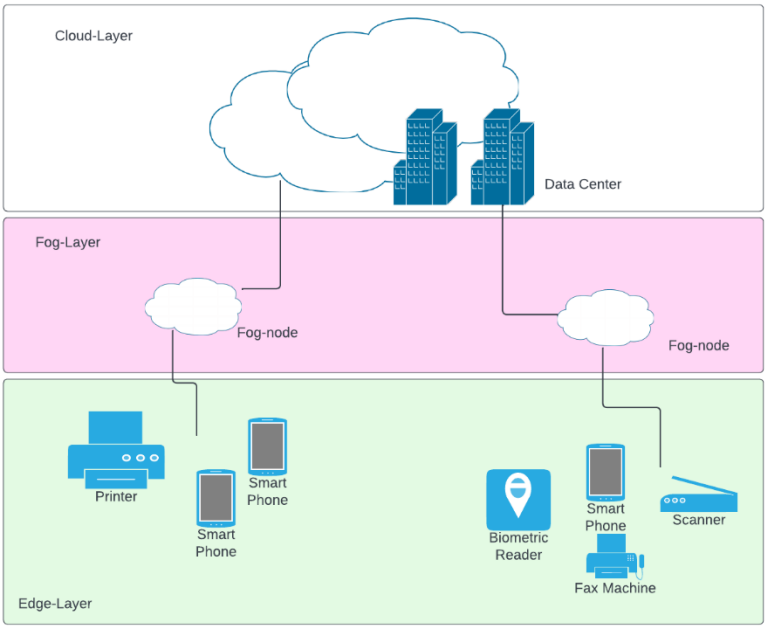

Figura 6. Capas que componen la arquitectura de la Fog Computing (Fuente: figura de elaboración propia)

Se observa en la Figura 6 que la arquitectura Fog computing, se divide en tres capas, en las cuales se enfoca el nivel en el que se procesan los datos. Una descripción acertada para estas capas proviene de Hurtado (2021):

- **Edge layer:** la cual comprende los dispositivos de borde, incluyendo dispositivos IoT; los datos que se procesan en este nivel se envían a la capa de nodos de niebla.
- **Fog layer:** se compone por una serie de servidores locales o en su defecto no tan remotos, en estos se procesa la información que posteriormente se enviaría a la capa de nube.
- **Capa de nube**: el punto más alto, representa a los servidores remotos y la gran infraestructura que ofrece la nube como plataforma

Observando esta composición vale la pena enfocarse en la capa del borde (edge layer), ya que es la capa más propensa a los ataques, y precisamente, relacionada a este concepto del borde, existe otra arquitectura que se enfoca al procesamiento de la información que ocurre con estos dispositivos en un entorno más local. De acuerdo con Hurtado, la edge computing (o cómputo de borde), consiste en un enfoque más perimetral, en la que los datos se procesan en la periferia de la red, mucho más cerca de la fuente de estos (2021); esta computación perimetral se ve impulsada por dispositivos móviles y una gran cantidad de dispositivos IoT.

Es decir, mientras la fog computing se enfoca en cómo se procesan los datos antes de alcanzar la nube, por otra parte, la edge computing se especializa en cómo se administra, almacena y procesa la información de estos dispositivos IoT. Además de ello, el cómputo de borde se mantiene en un nivel cercano a los dispositivos y el usuario; en palabras del consorcio de la OpenFog, a diferencia de la estructura jerárquica de la computación de niebla que abarca el almacenamiento, procesamiento y acceso de los dispositivos hacia la nube, la computación de borde tiende a limitar sus operaciones al borde (Ashkan et. al, 2019).

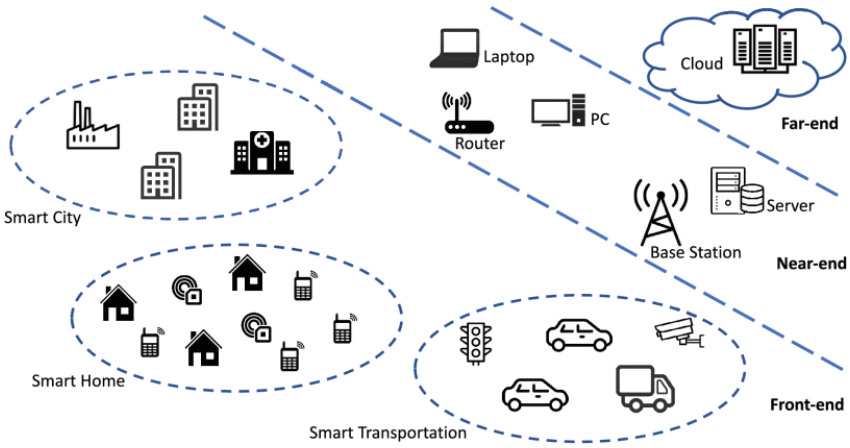

Figura 7. Estructura general de una arquitectura de cómputo de borde (Fuente: Kewei Sha, T. Andrew Yang, Wei Wei & Sadegh, 2017).

En la figura 7, se observan tres niveles principales en las arquitecturas edge computing, las cuales Kanai & Niwa (1999) describen apropiadamente:

- **Front-End:** Donde se encuentran dispositivos finales como sensores, actuadores y proveen mucha interacción entre sí mismos y con los usuarios, se caracteriza especialmente por ofrecer servicios en tiempo real, sin embargo, debido al poder de cómputo (low cost) que tienen estos dispositivos en muchas ocasiones recurren a infraestructura y servicios de servidores cercanos.
- **Near-End**: En esta capa encontramos los gateways que redirigen el tráfico de la capa anterior, este tráfico incluye el procesamiento de los datos, dato cache y descargas de cómputo.
- **Far-End:** A pesar de estar lejos, se encargan de proporcionar servicios, almacenamiento y poder de cómputo, haciendo posibles tareas que no son posibles para las dos capas anteriores, como es el caso de big data, procesamiento masivo de datos en paralelo, minería de datos y machine learning.

Al estar orientada al procesamiento de datos en tiempo real, uno de los principales problemas con esta arquitectura es la seguridad. Principalmente a que la capa Front-end, en donde coexisten múltiples dispositivos con diferente tecnologías y protocolos como peer to peer, inalámbrico, etc. Otro factor es el nivel de privacidad, de acuerdo con el artículo publicado por la IEEE: A Survey on the Edge Computing for the Internet of Things, "…la edge computing procesa datos en el perímetro, y dicha información sensible y asociada con los usuarios finales podría explotarse, debido a que la información proveniente de nodos IoT se almacenan en los nodos perimetrales, que pueden ser más vulnerables que los servidores en la nube." (Kanai & Niva, 1999).

En este contexto, la computación de borde cumple con las condiciones para ser un caso de estudio para este trabajo, ya que, al ser un paradigma en tendencia, se asemeja a muchas de las topologías con las que coexistimos, casi como un símil de un ambiente de red local. Por ello, el objetivo del presente trabajo es estudiar y evaluar a detalle el nivel de penetración que un dispositivo IoT de bajo costo es capaz de lograr, considerando lo sencillo que resulta para un usuario llegar a una red y colocar (o atacar) un dispositivo IoT y aprovecharse de las vulnerabilidades existentes.

4.2 Estado del arte y literatura al respecto

4.2.1 Sobre la ciberseguridad y el IoT

Es bien sabido que existe bastante literatura relacionada a la seguridad en comunicaciones informáticas, sin embargo, en términos de investigación realizada en arquitecturas enfocadas al IoT es reciente, como lo vemos en el artículo del 2020 por parte del *Department of Computing Science* de la UHCL, se discuten muchas de las brechas reportadas en años recientes, cómo es el caso de cámaras inteligentes usadas como botnets en ataques DDoS (Kewei Sha et. al, 2020). A pesar de la existencia de soluciones para este tipo de ataques (que son aplicadas en la mayoría de las infraestructuras industriales), estas medidas no son del todo funcionales en la mayoría de los dispositivos IoT debido a sus recursos limitados.

Como bien lo menciona Kewei Sha et. al, algunas investigaciones se han concentrado en desarrollar soluciones basadas en edge IoT security, como es el caso de firewalls, sistemas de detección de intrusos, protocolos de autentificación, etc. Sin embargo, en general, la investigación de seguridad

basada en arquitecturas edge aún se encuentra en desarrollo (2020).

De manera general, el interés por la ciberseguridad en dispositivos IoT es reciente, gracias a ataques de alto impacto. A partir del famoso ataque DDoS hacia la compañía Dyn (quien brinda servicios DNS) el 21 de octubre de 2016 el mundo en general experimento el daño que puede causar una inmensa botnet compuesta por IoT devices como Cámaras IP, impresoras, entre otros. El resultado de este ataque fue 1.2 terabits de peticiones por segundo, daños por más de 110 millones de dólares y servicios inhabilitados en California, el Medio Oeste y la Costa Oeste de Estados Unidos (Kochetkova, 2016).

A partir de este suceso se establecieron muchos principios y directivas para asegurar a los dispositivos IoT, como sugerencias hechas por el Departamento de Seguridad Nacional de los Estados Unidos, el *Industrial Internet Framework* propuesto por el consorcio Industria de Internet, el marco de seguridad de Sigma Designs S2 y otras iniciativas. Sin embargo, implementar políticas en dispositivos IoT es complicado debido a que es un conjunto de dispositivos muy variados, que operan con diferentes prestaciones técnicas (diferente procesamiento, memoria, sistema operativo, etc.) sumado a ello, el desconocimiento de los usuarios en general. Por ello, destaca una lista de vulnerabilidades recopiladas por la fundación OWASP: https://owasp.org/, con la finalidad de generar consciencia hacia a los usuarios (Molina, 2019):

A continuación, se resumen dichas vulnerabilidades:

1) Uso de contraseñas débiles o embebidas: la mayoría de las contraseñas débiles o por defecto son fácilmente rotas mediante ataques de fuerza o sencillamente ya se encuentran en internet publicadas.

2) Servicios de red inseguros: Entre más, es menos seguridad; evitar servicios de red innecesarios o inseguros es un factor clave a la hora de explotar vulnerabilidades ya que es posible que se estén ejecutando procesos en segundo plano con vulnerabilidades incluidas.

3) Interfaces inseguras en el ecosistema IOT: Generalmente las plataformas de administración de estos dispositivos como web interfaces o APIS en backend no están configuradas adecuadamente, ya que no poseen medidas de control ni mecanismos de encriptación

4) Falta de mecanismos de actualización seguros: Existe una carencia de mecanismos de validación para las versiones de firmware en estos dispositivos, sumado a ello, la instalación de versiones modificadas de firmware es común.

5) Uso de componentes inseguros o desactualizados: En muchas ocasiones el uso de tecnología obsoleta es un factor de explotación, tanto a nivel de software como en hardware. Es bastante común el uso de librerías de terceros y componentes mixtos en dispositivos IoT.

6) Insuficiente protección de la privacidad: El procesamiento de datos sensibles (y en general) en los dispositivos IoT no es competente e inclusive sin consentimiento, se recomienda implementar políticas para el acceso a la información necesaria únicamente.

7) Falta de seguridad en el almacenamiento y transferencia de datos: Existe un mínimo o en general nulo manejo de algoritmos de cifrado para proteger los datos almacenados, especialmente en ecosistemas de domótica.

8) Inadecuada gestión de dispositivos: Es común que no exista (o sea pobre) el control de los dispositivos en producción, incluyendo la monitorización de sistemas, políticas de privacidad, actualización, etc.

9) Configuración por defecto inseguras: En muchas ocasiones se conservan las configuraciones de fábricas, las cuales no son eficientes en términos de seguridad.

10) Falta de bastionado físico: La falta del control de acceso a los dispositivos físicos representa un problema serio, ya que es un muy fácil que un atacante intercepte un dispositivo y lo altere (o lo suplante).

Así mismo, por parte de la iniciativa privada existen diferentes fabricantes y proveedores de soluciones IoT que adoptaron diferentes estrategias de seguridad en sus productos, como es el caso de Hikvision, la cual, es una empresa fundada en el 2001 y se enfoca principalmente al mercado de la video vigilancia. (SYSCOM, SA). En este tenor, Hikvision público un "*white paper*" el cual detalla la postura que tiene sobre el tema de la ciberseguridad. En este documento, la empresa nos menciona las diferentes amenazas que existen en el campo de IoT, estas amenazas las agrupa en tres capas fundamentales: percepción, red y aplicación, las cuales agrupan de manera global los incidentes más comunes relacionados a la seguridad en dispositivos IoT (2018):

Amenazas de la capa de percepción

- **Robo del equipo o daño:** Se trata de un incidente bastante común debido a que generalmente los dispositivos IoT son colocados en posiciones remotas y sin ninguna protección física, los cuales los hace susceptibles a ser robados o dañados por algún agente externo.
- **Sobrecalentamiento o sobre enfriamiento de los dispositivos:** Como se mencionó en el punto anterior, la exposición a la que a veces son sometidos estos dispositivos sobre pasa sus características técnicas, al ser expuestos a temperaturas no adecuadas para su funcionamiento es posible que estos dispositivos se dañen, logrando brechas de seguridad en el sistema.
- **Ataques basados en vulnerabilidades conocidas:** Consiste en aquellas brechas de seguridad logradas a partir del uso de sistemas operativos desactualizados o versiones de firmware con vulnerabilidad no corregidas.
- **Ataques y mecanismos de bypass para burlar autenticación:** Consiste en aquellas malas prácticas relacionadas al uso de contraseñas débiles, las cuales en la mayoría de los casos permiten el acceso al sistema de manera sencilla.
- **Robo de información sensible:** Existen dispositivos que contienen información sensible dentro de su sistema, esta información en la mayoría de las veces de fácil acceso y representa un riesgo.
- **Ataques de dispositivos de control remoto:** En la mayoría de los dispositivos existen puertos asignados para debugging y testeo, es decir accesos al dispositivo libres de

restricciones para que el usuario ejecute pruebas o recupere la administración del dispositivo. Esta característica puede ser explotada para tomar acceso del equipo con fines maliciosos.

- **Robo de información privada:** Se refiere al robo de la información que pueda lograrse durante los procesos de comunicación que existen entre los dispositivos IoT, en muchas ocasiones se trata de una fuga de información debido a la falta de protocolos seguros para la transferencia.

Amenazas de la capa de red

- **Penetración de la red a través del medio inalámbrico:** Se trata del acceso indebido a comunicaciones e información privada debido al uso de protocolos inalámbricos no seguros, así como a la falta de mecanismos de autenticación durante la comunicación.

- **Ataques a comunicaciones de red no encriptadas:** En suma, al punto anterior, la falta de mecanismos de cifrado durante una comunicación, permiten la información pueda ser extraída de manera íntegra en caso de que estas sean interceptadas.

- **Ataques e intrusiones desde el internet:** Se refiere los ataques provenientes de sistemas que trabajan con el protocolo IP, así como ataques provenientes de internet.

- **Ataques de Denegación de servicios:** Consiste en ataques con el objetivo de denegar un servicio, existen muchas formas de ejecutar estos ataques, incluyendo virus informáticos.

Amenazas de la capa de aplicación:

- **Dificultades en la gestión de los dispositivos:** Vulnerabilidades y complicaciones derivadas de una mala administración en las actualizaciones de dispositivos IoT.

- **Riesgos a la privacidad y seguridad:** Riesgos a la seguridad causados por accesos no autorizados al sistema.

- **Riesgos causados por falta de actualizaciones:** En suma, al primer punto de esta sección, en ocasiones los dispositivos IoT son desplegados sin ningún plan de administración, mantenimiento o gestión. Lo cual los deja vulnerables a nuevos ataques y susceptibles a quedar obsoletos en términos de software.

De manera general, como se ha observado a lo largo de esta sección, la seguridad de los dispositivos IoT corresponde en gran medida al uso de buenas prácticas por parte del usuario. Una correcta administración, el cumplimiento de medidas de seguridad y el correcto posicionamiento de estos dispositivos en un ambiente seguro son factores determinantes al momento de determinar si un dispositivo IoT se puede convertir en un vector de ataque. No obstante, tecnológicamente hablando, la industria ha avanzado considerablemente, los protocolos han adquirido nuevas versiones que apelan a la seguridad de transmisión de datos y muchos otros han sido depreciados. Un claro ejemplo de esto es el protocolo 802.11 (protocolo encargado de la tecnología WiFi) que fue liberado en el año 1997 y estaba adecuado a la tecnología de la época, ha sufrido varios cambios.

Actualmente la versión más reciente de este protocolo es el 802.11ax (Wi-Fi 6) y es un protocolo totalmente diferente al de hace unas décadas, con la capacidad de soportar la tecnología 5G (Intel, 2021a).

Es entonces que los protocolos cobran un factor decisivo en términos de ciberseguridad, ya que dependiendo del protocolo que use un dispositivo IoT para comunicarse con su entorno, es el nivel de seguridad que este posee durante la ejecución de sus comunicaciones. De manera proporcional también influye el tipo de protocolos que son usados en la red, por ejemplo, si en dispositivo IoT que fue desplegado se conecta a un AP con protocolos descontinuados, claramente el canal de comunicación que estos usaran para transmitir información será vulnerable a muchos tipos de ataques.

Precisamente, conocer que protocolos de comunicación pueden utilizar estos dispositivos otorga un punto de partida para obtener el panorama en el cual se desenvuelven estos dispositivos en términos de ciberseguridad. De acuerdo con la guía elaborada por Azure, "Protocolos y tecnologías de IoT" existen varios protocolos designados para la comunicación entre estos dispositivos, algunos enfocados al desarrollo de comunicaciones en redes amplias, mientras que otros están destinados a comunicaciones de corto alcance. Sin embargo, un enfoque más apropiado para abordar estos protocolos es de acuerdo al modelo TCP/IP. A continuación, se presentan algunos de estos protocolos recopilados por Azure (2022):

Capa de aplicación
En esta capa se concentran los protocolos que se encargan de coordinar la interfaz entre el usuario y el dispositivo.

- **AMQP (Advanced Message Queuing Protocol):** Tiene la función de ayudar a la interoperabilidad entre aplicaciones y el middleware del dispositivo, permitiendo ejecutar funciones de mensajería.
- **Protocolo de aplicación restringida (CoAP):** Se trata de un protocolo especializado para funcionar en dispositivos de recursos limitados. Su función se centra en la comunicación entre maquinas. A su vez, es un protocolo de transferencia de documentos que funciona bajo el protocolo UDP.
- **Protocolo de Servicio de distribución de datos (DDS):** Es un protocolo enfocado a la comunicación punto a punto. Posee varias funciones relacionadas a la conexión entre dispositivos y redes.
- **Protocolo Message Queue Telemetry Transport (MQTT):** Se trata de un protocolo enfocado a la mensajería entre equipos con pocos recursos con conexiones de bajo ancho de banda. Una característica destacada de este protocolo es que es útil cuando la comunicación entre estos dispositivos es remota.

Capa de transporte
En esta capa se concentran los protocolos enfocados al transporte de los datos entre las diferentes capas del modelo TCP/IP.

- **Protocolo de control de transmisión (TCP):** Protocolo orientado a la conexión. Se encarga de la comunicación entre hosts y posee la capacidad de dividir los datos en paquetes individuales para posteriormente reensamblarlos.
- **Protocolo de datagramas de usuario (UDP):** Protocolo no orientado a la conexión. Su principal característica es que altamente eficiente en la velocidad de transferencia de datos.

Capa de red
Corresponde a aquellos protocolos encargados de la comunicación entre dispositivos finales y el enrutador.

- **Protocolo de Internet (IP):** Protocolo encargado de proveer IPv4 a los dispositivos IoT.
- **6LoWPAN:** Funciona de manera similar al protocolo IP, sin embargo, este protocolo se caracteriza por ser implementado en dispositivos con baja capacidad de procesamiento y potencia.

Capa de acceso a la red
En esta capa se concentran aquellos protocolos encargados de la transferencia de datos dentro de la arquitectura del dispositivo, así como de la correcta transmisión y recepción de los datos en el medio físico.

- **IEEE 802.15.4:** Protocolo enfocado a la conexión inalámbrica de bajo consumo.
- **LPWAN:** Protocolo especializado para redes de área extensa en dispositivos de baja potencia. Soporta comunicaciones de 500 metros de distancia y bajo ciertas condiciones de hasta 10 km.
- **Bluetooth Low Energy (BLE):** Se trata de un protocolo basado en el bluetooth original, sin embargo, se especializa en impactar mínimamente el consumo de energía y costo (en cuestión de recursos) de operación.
- **Wi-fi 802.11:** Se trata del protocolo clásico para la comunicación Wifi.
- **Z-Wave:** Protocolo especializado en redes de tipo malla. Su principal característica es que implementa ondas de radio de baja potencia para la comunicación entre dispositivos.
- **Zigbee (IEE 802.15.4):** Protocolo para la comunicación inalámbrica enfocado a pequeñas redes de área personal de baja potencia.

4.2.2 Hacking ético y ESP32

De manera específica, en la parte ética ofensiva existen autores como Richard Stehlík (2021) que documentan técnicas de ataque más comunes a realizar con dispositivos IoT. En su artículo: "Wi-Fi attacks using ESP32", Stehlík habla sobre ataques de autenticación WPA/WPA, ataques de fuerza bruta WPS PIN, ataques KRACK, captura PMKID, Pentesting, etc. La propuesta se realiza mediante el módulo lógico programable ESP32, el cual cuenta con tecnología bluetooth y Wi-Fi, haciéndolo bastante versátil para proyectos de domótica y electrónica.

Otras propuestas académicas incluyen proyectos interuniversitarios como el artículo "*Building a Low-cost and State-of-the-art IoT Security Hands-on Laboratory"* desarrollado entre la Universidad de Florida Central (Orlando), la Universidad de Florida (Gainesville) y la Universidad

de Lowell Massachusetts (Lowell). El artículo desarrolla un laboratorio IoT enfocado a la ciberseguridad y entre sus elementos se destaca el uso del dispositivo *ESP32*. Dicho laboratorio tiene como finalidad enseñar las propiedades de dicho dispositivo y emplear conceptos teóricos a la práctica ética del hacking, por ejemplo, entre los módulos de desarrollo se encuentran: el monitoreo de la red vía métodos de monitoreo aéreo, seguridad a nivel de hardware, almacenamiento de claves, módulos de cifrado, entre otros (Pearson et. al, 2020).

