UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

POSGRADOS

MAESTRÍA EN
SEGURIDAD DE LA INFORMACIÓN
RPC-SO-28-NO.669-2021

OPCIÓN DE TITULACIÓN:

ARTÍCULOS PROFESIONALES DE ALTO NIVEL

TEMA:

SEGURIDAD EN EL INTERNET DE LAS COSAS INDUSTRIAL (IIOT) EN LA INDUSTRIA 4.0

AUTORES:

ESTEFANIA DAYANNA ACOSTA GALLO
ANTHONY FABRICIO SÁNCHEZ BRICEÑO

DIRECTOR:

JUAN CARLOS DOMÍNGUEZ AYALA

CUENCA – ECUADOR
2024

Autores:

Estefania Dayanna Acosta Gallo
Ingeniera Electrónica.
Candidata a Magíster en Seguridad de la Información por la Universidad Politécnica Salesiana – Sede Cuenca.
eacostag1@est.ups.edu.ec

Anthony Fabricio Sánchez Briceño
Ingeniero Electrónico.
Candidato a Magíster en Seguridad de la Información por la Universidad Politécnica Salesiana – Sede Cuenca.
asanchezb2@est.ups.edu.ec

Dirigido por:

Juan Carlos Domínguez Ayala
Ingeniero de Sistemas.
Magister en Redes de Comunicaciones.
jdominguez@ups.edu.ec

DEDICATORIA

Dedico este articulo a mis padres porque han sido un pilar muy importante para mi vida, ya que con su esfuerzo y apoyo incondicional he podido cumplir mis metas.

A mis hermanos por siempre apoyarme en las buenas y en las malas y a mis sobrinos por siempre ser buenos conmigo.

Estefania Dayanna Acosta Gallo

DEDICATORIA

En primer lugar, dedico este trabajo a Dios, por brindarme la salud, sabiduría y fuerza para seguir adelante con mis metas y por poner en camino a personas increíbles que suman a mi crecimiento y desarrollo personal.

A mis padres, ya que por ellos he logrado llegar muy lejos y cumplir mis objetivos, ellos son un pilar fundamental en mi vida, con su amor y paciencia me han inculcado valores y principios que hacen de mi un hombre de bien y un buen profesional, este logro es de ustedes.

A mi hermano, por estar siempre apoyándome y darme ánimo cuando lo necesito, por alentarme a seguir adelante y por estar en los momentos más importantes de mi vida.

A toda mi familia, en especial a mis abuelitas Olinda y Maruja, por estar a mi lado en todo momento apoyándome y dándome mucho aliento y por mostrarme lo que es una familia unida a pesar de las circunstancias.

A mis más grandes amigos, que me han demostrado que a pesar del paso del tiempo y circunstancia puedo contar con ellos, y siempre están ahí, son los mejores, son para mí un tesoro y algo invaluable.

Anthony Fabricio Sánchez Briceño

AGRADECIMIENTO

Agradezco a mi compañero por el apoyo, tiempo y esfuerzo para realizar este artículo.

Agradezco a mis padres por siempre confiar en mí y apoyarme en esta trayectoria.

Un agradecimiento especial al Ingeniero Juan Carlos Domínguez por el apoyo para la realización de esta investigación.

Estefania Dayanna Acosta Gallo

AGRADECIMIENTO

Agradezco a mis padres, mi hermano y a toda mi familia, por su apoyo incondicional y por estar a mi lado en todo momento, son mi motor para seguir adelante.

A mi compañera por todo su esfuerzo y dedicación en la elaboración de este artículo.

A mis amigos que han estado ahí apoyándome y dándome ánimo para continuar con mis estudios.

Un agradecimiento especial a nuestro tutor el Ingeniero Juan Carlos Domínguez por el apoyo para la realización de esta investigación y a todos los docentes del masterado, cuyas enseñanzas y experiencia, han aportado y brindado muchos conocimientos enriquecedores a mi formación profesional.

Anthony Fabricio Sánchez Briceño

Tabla de Contenido

Seguridad en el Internet de las cosas Industrial (IIoT) en la Industria 4.0

Autor(es):

Estefania Dayanna Acosta Gallo

Anthony Fabricio Sánchez Briceño

Resumen

En este artículo se presenta un entorno de prueba que simula una industria embotelladora de agua, donde se evalúa los protocolos de comunicación, junto con técnicas de cifrado usadas en la industria 4.0 con IIoT, para lo cual, se utiliza un sensor Sharp, con el propósito de realizar el conteo de embotellamientos diarios. La programación de este sensor se coloca en un microcontrolador llamado Esp32, que va a ser el cliente dentro de la topología, el sensor Sharp envía señales análogas al Esp32 y este convierte la señal en digital ya que cuenta con un ADC incorporado.

El dispositivo Esp32 envía los datos hacia un servidor o también llamado bróker, el cual va a ser una Raspberry pi. El envío de estos datos se realiza por medio de los protocolos de comunicación MQTT, HTTP, COAP y MQTT TLS, estos protocolos se van a programar en el servidor y en el cliente para poder crear la comunicación entre ellos.

Por último, con la ayuda del software Wireshark se realiza un ataque de hombre en el medio o “Man in the Middle” en cada uno de los protocolos, para realizar una captura de datos, y así comparar cada protocolo y validar cuál de ellos se puede implementar con el uso de software libre de una forma segura en las Industrias 4.0, cumpliendo todos los términos de seguridad que son: Disponibilidad, Integridad y Confidencialidad.

Palabras clave:

Encriptación Esp32, IIot, Protocolos de comunicación, Open SSL, Rapsberry pi, TLS, Wireshark.

ABSTRACT

This article presents a test environment that simulates a water bottling industry, where the communication protocols are evaluated, along with encryption techniques used in industry 4.0 with IIoT, for which a Sharp sensor is used, with the purpose of counting daily traffic jams. The programming of this sensor is placed in a microcontroller called Esp32, which will be the client within the topology, the Sharp sensor sends analog signals to the Esp32, and it converts the signal into digital since it has a built-in ADC.

The Esp32 device sends the data to a server or also called a broker, which will be a Raspberry pi. The sending of this data is carried out through MQTT, HTTP, COAP and MQTT TLS communication protocols. These protocols will be programmed in the server and the client to create communication between them.

Finally, with the help of the Wireshark software, a man in the middle attack is carried out on each of the protocols, to capture data, and thus compare each protocol and validate which of them is used. can be implemented with the use of free software in a safe way in Industries 4.0, complying with all the security terms that are: Availability, Integrity and Confidentiality.

Palabras clave:

Communication protocols, Esp32 encryption, IIot, Open ssl, Rapsberry pi, TLS, Wireshark

1. Introducción

La evolución industrial ha contribuido en la mejora de los procesos y eficiencia por medio de la implementación del Internet de las cosas Industrial o IIot, esto provocó que surjan nuevas maneras de vulnerar, dañar o robar información de los sistemas de una industria 4.0, es ahí donde radica la importancia de utilizar mecanismos de seguridad robustos que permitan reducir los riesgos de sufrir un ataque cibernético.

En primera instancia se delimita el problema principal, para luego continuar con el capítulo 1, donde se detalla los desafíos que enfrenta la industria 4.0 en cuanto a ciberseguridad. En el capítulo 2 se indican las vulnerabilidades más comunes y ataques que se realiza a este tipo de entornos sistematizados, también se realiza una recopilación bibliográfica de los dispositivos IIoT que se usa en la industria 4.0. En el capítulo 3 se realiza una investigación sobre protocolos de comunicación y técnicas de cifrado, utilizados en una infraestructura industrial con IIoT.

Una vez recopilada la información necesaria sobre los desafíos, vulnerabilidades y riesgos, dispositivos, protocolos de comunicación y técnicas de cifrado ciberseguridad, se realiza las pruebas pertinentes en el capítulo 4, donde se establece un entorno de prueba simulando una industria embotelladora de agua, con los siguiente dispositivos: Raspberry Pi, módulo ESP32, sensor Sharp y el uso de Open SSL, donde se muestra la eficacia y viabilidad de cada protocolo de comunicación, por medio del software Wireshark, se realizara un ataque Man in The Middle , y para culminar se lleva a cabo una comparación y análisis de resultados para determinar que protocolo es el más indicado para ser aplicado en una industria 4.0 con IIoT.

2. Determinación del Problema

La revolución industrial alrededor del mundo ha tenido un avance importante en términos de digitalización de procesos, correspondiente a la industria 4.0, ya que es la generación que realiza la interconexión de dispositivos y sistemas por medio de la IIoT, cabe recalcar, que en los últimos años la industria 4.0 ha tenido que afrontar varios retos referentes a la ciberseguridad debido al auge de la digitalización y conexiones a través del uso del internet, en consecuencia, han existido una variedad de ataques cibernéticos a los dispositivos IIoT incorporados en la industria, por tal motivo surge la necesidad de llevar a cabo esta investigación para determinar qué medidas de seguridad informática son las más efectivas al momento de proteger la ciberseguridad de los procesos y los datos dentro de la industria 4.0, garantizando que se cumpla con los principios de confidencialidad, integridad y disponibilidad en el ámbito del internet de las cosas industrial (IIoT)

3. Capítulo 1: Desafíos de ciberseguridad

En esta sección de la investigación se redacta un estado del arte enfocado en los desafíos a los que se enfrenta la industria 4.0 en temas de ciberseguridad, también se detalla que dispositivos IIoT son usados dentro de la industria 4.0. con la finalidad de conocer a cabalidad la infraestructura de la industria y familiarizarse con el entorno de IIoT manejado industrialmente, cabe destacar que una vez estudiado los desafíos y conociendo la infraestructura en la cual se realizan los procesos más importantes de una industria 4.0, se continuará con un análisis de las vulnerabilidades más comunes y los riesgos que están ligados a las brechas conocidas en la industria, para seguir con una investigación de los protocolos y técnicas de cifrado que se utiliza en los entornos IIoT con la finalidad de obtener una base teórica y amplio conocimiento sobre cómo funciona la industria 4.0, que retos deben ser superados y conocer que dispositivos, protocolos y técnicas de cifrado son los escogidos dentro de una infraestructura industrial de cuarta generación

3.1 Estado del arte sobre los desafíos que enfrenta la industria 4.0

Para esta investigación se ha realizado una recopilación de varios casos de estudio sobre los desafíos que está afrontando la industria 4.0 en base a los dispositivos IIoT. para ellos se ha tenido que tomar en cuenta todos los ataques de menor a mayor gravedad para tener una base teórica sólida y una visión hacia qué desafíos existen en torno a la industria 4.0 y como enfrentar estos desafíos en términos de ciberseguridad por ellos también es necesario analizar los tipos de ataques más relevantes que tienen un impacto negativo directo a la industria 4.0, además se indica algunas medidas o consideraciones que se deben ejecutar para disminuir los atentados cibernéticos.

3.1.1 Desafíos de ciberseguridad con IIoT en la Industria 4.0

Actualmente con todos los nuevos avances de la revolución industrial, se ha generado nuevas herramientas IIoT, por lo tanto, han aumentado y generado nuevos métodos de

ataque y aprovechamiento de brechas dentro de los sistemas informáticos de la industria, por lo tanto, Los desafíos que debe superar la industria 4.0 en términos de ciberseguridad, principalmente son los vectores de ataques cibernéticos que se presentan con frecuencia, un ejemplo de ataque, puede ser realizado por medio de los controladores inteligentes conectados a la red, por ello es necesario tener conocimiento sobre ciberseguridad en los procesos de la industria y en su infraestructura, es decir, estudiar las amenazas y vulnerabilidades para tener conocimiento de los riesgos que puede existir y contrarrestar los ataques, implementando los estándares que son aprobados por los organismos de estandarización cenelec/etsi como por ejemplo isa/iec 62443, Iso 27001, bs 25999, nist sp 800-82, etc, para realizar un plan de seguridad para la protección tanto del sistema industrial como de infraestructura, también es importante tomar en cuenta el monitoreo de data centers, ya que mantienen sistemas de gestión de ciberseguridad industrial para verificar el acceso indebido a ciertas aplicaciones o también verificar que todos los firewalls estén actualizados, que mantenga licencias antivirus vigentes, y actualizados los softwares, etc. (Joyanes Aguilar, 2018)

4. Capítulo 2: Vulnerabilidades y riesgos

4.1 Vulnerabilidades y Riesgos más comunes en dispositivos IIoT - Industria 4.0

Una vez familiarizados con los desafíos, que enfrenta la Industria 4.0 dentro de su infraestructura IIot y conocidos los dispositivos más comunes con los cuales está conformada, se procede a echar un vistazo sobre las vulnerabilidades más comunes y los riesgos que representan dentro de un sistema industrial avanzado, ya que, se debe tener claro el panorama sobre el cual se va a implementar las seguridades pertinentes, o de ser el caso analizar o auditar una infraestructura industrial de alto nivel. Cabe destacar que hay varios tipos de industria 4.0 entre ellas las más críticas son: Industrias farmacéuticas, procesadoras de alimentos, embotelladoras de agua, plantas eléctricas, procesadoras de agua potable, etc. De ahí la importancia de tener un conocimiento sobre las vulnerabilidades que afectan directamente a la Industria 4.0 (Sotolani et al., 2022)