Existe una comunidad creciente y activa enfocada al desarrollo de proyectos en dispositivos IoT, solo basta con hacer una búsqueda rápida en YouTube y los resultados de búsqueda arrojaran múltiples títulos y autores relacionados a proyectos de domótica, electrónica, robótica, etc. Dentro de esta comunidad el uso de módulos programables es esencial, pues permiten un óptimo ejercicio ya que poseen las herramientas necesarias y la versatilidad para desarrollar todo tipo de proyectos. Existen los clásicos módulos programables como Arduino, Raspberry pi, ESP 8266 y el ya mencionado ESP32. Estos últimos destacan debido a que, a diferencia de las clásicas tarjetas como Arduino, cuentan con módulos ya integrados para la comunicación inalámbrica, y, en comparación con otras tarjetas como la Rasberry sus precios son bastante económicos. De la misma forma, existe una amplia variedad de librerías de código libre y especializado para explotar cada una de las capacidades de estos dispositivos, convirtiéndolos en herramientas capaces de adaptarse a casi cualquier tarea.

Precisamente dentro de esta comunidad la variedad de librerías, tutoriales, documentos dispersos en videos, foros, mercados y grupos hace posible el acceso código y diagramas que permiten el libre desarrollo de proyectos de todo tipo. Como es el caso del YouTuber Hak5 (2021), que en su video “HackByte: Build a Hackcble Router with a $5 ESP32”, muestra a manera de tutorial educativo como transformar esta tarjeta programable en un sniffer de tráfico dentro de una red Wi-Fi.

Volviendo al tema central, existe la documentación necesaria para que un usuario malicioso pueda desarrollar proyectos de ataque mediante módulos programables que son de fácil acceso, los hay desde la módica cantidad de $185 MXN, hasta herramientas especializadas cargadas con software y hardware especial con un costo de $60 USD (Halfacree, 2020). Ya sea que se compre o se desarrolle, existen los medios necesarios para crear proyectos de naturaleza maliciosa o ética. Los alcances que pueden tener estos proyectos son de naturaleza flexible, por ejemplo, el ataque descubierto en 2019, *fatal fury* es una técnica que se enfoca en una vulnerabilidad de tipo *Boot Flow* (flujo de arranque), es decir explota un error durante el ciclo de arranque del ESP32, haciendo posible engañar al sistema para lograr omitir algunos mecanismos de verificación o seguridad del sistema; esto nos podría permitir suplantar el firmware del dispositivo y obtener acceso no autorizado a un sistema, entre otras cosas (Harsha, 2019).

Otro ejemplo recae sobre el ataque *Forever-hack*, el cual permite lograr un *bypass* sobre el dispositivo y ejecutar una inyección de malware. Gracias a esto, el atacante obtiene la capacidad de encriptar y descifrar el firmware del dispositivo, lo cual le otorga el acceso de crear un nuevo firmware, así como extraer las llaves de seguridad del dispositivo (secure boot keys) (SA, 2019). Una de las cosas más interesantes sobre este ataque, es que es de naturaleza irreversible, de ahí su nombre. Un aspecto notorio es que ambos ejemplos anteriormente expuestos fueron descubiertos por *LimitedResults*, un miembro de la comunidad especializado a la ciberseguridad en IoT, el cual puede ser localizado en su red social de twitter a través del usuario @LimitedResults.

4.2.2 Delimitaciones

Existen múltiples formas y métodos para ejecutar ataques, como se desarrolló en el anterior apartado, como son el caso de ataques que funcionan directamente como el *boot flow* del dispositivo, así como otros que requieren una conexión física hacia la víctima. En este sentido, abordar toda clase de ataques extendería el presente trabajo (aunque bien puede desarrollarse otra clase de trabajo partiendo del análisis de este tipo de ataques), por lo tanto, durante las siguientes páginas, la intención es mostrar algunos de los ataques más populares que pueden ser ejecutados a través del medio inalámbrico, es decir, aquellos ataques que son posibles propagar y accionar a mediante tecnología inalámbrica.

4.3 Métodos y técnicas de ataque

La frase *"la mejor defensa es el ataque"* probablemente ha sido sacada de contexto la mayoría de las veces, o en su defecto, malinterpretada, sin embargo, para el caso de la ciberseguridad cobra cierto sentido, hablando del entendimiento sobre la gestión de la seguridad. Un especialista en ciberseguridad debe estar consciente de los ataques y técnicas que un atacante puede ser capaz de ejecutar, entender los medios y mecanismos a través de los cuales se pueden realizar estos ataques, permite identificar qué elementos se deben atender para salvaguardar la seguridad. Por ello, en esta sección se detallanlos métodos y técnicas de ataque más comunes.

4.3.1 Ataques tipo spoofing y de hombre en el medio

De acuerdo con The Computing Technology Industry Association, las técnicas de tipo spoofing son aquellas que pretender imitar la identidad de algún sistema o persona. Es decir, se trata de un método que consiste en engañar al sistema o usuario a través de un engaño para aparentar o suplantar a alguna identidad conocida o permitida (CompTIA, SAa). Existen varios tipos de spoofing, por ejemplo, el email-spoofing que va dirigido especialmente a engañar a un usuario, el gps spoofing que consiste en engañar a un sistema mediante una geolocalización falsa (un ejemplo de ello sucede en el famoso juego Pokémon Go, el cual basa su jugabilidad en la geolocalización del usuario), entre otros.

Una de las principales ventajas de los ataques de tipo spoofing es que dificultan la identificación y rastreo del atacante, por ello, no es extrañar que se trate de ataques bastante populares comunes, sin embargo, los ataques de tipo spoofing que abordaremos en esta sección corresponden únicamente a aquellos que podrían ser implementados en un dispositivo IoT. Precisamente debido a estas características este tipo de ataques se convierten en objetos de estudio para investigadores como Vaidya, et. al (2016) que hace un análisis sobre ellos y recupera a los más comunes:

ARP Spoofing

El protocolo ARP es uno de los más importante durante la comunicación a nivel LAN, ya que en un nivel local la comunicación ocurre gracias a las direcciones físicas (MAC) de los dispositivos,

por ello durante esta sección es necesario retomar un poco de su teoría de funcionamiento. El *Address Resolution Protocol* o ARP puede ser visto como un mecanismo de traducción entre direcciones MAC y direcciones IP. El cual consiste en dos tipos de mensajes:

- **Petición ("who-has"):** Especifica la dirección IP del host del cual queremos describir su dirección MAC. Este mensaje es enviado como un *broadcast*.

- **Respuesta ("is-at"):** Consiste en la repuesta del host, especificando la dirección MAC asociada al IP. Este mensaje es enviado como una respuesta de tipo *unicast*.

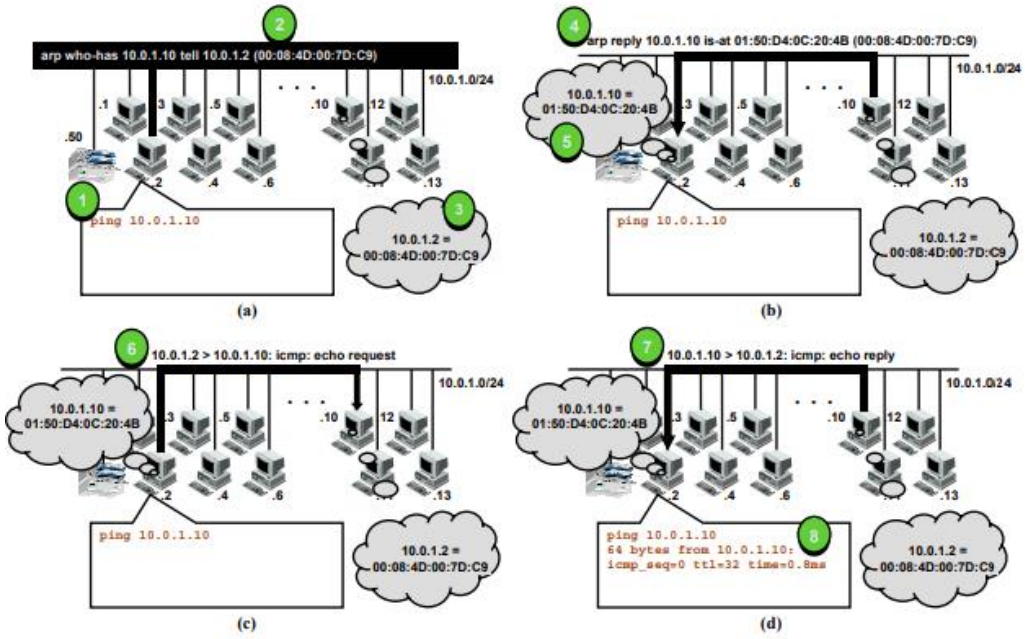

Figura 8. *Representación gráfica del funcionamiento del protocolo ARP (Fuente: Carnut & Gondim, 2003).*

Como se observa en la figura 8, el protocolo ARP se puede describir en los siguientes pasos:

1. El host 10.0.1.2 desea hacer ping al host 10.0.1.10 pero este no se encuentra en su tabla ARP.
2. Entonces host (10.0.1.2 hace una petición de tipo **"who-has**" a todos los hosts en la red, es decir envía un mensaje de tipo broadcast.
3. Todos los dispositivos en la red reciben el mensaje y asumen que la tabla ARP está vacía, sin embargo, los hosts que no posean la dirección 10.0.1.10 no responderán a la petición.
4. El dispositivo que tiene la dirección 10.0.1.10 responde a la petición con una respuesta **"is-at"**
5. La tabla ARP del host 10.0.1.2 asocia la dirección MAC del host 10.0.1.10

6. Ahora, con este conocimiento, el host 10.0.1.2 procede a lanzar el protocolo ICMP para realizar el ping y lograr comunicación con el 10.0.1.10
7. El host 10.0.1.10 responde de manera efectiva a esta comunicación
8. Ocurre con éxito la comunicación.

Utilizar una explicación detallada del protocolo ARP nos ofrece varias ventajas, la principal es que permite entender a detalle el funcionamiento del protocolo y, por lo tanto, nos provee de la capacidad de identificar anomalías en el proceso del llenado de la tabla ARP. Otra ventaja es que ahora que conocemos perfectamente en que consiste el protocolo podemos entender de qué forma se efectúa este tipo ataque:

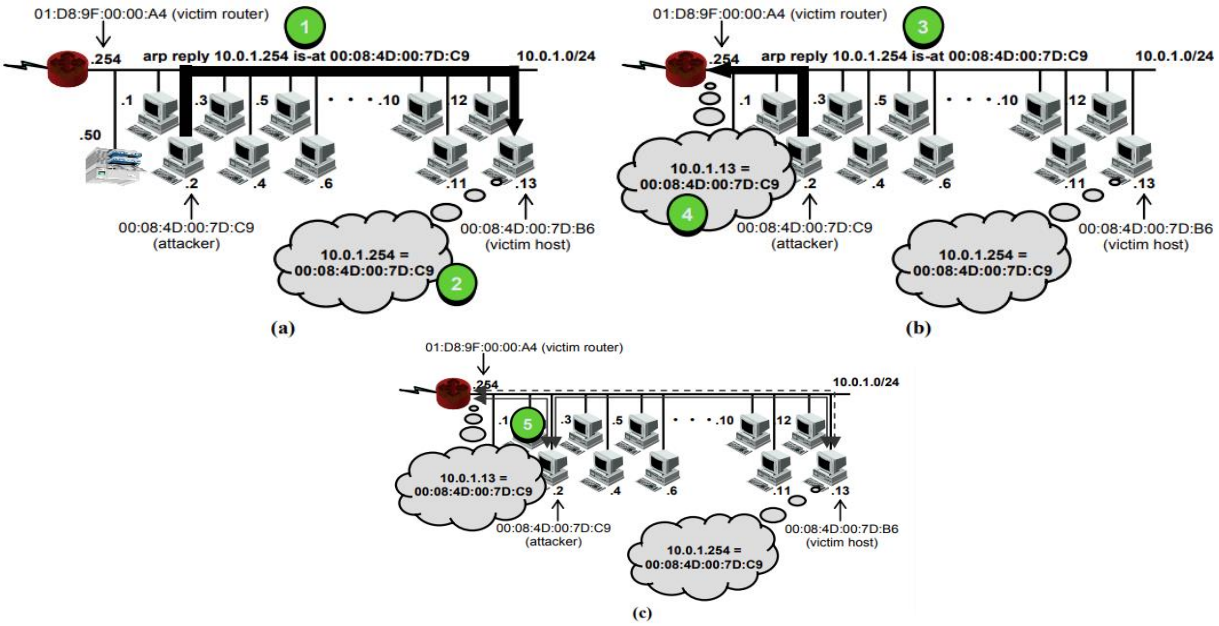

Figura 9. Representación gráfica del proceso que realiza el ataque ARP Spoofing (Figura: Carnut & Gondim, 2003).

En la figura 9, podemos identificar cinco momentos clave durante este ataque, y, retomando el ejemplo anterior, en esta ocasión el atacante será el host que posee la dirección 10.0.1.2

1. El atacante (10.0.1.2) envía un mensaje de tipo ***Respuesta (is-at)*** hacia la victima 10.0.1.13
2. A través del mensaje, la victima (10.0.1.13) actualiza su tabla ARP con los datos propuestos por el atacante: 10.0.1.254 = 00.08:4D:00:7D:C9. Es decir, el atacante envió una Respuesta asignando su dirección Física a la dirección del gateway de la red.
3. El atacante repite el proceso de ataque, pero ahora dirigido hacia el gateway de la red, el cual posee la dirección 10.0.1.254
4. El gateway victima actualiza tu tabla ARP con los datos propuestos por el atacante: 10.0.1.13 =00.08:4D:00:7D:C9. Es decir, el atacante envió una Respuesta asignando la

dirección de la víctima del paso 2 a su dirección física.

5. Se efectúa un ataque de tipo man in the middle debido a que gracias a estas asignaciones el Gateway de la red (10.0.1.254) piensa que estas enviando datos al 10.0.1.13 y viceversa.

Este tipo de ataques son efectivos debido a que el protocolo ARP se denomina *stateless,* es decir no verifica las peticione y respuestas, y, por lo tanto, el protocolo acepta respuestas aun si no se han hecho solicitudes.

DHCP Spoofing

Otro ataque enfocado a la suplantación de identidad (en el marco de la comunicación entre dispositivos) es el envenenamiento de DHCP. The Dynamic Host Configuration Protocol se encarga de brindar parámetros de configuración a los hosts, de manera específica se trata de un protocolo construido sobre un modelo de cliente servidor que realiza dos acciones principalmente: la primera consiste en la entrega parámetros configuraciones para los hosts, y la segunda, asignar estos parámetros a dichos hosts. Para ejemplificarlo, se muestra la figura 10:

.

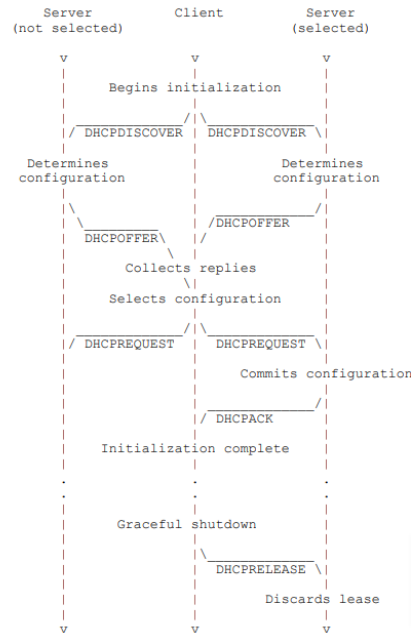

Figura 10. Representación gráfica del funcionamiento del protocolo DHCP (Figura: R. Droms, 1993).

Como se denota en la figura 10, el protocolo DHCP consiste de una comunicación entre un cliente y un servidor, la cual consta de los siguientes pasos:

1. El cliente realiza un broadcast con el mensaje DHCPDISCOVER en la subred donde se encuentra.

2. El servidor (o los servidores) puede responder con un mensaje DHCPOFFER que contiene una dirección de red disponible.

3. El cliente recibe un mensaje DHCPOFFER, el cual contiene un identificador del dhcp server, mediante esta información el cliente envía un mensaje broadcast tipo DHCPREQUEST.

4. Los servidores DHCP reciben el DHCPREQUEST, de esta forma los demás servidores reciben la notificación respecto a la decisión del cliente. Entonces, el servidor seleccionado publica la configuración del cliente y envía un mensaje DHPCACK que contienen los parámetros de configuración para el cliente.

5. Una vez concluido el servicio el cliente envía un mensaje DHCPRELEASE hacia el servidor, con el objetivo de descartar el préstamo de la configuración dado al cliente

Por otra parte, el ataque DHCP Spoofing es una técnica que consiste en redirigir el tráfico de una red mediante el envenenamiento del protocolo DHCP. Este Spoofing o envenenamiento se realiza a través de un servidor DHCP falso en la red. Es decir, el ataque consiste en posicionar un servidor DHCP falso en la red de la víctima, el cual se encargará de entregar configuraciones maliciosas a los usuarios. Gracias a estas configuraciones maliciosas el atacante puede obtener credenciales, comunicaciones y datos que sean enviados durante las comunicaciones de la red, especialmente aquellas que se ejecuten a través de protocolos que no posean servicios de cifrado.

Rogue AP

Para comprender la naturaleza de este ataque es necesario entender el funcionamiento de un AP. Un Access Point (AP), es un dispositivo que permite a otros dispositivos conectarse a una red, esta conexión ocurre gracias a que este dispositivo propaga una red inalámbrica, y entonces, los dispositivos con capacidad inalámbrica (aquellos una tarjeta de red inalámbrica o WiFi) pueden conectarse a ella y acceder a la red (Cisco, 2022b).

En resumen, un AP se trata de los dispositivos más comunes y que desempeñan una de las tareas más frecuentes en la actualidad: conectarse a una red WiFi. Generalmente accedemos a esta red a través de nuestros teléfonos, laptops y demás dispositivos con WiFi. Generalmente estos dispositivos son de uso comercial e industrial, por ejemplo, en oficinas, escuelas y fábricas; si bien un “modem” (o router inalámbrico) tiene la capacidad de propagar una red inalámbrica no se trata de su especialización, al contrario de un (AP) que se enfoca totalmente en la propagación de una red inalámbrica en entornos exigentes, donde se requiere proveer de servicio a una gran cantidad de usuarios.

Por otra parte, un Rogue AP se trata de un AP que ha sido colocado de manera maliciosa y no autorizado dentro de la red. Esta colocación puede hacerse de dos formas principalmente (García, G., 2015):

- **De manera interna,** refiriéndose a la colocación de un Rogue AP dentro de la red y teniendo una administración por alguien no autorizado.
- **De manera externa,** refiriéndose a la colocación de un Rogue AP que no está conectado a la red y que tiene como objetivo hacerse pasar como un AP autorizado.

Un ejemplo de aplicación de estas tácticas puede ser el escenario en donde un usuario se encuentre en un restaurante o una plaza pública y desee conectarse a una red Wifi; cuando el usuario busque alguna red disponible en su teléfono o laptop, seguramente aparecerá alguna red disponible con la leyenda *"WIFI GRATIS ZOCALO"* o *"WIFI GRATIS PARA CLIENTES*", entonces, el usuario accederá a esta red y obtendrá el acceso fácilmente, sin saber que en realidad puede estar conectado a un Rogue AP que tiene como propósito robar u obtener sus datos.

Sniffing

Generalmente las herramientas llamadas "*sniffers*" son utilizadas con el consentimiento del usuario y con la finalidad de atender a alguna necesidad de los usuarios o administradores de la red, sin embargo, en otros tenores, también es posible utilizar esta herramienta con fines maliciosos. Pero, ¿Qué es un Sniffer? Como lo menciona el fabricante en soluciones de ciberseguridad AVAST, un sniffer consiste en una herramienta de software o hardware con la capacidad de capturar todo el tráfico de la red en tiempo real y analizarlo (Belcic, I., 2022).

En otras palabras, un sniffer puede ser utilizado como una herramienta de espionaje, ya que posee la capacidad de interceptar comunicaciones y visualizar su contenido. Por supuesto, actualmente gracias al desarrollo de protocolos de seguridad a nivel de red como HTTPS (Protocolo seguro para transferencia de hipertexto) existe cierto grado de privacidad entre las comunicaciones que ocurren en la red, así como otros protocolos encargados del cifrado de las comunicaciones. Por lo tanto, el uso de sniffers se ve relegado a aquellos usuarios con el suficiente conocimiento para entender a nivel teórico/practico las comunicaciones entre los dispositivos y a través de ese conocimiento extraer información útil para fines maliciosos.

Por otra parte, la estandarización, uso de protocolos y soluciones enfocadas a la protección de comunicaciones podría ejercer un contra peso en contra de estas herramientas, sin embargo, es toda una realidad que estas buenas prácticas no se utilizan en todos los casos, de tal forma, que el uso de sniffers en entornos inseguros podría resultar en la obtención de contraseñas, información de pagos, información delicada resultado del envío de correos electrónicos en servidores locales, información clave para lograr estafas mediante ingeniería social, etc. En general, existen dos vertientes de ejecución en el sniffing:

> **Sniffing pasivo**: Se trata de una técnica que se ejecuta cuando existen repetidores o concentradores en la red (Hubs), los cuales se encargan de redirigir el tráfico de manera general, sin algún mecanismo que regule y segmente el tráfico. Debido a esta característica, cada dispositivo conectado recibe todo el tráfico de red y determina si le corresponde o no, y, por lo tanto, un sniffer colocado en uno de estos dispositivos será capaz de absorber todo el tráfico de la red de manera sin restricción alguna.

Sniffing activo: Al contrario del sniffer pasivo, en la mayoría de las infraestructuras de la red existen dispositivos encargados de regular y segmentar la red, es decir routers y switches. Entonces, el tráfico que cada dispositivo llega a percibir está limitado y únicamente corresponde a sus funciones (como una buena práctica de seguridad de red). Es entonces que el sniffing activo entra en juego, el cual consiste en utilizar medios y técnicas para lograr un direccionamiento que le permita alcanzar todos los segmentos de red y de esta forma alcanzar a percibir más tráfico.

4.3.2 Ataques de denegación de servicio

Ampliamente conocidos como Denial of Services (DoS), la principal característica de este tipo de ataque es inhabilitar el uso de un sistema, proceso o servicio (OSI, 2018). En general, la fama de estos ataques se debe a otra técnica de la misma naturaleza, Distributed Denial of Services (DDoS), el cual posee el mismo objetivo que los DoS, pero con la diferencia que este se ejecuta a través de múltiples peticiones al mismo tiempo. Para aclarar este punto, Verdejo (SA) nos presenta dos definiciones muy útiles:

- Ataques tipo **Deny Of Services** (DoS): Ataques destinados a eliminar total o parcialmente la presencia en Internet de un ordenados o servicio, se caracterizan por tener un único origen del ataque.
- Ataques tipo **Distributed DOS** (DDoS): De manera similar a DOS, sin embargo, el ataque realizado por esta técnica se realiza a través de varias fuentes coordinadas para realizar un ataque total.