Un ejemplo claro de aprovechamiento de vulnerabilidades dentro de una industria tenemos, el ocurrido en el año 2016 a la red eléctrica de Ucrania, donde un programa con codificación maliciosa o “malware” conocido como “Black Energy”, aprovechó una vulnerabilidad en las versiones de software y demás brechas existentes dentro del sistema operativo de las computadoras de la empresa eléctrica , el malware creó un “backdoor” en el sistema, lo que provocó la toma de control del sistema eléctrico y finalizando con el evento del apagón a nivel nacional. Según los investigadores de la empresa “Eset” debido a un ataque de phishing por el cual un operador de la empresa eléctrica ucraniana lamentablemente dio acceso al programa maligno mencionado permitiendo la explotación de las brechas del sistema. (Sotolani et al., 2022)

Adicionalmente, se debe tener en cuenta que, como primera vulnerabilidad y desencadenante principal de brechas en un sistema, es el “error humano” y “malas prácticas”, ya que, un operador o encargado del sistema puede realizar una mala configuración, o es expuesto a ataques como el “Phishing” o “Ataques de ingeniería social”,

siendo el mayor riesgo al cual se expone un sistema informático. Sin embargo, para identificar las vulnerabilidades más comunes dentro de un sistema industrial, es recomendable y necesario, tener un inventario de todos los dispositivos de manera organizada, de tal manera, que se separe los componentes según la criticidad e importancia dentro de la organización industrial, una vez que se tenga claro los dispositivos de vital importancia dentro de la industria, se procedería a analizar las amenazas a las que están expuestos los dispositivos o componentes, por medio de un análisis de vulnerabilidades que afecten directamente a la integridad de todo el sistema industrial. (Sotolani et al., 2022)

Si bien es cierto, los dispositivos que se implementan con "IIoT" favorecen en gran medida a las industrias y sus procesos, de igual manera no están exentos a sufrir algún tipo de ataque cibernético e interferir con el correcto funcionamiento o incluso deteniendo la funcionalidad de estos sistemas, por lo cual, se debe tener en cuenta las siguientes vulnerabilidades que son de mayor importancia y que son recurrentes o muy conocidas dentro de un sistema industrial (Colazo Ornella & Fabbri Lucia, 2023)

4.1.1 Vulnerabilidad de gestión de contraseñas y certificados

En este caso es una vulnerabilidad muy común el manejo incorrecto de las contraseñas y su gestión, ya que por poner un ejemplo, por temas de comodidad o manejo de varios aplicativos, se utiliza la misma contraseña para todos los softwares, adicionalmente, se tiene el caso en el que no se cambia la contraseña por defecto de un dispositivo, otro ejemplo es el guardado de contraseñas dentro de los navegadores o en bloc de notas o documentos sin ningún tipo de cifrado o contraseña maestra, lo cual conlleva el riesgo de que en un ataque cibernético de cualquier tipo, el ciber delincuente o atacante tenga acceso completo a la información de usuario y contraseñas. (Colazo Ornella & Fabbri Lucia, 2023)

4.1.2 Cambios de hardware y su configuración

Otra vulnerabilidad muy común dentro de un entorno industrial es el cambio indebido de dispositivos y mala configuración de estos, sin llevar un manejo adecuado, ya que, en una industria de alto nivel, se tiene dispositivos finales como sensores actuadores y controladores que son de ultima gama y ya poseen conexión a internet, es decir, forman parte de IIoT. Por lo general este tipo de dispositivos finales utilizan protocolos de comunicación, como se verá más adelante, sin embargo, debido a malas configuraciones del hardware mencionado, eleva el riesgo de que el sistema industrial sea vulnerado y

aprovechado por agentes externos lo cual podría dejar inoperativa un área de la organización industrial o incluso a la planta industrial completa. (Colazo Ornella & Fabbri Lucia, 2023)(Hoffman, 2019)

4.1.3 Configuración errónea del sistema de control y gestión de dispositivos

Esta vulnerabilidad también es muy común, ya que, debido a mala configuración de los sistemas, softwares de control de equipos finales, sistemas o softwares desactualizados, etc. Esto podría afectar gravemente a la planta industrial y conllevaría el riesgo de sufrir un ataque activo o pasivo, donde tanto las maquinas donde alojan los sistemas de control pueden ser usados como "botnet" y tomarían el control sobre los sistemas críticos de la organización y equipos finales como los sensores, ahí la importancia de llevar un correcto uso de protocolos de seguridad y mantener siempre actualizados y bien configurados los sistemas de control de la organización industrial. (Colazo Ornella & Fabbri Lucia, 2023)(Hoffman, 2019)

Una vez identificadas las vulnerabilidades más comunes dentro de un entorno industrial de cuarta generación, a continuación, se indicará una pequeña reseña investigativa sobre los protocolos de seguridad informática y técnicas de cifrado usados dentro de un entorno IIoT en la Industria 4.0

4.1.4 Tipos de ataques cibernéticos en la Industria 4.0

Ahora teniendo en cuenta los principales desafíos, se debe recalcar que industria 4.0 ha logrado unificar los proceso de producción y la tecnología de la información y comunicación (tics), por lo cual, se han registrado varios ataques cibernéticos de distintas categorías, es decir, desde los menos graves hasta los de mayor riesgo que pueden dejar inhabilitados los sistemas, como por ejemplo, el ataque más famoso y usado alrededor del mundo es el ataque de phishing, este tipo de ataque es de alta consideración, ya que, por medio de este ataque, los ciberdelincuentes buscan recolectar información y credenciales de acceso lo cual permite al atacante el ingreso a todas las plataformas y posible escalamiento de privilegios en el sistema, lo cual es muy grave y puede vulnerar de sobremanera los procesos industriales.

Otro tipo de ataque muy conocido es el de Ataque de Hombre en el Medio, también conocido como Ataque MITM (Man in the Middle), este ataque tiene como finalidad

capturar el tráfico en una comunicación entre dos usuarios, en el caso de la industria 4.0 el objetivo sería capturar tráfico entre dos dispositivos IIot, de esta manera el atacante podría observar la información que se envía entre los dispositivos, e incluso puede alterarla a su gusto, lo cual afecta a la confidencialidad, integridad y disponibilidad de los datos. Otro tipo de ataque se tiene el que se realiza por medio de Rootkit, estos bajan el rendimiento de los procesos, infiltrándose de tal manera que no puedan ser detectados y así obtener el control total del sistema, antes de ser detectados por algún firewall de ciberataques, la pérdida del control puede ser perjudicial para la empresa en el ámbito económico ya que sufrirían varias perdidas de producción cuando el Código llega a ser alterado por los ciberdelincuentes, en la industria 4.0 el ataque que se ha registrado con mayor frecuencia es el malware, por ello es importante realizar en las industrias planes de seguridad de la información para poder combatir los ciberataques. (Ayerbe, 2018)(Tavares Jonathas, 2024). Los ataques cibernéticos se los realiza por medio de algunas herramientas de software conocidas, a continuación, se enlista a manera de ejemplos las herramientas usadas por los atacantes para robar información o ganar acceso a una red: (Rodríguez Llerena, n.d.)

- Metasploit: Herramienta de Kali Linux, con un repositorio o "biblioteca" de "exploits y payloads" que permiten aprovechar las vulnerabilidades de un sistema y ganar acceso o privilegios de administrador.
- Nmap: Es una herramienta del sistema Kali Linux, que permite el escaneo de puertos abiertos, además de brindar información sobre versiones de software, entre otros.
- Wireshark: Capturador de tráfico por excelencia, permite observar los paquetes que se envía a través de una red de comunicación, además de ver el tipo de paquete y protocolos de comunicación, además de otro tipo de información que recopila esta herramienta.
- BetterCap: Herramienta utilizada para realizar ataques de hombre en el medio MITM, adicionalmente permite alterar la información TCP, trafico Http, entre otras funcionalidades, que permiten al atacante.

4.1.5 Consideraciones para tener en cuenta sobre ciberseguridad en la industria 4.0

Para afrontar los desafíos en ciberseguridad en la industria 4.0 es importante considerar una segmentación de los procesos de producción, es decir, implementar un firewall que pueda

examinar todas las capas de modelo OSI, creando una red multi capa para que las comunicaciones sean seguras e integras, mantener siempre todos los softwares actualizados del servidor, de los dispositivos IIoT y de la programación Scada, también es recomendable tener un antivirus original y actualizado para poder detectar amenazas, con respecto a la protección de los protocolos es preciso colocar un analizador sintáctico el cual previene la suplantación de comandos y de paquetes falsos, también neutraliza protocolos que no sean necesarios para el proceso. en los dispositivos IIoT se debe deshabilitar todos los puertos y prohibir los accesos sin credenciales establecidas. (Ayerbe, 2018)

4.2 Dispositivos usados dentro de la industria 4.0

Una vez realizado un estudio enfocado en todos los desafíos que debe superar las organizaciones industriales 4.0, vamos a Elaborador una pequeña reseña de los dispositivos que se encuentran involucrados dentro de los procesos de producción de una industria de cuarta generación, ya que, debido a la implementación del internet de las cosas industrial (IIoT), se debe tener una idea clara de que dispositivos formarían parte de una infraestructura de la Industria 4.0 y se debe conocer qué tipo de dispositivos específicos tienen un funcionamiento que va ligado con IIoT, cabe destacar, que gracias al avance en IIoT, se ha logrado digitalizar todos los procesos dentro de la industria, por lo tanto, con esta interconexión de las maquinas a internet, se ha logrado tener un control y análisis de datos de la productividad de las maquinas industriales, lo cual permite a los operarios mejorar la calidad de productos y su eficiencia de productividad, a continuación se indica las diferentes tecnologías relacionadas con IIoT dentro de una industria 4.0. En primer lugar, tenemos un estudio de este tipo de tecnologías los cuales son la base de la identificación de estos dispositivos, estos dispositivos están distribuidos los siguientes tipos de tecnologías como: (Ávila-Camacho & Moreno-Villalba, 2023)

4.2.1 RFID (Identificación por radiofrecuencia)

Este tipo de tecnología es utilizada para la identificación de diferentes dispositivos u objetos de manera automatizada gracias al uso de la radiofrecuencia, lo cual nos permite tener visibilidad y seguimiento de los dispositivos en tiempo real dentro de cualquier escenario. (Ávila-Camacho & Moreno-Villalba, 2023)

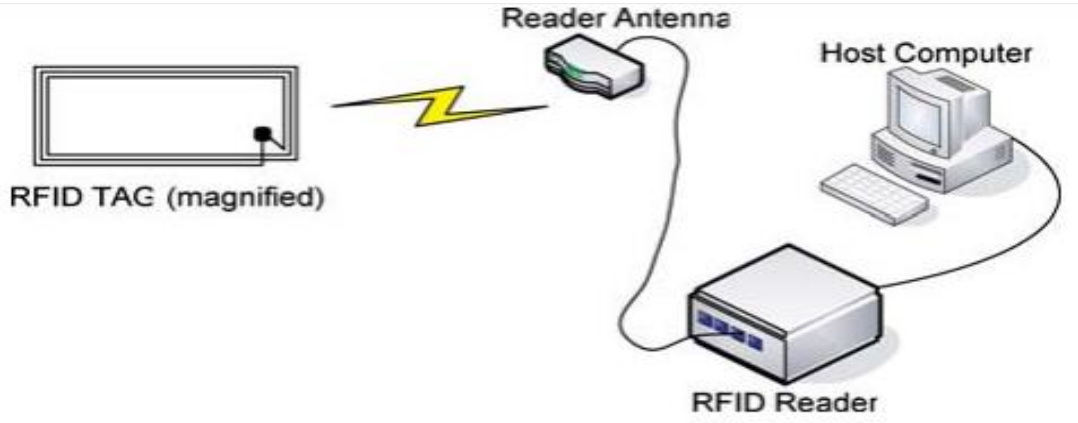

Figura 1. Conexión común de identificación RFID (Kabachinski Jeff, n.d.)