Probablemente este tipo de ataques son los más famosos debido a su principal característica: colapsar servicios. Por ello, es fácil encontrar que los ataques más famosos en el internet son de tipo DDoS, como es el caso del gusano “Morris”, el cual tenía como finalidad investigar en el tamaño del internet en la década de los 80’s; se trataba de un experimento realizado por Robert Tappan Morris que tenía la finalidad de detectar cada terminal conectada a internet y de esta forma hacer un mapeo del total de dispositivos conectados a internet, sin embargo, debido a un error en el código el gusano creció de forma ilimitada hasta que finalmente colapso toda la red ARPANET (Rubén Andrés, 2020).

En general hay muchos tipos de ataques de denegación de servicio ya que existen muchas formas de colapsar un servicio, los hay como el famoso caso de la compañía Dyn que se mencionó en la sección *3.2.1* y los hay en una menor escala, lo suficiente como para ejecutarse de manera individual en un pequeño dispositivo IoT, como es el caso de los ataques de tipo DoS.

La Computing Technology Industry Association (CompTIA), establece algunos de los principales ataques tipo Denial of Services (SAb):

Inundaciones de paquetes SYN

Dado que los ataques DoS se caracterizan por realizarse desde una sola fuente, en este sentido, este ataque consiste en usar una sola fuente para generar una inundación de paquetes SYN, la cual es posible mediante la manipulación del handshake de tres vías, correspondiente al protocolo TCP/IP.

Para entender en que consiste este ataca es necesario abordar características del protocolo TCP. Debido a que el protocolo TCP está orientado a la conexión, es decir, se enfoca a verificar que antes de que comience la comunicación, exista una conexión verificable entre ambas partes de la comunicación. Por otra parte, durante la comunicación siempre existirían parámetros de control para verificar que existe una comunicación entre el cliente y el servidor. Por lo tanto, para garantizar estos aspectos de la comunicación se utilizan varios parámetros, entre los cuales destacan (Deland-Han, et. al, 2021):

- **ACK**: campo de control asignado para el reconocimiento o "acknowledgment"
- **RST**: campo de control para el reinicio de la conexión
- **SYN**: campo de control que contiene números secuencia para sincronización
- **FIN**: campo de control para indicar el final de la conexión

La importancia del handshake recae en sus dos principales funciones. La primera, establecer una conexión, y la segunda, finalizar una conexión. De acuerdo con Pérez (2018) el proceso de handshaking de tres vías se realiza a través de tres pasos principalmente:

1. El cliente establece una conexión con el servidor a través del envió de un paquete SYN. Este paquete contiene un numero de secuencia.
2. El servidor recibe el paquete y establece su propia secuencia de números para responder al cliente, y establece el reconocimiento del mensaje del cliente incrementando en 1 el número de secuencia del cliente.
3. El cliente reconoce el mensaje del servidor incrementando en 1 el mensaje y enviándolo de vuelta.

De manera gráfica se obtiene la siguiente conversación:

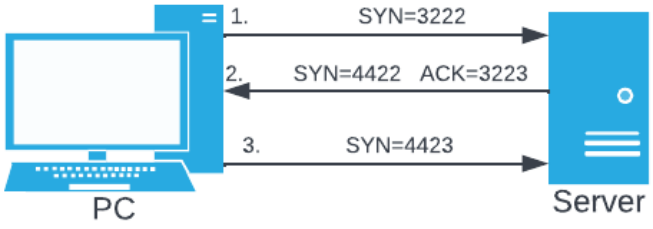

Figura 11. Representación gráfica del comportamiento del handshake de tres vías (Fuentes: Figura de elaboración propia).

Tomando en cuenta la figura 11, se aprecia que el proceso del handshake de tres vías es un proceso relativamente sencillo, de tal forma que es natural preguntarse ¿de qué forma puede ser explotado un proceso tan sencillo para generar un ataque de tipo DoS? De igual manera, la respuesta es sencilla, ya que una inundación de paquetes SYN (SYN Flood attack) sucede cuando un host es sobresaturado por mensajes SYN que intentar iniciar una conexión.

De manera específica el ataque se desarrolla en el paso 1 del handshake, cuando un host atacante envía un mensaje de tipo SYN al servidor, sin embargo, este host es capaz de enviar múltiples mensajes SYN con diferentes direcciones inexistentes o inalcanzables, y por lo tanto el servidor enviara múltiples respuestas SYN/ACK hacia dichos hosts inalcanzables y quedara en espera de su repuesta, mientras su memoria se sobresatura poco a poco. Una vez que la memoria del servidor se encuentre llena, este será incapaz de establecer nuevas conexiones con otros clientes y entonces, ocurre una denegación en los servicios (JUNIPER NETWORKS, 2021).

“The ping of death” & ping flood

Durante las primeras décadas del internet uno de los primeros ataques de tipo DoS que surgieron en internet fue el llamado “ping de la muerte” (Digital Guide IONOS, 2021)), el cual consistía en aprovecharse del protocolo de red ICMP para enviar un paquete de datos alterado a la víctima y de esta forma lograr bloquearlo. Actualmente se trata de un ataque obsoleto, sin embargo, actualmente existen otras variantes de este mismo ataque como es el caso del ping flood o ICMP flood, que en esencia siguen la misma filosofía, explotar el protocolo ICMP.

El protocolo ICMP o Internet Control Message Protocol forma parte de la implementación del protocolo IP (Protocolo de internet), ambos operan a nivel de la capa de red y son de suma importancia. Si bien resulta obvio que el protocolo IP es el más importante de la capa de red, el protocolo ICMP es una herramienta vital para todo administrador de redes, ya que permite, probar si un host o destino se encuentra “vivo” y es “alcanzable”. Por ejemplo, cuando se realiza el despliegue de un nuevo dispositivo en la red y necesitamos corroborar que efectivamente este dispositivo se encuentra “en línea” precisamente utilizamos el comando ping para averiguarlo, es decir, se utiliza el protocolo ICMP. Otras funciones del protocolo ICMP corresponden a informar de problemas en los parámetros de las cabeceras del datagrama, encargarse de la sincronización del reloj y el tiempo de envió del mensaje, obtener direcciones de Internet y máscaras de subred, entre otras aplicaciones (IBM, 2021).

En contra parte, el ataque ICMP flood consiste en hacer caer un host mediante una sobre saturación de peticiones ICMP del tipo echo, mejor conocidas como “pings”. La saturación de pings ocurre debido a que cuando un host recibe una petición tipo ping, este debe responder con la misma cantidad de peticiones que recibe, por ejemplo, cuando queremos verificar que existe una conexión continua, se ejecuta un ping extendido (mediante el comando *–t*), lo cual significa que enviaremos peticiones ICMP de manera indefinida, y por lo tanto, el dispositivo (si es alcanzable y está en línea) responderá de manera indefinida, al menos hasta que cancelemos el ping extendido o el dispositivo pierda conexión.

En este sentido, el ataque se basa en ese principio de respuesta propio de las peticiones echo, existen varias formas de ejecutar este ataque, dependiendo si se conoce o no la dirección IP de la víctima o si la víctima en específico se trata de un host o no, de igual forma existen muchas herramientas para lograr este ataque, se puede ejecutar con ayuda de herramientas disponibles en Kali Linux (sistema operativo, que posee facilidades para el pentesting), software especializado, programación en código, entre otros.

Deauthentication attack

La tecnología WiFi es relativamente nueva, su auge y popularización corresponde a décadas recientes debido a la popularidad de nuevos paradigmas como es el caso del Internet of Things (Kristiyanto & Ernastuti, 2020). El uso de la tecnología WiFi o más bien, la incorporación del estándar 802.11 es bastante conveniente para conectar a múltiples dispositivos como sensores, tablets, smartwaches, laptops, dispositivos compactos y demás dispositivos IoT en espacios amplios y con la ventaja de contribuir al ahorro de espacio e infraestructura debido a que se suprime la necesidad de cables para conectar a los dispositivos entre sí.

Otro de los aspectos que contribuye al uso de estas tecnologías es que únicamente basta con que exista un dispositivo tipo AP (Access point) que este dentro del rango del dispositivo y listo, existe la posibilidad de conexión. Para lograr esta conexión los dispositivos IoT necesitan cumplir con un procedimiento sencillo en el que el cliente (el dispositivo a conectar) realiza un procedimiento de autenticación con el AP y posteriormente lograr una asociación (Ananay Arora, SA).

Podemos entender al proceso autenticación como el primer paso para lograr una conexión con el AP, este proceso básicamente tiene el objetivo de establecer la identidad del dispositivo que pretenderse conectarse a la red. Por otra parte, la asociación se refiere al registro exitoso de los dispositivos con el AP, permitiéndoles el acceso a la red (Intel, 2021b).

Por ello, para comprender este procedimiento es necesario definirlo en pasos, tal como lo hace Cisco (2021). Existen tres estados para determinar el estado en el que encuentra un dispositivo:

1. No Asociado o no autenticado
2. Autenticado, pero aún no asociado
3. Autenticado y asociado

Dependiendo de los frames o mensajes que envié el cliente hacia el AP y viceversa el dispositivo tendrá un determinado estado, sin embargo, existen dos tipos de mensajes que destacan durante este proceso de comunicación. El primero es un frame de disociación que se envía cuando el cliente quiere romper la conexión con el AP. El segundo es el mensaje de des-autenticación, que es enviado cuando el cliente rompe la conexión con el AP de manera abrupta (Ananay Arora, SA).

La importancia de estos mensajes se relaciona directamente con el ataque conocido como *deauthentication attack*, el cual es un ataque que funciona sobre el estándar 802.11 y su función es romper una comunicación entre un cliente y un AP. Esto es posible debido a que el estándar 802.11 no posee mecanismos de autenticación durante su comunicación y por lo tanto es posible enviar

mensajes al servidor con direcciones falsas (spoofing) para engañar al AP y desconectar a clientes víctimas de manera involuntaria. (Kristiyanto & Ernastuti, 2020). Si bien, hoy en día existen diferentes protocolos de la familia 802.11 que atienden a la vulnerabilidad mediante mecanismos de autenticación durante la comunicación, se trata de un ataque vigente debido a que existen muchas oficinas que utilizan modelos de APs obsoletos que basan su funcionamiento en versiones inseguras del protocolo 802.11.

4.3.2 Ataques de ingeniería social

Actualmente el desarrollo de soluciones de ciberseguridad es un mercado activo, con múltiples fabricantes y productos que cubren cada aspecto, medio y posible brecha que pueda ser atacada por algún malware, ransomware o ataque de cualquier tipo. Sin embargo, hay un factor que no pueden cubrir estas soluciones, al menos, no de forma nativa. Como bien menciona Joan Cognom, el método el máximo de seguridad posible es no tener sistemas que vulnerar, después de todo "una computadora apagada es una computadora segura", sin embargo, a pesar de que tales condiciones existan, siempre habrá otra brecha, una siempre será vulnerable: el factor humano.

La ingeniería social se define como el conjunto de mecanismos, métodos, acciones o conductas para obtener información de usuarios que estén relacionados a un sistema, a través de esta información es posible ganar acceso a algún sistema o en su defecto obtener algún beneficio (Cognom, 2009). Existen infinidad de técnicas posibles para realizar ataques de ingeniería social debido a que su objetivo es engañar al usuario para que este otorgue sus datos de manera voluntaria. Esto puede lograrse, por ejemplo, a través de una página web falsa que emule la apariencia de un portal de banco, o quizás mediante algún correo falso que tenga como objetivo simular un mensaje de hacienda. Las técnicas de ingeniería social no se ven limitas a algún medio, sino más bien a la imaginación del atacante, después de todo su objetivo principal engañar al usuario, por ello de trata del tipo de ataque más efectivo.

5. Desarrollo

Una vez explorada la teoría necesaria para abordar el presente proyecto, a continuación, se presentan los elementos utilizados para su implementación:

5.1 Recursos y medios a utilizar

5.1.1 ESP32

El módulo ESP32 se puede definir como un mCU (microController Unit), mejor conocido como circuito integrado programable, cuenta con una unidad de procesamiento, memoria y pines para incorporar periféricos (Ben Lutkevich, 2019). De manera específica los ESP32 son especiales debido a que su construcción contempla funciones W-Fi y conectividad Bluetooth. Su diseño le permite soportar ambientes industriales (-40 °C y 125 °C), requiere un consumo bajo de energía, posee un nivel de integración en cualquier proyecto y puede funcionar como un dispositivo standalone o como un host MCU en sistemas de comunicación (Espressif Systems, 2022a).

El módulo ESP32 de manera nativa soporta protocolos TCP/IP, 802.11 Wi-Fi, transmisión de 2.4GHz y soporte para funciones de antena. De la misma forma, soporta el protocolo 802.3 para comunicaciones ethernet (Espressif Systems, 2022b). Entre algunas funciones interesantes también soporta Generadores de Números Aleatorios (RNG) para operaciones de cifrado y descifrado, incluyendo algunos algoritmos de cifrado como AES, RSA y SHA (Espressif Systems, 2022c).

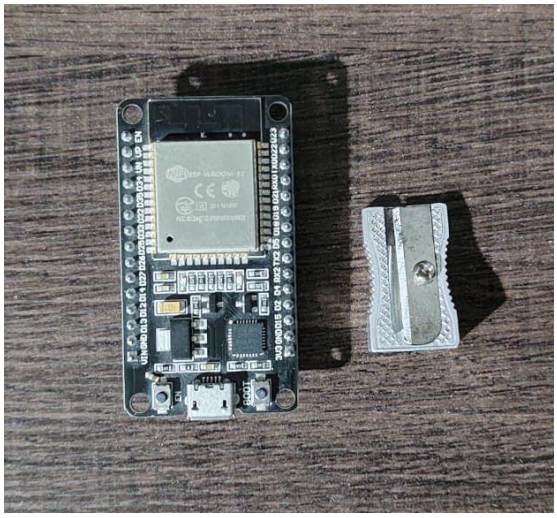

Figura 12. Relación de tamaño del ESP32 (Fuente: Imagen de elaboración propia).

En la figura 12, se aprecia un ESP32 en su presentación devkit1. Este modelo se trata de un kit de desarrollo, es decir posee un conjunto de pines que pueden ser utilizados para acceder a los diferentes componentes electronicos a la ESP32:

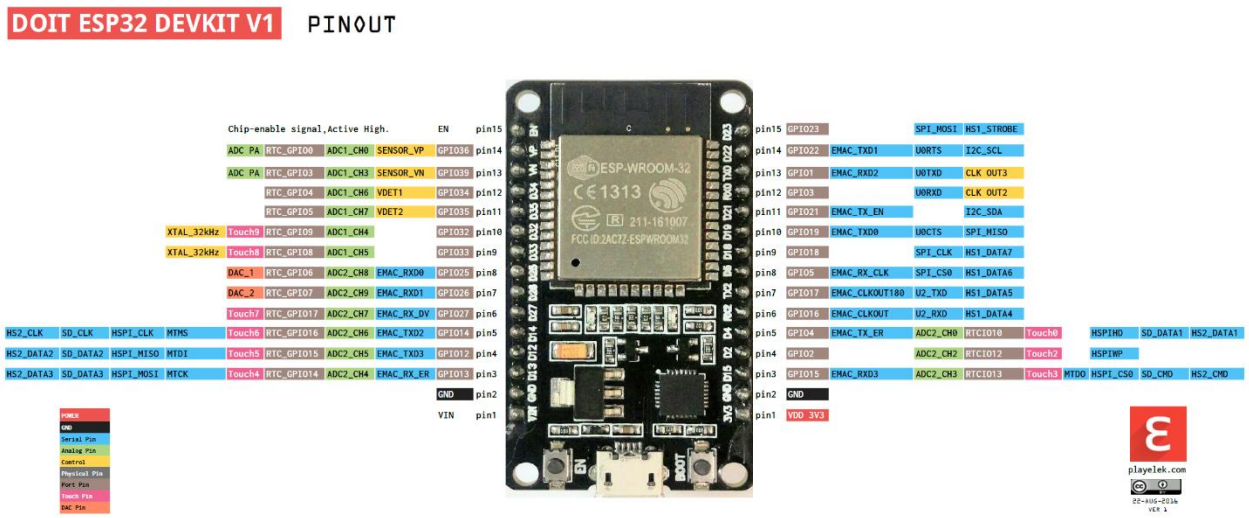

Figura 13. Diagrama de pines correspondientes al ESP32 Devkit1 (Fuente: playelek.com, 2016).

Como se observa en la figura 13, a través de estos pines es posible crear circuitos electricos con diferentes componentes electronicos, dandole la capacidad de trabajar con motores, pantallas lcd, leds, sistemas electricos, sensores e infinidad dispositivos. Precisamente es esta flexibilidad que posee la que permite a este dispositivo desempeñarse en multitud de proyectos telacionados a la domotica y el IoT, a continuación se presentan las caracteristicas generales del dispositivo así como sus especificaciones de transmisión inalambrica:

Tabla 1:
Especificaciones técnicas del ESP32

ESP32 DEVKIT V1	
Propiedades electricas	
Caracteristica	Propiedad
Voltaje de Alimentación (USB):	5V DC
Voltaje de Entradas/Salidas	3.3V DC
Consumo de energía	5µA en modo de suspensión
Conversor Analógico Digital:	Dos ADC de 12bits tipo SAR, soporta mediciones en

	hasta 18 canales, algunos pines soportan un amplificador con ganancia programable
Pines Digitales GPIO	24
Propiedades de conectividad	
Módulo Wifi	802.11 b/g/n/e/i (802.11n @ 2.4 GHz hasta 150 Mbit/s)
Poder de Transmisión Wifi	20.5 dBm
Módulo Bluetooth	4.2 BR/EDR BLE Modo de control dual
Seguridad:	IEEE 802.11, incluyendo WFA, WPA/WPA2 y WAPI
Criptografía acelerada por hardware	AES, SHA-2, RSA, criptografía de curva elíptica (ECC), generador de números aleatorios (RNG)
Propiedades de procesamiento	
Chip USB-Serial:	CP2102
CPU principal:	Tensilica Xtensa 32-bit LX6
Procesador secundario:	Operaciones básicas
Memoria:	448 KByte ROM, 520 KByte SRAM, 6 KByte SRAM en RTC y QSPI admite múltiples chips flash /SRAM
Frecuencia de Reloj	hasta 240Mhz

Nota: Obtenido de *ESP32 DEVKIT V1 30 Pines Wifi + Bluetooth,* por UNIT ELECTRONICS, (2022).

A la vista de las propiedades expuestas en la Tabla 1 es posible conocer las especificaciones técnicas del dispositivo, por consiguiente, es posible conocer en qué condiciones puede ser desplegado, en este caso el dato más relevante corresponde al voltaje de alimentación del dispositivo, que es de 5V. Por otra parte, se observa que el dispositivo es compatible con tecnología WiFi.

5.1.2 Estrategia de despliegue del proyecto

Puntualmente, el proyecto únicamente utilizó las propiedades de conectividad inalámbrica del dispositivo ESP32, es decir, no se utilizaron componentes electrónicos adicionales para realizar el proyecto. Por lo tanto, la principal estrategia que fue utilizada para desarrollar el proyecto fue la siguiente:

- Usar el microcontrolador ESP32 como una herramienta para cargar y ejecutar código que hace uso del protocolo WiFi

A continuación, se describe el flujo de desarrollo del proyecto:

1. Conectar la ESP32 a una computadora para cargar código
2. Desarrollar el código a través del IDE oficial de Arduino
3. Cargar el código
4. Desconectar la ESP32 de la computadora
5. Conectar la ESP32 a una fuente de alimentación de 5V para su despliegue operativo

Por último, se observa en el paso 5 que se necesita de una fuente de alimentación de 5V para la operación de la ESP32, es decir, el único requisito para que un ESP32 ejecute el código que le fue cargado es conectarlo a una fuente de alimentación soportada. Esta característica fue perfecta para el desarrollo de la presente ya que se traduce en un dispositivo que posee autonomía operativa.

5.1.3 Plataformas de desarrollo

Como ya se mencionó, la estrategia principal es usar la ESP32 para ejecutar código, a continuación, se describen las herramientas designadas para este proceso:

1. Laptop/computadora con acceso a internet

El ESP32 posee una entrada micro USB compatible con cables USB por lo cual es fácil de manipular, alimentar y conectar a computadoras para su programación. Por otra parte, también se necesita de acceso a internet para hacer uso de herramientas didácticas, búsqueda de información, consulta de librerías, consulta de código, ejecución de pruebas a través de la red. Dado que el módulo programable se conectará a internet, el uso de una computadora es esencial para corroborar las configuraciones aplicadas del por el ESP32.

2. Arduino IDE

El módulo ESP32 es compatible con el entorno de desarrollo oficial de Arduino, la integración que posee con este IDE es nativa, por lo tanto, fue el seleccionado para cargar y compilar el código necesario para el proyecto. En general, el hardware (ESP32) es compatible con lenguaje Arduino, esp-idf, MicroPython, Simba Embedded, Javascript, RTOS, entre otros. Sin embargo, el lenguaje que se maneja normalmente en este IDE es Arduino, el cual está basado en C/C++. A través del Arduino IDE se realizará toda la programación del hardware de acuerdo con las necesidades del proyecto. Por otra parte, el Arduino IDE fue usado como plataforma de pruebas y de depuración del código hasta obtener el código deseado

3. Programación

De manera general se puede decir que existen dos maneras de programar una ESP32, mediante

métodos nativos y no nativos. En este orden de ideas, los métodos nativos son aquellos que están en línea con la documentación oficial por parte del fabricante y que no requieren de alguna modificación externa al microcontrolador para que estos funcionen. Es decir, a través de estos métodos nativos solamente es necesario descargar algún IDE de preferencia, usar las librerías correspondientes y conectar la tarjeta.

Un ejemplo de estos métodos nativos corresponde a las librerías oficiales para la gestión de su arquitectura, cortesía de Espressif Systems: https://github.com/espressif. Estas librerías permiten manipular cada una de las funciones del dispositivo; su uso es tan amplio que es compatible con HTML, de tal forma que es posible crear interfaces web para la entrada de datos y operaciones web.

Por su parte, se pueden denominar métodos no nativos a aquellos que no forman parte de la documentación oficial del fabricante y que requieren de alguna modificación u proceso especial en el dispositivo para lograr una compatibilidad. Una manera de ejemplificar estos métodos es con el video de YouTube *"Flashing Esp32 & Esp8266 For Wifi Hacking by Eviltwin & Deauther Attack"* hecho por el usuario Chhatramani Yadav (2022). En este video nos muestra cómo realizar el proceso de "flasheo" en el ESP32, el cual es un proceso especial que altera las propiedades de fábrica del ESP32 y permite la carga de código especializado, en este caso librerías de código enfocadas al WiFi hacking

En contraposición con los métodos nativos, los no nativos requieren de procesos especializados sobre la tarjeta, generalmente estas librerías de código compatibles con estos métodos son desarrolladas por la comunidad y corresponde a un desarrollo abierto bajo la responsabilidad de cada usuario. Esta versatilidad open source permite la existencia de infinidad de librerías y código disponible desarrollado por la comunidad, haciendo posible el desarrollo de librerías especializadas al hacking.