4.2.2 NFC (Comunicación de campo cercano)

Es un tipo de comunicación de corto alcance, que utiliza radiofrecuencia, adicional es un tipo de conexión semidúplex, este tipo de tecnología se utiliza para realizar una conexión entre un emisor y un receptor permitiendo el intercambio de datos, por ejemplo, entre un dispositivo móvil o smartphone y una tarjeta NFC. (Ávila-Camacho & Moreno-Villalba, 2023)

Adicional cabe destacar que hay 3 tipos de conexión NFC:

- Modo de lectura
- Modo punto a punto
- Modo de Emulación de tarjeta NFC

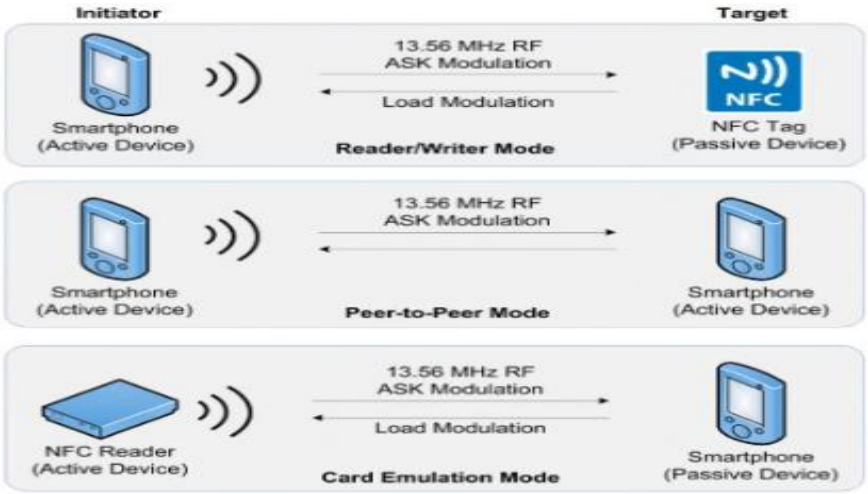

Figura 2. Tipos de conexión NFC (Ávila-Camacho & Moreno-Villalba, 2023)

4.2.3 MtoM (Comunicación Maquina a Maquina)

Es un método de comunicación muy común dentro de una infraestructura de red, como por ejemplo interconexión entre computadoras, teléfonos, controladores, sensores inteligentes, etc. Este tipo de comunicación puede ser intervenida o no por un operario, además este tipo de comunicación está determinada por una infraestructura de red que cuenta con un servicio de cloud o nube y una gran cantidad de máquinas, donde se utiliza un tipo de módulo de red el cual puede utilizar un medio inalámbrico o por cable de alta velocidad, para conectar varios servicios como bases de datos, servidores o aplicaciones hacia el servicio de cloud. (Ávila-Camacho & Moreno-Villalba, 2023)

4.2.4 CPS (Sistema Ciber físico)

Es un tipo de tecnología IIoT, que tiene la capacidad de conectar diferentes tipos de dispositivos y maquinas con un software o aplicativo que tenga acceso a internet, es decir, es una agrupación de herramientas físicas y digitales interactivos entre sí, que pueden ser distribuidos de tal manera que permitan la correcta detección, control y conexión con la infraestructura de red, ya que, se encarga de la recopilación de información por medio de distintos sensores, actuadores, etc. (Valencia & Portilla, 2019)

4.2.5 Cloud Computing o Computación por medio de la nube

Este tipo de conexión es un servicio clave en la conexión de varios servicios y dispositivos, actualmente es un método de comunicación e intercambio de datos muy usado, ya que permite la conexión de varios dispositivos para tener el acceso a diferentes recursos como computadoras, dispositivos IIoT, servidores de manera remota sin necesidad de estar dentro de la infraestructura industrial. (Valencia & Portilla, 2019)

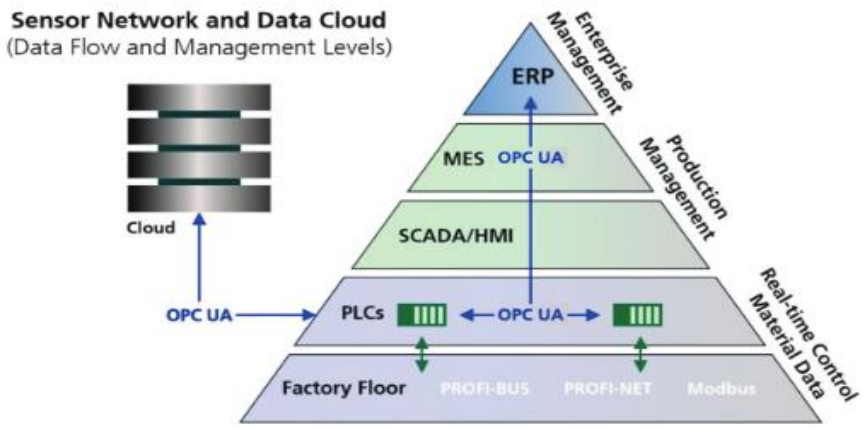

Figura 3. Interacción entre dispositivos y la nube (Valeske et al., 2020)

4.2.6 Sensores y controladores en la industria 4.0

Para finalizar los dispositivos IIoT más comunes dentro de un entorno de industria 4.0 son dispositivos inteligentes que nos permiten la recolección de datos dentro de la industria, ya que permiten el monitoreo constante de la producción de los productos o la recepción de datos e información, a continuación, se indica los dispositivos más comunes dentro de la industria 4.0: (Valencia & Portilla, 2019)(Steve Gómez-Meza et al., n.d.)

- Rapberry pi: Dispositivo utilizado en industria 4.0 que permite la conexión a internet además de otras utilizades como programación de código scratch y phyton.
- PLC: Es un dispositivo que permite la automatización de procesos industriales con el fin de controlar la maquinaria de un entorno industrial.
- Sensores PIR: Dispositivo utilizado como un sensor piro eléctrico el cual está diseñado para medir luz infrarroja de otro dispositivo emisor
- Sensores finales de carrera: Son conocidos como “interruptores de límite” que permiten controlar la ubicación y el límite de movimiento de un objeto.
- Sensor óptico: Este dispositivo nos permite detectar cuando un objeto pasa frente al sensor, de esta manera se puede determinar variables como velocidad y posición, al momento que un objeto pasa en frente del haz de luz que emite este tipo de sensor.
- Sensor Sharp: Dispositivo para recolección de datos infrarrojo, utilizado en la industria, por lo general para realizar el conteo de materia prima procesada, entre otras utilidades.
- Modulo ESP32: Microcontrolador que permite ser programado para diferentes utilidades dentro de la industria, se basa en la comunicación por medio de Bluetooth o Wifi.

5. Capítulo 3: Protocolos y Técnicas de Cifrado

5.1 Protocolos de comunicación de IIoT utilizados en la Industria 4.0

Los protocolos más usados en la industria 4.0 se tiene el MQTT, COAP, HTTP, AMQP, los cuales se detallarán a continuación.