De la misma forma, existen librerías oficiales enfocadas al control de las funciones Wi-Fi del dispositivo, así como para la comunicación y procesamiento de datos de este. Al contar con muchas librerías de código disponible, es posible programar el dispositivo para realizar infinidad de actividades, por lo que también existe una comunidad activa dedicada a desarrollar código y nuevas librerías basadas en métodos nativos.
Para el presente proyecto se planea usar métodos nativos. Esta decisión fue tomada con la finalidad de apoyar a la premisa de tesis, ya que el punto central es demostrar el potencial que ataque y defensa de estos dispositivos, y que mejor forma de hacerlo que a través librerías oficiales del fabricante. De esta manera, se puede demostrar que cualquier usuario es capaz de desarrollar proyectos de esta naturaleza.

Finalmente, se presenta el conjunto de librerías (oficiales por parte del fabricante) en la cual fue desarrollado el proyecto:

- Framework oficial arduino-esp32: https://github.com/espressif/arduino-esp32
- Librerías principales: https://github.com/espressif/arduino-esp32/tree/master/libraries

5.1.4 Modos de operación del ESP32

Una vez establecidas las librerías y el framework que fue utilizado, corresponde hablar de las configuraciones disponibles en las que puede operar un ESP32. Los modos de operación responden a la implementación de la librería <WiFi.h>, la cual encarga de la conexión de red local y a internet a través de la tecnología WiFi del dispositivo.

Estos modos de operación están descritos en la documentación oficial del framework Arduino-esp32 (ESPRESSIF, 2022). Específicamente esta librería atiende al funcionamiento del driver encargado del protocolo 802.11b/g/ (responsable de la comunicación WiFi). A través de estas configuraciones es posible manipular las funcionalidades inalámbricas del dispositivo:

1. **Modo de operación AP**

   El modo Access Point permite al dispositivo emular las propiedades de un AP, permitiéndole generar una red a la cual se pueden conectarse clientes. De manera específica, esta configuración es ideal para conectar a otros ESP32 entre sí, permitiendo que otras tarjetas que operen en modo STA se puedan comunicar con un ESP32 en modo AP. Es decir, a través de esta propiedad la tarjeta puede recibir conexiones provenientes de otros dispositivos. Este modo de operación es importante para la naturaleza del proyecto debido a que hace posible que variedad de dispositivos puedan establecer una comunicación con el ESP432, como es el caso de teléfonos, computadoras y cualquier dispositivo que posea conexión WiFi.

   En otras palabras, esta funcionalidad permite emular el funcionamiento de un AP, y, por lo tanto, permite la configuración de sus propiedades más características como es el caso de operaciones para definir una dirección IP, definir la dirección IP de los clientes, establecer un nombre SSID, establecer una contraseña de acceso para el SSID, gestionar la comunicación, configurar canales de transmisión, entre otras.

   Un aspecto a considerar es que, dadas las capacidades de procesamiento del dispositivo, la cantidad de clientes que la tarjeta puede atender responde a estas especificaciones, haciendo posible el manejo máximo de 8 clientes. Por lo tanto, esta configuración permite habilitar al ESP32 como un AP, capaz de manejar varios clientes, como se ejemplifica en la figura 14:

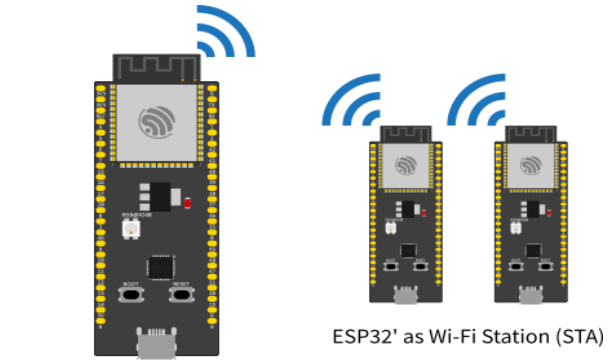

Figura 14. Modo de Operación Access Point (Fuente: ESPRESSIF, 2022).

Para acceder a esta funcionalidad, la palabra reserva en código corresponde a la siguiente nomenclatura: WIFI_AP

2. Modo de operación STA

El modo estación o STA, permite la configuración de un ESP32 como cliente (Figura 15). A través de esta función el dispositivo puede establecer comunicación con servidores, redes WiFi y por supuesto, a internet.

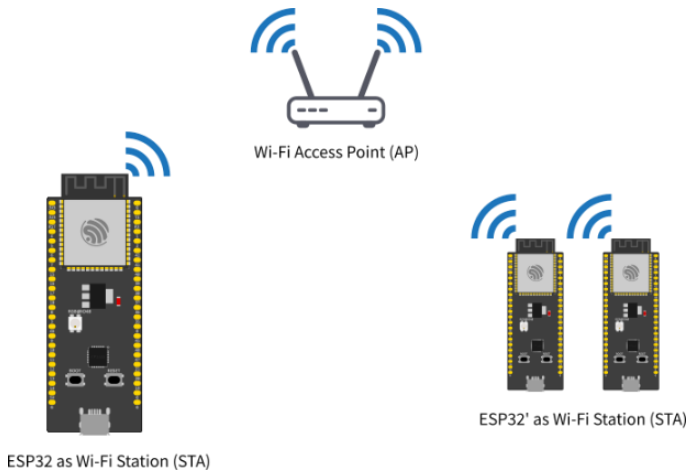

Figura 15. Modo de Operación Estación (Figura: ESPRESSIF, 2022).

A través de esta configuración el dispositivo puede conectarse a una red WiFi, y por tanto acceder a los servicios disponibles a los que este tenga acceso, siendo compatible con el direccionamiento disponible.

Para acceder a esta funcionalidad, la palabra reserva en código corresponde a la siguiente nomenclatura: **WIFI_STA**.

3. Modo de operación Híbrido

Finalmente, el modo de operación hibrido se trata del uso de ambas configuraciones al mismo tiempo (Figura 16), es decir, permite al ESP32 operar como un AP y como un cliente al mismo tiempo. Este modo fue el más importante para el proyecto debido a que permite una comunicación de dos sentidos, permitiendo que otros dispositivos se conecten al ESP32 mientras que al mismo tiempo el ESP32 se puede conectar a una red.

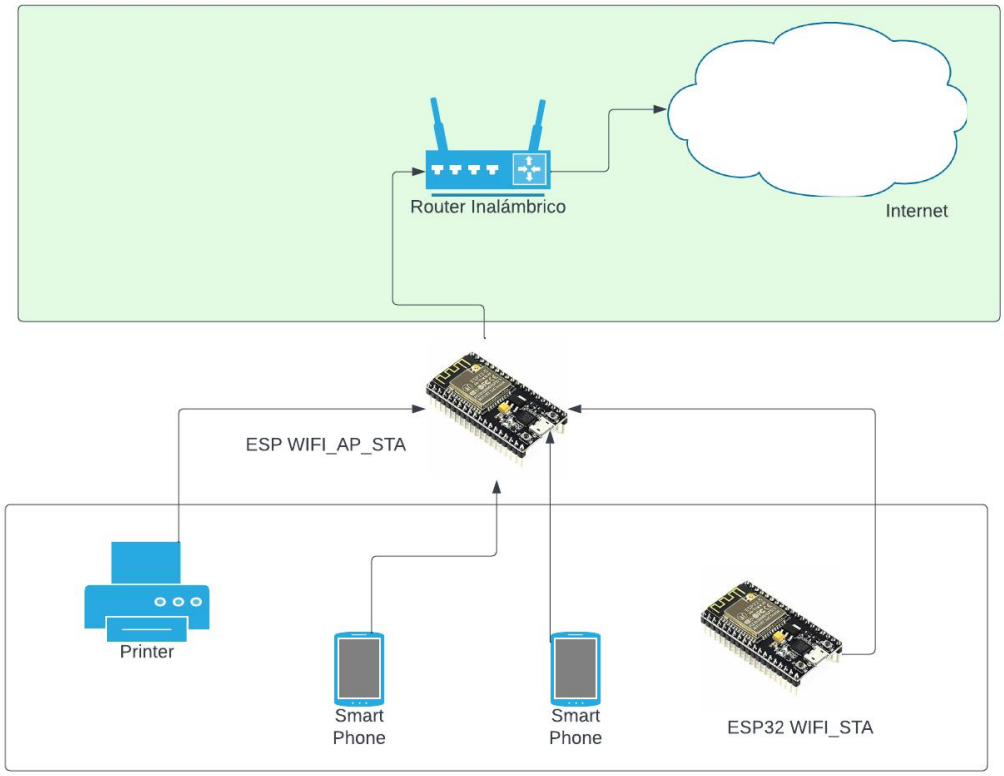

Figura 16. Modo de Operación Híbrido (Fuente: Imagen de elaboración propia)

Para acceder a esta funcionalidad, la palabra reserva en código corresponde a la siguiente nomenclatura: **WIFI_AP_STA**

5.2 Metodología

5.2.1 Framework a utilizar

Como fue expuesto en la sección *3.1 Justificación*, existen varios frameworks para desarrollar una prueba de pentesting apropiadamente, como es el caso de The *Open Source Security Testing Methodology*, sin embargo, realizar un ejercicio de esta naturaleza requiere de un objetivo claro que responda a una necesidad formal, por lo tanto, otro framework que se ajusta más a la naturaleza didáctica y demostrativa del proyecto es la cadena de eliminación informática.

The *cyber kill chain* es un framework enfocado a la prevención e identificación de las intrusiones cibernéticas. Su nombre proviene del entendimiento de las fases que tiene que lograr un adversario (usuario malicioso) para lograr su objetivo, por lo, se identifican las fases que desarrollan para efectuar un ataque exitoso (Lockheed Martin., 2015).

La cadena de eliminación consta de siete fases:

Figura 17. Etapas de un ataque, basado en la cadena de eliminación (Fuente: Imagen de elaboración propia)

Conforme a las fases observadas en la figura 17 se describe lo siguiente:

1. Reconocimiento: Se trata de la fase de planeación, en donde se realiza una investigación sobre los objetivos a atacar. El objetivo principal es recolectar información que beneficie al ataque. Algunas técnicas para esta fase son: recolección de correos electrónicos, búsqueda de los objetivos vía redes sociales, identificación de servidores, recolección de datos relacionados con los objetivos y en general cualquier técnica que ayude a identificar vulnerabilidades.

2. Preparación: Una vez identificados los objetivos a atacar comienza la preparación del ataque a través de herramientas automatizadas, técnicas de ataque o herramientas con la capacidad de entregar código malicioso que se mantenga a la espera de ser explotado. Algunas técnicas relaciones a esta fase son: preparación de código malicioso para implantar a través de backdoors, preparación de archivos maliciosos, preparación de paylods, entre otros.

3. Entrega: Consiste en la entrega del medio para detonar el ataque, por ejemplo, sí el ataque consiste en malware, esta fase consistirá de entregar el medio que contiene este malware implantado. Algunos ejemplos de técnicas propias de esta fase son: envió de correos maliciosos, malware implantado en USB, entrega de vínculos maliciosos a través de redes sociales y cualquier técnica que sirva a este propósito.

4. Explotación: La fase de explotación probablemente es la importante debido a que consiste en la detonación del código malicioso o técnica del ataque preparada en la fase dos. Un concepto apropiado para esta fase es el de “día cero”; como lo menciona el fabricante en ciberseguridad Kaspersky (2022a), día cero hace referencia a una vulnerabilidad descubierta por un atacante, pero desconocida por el administrador del sistema, es decir, que este ha tenido “cero días” para corregirla. De la misma forma, a esta definición se le pueden añadir los términos exploits de día cero (el método para atacar la vulnerabilidad) y ataque de día cero (el uso de un exploit de día cero).

Algunos ejemplos de estas técnicas de explotación son: cuando un usuario da clic a un enlace malicioso a través de correo electrónico, ejecución de virus a través técnicas de ingeniería social, descarga de archivos maliciosos, ejecución de aplicaciones maliciosas.

5. Instalación: Una vez realizado el ataque de día cero, la técnica de ataque ejecutara lo necesario para lograr un acceso persistente en el sistema de la víctima, algunos ejemplos de esto son: la

instalación de un backdoor, la implantación de un key logger, la incrustación un webshell en un servidor u otras técnicas que permitan el acceso al sistema sin levantar alarmas de la intrusión.

6. Control: La fase cinco fue la responsable de crear la infraestructura necesaria para que el atacante logre entrar al sistema de la víctima, ahora, bajo estas condiciones el atacante posee la capacidad de crear canales de comunicación directos y de tomar control de la infraestructura de la víctima a través de diversos canales como puertos de comunicación, la misma red, mediante algunos hosts.

7. Objetivo disponible: Finalmente, un atacante que haya logrado completar la fase seis posee la capacidad de tomar control del sistema dándole la posibilidad de lograr su objetivo. Algunos ejemplos de objetivos finales pueden ser: Obtener acceso a sistemas de información, obtener privilegios en el sistema, realizar un mapeo detallado de la infraestructura comprometida, hacer movimientos laterales para tomar control de todas las redes del sistema, destruir el sistema, entre otros.

Esta metodología permite entender de manera global las fases de un ataque, al mismo tiempo que ayuda a la defesa contra un ataque, puesto que, al identificar las fases de cada ataque permite tomar acciones apropiadas para evitar que la fase seis se complete. Un aspecto para considerar es que se requiere de un conocimiento amplio de las técnicas de ataque disponibles, tanto en el rol de atacante, como el rol defensivo. En este sentido, es recomendable hacer uso de las matrices de MITRE ATT&CK, las cuales son un cúmulo de conocimiento de todas las técnicas de ataque registradas y al mismo tiempo se trata de un organismo al servicio de la comunidad con el objetivo de apoyar a la seguridad informática, con conocimiento y modelado de amenazas disponible para todo mundo (2022b).

Entonces el desarrollo del presente proyecto se basa en el uso de los conocimientos expuestos en secciones anteriores, apoyándose en el framework de la cadena de eliminación, así como en las definiciones de ataques que se encuentran en la matriz de MITRE ATT&CK.

5.2.2 Definición del ejercicio de ataque

De acuerdo con todas las bases teóricas mostradas se establece lo siguiente:

Tabla 2
Definición formal del ataque a ejecutar

Fase de implementación definida en la cadena de eliminación	**Reconocimiento**				
Táctica definida en MITRE ATT & CK	1. Enterprise 2. Mobile				
Técnica definida en matriz MITRE ATT & CK	Gather Victim Identity Information	Sub técnica disponible	Sí	Sub técnica	Email Address

		Técnica relacionada	Sí	Técnica	Input Capture
Alcance	Entornos Edge computing de naturaleza pública y empresarial				
Canal de ataque	Inalámbrico, a través de protocolo 802.11 b/g/n				
Método de entrega	Rogue Access Point externo				
Vector	Ingeniera social				
Reglas de compromiso	Recolección de información en un laboratorio controlado, con datos demostrativos.				

Nota: De elaboración propia

Partiendo de la información definida en la tabla 2, la técnica *Gather Victim Identity Information*, consiste en la recolección de información correspondiente a la identidad de las víctimas, como es el caso de nombres, correos, números telefónicos y cualquier dato que pueda identificar al objetivo, y, generalmente se ejecuta mediante técnicas de ingeniera social (MITRE ATT&CK, 2022c). Por otra parte, la sub técnica *Gather Victim Identity Information: Email Addresses*, se especializa en recolectar correos electrónicos que pueden ser utilizados en fases y técnicas posteriores (MITRE ATT&CK, 2022d).

Por otra parte, la técnica *Input Capture*, consiste en aquellos métodos para obtener credenciales e información a través de mecanismos como páginas de acceso, portales de identificación y formularios de internet, este tipo de técnica corresponde a aquellas tácticas que pueden ser ejecutas en dispositivos móviles (MITRE ATT&CK, 2022e).

5.3 Ejecución

5.3.1 Escenario de implementación

Puesto que, una de las principales características del Edge computing es el procesamiento de los datos en el borde de la red (Near end), antes de alcanzar el Far End, entonces, el ataque efectuado corresponde al siguiente escenario:

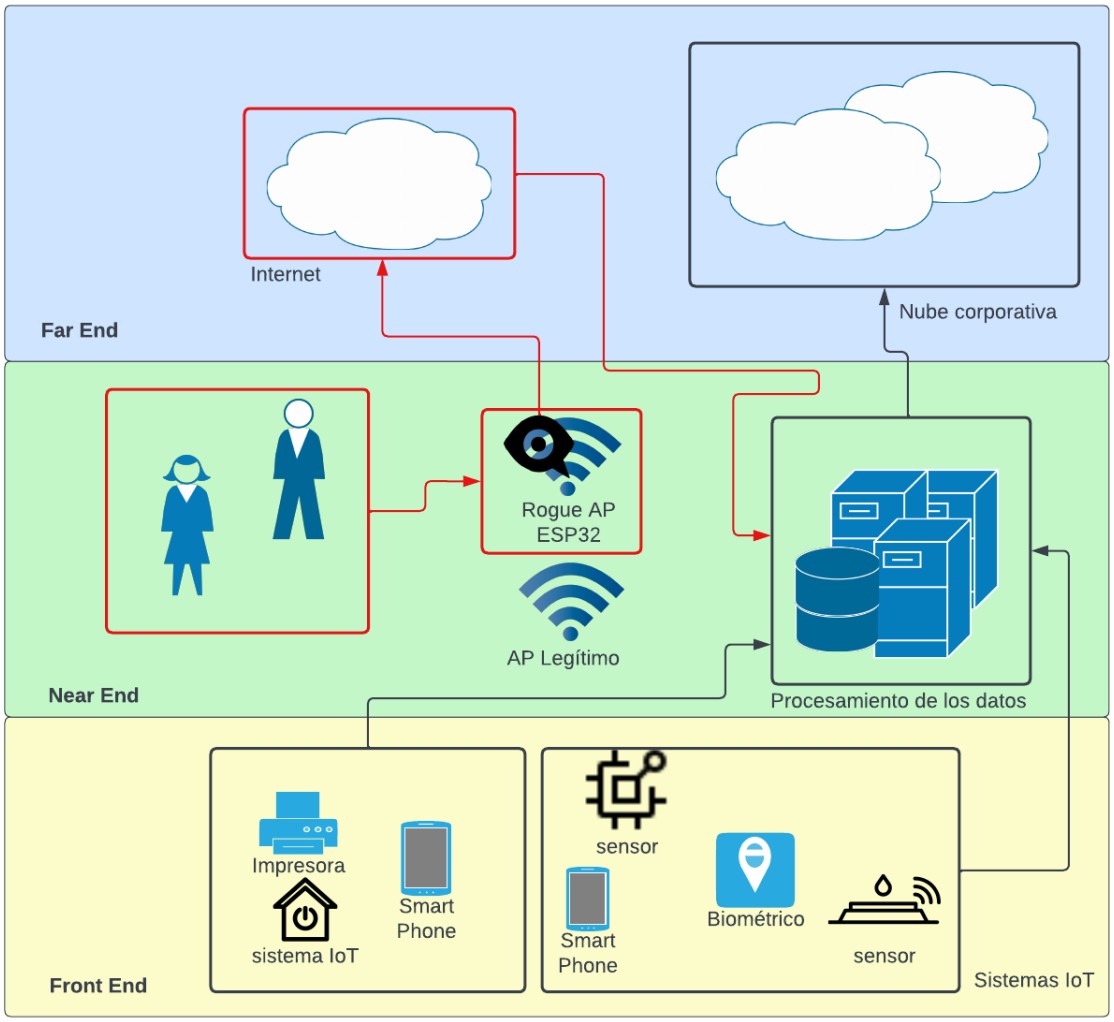

Figura 18. Representación del escenario de implementación (Fuente: Imagen de elaboración propia).

Se aprecia en la figura 18 que en el Front End se generan los datos recopilados por los sistemas IoT, los cuales se enviaran para su procesamiento en el Near End; en esta capa se encuentran los servidores, bases de datos y sistemas en el borde de la red encargamos de tratar estos datos. Finalmente, estos sistemas de borde podrán enviar estos datos a la instancia correspondiente, para el caso de este ejercicio se contempló una nube corporativa. Este proceso, se puede ver descrito en el diagrama por el flujo de cajas y flechas de color negro.

Por otra parte, en el diagrama se observa otra simbología de flujo, indicada con cajas y flechas de color rojo, las cuales indican el flujo del ataque. En este caso, el ataque se efectúa en el Near End, donde se encuentran los elementos encargados de la administración de la red, por lo que puede integrarse de manera orgánica a través de un Rogue AP externo para no levantar sospechosas. Entonces, se pretende que a través de la interacción de los usuarios con el Rogue AP se recopile información suficiente que será almacenada en una nube publica con la finalidad de obtener acceso a los sistemas de procesamiento de los datos.

Mediante este despliegue se pretende recopilar la información suficiente para posteriormente desarrollar las fases de preparación y entrega, a fin de lograr acceso a los sistemas de procesamiento de estos datos.

El objetivo de esta implementación responde a 3 objetivos:

1. Desarrollar una fase de reconocimiento para recolectar correos electrónicos, que podrán ser utilizados en la fase entrega:
2. Conformar una base de datos de identidades (conformada de nombres, correos electrónicos, teléfonos y género)
3. Almacenar la base de datos obtenida en un medio externo, en este caso el Far End, en una nube pública para su uso posterior.

5.3.2 Datos a recolectar

De acuerdo al objetivo dos, se presenta el formato usado para recolectar dichos datos:

Figura 19. Formato de recolección de datos (Fuente: Imagen de elaboración propia).

Conforme a la figura 19, se consideran los datos de *nombre*, *teléfono* y *género* para apoyar al método de entrega, ya que, al tratarse de la recolección de correos con fines maliciosos es necesario tener la mayor cantidad de datos del usuario para lograr cierto grado de credibilidad en la fase de entrega; por ejemplo, el dato del *género* está incluido ya que es probable que el usuario seleccione apropiadamente este dato al sentirse identificado, este dato es vital ya que puede ayudar a la redacción de un correo personalizado en una fase de entrega; por otra parte, el dato *nombre*, apoya a este argumento, mientras que el número de *teléfono* es un dato extra a obtener, el cual, no es relevante para la implementación, pero, en caso de obtenerlo será valioso.

En última instancia se aclara que los datos a recolectar en este aspecto son demostrativos, ya que, para una ejecución más apegada a la realidad, se podrían incluir requisitos de dependencia o departamento para identificar al usuario y su alcance dentro de una estructura corporativa. Este

formato será almacenado dentro de una variable HTML para que al momento de que un usuario acceda al Rogue AP visualice este formato.

5.3.3 Implementación

Recapitulando lo expuesto en las secciones *5.1.3 y 5.1.4*, se pueden establecer los siguientes requisitos de implementación:

1. El ESP32 necesita funcionar como un Access Point
2. El ESP32 requiere funcionar como un portal cautivo
3. Es necesario que el ESP32 funcione como un Servidor que gestione las peticiones de las victimas conectadas en el Near End, y, al mismo tiempo, como un cliente que mantiene comunicación con una base de datos en el Far End.

Para lograr el primer y segundo objetivo, se hará uso, de las librerías: *DNSServer* y *WiFi*. En un primer momento, la librería *DNSServer* permite la creación de un portal cautivo, mientras que la librería *WiFi* nos permite acceder a los modos de operación del ESP32. Sin embargo, el objetivo más complejo es el último, en un primer momento, se consideró la implementación de la librería de la librería WebServer, sin embargo, esta librería necesita del puerto 80 (http) para su ejecución. Debido a que la librería *DNSServer* requería del mismo puerto existía una incompatibilidad de operación. Entonces, se decidió utilizar otra librería enfocada a la comunicación asíncrona basada en el protocolo TCP.

ESPAsynWebServer es una librería creada por *Me No Dev*, un desarrollador de código basado en el framework de Arduino que define comunicación asíncrona como el manejo de múltiples conexiones al mismo tiempo, de tal forma que este proceso ocurre por turnos (2022). Por otra parte, la integración de HTML con esta librería es óptima, por lo tanto, se trata de una elección aceptable. Por último, respecto a la comunicación con la base de datos en el Far End, se utilizó la librería Firebase-ESP32 basada en el framework de Arduino y desarrollada por el usuario Suwatchai K (2022); a través de esta librería se establece la comunicación con la base de datos en la nube Firebase.

A continuación, se presentan los aspectos más importantes del código implementado:

1. Funcionamiento del ESP32 como AP

```
WiFi.mode(WIFI_AP_STA);
WiFi.softAP("ACCESSTREK_Wifi Gratis");
WiFi.softAPConfig(apIP, apIP, subnet);
```

Extracto de código 1: Configuración del ESP32 como AP

Se observa en el extracto 1 la definición de operación del ESP32 como modo hibrido, por otra parte, la función *SoftAP* establece el funcionamiento como un AP. En este caso se establece el SSID de la red a propagar y el direccionamiento designado para los clientes (dirección ip, gateway,

máscara de red).

Para este caso se consideró un SSID que aparente un nombre corporativo para huéspedes, sin embargo, se aclara que para que el ejercicio sea más efectivo el nombre de SSID debería responder a acceso del internet de algún departamento o sección que corresponda según sea caso, no obstante, al tratarse de un ejercicio didáctico se completó el nombre “ACCESSTREK_Wifi Gratis” como un ejemplo global.

2. Funcionamiento del ESP32 como Portal cautivo

```
const byte DNS_PORT = 53;
dnsServer.start(DNS_PORT, "*", WiFi.softAPIP());
```

Extracto de código 2: Configuración del portal cautivo

En el extracto 2 se define el puerto de operación del portal cautivo (53). También, se establece la configuración la configuración del portal cautivo a través de la función dns.Server. Cabe aclarar que la razón por la que esta función hace referencia a “server” es porque precisamente se trata de eso; la configuración de captive portal en el ESP32 fue construida como un servidor DNS, de esta forma logra redireccionar la comunicación cuando se establece como un AP, de ahí la función *softAPIP*. Siendo así, por ello se establece el puerto 53, dado que la operación del ESP32 como servidor opera en el puerto 80.

3. Funcionamiento del ESP32 como servidor que atiende peticiones de clientes

```
void DeployESP32Server(){
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
     request->send_P(200, "text/html", index_html);
  });

server.addHandler(new
CaptiveRequestHandler()).setFilter(ON_AP_FILTER);
server.begin();
```

Extracto de código 3: Configuración de servidor para manejar las peticiones de las victimas

En el extracto 3, el método *DeployESP32Server*, se encarga de configurar la respuesta a las peticiones realizadas por las victimas que acceden al portal cautivo; cuando esto sucede, se envía un código de estatus al servidor http (con el código 200, que significa OK), para así desplegar un formulario HTML (variable index_html) que simula un portal de acceso para el usuario.

Asimismo, la función *server.addHandler* configura el manejo las peticiones, es decir, cuando una petición sucede, esta función se encarga de atenderla y desplegar los métodos correspondientes para su respuesta. Finalmente la función *server.begin* mantiene el servidor a la escucha de las peticiones.

4. Funcionamiento del ESP32 como un cliente que mantiene comunicación con una base de datos

```
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.reconnectWiFi(true);

 while(peticion){
   Firebase.pushString(firebaseData, nodo + "/Usuarios", nombre);
   delay(100);
   Firebase.pushString(firebaseData, nodo + "/Usuarios", correo);
   delay(100);
   Firebase.pushString(firebaseData, nodo + "/Usuarios", telefono);
   delay(100);
   Firebase.pushString(firebaseData, nodo + "/Usuarios", gen);
   delay(100);

   peticion = false;
        }
}
```

Extracto de código 4: Configuración como cliente de una base de datos en la nube

En el extracto 4, la función *WiFi.begin* establece conexión a internet mediante el canal inalámbrico, mientras que con la función *Firebase.begin* se establece comunicación con la base de datos en la nube (usando las credenciales correspondientes). Por otra parte, a través de las funciones *pushString* se añaden los datos a la base de datos, en el caso de que una víctima envié sus datos a través del portal cautivo.

6. Resultados

6.1 Resultados obtenidos de la implementación

En función de lo definido en la sección *5.3 Ejecución*, se obtiene el siguiente diagrama de secuencia que explica el funcionamiento del código y su interacción con el usuario:

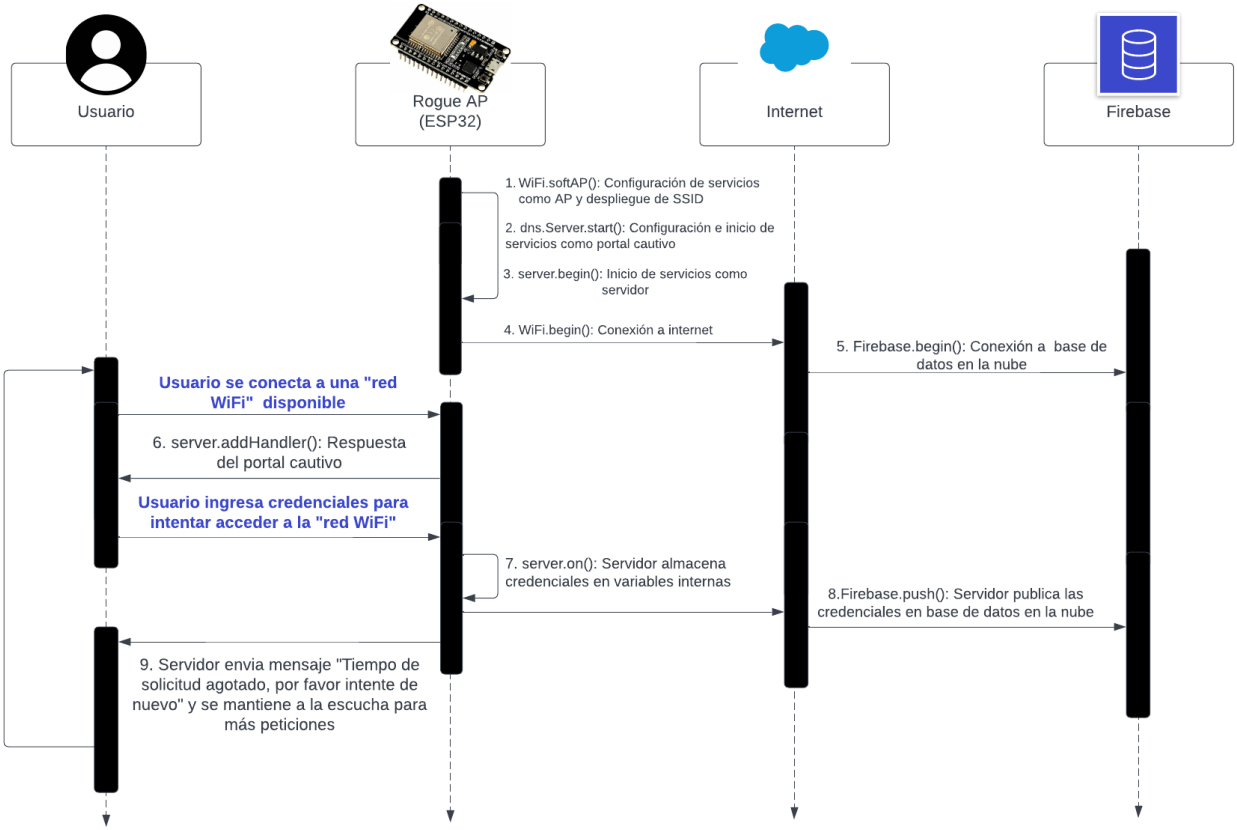

Figura 20. Diagrama de secuencia de identidades durante la ejecución del ataque (Figura: diagrama de elaboración propia)

Conforme a lo expuesto en la Figura 20, el ataque ocurre a través de 4 identidades:

1. El usuario
2. El Rogue Access Point
3. La internet
4. La base datos en la nube Firebase

Las interacciones del usuario se denotan con el texto en color azul y se limitan a acceder al Rogue AP e ingresar sus datos, mientras que la ESP32 mantiene contacto con todas entidades; en primera instancia establece comunicación con la internet para poder establecer comunicación con la base de datos en la nube, una vez alcanzada esta base de datos, mantiene comunicación de un solo sentido con ella, la cual se ve condicionada por la interacción del usuario.

De tal forma que por parte del usuario se puede observar lo siguiente:

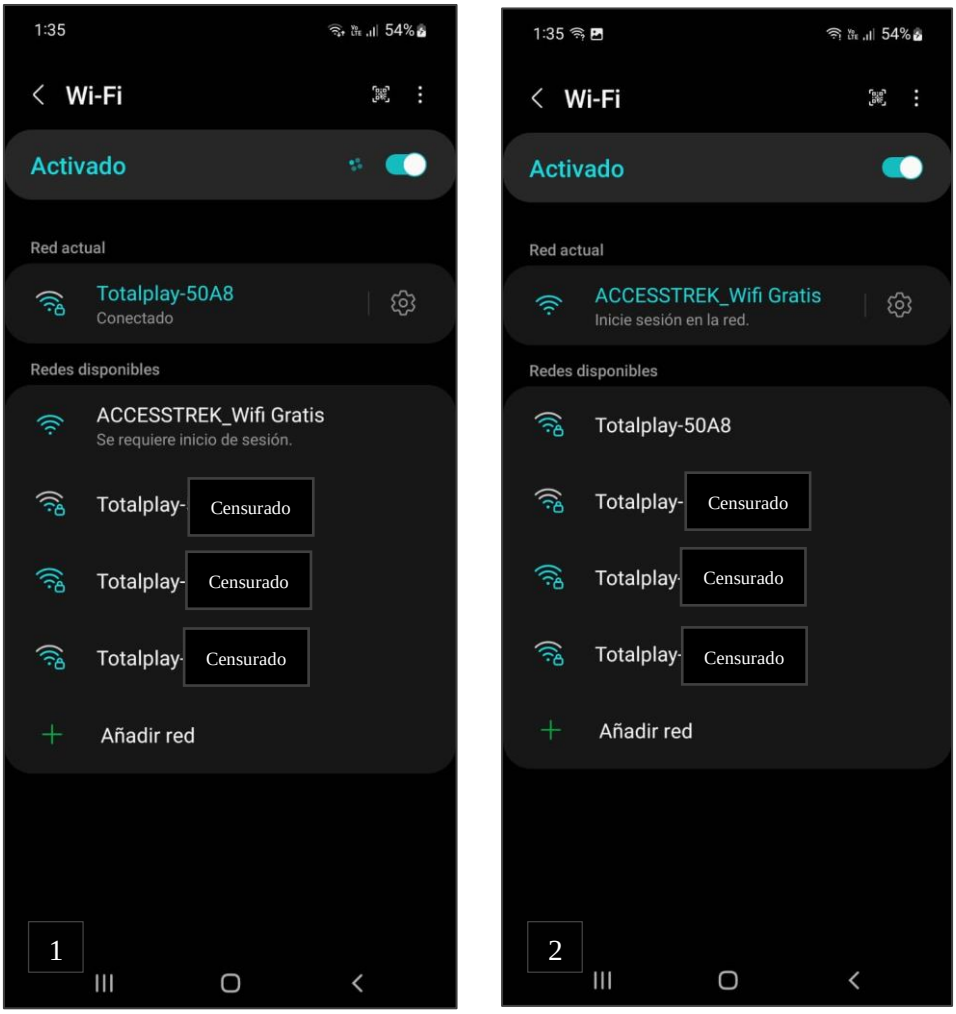

Figura 21. Capturas de pantalla 1 y 2, vista del usuario. (Fuente: Captura de pantalla).

En Figura 21 (1) se observa que aparecen una seria SSID correspondientes a las redes que se encuentran disponibles, entras ellas, el correspondientes al ROGUE AP, *"ACCESTREK_Wifi Gratis"*. Nótese que en esta red aparece la leyenda *"Se requiere inicio de sesión"*, la cual es normal tratándose de portales cautivos. Posteriormente, se muestra la pantalla que observa el usuario cuando selecciona la red el Rogue AP, indicando que debe acceder dar clic para acceder a la red (2).

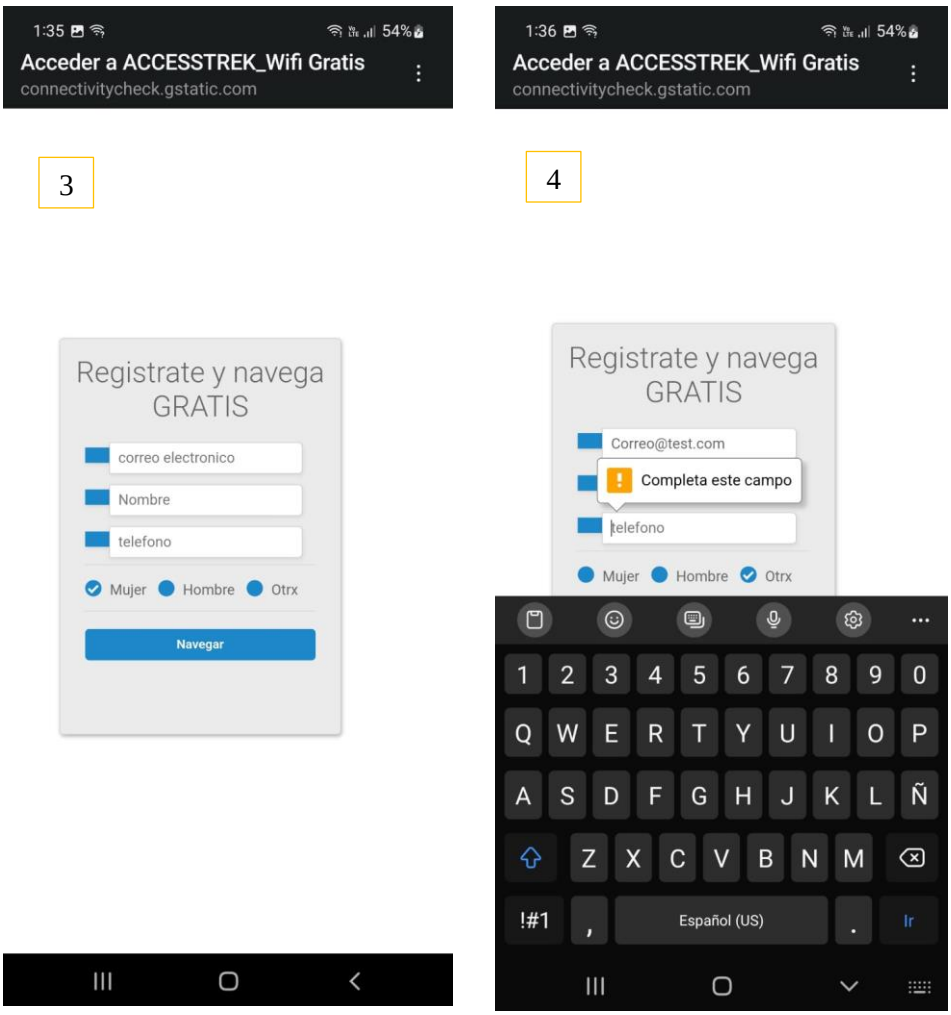

Figura 22. Vista del usuario, pantallas 3 y 4 (Fuente: Captura de pantalla).

De acuerdo con lo observado, en la figura 22, una vez pulsado el inicio de sesión el usuario observa una pantalla para ingresar sus datos para poder acceder a la red (3), una vez ingresados estos datos solo resta presionar el botón navegar, sin embargo, en que caso de que el usuario no ingrese alguno de los datos solicitados, aparecerá el mensaje “Completa este campo” (4).

Una vez que el usuario ingrese los datos correspondientes en el formulario, el sistema continuará con el proceso descrito en la *figura* 19, en este caso, nótese que los datos ingresados para esta prueba fueron:

Correo electrónico: Correo@test.com
Nombre: Tester
Teléfono: 0986580966
Género: Otrx

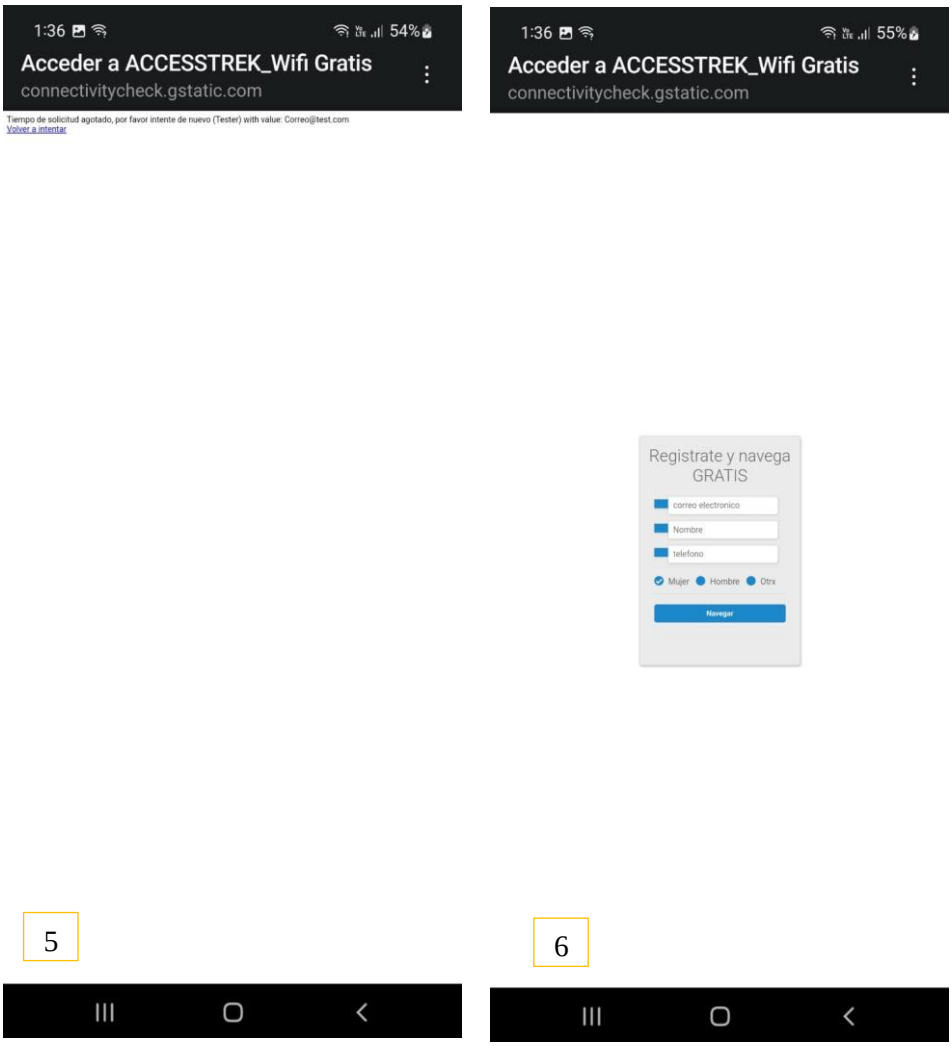

Figura 23. Capturas de pantalla de la vista del usuario (Fuente: Captura de pantalla).

Posterior al llenado del formulario, en correspondencia con la figura 23, una vez pulsado el botón *Navegar,* el ESP32 mostrará el mensaje “Tiempo de solicitud agotado, por favor intente de nuevo”, junto el nombre y el correo ingresado por el usuario (5); cuando el usuario visualiza este mensaje posee la opción de volver a la pantalla de inicio, en la cual, sí se pulsa, se volverá a visualizar el formato de acceso para el portal cautivo (6).

Por el otro lado, en la base datos en la nube (Firebase) se puede observar lo siguiente:

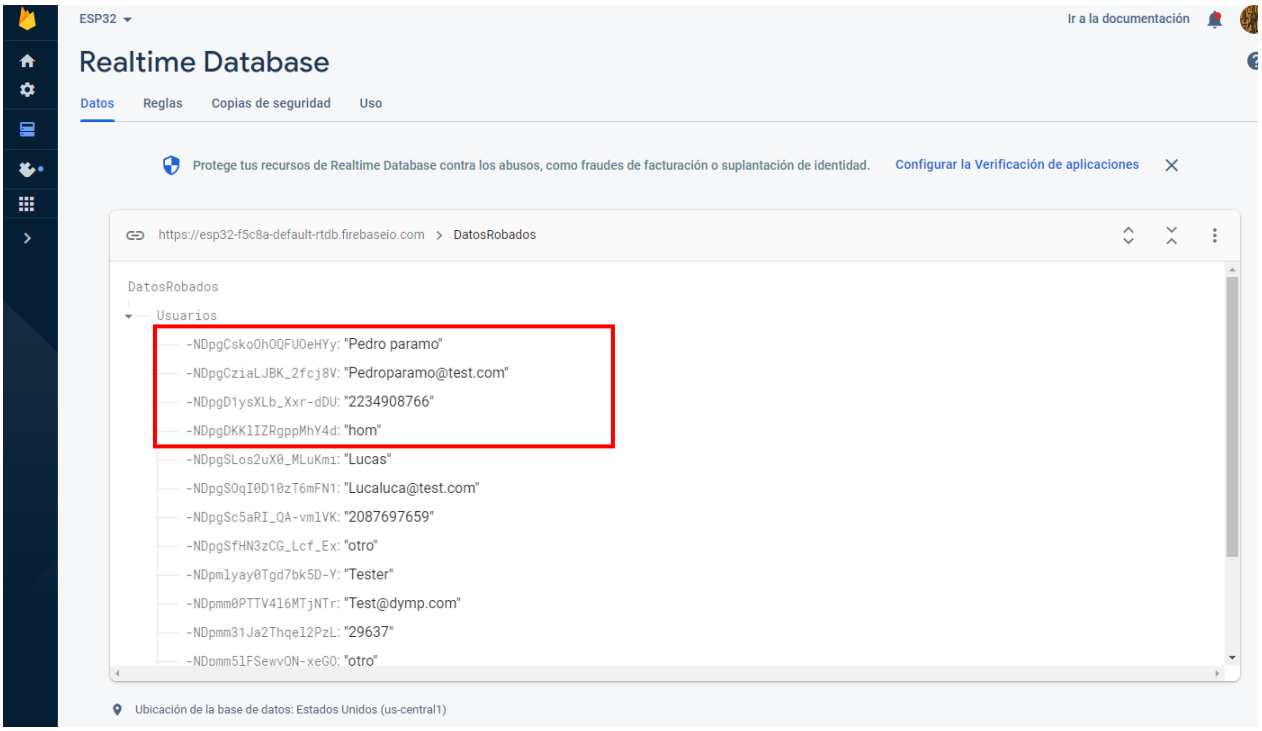

Figura 24. Captura de pantalla de la consola de Firebase (Fuente: Captura de pantalla).