5.1.1 Protocolo MQTT en la industria 4.0

MQTT es un protocolo de comunicación de maquina a máquina(M2M), el cual se basa en la suscripción y publicación de topics hacia un bróker. Los topics son un tipo de filtro de mensajes el cual está conformado por una cadena de caracteres que puede tener un tamaño máximo de 65535 caracteres y un Broker es básicamente un servidor. En MQTT el proceso de envío de información se realiza de la siguiente manera: en primer lugar, un dispositivo se suscribe al topic del cual desea recibir la información y al mismo tiempo puede realizar una publicación, para que otros dispositivos inscritos a ese topic puedan recibir una respuesta, una vez suscritos los dispositivos, el bróker se encarga de gestionar las suscripciones de los topics a las publicaciones, adicionalmente considera los niveles de calidad de servicio hacia los clientes, esto también se conoce como Quality of service (QoS), los cuales se detallaran a continuación. (Neco Villegas Saiz, 2021):

- **QoS 0:** Este mensaje solo se envía una vez
- **QoS 1:** Esta seguridad espera hasta que el mensaje sea recibido correctamente. Este mensajea se puede recibir varias veces.
- **QoS 2:** Esta seguridad garantiza que el mensaje haya sido entregado una sola vez al suscriptor.

La seguridad del protocolo MQTT utiliza la capa de trasporte, para esto el protocolo utiliza dos tipos de seguridades los cuales son: “secure sockets layer (SSL)” y “transport layer security (TLS)”, las seguridades mencionadas permiten colocar un método de cifrado que viene incluido en el protocolo, ya que, permite tener una autenticación por medio de usuario y contraseña o por medio de un certificado de seguridad. (Neco Villegas Saiz, 2021)

5.1.2 Protocolo COAP en la Industria 4.0

Es un protocolo de comunicación M2M con el uso de dispositivos de baja potencia los cuales pueden comunicarse por medio de internet, su estructura está basada en la interacción de cliente a servidor de una manera asíncrona utilizando la capa de transporte UDP para poder reducir la sobrecarga de tráfico en TCP, este protocolo se parece al HTTP sin embargo presenta mejoras con respecto a HTTP (Castro Heredia & Losilla López, 2014). La forma en que se manejan los mensajes en este protocolo es el solicitud y respuesta de mensajes por medio de UDP, para poder validar si el mensaje fue recibido correctamente, se puede validar con el mensaje de asentamiento (ACK) el cual mantiene la ID con la que se envió el mensaje, si es que no fue recibido se recibirá una respuesta con un "Reset (RST)" y si no requiere de confirmación de llegada de la mensajería se recibirá un "No confirmable (NON)" o un RST y esto indica que el mensaje fue recibido con éxito. Los mensajes que son enviados por medio de este protocolo son divididos en tres partes que son cabecera, opciones y carga útil. La cabecera es en donde se encuentra toda la información como versión, tipo, código e ID para el mensaje a enviar, la Opción es la que mantiene los parámetros de las peticiones y para finalizar la carga útil es el que está constituido por el cuerpo del mensaje. (Castro Heredia & Losilla López, 2014)

5.1.3 Protocolo HTTP en la industria 4.0

Este protocolo es muy usado en el ámbito de la comunicación mediante internet, el cual permite la transferencia de información en la web o "World Wide Web", está conformado con la estructura de servidor/cliente, es decir el servidor es el que se encarga de solventar la petición que requieran los clientes ya que puede ser solo uno o vario a la vez. HTTP es un protocolo de hipertexto del modelo "TCP/IP", el cual contiene links, videos, texto, sonido y su puerto de conexión que es asignado por defecto es el 80 en TCP. El mensaje que envía el protocolo HTTP es de texto plano hacia un servidor , el cual guarda los estados de cada uno de los dispositivos , el servidor en cambio utiliza el estándar REST ya que ofrece la capacidad de intercambiar datos de manera segura entre aplicaciones y sistemas que existen en la web, por medio de un método de acceso POST el cual permite añadir un dispositivo ya que pide permiso al servidor que acepte la petición ya que este es el único que puede aceptar o rechazar , PUT es el que actualiza la información, GET obtiene la información y DELETE solicita al servidor que elimine el dispositivo. También se utiliza JSON Y XML, ya que estos estándares permiten la transmisión de información entre aplicaciones que se encuentra en

la navegación web. Este protocolo HTPP ha sido diseñado de tal forma que todos los navegadores puedan interpretarlo y enviar peticiones con el método GET por lo cual todos los dispositivos IIoT puede utilizar este protocolo en cualquier navegador, permitiendo crear una interfaz gráfica para el uso del cliente. (Muralles Eduardo, 2020)

5.1.4 Protocolo AMQP en la industria 4.0

Este protocolo AMQP tiene la estructura de publicar y suscribir mensajería, pero este protocolo sirve para las entidades financieras, ya no es recomendable para la industria 4.0, pero este protocolo soporta hasta transacciones, en comparación con el protocolo MQTT este puede confirmar transacciones completas. (Caiza et al., 2019)

5.2 Técnicas de cifrado utilizadas en IIoT en la Industria 4.0

En la actualidad es importante el envío de mensajes que debe ser de forma segura, en IIot, existen protocolos de cifrado para poder enviar estos mensajes encriptados mientras se trasmiten, para que las personas no autorizadas, no puedan robar esta información crítica de una organización. Para utilizar los protocolos de cifrado se lo realiza mediante certificados SSL, los cuales son proporcionados por autoridades de certificación, existen autoridades de paga como por ejemplo “Digicert” sin embargo, existen autoridades certificadoras completamente gratuitas llamadas “Open SSL” y “Let’s Encrypt”. (Digicert, 2024; Let´s Encrypt, 2024; Open SSL Corporation, 2024) Cabe destacar que los protocolos más usados en IIot es el Secure Sockets Layes (SSL) y Transport Layer Security (TLS), a continuación, se detalla una breve reseña sobre cada técnica de cifrado:

5.2.1 Técnica SSL

Este protocolo se utiliza en el envío de mensajes en sitios web y correo electrónicos de forma segura el cual fue creado en 1994 por Netscape, durante el periodo de estar en auge este protocolo se dieron cuenta que existía vulnerabilidades en el protocolo por lo cual tuvieron que remplazarlo por el protocolo TLS, el cual es una versión mejorada estandarizado en la Industria para poder tener comunicaciones, que sean por medio de internet y que estas sean seguras en los dispositivos cuando se realicen aplicaciones, envío de mensajes por medio de correo electrónico, transmisión de archivos, etc. (Cruz Lucas et al., 2022)