En conformidad con la figura 24, se muestra que existen varios registros en la base datos, estos corresponden a varios ejercicios realizados con el Rogue AP, estos datos fueron agregados a la base de datos en el orden de los siguientes datos: nombre, correo, teléfono y género; por lo que cada secuencia de datos se idéntica por el dato correspondiente al nombre, tal y como se resalta a través del recuadro rojo.

Por consiguiente, recapitulando los datos ingresados para esta prueba, se muestran en la siguiente figura:

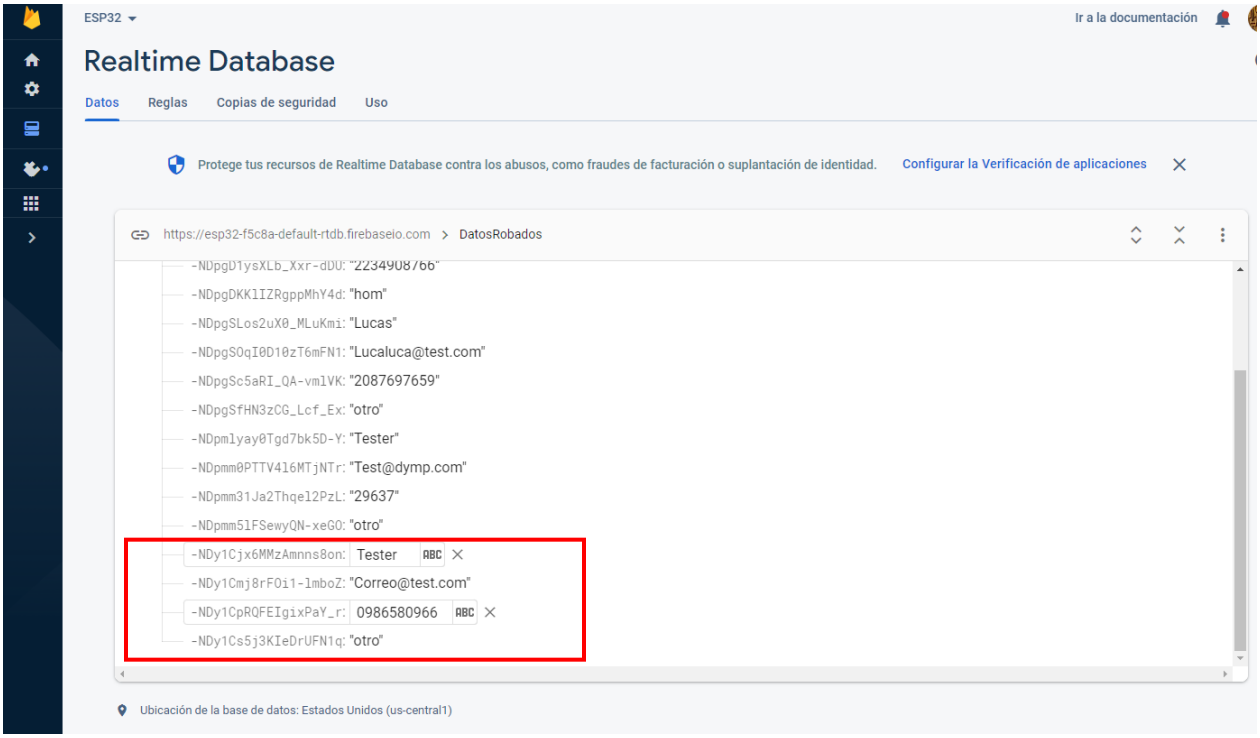

Figura 25. Captura de pantalla de los datos ingresados durante ejercicio (Fuente: Captura de pantalla).

Conforme a lo expuesto en el ejercicio, en la figura 25 se muestran dichos datos, resaltados mediante el recuadro rojo.

6.2 Contramedida defensiva propuesta

Una vez efectuado el ataque, se propone una implementación para contrarrestar este tipo de ataques. La contramedida se construye a través del ESP32, pero en una configuración diferente, en donde la tarjeta sea programada para desempeñar funciones de un escáner de red y de esta forma ayude al administrador de la red a detectar incidencias como la anteriormente desarrollada.

A continuación, se presenta un diagrama de su implementación:

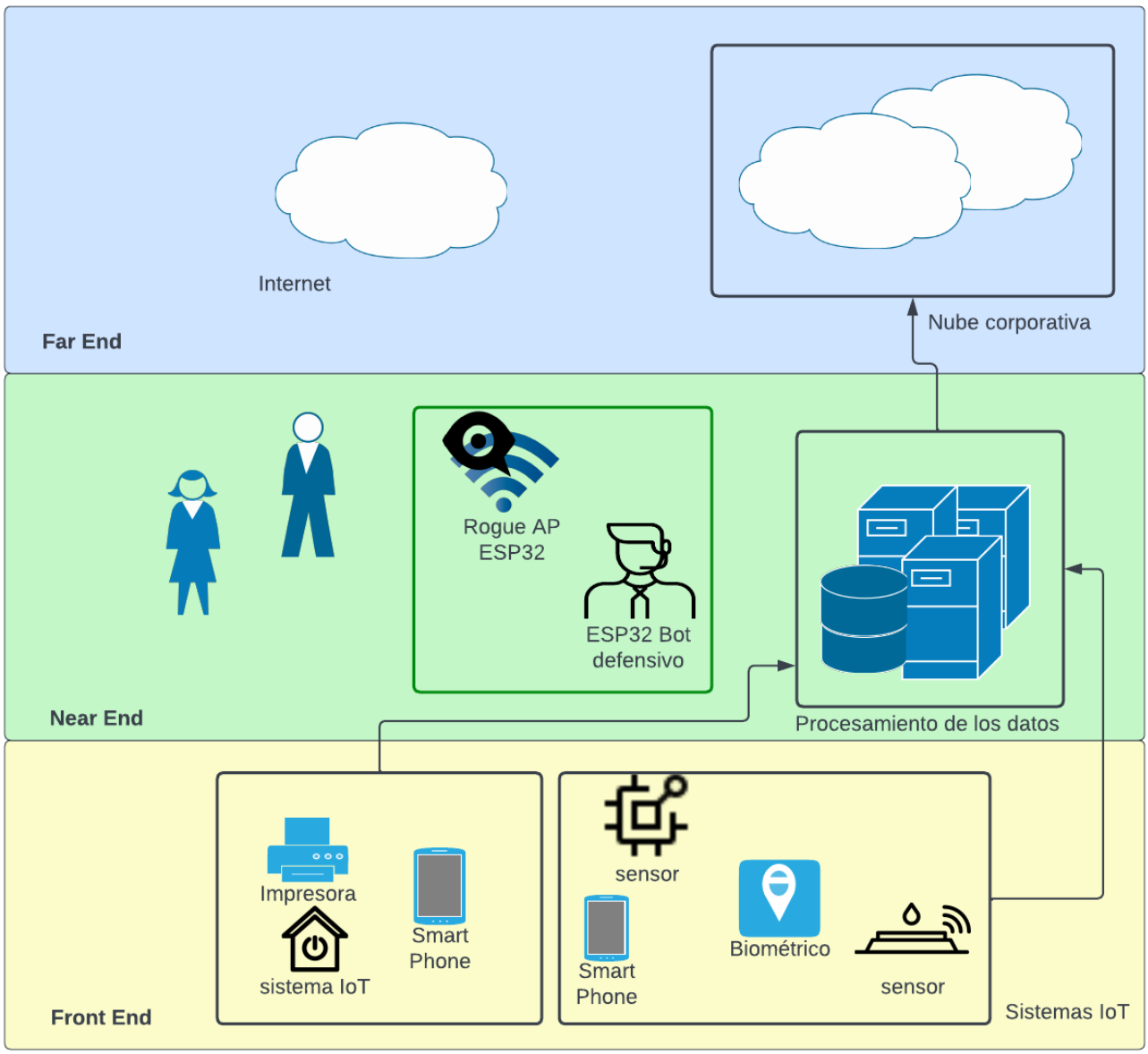

Figura 26. Diagrama de implementación de contramedida propuesta para combatir un Rogue AP (Fuente: diagrama de elaboración propia).

Debido a que el ataque es implementado ocurre en el Neard End, se propone el despliegue de un bot que efectué operaciones de escaneo de redes, es decir, un bot con la función de realizar escaneos mediante tecnología WiFi para detectar las redes inalámbricas que están operativas en esta capa y de esta forma ayudar al administrador a detectar redes inalámbricas maliciosas que pudiesen ser motivo de alerta. En correspondencia con la figura 26, la propuesta se trata de un ESP32 configurado como un bot que alerte al administrador sobre la presencia de una red maliciosa que esté operativa, de esta forma, el administrador de la ciberseguridad podrá efectuar las acciones correspondientes.

De acuerdo con el fabricante kaspersky, un bot se puede definir como un programa con la capacidad de efectuar tareas repetitivas o automatizadas; estas tareas se definen a través de código y se ejecutan de forma automatizada (2022b). Por lo tanto, se pueden definir las siguientes condiciones:

Para implementar esta estrategia fue necesario definieron lo siguientes requisitos:

1. Configurar la ESP32 como un bot que sea capaz de interactuar con el administrador.
2. El usuario debe ser de enviarle instrucciones al bot vía remota.
3. El bot debe responder a las instrucciones del administrador y enviarle la información correspondiente.

En conformidad con los requisitos descritos, se optó por implementar un bot basado en una plataforma de mensajería instantánea Telegram. Esta decisión fue tomada debido a que era necesario una plataforma gratuita y accesible para establecer comunicación con un posible administrador. Por otra parte, Telegram cuenta con soporte para la gestión de APIs. Como lo menciona la plataforma, el *Bot API* es una plataforma de desarrollo para desarrollar bots basados en el protocolo HTTP compatibles con Telegram (Telegram, SA).

6.2.1 Implementación de la contramedida defensiva

Para la implementación del bot primero fue necesario crear un bot a través de la plataforma Telegram, puntualmente a través del BotFather:

Figura 27. BotFather (Fuente: captura de pantalla).

En la figura 27 se muestra la herramienta BotFather, para crear bots de manera nativa en la plataforma Telegram, de tal forma, que cuando es necesario crear uno se debe acudir a esta entidad para su creación y registro:

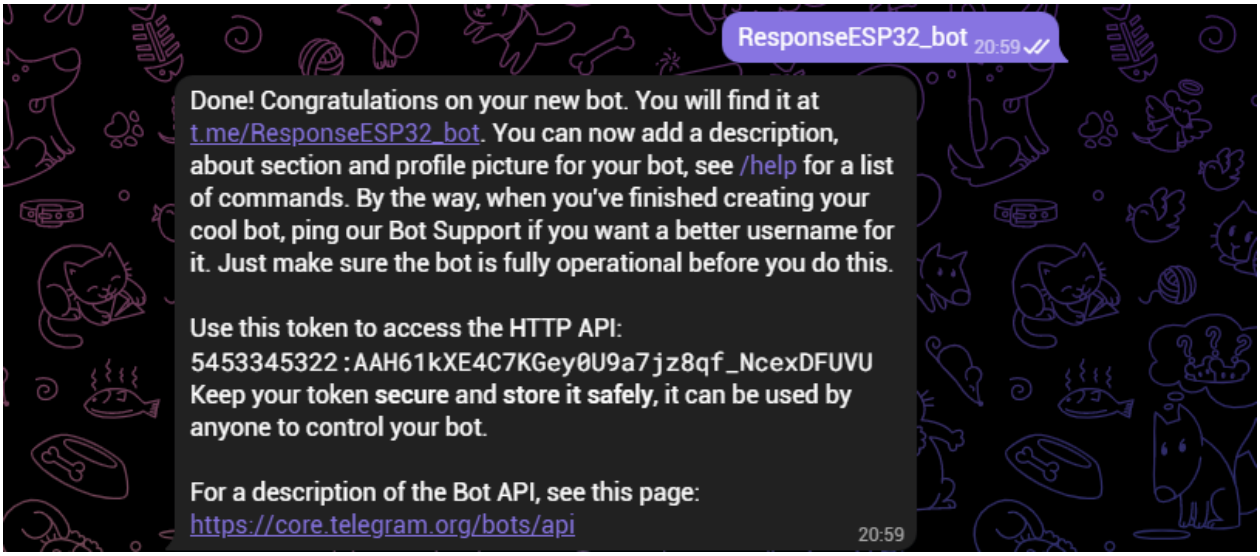

Figura 28. Registro de bot a través del BotFather (Fuente: captura de pantalla).

Como se muestra en la figura 28, mediante el registro con el BotFather, se creó el bot: *ResponseESP32_bot*, el cual, responde a las directrices del BotFather. Un aspecto a resaltar de este proceso, es que el primer dato que se proporciona al crear un bot, es el token: *5453345322:AAH61kXE4C7KGey0U9a7jz8qf_NcexDFUVU*. Mediante esta clave o token, se realiza la autenticación para conectar con la API de Telegram encargada de gestionar los bots, y de esta forma habilitar su funcionamiento en la plataforma.

Entre otras funcionalidades, el BotFather es capaz de ofrecer funciones de administración del bot creado, como es el caso de definir comandos, revocar el token, establecer descripciones, eliminar el bot, incluir el bot en grupos e incluso establecer una foto de perfil para el bot (Figura 29):

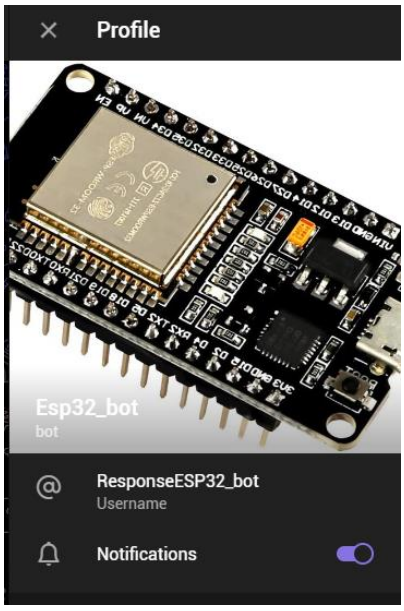

Figura 29. bot ResponseESP32 (Figura: captura de pantalla).

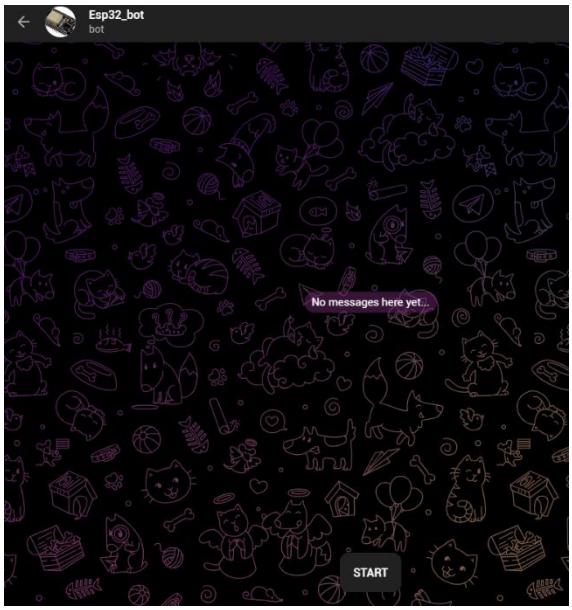

Figura 30. Interfaz de chat del bot (Figura: captura de pantalla).

La administración de los bots creados es directamente con el BotFather o mediante un chat propio con el bot, como si se tratase de otro un usuario de la plataforma, como se aprecia en la figura 29. De igual manera, la forma de interactuar con él es a través de un chat, como se denota en la figura 30. Por lo tanto, el medio de comunicación del administrador hacia el bot para comandar órdenes y recibir datos de este, es mediante un chat, el cual podrá efectuarse a través de la aplicación para dispositivos móviles o mediante la plataforma web de Telegram.

Puesto que ya fue cubierta la parte correspondiente a la creación del bot, corresponde mostrar el desarrollo de la programación de la ESP32:

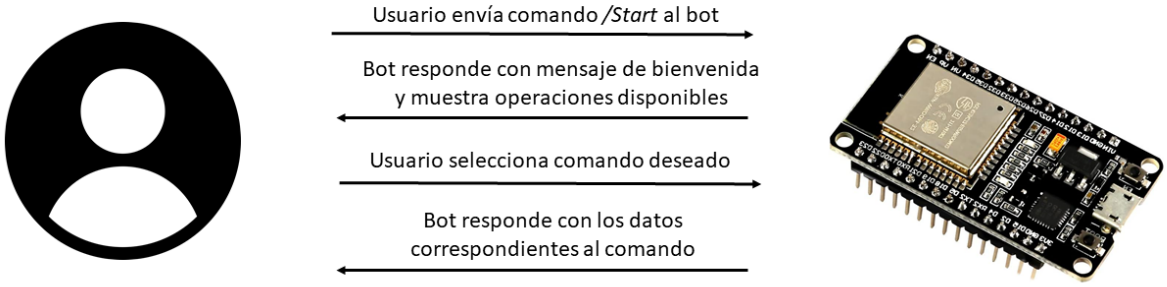

Figura 31. Diagrama de interacción entre el administrador y la ESP32 como bot defensivo (Fuente: diagrama de elaboración propia).

La programación efectuada sobre la ESP32 responde a la interacción descrita en la figura 31, donde se describe que el administrador selecciona un comando disponible, en este caso los comandos disponibles programados fueron:

- **/Mostrar_redes**: Comando encargado de realizar un escaneo de todas las redes WiFi-disponibles dentro del rango de operación del ESP32.
- **/5_redesActivas:** Comando encargo de realizar un escaneo de las redes WiFi-disponibles, únicamente sí existen más de 5 redes activas bien conocidas por el administrador.

Los dos comandos anteriormente descritos fueron definidos de acuerdo con dos criterios:

1. El administrador debe efectuar un escaneo de las redes para conocer que redes se encuentran operativas, de esta forma podrá comprobar que todo está en orden.
2. Sí el administrador conoce el número de redes que se encuentran operativas (en este caso, a manera de ejemplo son 5 redes), entonces, debe seleccionar el segundo comando, de esta forma, se ejecuta el análisis de la red y, en caso de que existan más de 5 redes operativas, se muestran los resultados de dicho análisis.

De tal forma que se efectuó la siguiente programación:

```
#include <UniversalTelegramBot.h>
```

Extracto de código 5: Librería Universal TelgramBot

Para efectuar la programación de la tarjeta se utilizó la librería Universal- Arduino-Telegram-Bot, basada en el framework Arduino y creada por el usuario Brian Lough, la cual es compatible con la API oficial de bots creada por Telegram (2021). Por otra parte, se adaptó un ejercicio clásico de la librería WiFi (anteriormente mostrada): ***WiFiScan***.

Por lo cual, se tienen los siguientes fragmentos de código:

1. Configuración de la ESP32 y conexión a la API bot de Telegram

```
WiFi.mode(WIFI_STA);
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

WiFiClientSecure secured_client;
UniversalTelegramBot bot(BOT_TOKEN,
secured_client);
```

Extracto de código 6: Definición del ESP32 en modo de operación estación y conexión a la API de Telegram, a través del token otorgado por el BotFather.

2. Método para procesar los mensajes provenientes del administrador a través del chat de Telegram

```
void handleNewMessages(int numNewMessages)
{
  for (int i = 0; i < numNewMessages; i++)
  {
    String chat_id = bot.messages[i].chat_id;
    String text = bot.messages[i].text;
}}
```

Extracto de código 7: Definición de método para almacenar el contenido de los mensajes del administrador, en este caso en una variable *text,* para posteriormente ser procesada.

3. Definición de comandos disponibles para el administrador: *Comando /Start*

```
if (text == "/start"){
      String welcome = "Bienvenido al bot ESP32, mi función es
hacer escaneos de redes inalambricas, elige la operación que
desees , " + from_name + ".\n";
      welcome += "Elige alguna de las siguientes opciones.\n\n";
      welcome += "/Mostrar_redes : para ver que SSID estan en
operativo\n";
      welcome += "/5_redesActivas : para recibir por mensaje sí hay
más de 5 redes activas\n";
      bot.sendMessage(chat_id, welcome);
    }
```

Extracto de código 8: El comando /Star es el encargado de inicializar el bot, mostrando el menú principal donde se encuentran los demás comandos disponibles.

4. Definición de comandos disponibles para el administrador: Comando /*Mostrar_redes*

```
if (text == "/Mostrar_redes"){
  n = WiFi.scanNetworks();
  Serial.println("scan done");
          if (n > 0) {
              Serial.print(n);
              Serial.println(" networks found");
              for (int i = 0; i < n; ++i) {
              delay(10);
                  bot.sendMessage(chat_id, WiFi.SSID(i));
              delay(10);
              }
           }
  }
```

Extracto de código 9: El comando ejecuta un escaneo de las redes y posteriormente envía los SSIDs de las redes registradas a través del chat establecido con el administrador.

5. Definición de comandos disponibles para el administrador: Comando /5_redesActivas

```
if (text == "/5_redesActivas"){
        while(p){
        n = WiFi.scanNetworks();
        Serial.println("scan done");
             if (n > 5) {
                 Serial.print(n);
                 Serial.println(" networks found");
                 for (int i = 5; i < n; ++i) {
                 delay(10);
                    bot.sendMessage(chat_id, WiFi.SSID(i));
                 delay(10);
                  }p= false;
                 } }
   }
```

Extracto de código 10: El comando fue desarrollado partiendo de la idea de que el administrador conoce le número de redes operativas (en este ejemplo, cinco), y por lo tanto efectúa este comando con la finalidad de obtener registros SSIDs únicamente sí existen más de 5 redes operando.

6.2.2 Evidencias de la Implementación de la contramedida defensiva

Por parte del administrador se observa lo siguiente:

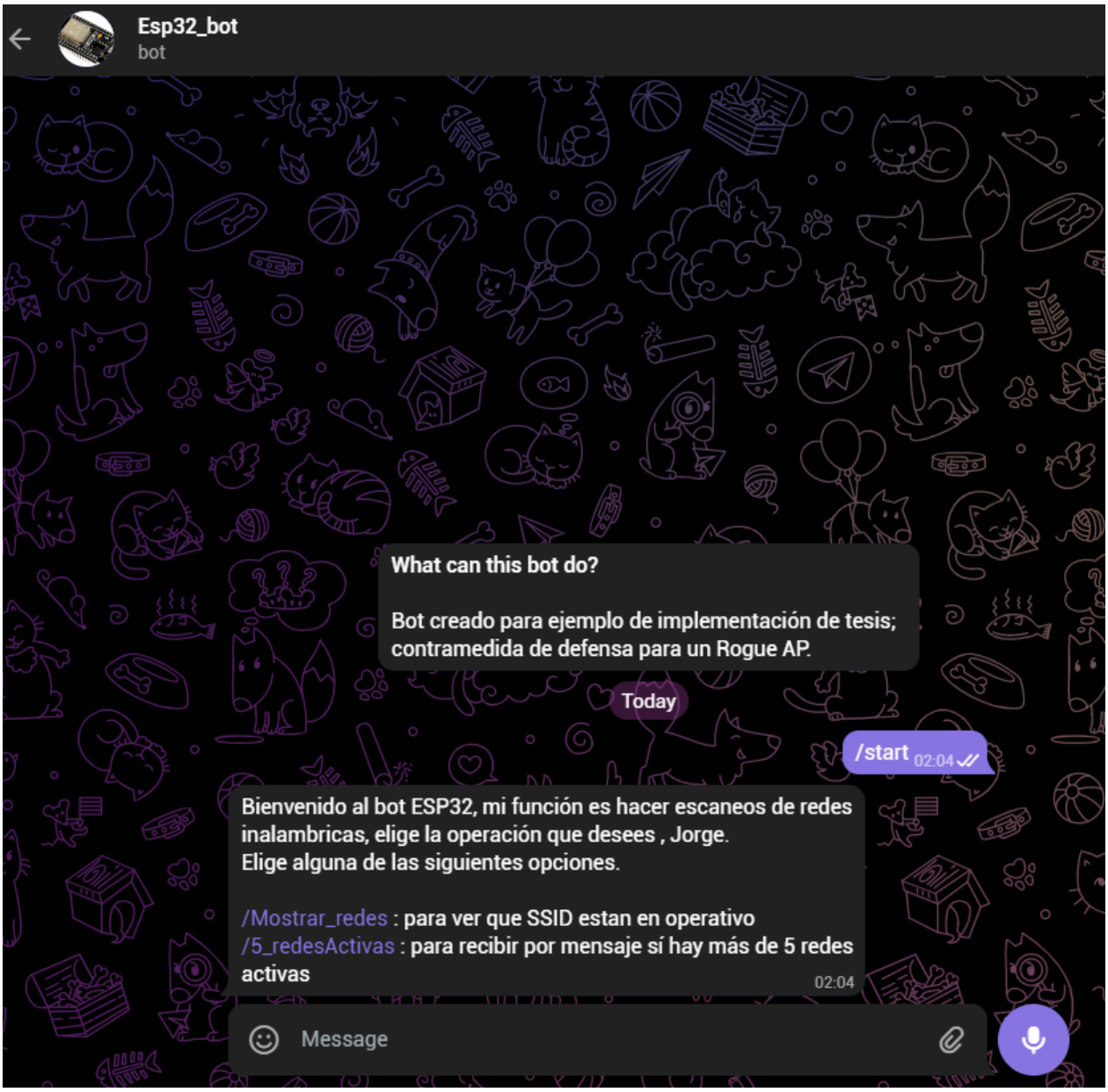

Figura 32. Pantalla del administrador, vía chat de Telegram con bot ESP32 (Fuente: captura de pantalla).

Se denota en la figura 32 que al interactuar con el bot mediante el comando */start*, el bot muestra un mensaje de bienvenida e indica las operaciones disponibles. En este caso, las correspondientes a */Mostrar_redes* y */5_redesActivas.*

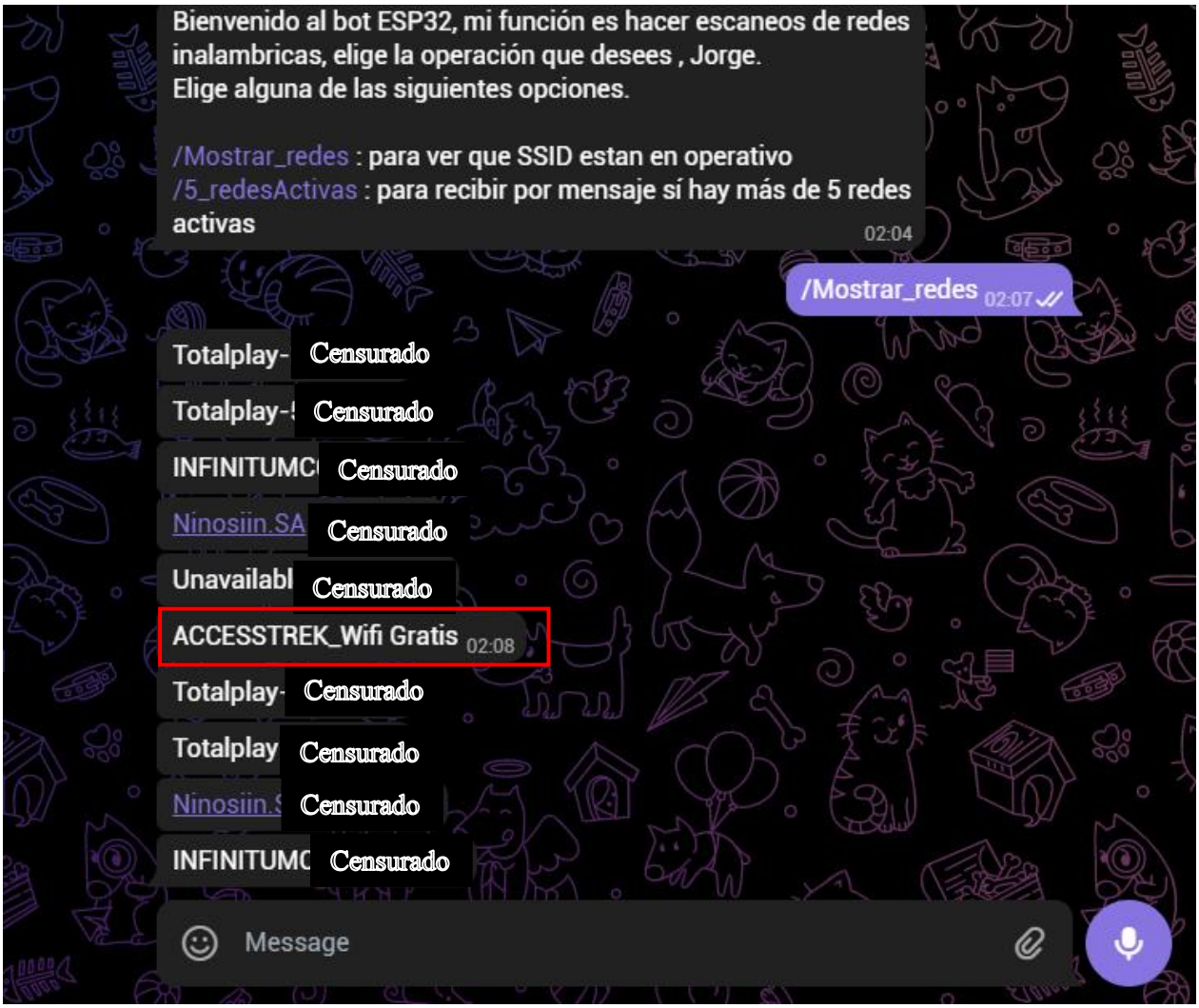

Figura 33. Ejecución de operación /Mostrar_redes (Fuente: captura de pantalla).

Como se observa en la figura 33 los resultados del comando indican al administrador todas las redes inalámbricas disponibles, entre ellas la red correspondiente (remarcada con un recuadro rojo) al Rogue AP implementado, es decir, *"ACCESSTREK_Wifi GRATIS"*.

Por otra parte, al ingresar el comando /5_redesActivas, el bot mostrará aquellas redes que se encuentran activas, únicamente sí existen 5 redes activas; por tanto, el comando tiene la intención de mostrar aquellas redes que no están definidas. Sin embargo, este criterio no es preciso, ya que es posible que el escaneo de redes no coincida con la activación del Rogue AP, dando el siguiente registro:

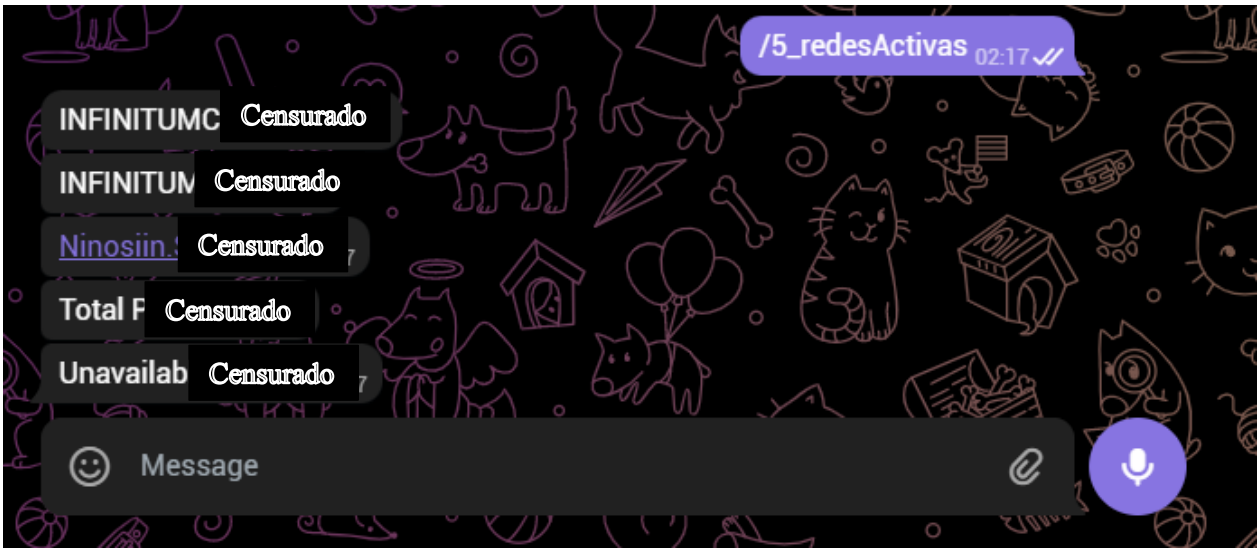

Figura 34. Resultados de comando /5_redesActivas (Fuente: Captura de pantalla).

En la figura 34 que el comando /5_redesActivas no muestra el SSID creado para el ataque, *"ACCESSTREK_Wifi GRATIS".* A través de este ejemplo, se pretende mostrar las limitaciones que existen al momento de programar con el ESP32, por lo cual, se debe considerar que la programación en este dispositivo tiene alcances bien definidos, por su supuesto, es posible que existan formas más eficientes de desarrollar este ejercicio, sin embargo, eso requiere de acciones concretas, como crear nuevas librerías o alejarse del framework que ofrece Arduino, lo cual escapa de los objetivos del proyecto.

6.3 Análisis de los resultados

Con base en los resultados anteriormente mostrados, se hacen las siguientes observaciones:

6.3.1 Respecto a sus capacidades ofensivas:

Tabla 3
Análisis del ataque ejecutado

Técnica empleada	ROGUE AP	
Función	Robar datos de identidad del usuario	
Canal	Inalámbrico	
Condiciones de éxito	**Condición 1**	El esp32 debe ser lanzado exitosamente y mantenerse a la escucha
	Condición 2	El ESP32 debe mantener una conexión a internet

	Condición 3	El usuario debe conectarse al SSID generado
	Condición 4	El usuario debe llenar el formulario del portal cautivo
	Condición 5	El usuario debe pulsar el botón para acceder al servicio
Nivel de efectividad observado	Dependiente del usuario	
Capacidad de manejo de usuarios	Limitado a 8 usuarios máximo	
Capacidad almacenamiento de datos	Limitado a la capacidad de la base de datos en la nube	
Tiempo de operación posible	Limitado a la fuente de poder, 5V	

Nota: De elaboración propia

A tenor de la tabla 3, existen una serie de condiciones para efectuar el ataque, a la vista de ellos, se establecen los siguientes puntos:

- **Desventajas**

  1. La capacidad ofensiva de la ESP32 depende de sus prestaciones de hardware, por ejemplo, para el caso el ROGUE AP, solo puede atender un máximo de 8 clientes al mismo tiempo, lo cual lo convierte en una técnica con límites de recolección bien definidos.

  2. Debido a que es necesario garantizar una fuente de acceso a internet para el dispositivo, si este servicio es retirado, la técnica perderá efectividad.

  3. En suma al punto anterior, la conexión con la base de datos por parte de la tarjeta se convierte en un factor inconveniente, ya que, si la tarjeta pierde acceso a la base de datos, entonces, no existe forma de registrar los datos robados.

  3. El ROGUE AP depende del facto ingeniería social, por lo cual, su tasa de éxito depende de aquello usuarios que no estén familiarizados con técnicas de ingeniería social.

  4. Es posible que los usuarios ingresen datos falsos, por lo cual el ROGUE AP es susceptible a recibir información falsa o no relevante para su propósito.

  5. Gracias a algunas pruebas de laboratorio, realizadas con soluciones tipo Endpoint mobile (para smartphones), se detectó que el ROGUE AP no es efectivo contra dispositivos que tengan soluciones instaladas, se debe aclarar que solo se hizo esta prueba con una solución.

Al mismo tiempo, en consideración a que se trata de un ataque implementado en un

microcontrolador con prestaciones limitadas se debe considerar lo siguiente:

- **Ventajas**

  1. Dado que se trata de una tarjeta que tiene un costo menor a 200 MXN, el hecho que permita realizar funciones de cliente y servidor, la convierte en un dispositivo potente, capaz de realizar operaciones complejas, como actuar como un ROGUE AP.

  2. Debido a que la ESP32 puede ser configura como un ROGUE AP únicamente con librerías y frameworks oficiales la convierte en una herramienta flexible para usuarios con poco expertis, brindando la posibilidad de desarrollar ataques sin la necesidad de requerir conocimientos avanzados en programación, ni utilizar herramientas de hacking avanzadas.

  3. Dado que se trata de un dispositivo IoT, es difícil que el público en general sospeche que una tarjeta de este tipo pueda desempeñarse como un AP, por cual, aumenta el factor de credibilidad ante las victimas al momento de visualizar una red WiFi disponible.

  4. En consideración a que la ESP32 puede acceder a servicios de nube, como fue el caso de la base de datos Firebase, la convierte en una herramienta de ataque capaz de solventar sus carencias, como es caso, de la falta de características de almacenamiento, en este sentido, es posible compensar otras carencias, como poder de cómputo, a través de la conexión de servicios a la nube.

  5. Dada la compatibilidad de la tarjeta con HTML la convierte en una herramienta capaz de desempeñar funciones de ingeniería social, debido a que puede replicar formatos comunes en páginas web.

  6. Debido a su tamaño, es posible transportar la ESP32 con suma facilidad, siendo posible transportarla en un bolsillo y plantar la tarjeta en algún punto estratégico para que desempeñe su función de ROGUE AP sin despertar sospechas.

Conforme a los aspectos ya descritos se llega a la conclusión que el Rogue AP ESP32 es un método de ataque aceptable de acuerdo a sus limitaciones, es decir, únicamente bajo el argumento de que se trata de un dispositivo con pocos recursos tecnológicos. La técnica realiza un trabajo aceptable y potencialmente exitoso. Sin embargo, en comparación con otros medios y dispositivos, se trata de un ejercicio poco eficiente, ya que existen herramientas más eficaces (y especializadas) para llevar a cabo estas tareas; por consiguiente, esta implementación únicamente responde a métodos demostrativos y didácticos.

6.3.1 Respecto a sus capacidades defensivas:

Tabla 4
Tabla descriptivita de método de defensa ejecutado

Técnica empleada	Bot - Scaner de redes	
Función	Monitorear la red y notificar resultados	
Canal	Inalámbrico	
Condiciones de éxito	**Condición 1**	El esp32 debe ser desplegado exitosamente y mantenerse a la escucha
	Condición 2	El ESP32 debe mantener una conexión a internet para establecer comunicación con la API de Telegram
	Condición 3	EL ESP32 debe recibir un comando por parte del administrador
Nivel de efectividad observado	Dependiente de la comunicación con la API de Telegram	
Tiempo de operación posible	Limitado a la fuente de poder, 5V	

Nota: De elaboración propia

En la tabla 4 se expone que existen una serie de condiciones para efectuar el método defensivo, a la vista de ellos, se establecen los siguientes puntos:

- **Aspectos negativos**

  1. Debido a que se requiere que el dispositivo se encuentre en desplegado en un punto estratégico, el éxito del método es arbitrario, puesto que el método defensivo puede ser desplegado en una zona diferente a la del ROGUE AP.

  2. El método defensivo requiere de una conexión a internet para establecer comunicación con la API de Telegram, en este caso, sí la tarjeta pierde acceso a internet, la efectividad del método se ve comprometida.

  3. Durante las pruebas ejecutadas, se observó que existen ciertos errores sincronización en los mensajes que se envía la ESP32 a Telegram, por lo cual, los datos mostrados en el chat del bot no son del todo confiables.

  4. La detección de las redes por parte de la ESP32 depende de la disponibilidad de estas, en este sentido, sí al momento del escaneo el ROGUE AP no se encontraba activo, puede generar resultados negativos falsos.

  5. Durante las pruebas desarrolladas, se contempló la implementación un escáner de red en escucha permanente, para detectar cualquier tipo de cambio en las redes operativas, sin embargo, debido a que existen errores de sincronización con la API de Telegram, esta función fue descartada.

No obstante, se rescatan los siguientes puntos favorables:

- **Aspectos positivos**

  1. Gracias a que el ESP32 puede establecer conexión con servicios de API para interactuar con usuarios, la convierte en una tarjeta flexible y multipropósito en el contexto del IoT.

  2. La versatilidad que posee para crear instrucciones y ser compatible con diversos dispositivos, la convierte un elemento a tomar en cuenta para monitorear procesos de forma automatizada.

  3. Dado que la ESP32 puede ser alcanzada a través de la red local, la convierte en una herramienta de análisis remoto eficiente para un administrador, el cual puede estar al pendiente de diferentes espacios del sistema de forma remota.

  4. Acorde a las capacidades de la ESP32 para ser reprogramada, se pueden establecer múltiples directrices y funciones de monitoreo, de acuerdo con los requerimientos del administrador.

  5. Como se observó durante su implementación, el ROGUE AP fue detectado por la propuesta, por lo cual, se puede tomar en cuenta como una medida de monitoreo factible.

En consecuencia, la propuesta de implementación es una técnica factible para detectar ataques de este tipo, sin embargo, se debe considerar que no es la más eficiente, y, por lo tanto, se deben considerar otras medidas de implementación. No obstante, se debe recalcar que como herramienta de monitoreo para escenarios no críticos ofrece al administrador las bondades suficientes para obtener datos e información relevante en un escenario donde se requiriere obtener información de forma remota.

6.4 Análisis de los resultados respecto a otras literaturas

Como se mencionó en la sección *4.2 Estado del arte y literatura al respecto,* existen trabajos académicos que documentan la implementación de las capacidades ofensivas de la ESP32. En este caso, se recuperan los trabajos elaborados por Richard Stehlík y la colaboración desarrollada por Bryan Pearson, Lan Luo, Cliff Zou, Jacob Crain, Yier Jin, y Xinwen Fu.

En el caso de Stehlík, se trata de una serie de laboratorios desarrollados con métodos no nativos, en su artículo “Wi-Fi attacks using ESP32” (SA), muestra la ejecución de los ataques: deauthentication attack, WPA/WPA2 handshake capture, además de documentar otros ataques existentes como es caso de PMKID capture and brute-force attack, WPS PIN attack KRACK attack. A través de este trabajo se puede corroborar la potencia de la ESP32 cuando se utilizan métodos no nativos, específicamente, a través de una programación basada en Python. De igual manera, el autor hace énfasis sobre las limitaciones de la tarjeta, las cuales son naturales debido a

sus características, a pesar de ello, se tratan de métodos efectivos tomando en cuenta estas restricciones. Por otra parte, en el caso de la colaboración plasmada en el artículo “Building a Low-cost and State-of-the-art IoT Security Hands-on Laboratory” (2020), se desarrolla la propuesta de implementar un laboratorio de ciberseguridad, con fines didactas. El laboratorio se basa en el desarrollo de las capacidades ofensivas de la ESP32; puntalmente, el desarrollo de estos ejercicios se realiza a través de métodos nativos y no nativos, por lo cual, se exploran las propiedades de cifrado del dispositivo, funciones de su procesador e integración con otros módulos electrónicos.

Sin embargo, en consideración a los resultados obtenidos en estos artículos, (los cuales son excelentes), no se contempló el desarrollo de métodos de ingeniería social, los cuales, han demostrado ser una de las técnicas más efectivas en la actualidad. Por otra parte, el desarrollo mostrado en estos trabajos corresponde a la exploración de propiedades específicas de la tarjeta, así como procedimientos que requieren cierto grado de conocimiento, por lo cual, en contra posición del presente trabajo, se exploran técnicas que pueden ser desarrolladas por usuarios casuales y que pueden ser ejecutadas mediante técnicas disponibles en la documentación oficial, complementando así el alcance de implementación que posee la ESP32 y brindando un panorama general de los proyectos relacionados a la ciberseguridad que se pueden desarrollar a través de métodos nativos y no nativos.

7. Conclusiones

Las propiedades que posee la ESP32 permiten el desarrollo de diversos métodos de ataque, gracias, a la facultad que posee de alteración sus propiedades de fábrica, sin embargo, en correspondencia a todo lo expuesto anteriormente, estas implementaciones se ven limitadas a las mismas características de la tarjeta. Partiendo de esta consideración, es necesario contemplar que existen herramientas más eficientes para ejecutar ataques, como es el caso de las técnicas y herramientas disponibles en sistemas especializados como *kali Linux*, por ello, las implementaciones de estos métodos de ataque basados en la ESP32 deben efectuarse considerando sus límites computaciones y de hardware.