5.2.2 Técnica TLS

Es un protocolo mejorado del SSL el cual se convirtió en un estándar para la seguridad de las comunicaciones mediante internet, esta técnica tiene una seguridad mediante algoritmos de cifrado y verificación de identidad del servidor , con el uso de certificados digitales, para el uso de sitios seguros en la red es importante tener certificados digitales reconocidos, estos certificados mantienen información del propietario del sitio web y la clave que se utiliza para cifrar los datos de transmisión, es decir cuando el cliente ingresa a este sitio web, el navegador valida el certificando que sea reconocido y pueda mantener una conexión segura , por medio del uso de la clave . Este protocolo también garantiza la privacidad de los datos del correo electrónico y de conexiones de VPN mediante los certificados, los cuales protegen la privacidad de las comunicaciones en aplicaciones y mensajerías, mediante la autentificación de datos en el servidor web por medio del cifrado de mensajes. (Cruz Lucas et al., 2022)

6. Capítulo 4: Materiales y metodología

Una vez que se realizó una revisión bibliográfica sobre las diferentes vulnerabilidades y riegos, protocolos, técnicas de cifrado y se dio un vistazo de cómo se conforma una Industria con IIot. Se procedió a evaluar la viabilidad y eficacia de los protocolos en términos de seguridad y protección de datos, por lo que se creó un ambiente de prueba simulado una planta de agua, donde se realizará un conteo de botellas diarias que embotella la planta y se le enviará estos datos por medio de varios protocolos como el MQTT, MQTT TLS, HTTP y COAP, para determinar la confidencialidad, integridad y disponibilidad de los datos.

6.1 Materiales para el entorno de Prueba

Como parte inicial se seleccionó los materiales a ocupar para elaborar el entorno de prueba y realizar las verificaciones pertinentes de cada protocolo, a continuación, se detalla los materiales utilizados:

- ✓ Sensor Sharp
- ✓ Raspberry pi
- ✓ Módulo Esp32
- ✓ Software Node-RED
- ✓ Software Arduino IDE
- ✓ Computadora

Una vez seleccionados los materiales a usar para el entorno de prueba, se procedió a desarrollar las diferentes topologías, donde se puso a prueba cada protocolo en términos de seguridad de la información, como se muestra a continuación.

6.2 Topología protocolo MQTT

A continuación, en la Figura 4 se muestra la topología aplicada para el protocolo MQTT, en el cual se indica el proceso de obtención de datos por medio de un sensor Sharp que envía los valores análogos al microcontrolador Esp32 que convierte la señal análoga en digital y se publica en la Raspberry Pi. Los dispositivos que se suscriban a los diferentes temas que se publicarán en la Raspberry Pi, recibirán la información solicitada, en este caso, si los

dispositivos desean saber cuántas botellas diarias se embotellaron, se recibirá el mensaje con el valor recolectado diario, de esta manera se podrá visualizar en el dispositivo final.

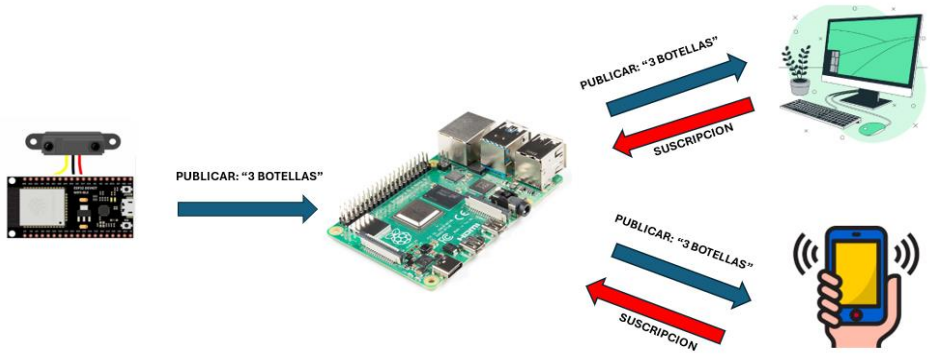

Figura 4. Topología MQTT Elaborado por: Estefania Acosta & Anthony Sánchez

A continuación, se detalla el proceso a seguir en cada dispositivo de la topología, para la obtención de resultados.

6.2.1 Programación en el ESP32

En este dispositivo se realiza la programación del sensor Sharp, el cual va a ser el cliente en el entorno de prueba, ya que va a censar la cantidad de botellas que se va a embotellar diariamente, como se muestra a continuación en la Figura 5, se utiliza la librería de Arduino del protocolo MQTT y se coloca credenciales de acceso al wifi del hogar y se procede con la configuración del sensor.