En otro tenor, dado el material documentado es posible realizar más implementaciones, lo cual abre paso a futuros trabajos. La versatilidad de la tarjeta para adaptarse a diversas herramientas es abrumadora, existiendo incluso, librerías nativas compatibles con servicios de nube como AWS. Sin embargo, para los alcances del presente trabajo se establece que se desarrollaron con éxito los objetivos, demostrando así las capacidades ofensivas y defensivas de la ESP32, así como sus limitaciones.

En última instancia, a manera de reflexión, es preciso tomar conciencia acerca de los peligros que representan los dispositivos IoT, en especial aquellos con la capacidad para ejecutar código, pues como se mencionó, existe una cultura general acerca de los riesgos a la ciberseguridad que existen en internet, pero poco se habla de los riesgos que representan los dispositivos IoT. Después de todo, ¿quién podría imaginarse que, al momento de conectarse a una red WiFi gratuita, en realidad se trate de un dispositivo IoT malicioso?, y, al mismo tiempo, ¿vale la pena ingresar nuestros datos personales en algún portal cautivo, a cambio de obtener acceso a internet gratuito?, naturalmente, quizás se trate de algún servicio confiable, pero, no es posible tener la certeza acerca de quien accede a estos datos, y, en el peor de casos, si realmente se trata de una red legitima.

Referencias bibliográficas

Lista de Figuras

Figura 24. Captura de pantalla de la consola de Firebase.
Fuente: Captura de pantalla

Figura 25. Captura de pantalla de los datos ingresados durante ejercicio.
Fuente: Captura de pantalla

Figura 26. Diagrama de implementación de contramedida propuesta para combatir un Rogue AP
Nota: Diagrama de elaboración propia

Figura 27. Botfather
Fuente: Captura de pantalla

Figura 28. Registro de bot a través del BotFather
Fuente: Captura de pantalla

Figura 29. bot ResponseESP32
Fuente: Captura de pantalla

Figura 30. Interfaz de chat del bot
Fuente: Captura de pantalla

Figura 31. Diagrama de interacción entre el administrador y la ESP32 como bot defensivo
Fuente: De elaboración propia

Figura 32. Pantalla del administrador, vía chat de Telegram con bot ESP32
Fuente: Captura de pantalla

Figura 33. Ejecución de operación /Mostrar_redes
Fuente: captura de pantalla

Figura 34. Resultados de comando /5_redesActivas
Fuente: Captura de pantalla

Tablas

Extractos de código

Extracto de código 1: Configuración del ESP32 como AP.
Nota: Funcionamiento del ESP32 como AP.

Extracto de código 2: Configuración del portal cautivo.
Nota: Funcionamiento del ESP32 como Portal cautivo.

Extracto de código 3: Configuración de servidor para manejar las peticiones de las víctimas.
Nota: Funcionamiento del ESP32 como servidor que atiende peticiones de clientes.

Extracto de código 4: Configuración como cliente de una base de datos en la nube.
Nota: Funcionamiento del ESP32 como un cliente que mantiene comunicación con una base de datos.

Extracto de código 5: Librería Universal TelgramBot.
Nota: Librería de código

Extracto de código 6: Definición del ESP32 en modo de operación estación y conexión a la API de Telegram, a través del token otorgado por el BotFather.
Nota: Configuración de la ESP32 y conexión a la API bot de Telegram.

Extracto de código 7: Definición de método para almacenar el contenido de los mensajes del administrador, en este caso en una variable text, para posteriormente ser procesada.
Nota: Método para procesar los mensajes provenientes del administrador a través del chat de Telegram.

Extracto de código 8: El comando /Star es el encargado de inicializar el bot, mostrando el menú principal donde se encuentran los demás comandos disponibles.
Nota: Definición de comandos disponibles para el administrador: Comando /Start.

Extracto de código 9: El comando ejecuta un escaneo de las redes y posteriormente envía los SSIDs de las redes registradas a través del chat establecido con el administrador.
Nota: Definición de comandos disponibles para el administrador: Comando /Mostrar_redes.

Extracto de código 10: El comando fue desarrollado partiendo de la idea de que el administrador conoce le número de redes operativas (en este ejemplo, cinco), y por lo tanto efectúa este comando con la finalidad de obtener registros SSIDs únicamente sí existen más de 5 redes operando.
Nota: Definición de comandos disponibles para el administrador: Comando /5_redesActivas.

Referencias

Abad Domingo, A. (2013). Redes locales. McGraw-Hill España. https://elibro.bibliotecabuap.elogim.com/es/lc/bibliotecasbuap/titulos/50228

Adell, J. (noviembre, 1995). La navegación hipeertextual en el World-Wide Web: implicaciones para el diseño de materiales educativos. [Comunicación presentada]. II Congreso de Nuevas Tecnologías de la Información y Comunicación para la Educación. Palma de Mallorca, España. https://www.um.es/innova/OCW/diseno_y_evaluacion_materiales_didacticos/mpaz/utilidades/pdf/18.pdf

Ananay Arora. (SA). Preventing wireless deauthentication attacks over 802.11 Networks. Obtenido de: https://arxiv.org/ftp/arxiv/papers/1901/1901.07301.pdf

Andrade, R. O., & Gualli, T. G. (2019). Los retos de la ciberseguridad en ciudades inteligentes. Ciudades Inteligentes. https://sistemas.acis.org.co/index.php/sistemas/article/view/29

Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong & Jason P. Jue. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. Journal of Systems Arquitecture. (98), 289-330. https://doi.org/10.1016/j.sysarc.2019.02.009

Azure. (2022). Protocolos y tecnologías de IoT: Guía de protocolos y tecnologías de IoT. Obtenido de: https://azure.microsoft.com/es-es/solutions/iot/iot-technology-protocols/

Barrio Andrés, M. (2018). Internet de las cosas. Editorial Reus. https://elibro.bibliotecabuap.elogim.com/es/lc/bibliotecasbuap/titulos/121519

Belcic, I. (2022). ¿Qué es un sniffer y cómo puede protegerse? Avast Academy. Obtenido de: https://www.avast.com/es-es/c-sniffer

Ben Lutkevich (2019). Microcontroller (MCU). TechTarget | IotAgenda. Obtenido de: https://internetofthingsagenda.techtarget.com/definition/microcontroller

Borghello, C. (2009). El arma infalible: la Ingeniería Social. ESET Latinoamérica. Obtenido de: http://www.eset-la.com/pdf/prensa/informe/arma_infalible_ingenieria_social.pdf

Brian Lough. (2021). Universal-Arduino-Telegram-Bot. Obtenido de: https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot

Carnut, M., & Gondim, J. (2003). ARP SPOOFING DETECTION ON SWITCHED ETHERNET NETWORKS: A FEASIBILITY STUDY. Obtenido de: https://www.researchgate.net/publication/340389502_ARP_SPOOFING_DETECTION_ON_SWITCHED_ETHERNET_NETWORKS_A_FEASIBILITY_STUDY

Cartuche, J. J., Hernández, D., l., Morocho, R., F., & Radicelli Garcia, C., D. (2020). SEGURIDAD IOT: PRINCIPALES AMENAZAS EN UNA TAXONOMÍA DE ACTIVOS. Hamut'ay. Revista cuatrimestral de divulgación cientifica de la Universidad Alas Peruanas, 7 (1), 51-59 http://dx.doi.org/10.21503/hamu.v7i3.2192.

Castaño Ribes, R. J. (2013). Redes locales. Macmillan Iberia, S.A. https://elibro.bibliotecabuap.elogim.com/es/lc/bibliotecasbuap/titulos/43257

Chai, W. & Irei, A. (2021). Network protocol. TechTarget | SearchNetworking. Obtenido de: https://www.techtarget.com/searchnetworking/definition/protocol

Chhatramani Yadav. (2022). Flashing Esp32 & Esp8266 For Wifi Hacking by Eviltwin & Deauther Attack. Obtenido de: https://www.youtube.com/watch?v=5Jj2OenFJaU

CISCO. (2020). 802.11 Association Process Explained. Meraki. CISCO. Obtenido de: https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/802.11_Association_Process_Explained

Cisco. (2022a). What Is Edge Computing? CISCO. https://www.cisco.com/c/en/us/solutions/computing/what-is-edge-computing.html

Cisco. (2022b). ¿Qué es un access point? CISCO. Obtenido de: https://www.cisco.com/c/es_mx/solutions/small-business/resource-center/networking/what-is-access-point.html#~tipos-de-access-points

CompTIA. (SAa). What Is Spoofing? CompTIA. Obtenido de: https://www.comptia.org/content/articles/what-is-spoofing

CompTIA. (SAb). WHAT IS A DDOS ATTACK AND HOW DOES IT WORK? CompTIA. Obtenido de: https://www.comptia.org/content/guides/what-is-a-ddos-attack-how-it-works

Cueva Lovelle, J. M., Rodriguez Molano, J. I., & Montenegro Marin, C. E. (2015). INTRODUCCIÓN AL INTERNET DE LAS COSAS. Redes de Ingeniería, 6. https://doi.org/10.14483/2248762X.8505

Deland-Han, helenclu, v-phoebh, v-lianna & simonxjx. (2021). Explanation of the three-way handshake via TCP/IP. Microsoft. Obtenido de: https://docs.microsoft.com/en-us/troubleshoot/windows-server/networking/three-way-handshake-via-tcpip

Digital Guide IONOS. (2021). El ping de la muerte: uno de los primeros ataques de red. Seguridad. Digital Guide IONOS. Obtenido de: https://www.ionos.mx/digitalguide/servidores/seguridad/ping-de-la-muerte/

Evans, D. (2011). The internet of Things How the Next Evolution of the Internet Is Changing Everything. Cisco Internet Bussiness Solutions Group (IBSG).

https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf
ESPRESSIF. (2022). Wi-Fi API. Obtenido de: https://docs.espressif.com/projects/arduino-esp32/en/latest/api/wifi.html

Espressif Systems (2022a). ESP32. Obtenido de: https://www.espressif.com/en/products/socs/esp32

Espressif Systems. (2021b). ESP32 Series Datasheet. Obtenido de: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Espressif Systems. (2021c). ESP32 Technical Reference Manual. Obtenido de: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

Fresno Chávez, C. (2018). ¿Cómo funciona Internet? Ciudad Educativa. https://elibro.bibliotecabuap.elogim.com/es/lc/bibliotecasbuap/titulos/36728

García, G. (2015). Qué es un Rogue AP y cómo protegernos. Naps Tecnología y educación. Obtenido de: https://naps.com.mx/blog/que-es-un-rogue-ap-y-como-protegernos/

Hak5. (2021). HakByte: Build a Hackable Router with $5 ESP32. [Archivo de video]. YouTube. Obtenido de: https://www.youtube.com/watch?v=41Lymi6rXA8

Harsha, S. (2019). ESP32 Fatal Fury Attack — What you should know. MEDIUM. https://medium.com/swlh/esp32-fatal-fury-attack-what-you-should-know-6aa342acec09

Halfacree, G. (2020). ESP32 Marauder Puts a Bluetooth and Wi-Fi Pen Testing Toolkit in Your Pocket. hackster.io. https://www.hackster.io/news/esp32-marauder-puts-a-bluetooth-and-wi-fi-pen-testing-toolkit-in-your-pocket-32d389f6e66f
Herzog, P. (2010). OSSTMM 3 The Open Source Security Testing Methodology Manual. Contemporary Security Testing and Analysis. INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES. https://www.isecom.org/OSSTMM.3.pdf

HIKVISION. (2018). Hikvision Cybersecurity White Paper. 2018 Hangzhou Hikvision Digital Technology Co., Ltd. Obtenido de: https://www.hikvision.com/content/dam/hikvision/en/brochures/cybersecurity/Hikvision%20Cyber%20Security%20White%20Paper.pdf

Hurtado, J., J. (2021). ESTUDIO COMPARATIVO DE LAS TECNOLOGÍAS ALTERNATIVAS: COMPUTACIÓN EN LA NUBE, COMPUTACIÓN EN EL BORDE Y COMPUTACIÓN EN LA NIEBLA, PARA LAS PYME EN PANAMÁ. UNIVERSIDAD INTERNACIONAL DE CIENCIA Y TECNOLOGÍA INGENIERÍA EN REDES DE COMUNICACIÓN. [PROYECTO DE TRABAJO PARA OPTAR AL GRADO DE LICENCIADO EN INGENIERÍA EN REDES DE COMUNICACIONES
CON ÉNFASIS EN SEGURIDAD, Universidad Internacional de Ciencia y Tecnología]. http://www.idi-unicyt.org/wp-content/uploads/2021/03/Informe-del-Trabajo-de-Grado-de-Juan-Hurado-DEFINITIVA.pdf

IBM. (SAa). ¿What is edge computing? IBM. Obtenido de: https://www.ibm.com/es-es/cloud/what-is-edge-computing.

IBM. (SAb). ¿Qué es la ciberseguridad? IBM. Obtenido de: https://www.ibm.com/mx-es/topics/cybersecurity#:~:text=La%20ciberseguridad%20es%20la%20pr%C3%A1ctica,confidencial%20de%20los%20ataques%20digitales.

IBM. (SA). Cloud computing: Guía completa. Obtenido de: https://www.ibm.com/es-es/cloud/learn/cloud-computing-gbl

IBM. (2021). Protocolo de mensajes de control de Internet (Internet Control Message Protocol). IBM. Obtenido de: https://www.ibm.com/docs/es/aix/7.2?topic=protocols-internet-control-message-protocol

Intel. (2021a). Diferentes protocolos de Wi-Fi y velocidades de datos. Obtenido de: https://www.intel.la/content/www/xl/es/support/articles/000005725/wireless/legacy-intel-wireless-products.html

Intel. (2021b). Comprender la asociación y autenticación de IEEE* 802.11. Asistencia. Intel. Obtenido de: https://www.intel.la/content/www/xl/es/support/articles/000006508/wireless/legacy-intel-wireless-products.html

Internet Society. (2022). Brief history of the internet. Obtenido de: https://www.internetsociety.org/es/internet/history-internet/brief-history-internet/

JUNIPER NETWORKS. (2021). Network DoS Attacks. Attack Detection and Prevention User Guide for Security Devices. JUNIPER NETWORKS. Obtenido de: https://www.juniper.net/documentation/us/en/software/junos/denial-of-service/topics/topic-map/security-network-dos-attack.html

Kanai, A. and Niwa, k. (1999) "Knowledge-management system including viewpoints as an external constraint," IEEE SMC'99 Conference Proceedings. IEEE International Conference on Systems, Man, and Cybernetics pp. (4),147-152. 10.1109/ICSMC.1999.812391

kaspersky. (2022a). ¿Qué es un ataque de día cero?: definición y explicación. Obtenido de: https://latam.kaspersky.com/resource-center/definitions/zero-day-exploit

kaspersky. (2022b). ¿Qué son los bots? Definición y explicación. Obtenido de: https://latam.kaspersky.com/resource-center/definitions/what-are-bots

Kewei Sha, T. Andrew Yang, Wei Wei, Sadegh Davari (2020). A survey of edge computing-based designs for IoT security. Digital Communications and Networks. 6(2), PP: 195-202. https://doi.org/10.1016/j.dcan.2019.08.006

Kochetkova, K. (2016). Cómo evitar que el Internet se caiga. Kaspersky daily. Obtenido de: https://latam.kaspersky.com/blog/attack-on-dyn-explained/7901/

Kristiyanto, Y. & Ernastuti, E. (2020). Analysis of Deauthentication Attack on IEEE 802.11 Connectivity Based on IoT Technology Using External Penetration Test. CommIT (Communication and Information Technology) Journal. 14. 45. Obtenido de: https://www.researchgate.net/publication/343232590_Analysis_of_Deauthentication_Attack_on_IEEE_80211_Connectivity_Based_on_IoT_Technology_Using_External_Penetration_Test

Lockheed Martin. (2015). GAINING THE ADVANTAGE: Applying Cyber Kill Chain. Lockheed Martin Corporation. Obtenido de: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/Gaining_the_Advantage_Cyber_Kill_Chain.pdf

Lutkevich, B. (2019). Microcontroller (MCU). TechTarget | IotAgenda. Obtenido de: https://internetofthingsagenda.techtarget.com/definition/microcontroller

Maurushat, A. (2019). Ethical Hacking. University of Ottawa Press. https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2107145&lang=es&site=ehost-live

Me No Dev. (2022). ESPAsyncWebServer. GitHub. Obtenido de: https://github.com/me-no-dev/ESPAsyncWebServer

MITRE|ATT&CK. (2022a). Adversary-in-the-Middle: DHCP Spoofing. MITRE|ATT&CK. Obtenido de: https://attack.mitre.org/techniques/T1557/003/

MITRE ATT&CK. (2022b). MITRE ATT&CK. Obtenido de: https://attack.mitre.org/

MITRE ATT&CK. (2022c). Gather Victim Identity Information. Obtenido de: https://attack.mitre.org/techniques/T1589/

MITRE ATT&CK. (2022d). Gather Victim Identity Information: Email Addresses. Obtenido de: https://attack.mitre.org/techniques/T1589/002/

MITRE ATT&CK. (2022e). Input Capture. Obtenido de: https://attack.mitre.org/techniques/T1417/

Molina, J.A. (2019). La importancia de la gestión de riesgos y seguridad en el internet de las cosas (IOT). Universidad Piloto de Colombia Bogotá, Colombia. http://repository.unipiloto.edu.co/handle/20.500.12277/6754

Molina Robles, F. J. & Polo Ortega, E. (2015). Servicios de red e Internet. RA-MA Editorial. https://elibro.bibliotecabuap.elogim.com/es/lc/bibliotecasbuap/titulos/62478

Molina Robles, F. J. (2015). Redes locales. RA-MA Editorial. https://elibro.bibliotecabuap.elogim.com/es/lc/bibliotecasbuap/titulos/62450

Muncaster, P. (2021). Qué es una superficie de ataque y cómo podemos reducirla. WELIVESECURITY by ESET. https://www.welivesecurity.com/la-es/2021/09/14/que-es-superficie-ataque-como-reducirla/

National Cyber Security Centre. (2017). Penetration Testing
Advice on how to get the most from penetration testing. https://www.ncsc.gov.uk/guidance/penetration-testing

NIST. (SA). CYBERSECURITY FRAMEWORK. https://www.nist.gov/cyberframework

OSI. (2018). ¿Qué son los ataques DoS y DDoS? Oficina de Seguridad del Internauta. Obtenido de: https://www.osi.es/es/actualidad/blog/2018/08/21/que-son-los-ataques-dos-y-ddos

OWASP. (2022). OWASP. https://owasp.org/

Pearson, B., Luo, L., Zou, C., Crain, J., Jin, Y. & Fu, X. (2020). Building a Low-Cost and State-of-the-Art IoT Security Hands-On Laboratory. University of Central Florida, University of Florida, Gainesville & University of Massachusetts Lowell. https://www.cs.ucf.edu/~czou/research/IFIPIoT_Education-2019.pdf

Perez, O. (2018). Three Hand Shake. Universidad del Caribe. Informatica II (INF-102). Obtenido de: https://www.studocu.com/es-mx/document/universidad-del-caribe/informatica-ii/three-hand-shake/7694981

R. Droms. (1993). Dynamic Host Configuration Protocol. RFC 1531. Bucknell University. Obtenido de: https://www.rfc-editor.org/pdfrfc/rfc1531.txt.pdf

Riso, H. & Saibene, O. (2020). Redes de Telecomunicaciones. Jorge Sarmiento Editor - Universitas. https://elibro.bibliotecabuap.elogim.com/es/lc/bibliotecasbuap/titulos/174559

Rubén Andrés. (2020). Los 7 mayores ataques DDoS de la historia de Internet. Computer Hoy. Obtenido de: https://computerhoy.com/listas/tecnologia/mayores-ataques-ddos-historia-internet-555187

SA. (2019). Wi-Fi Chip ESP32 Subjected to Forever-Hack. myce. https://myce.wiki/news/wi-fi-chip-esp32-subjected-to-forever-hack-89147/

Salinas Anaya, Y. D., Galván Rodríguez, D. G., Guzmán Prince, I., & Orrante Sakanassi, J. A. (2022). El impacto del internet de todas las cosas (IoT) en la vida cotidiana. Ciencia Latina Revista Científica Multidisciplinar, 6(2), 1369-1378. https://doi.org/10.37811/cl_rcm.v6i2.1959

Sánchez Rubio, M. Barchino Plata, R. & Martínez Herráiz, J. J. (2020). Redes de computadores. Editorial Universidad de Alcalá. https://elibro.bibliotecabuap.elogim.com/es/lc/bibliotecasbuap/titulos/131606

Santana González, ARD, C. de N. R. (2021). Costo económico de los ciberataques no tipificados en las leyes dominicanas. Seguridad, Ciencia &Amp; Defensa, 5(5), 32–39. http://35.190.156.69/index.php/rscd/article/view/59

Stehlík, R. (2021). Wi-fi attacks using ESP32. BRNO UNIVERSITY OF TECHNOLOGY.

Faculty of Information Technology. http://excel.fit.vutbr.cz/submissions/2021/048/48.pdf

Suwatchai K. (2022).Firebase Realtime Database Arduino Library for ESP32. GitHub. Obtenido de: https://github.com/mobizt/Firebase-ESP32

Syscom. (SA). HIKVISION Digital Technology Co., Ltd. is. Obtenido de: https://www.syscom.mx/principal/listadopormarca/hikvision

Tapia, V. (2017). Industria 4.0 – Internet de las Cosas. UTCiencia "Ciencia y Tecnología al servicio del pueblo". Recuperado de http://investigacion.utc.edu.ec/revistasutc/index.php/utciencia/article/view/6

Telegram. (SA). Telegram Bot API. Obtenido de: https://core.telegram.org/bots/api

Valle, B., Ronaldo, A., A. (2019). COMPUTACIÓN EN LA NIEBLA: ARQUITECTURA, APLICACIONES Y SEGURIDAD. UNIVERSIDAD DE QUINTANA ROO, DIVISIÓN DE CIENCIAS E INGENIERÍA. PP: 18. Obtenido de: http://risisbi.uqroo.mx/handle/20.500.12249/2712

Vaidya, A., Jaiswal, S., Motghare, M. (2016). A review paper on spoofing detection methods in wireless LAN. 1-5. 10.1109/ISCO.2016.7727054.

Verdejo, G. (SA). CAPITULO 2. Denegación de Servicio: DOS / DDOS. Obtenido de: https://www.cs.upc.edu/~gabriel/files/DEA-es-2DOS-DDOS.pdf

Villarino Marzo, J. (2018). La privacidad en el entorno del cloud computing.. Editorial Reus. https://elibro.bibliotecabuap.elogim.com/es/lc/bibliotecasbuap/titulos/121544