mqtt §

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
// Replace the next variables with your SSID/Password combination
const char* ssid = "CELERITY_SOFY";
const char* passwo[redacted]e";
// Add your MQTT Broker IP address, example:
const char* mqtt_server = "192.168.1.46";
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
float temperature = 0;
float humidity = 0;
const int ledPin = 4;
void setup() {
  Serial.begin(115200);
   setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  pinMode(ledPin, OUTPUT);
}
void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
```

Figura 5. Programación Esp32 en Arduino Elaborado por: Estefania Acosta & Anthony Sánchez

6.2.2 Programación en la Raspberry pi

En la Raspberry pi se realiza la programación del protocolo MQTT en Node-RED, para visualizar los datos que envía el dispositivo Esp32, con el conteo de botellas diarias, como se muestra a continuación en la Figura 6 y Figura 7. Para este protocolo se utiliza el puerto 1883 y la ip de la Raspberry Pi o servidor es 192.168.1.46, en la Figura 8, se puede observar el envió de datos desde el Esp32 hacia la Raspberry pi por medio del protocolo MQTT.

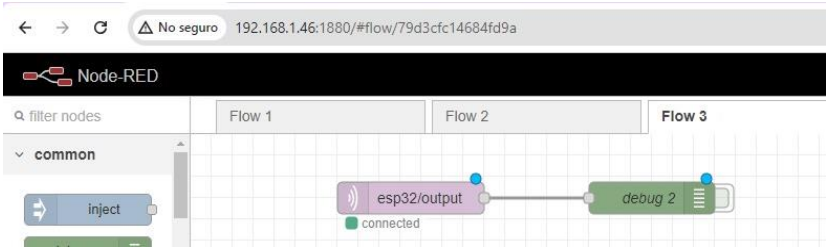

Figura 6. Configuración del bróker con Node-RED Elaborado por: Estefania Acosta & Anthony Sánchez

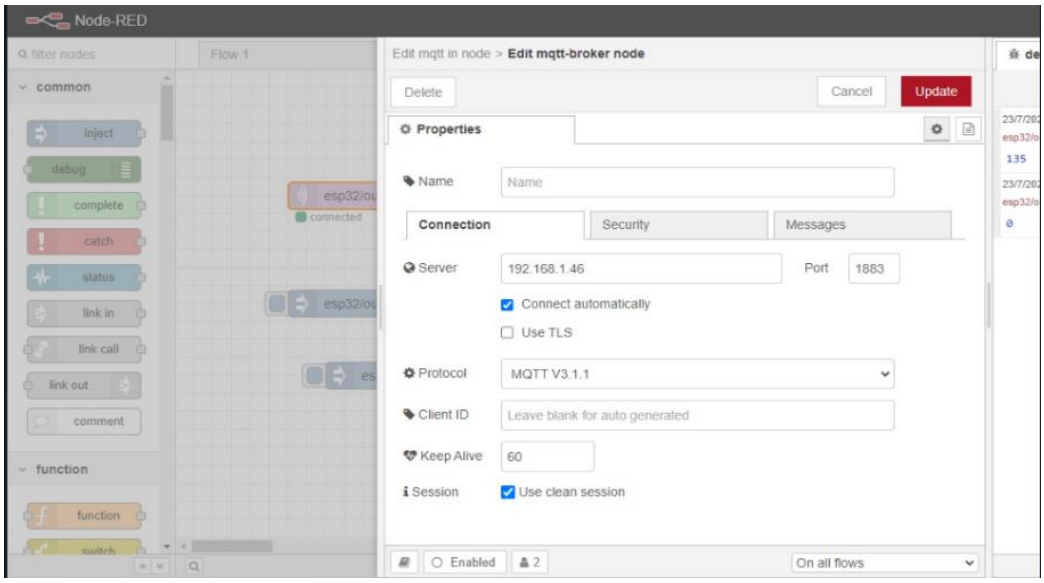

Figura 7. Configuración del servidor Elaborado por: Estefania Acosta & Anthony Sánchez

Figura 8. Recolección de datos con uso de protocolo MQTT Elaborado por: Estefania Acosta & Anthony Sánchez

6.2.3 Prueba de viabilidad del Protocolo MQTT

Para comprobar la viabilidad del protocolo MQTT, se utilizó el software Wireshark, ya que permite interceptar los paquetes que se envía desde el cliente al servidor, este ataque se llama Man in the Middle o MITM como se muestra en la Figura 9 se busca el protocolo MQTT.

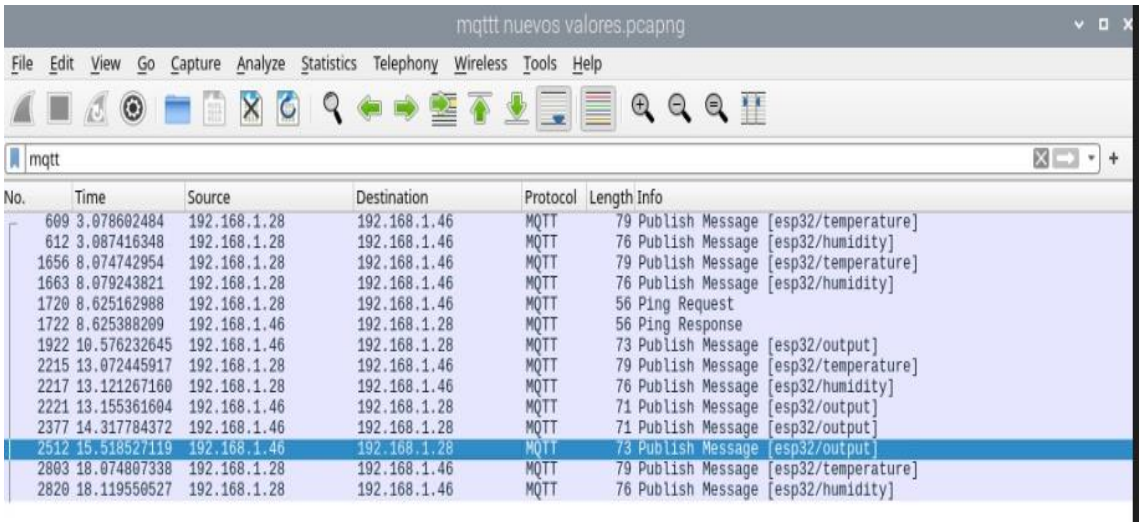

Figura 9. Ataque Man in the Middle con Wireshark en protocol MQTT Elaborado por: Estefania Acosta & Anthony Sánchez

6.3 Topología MQTT TLS

A continuación de la Figura 10 se muestra la topología MQTT TLS la cual es similar a la topología anterior del MQTT, pero en este caso se utiliza llaves y certificados de autoridad para cifrar y descifrar los mensajes que utilizan los dispositivos ESP32 y bróker que en esta situación es la Raspberry Pi. Se debe colocar los certificados que otorga la autoridad de certificación tanto en el bróker como en el ESP32 para que de esta manera los dispositivos puedan realizar intercambio de mensajes de forma segura evitando de esta forma un posible ataque, como por ejemplo un ataque Man in the Middle. Al hacer uso de la Raspberry Pi se cuenta con un sistema operativo basado en Linux en el cual se puede instalar el programa Open SSL, que permite al usuario generar certificados de autoridad de forma gratuita ya que es un programa basado en código abierto.

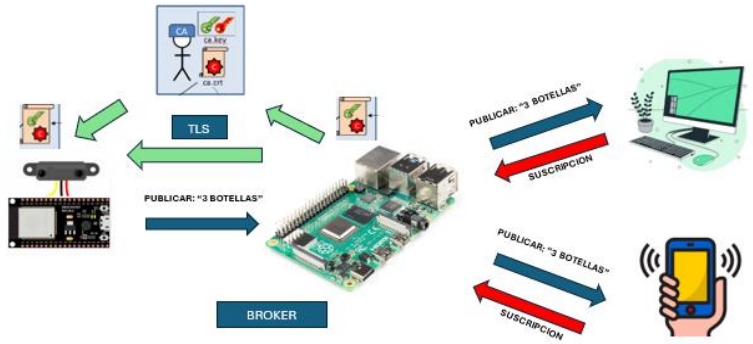

Figura 10. Topología MQTT TLS Elaborado por: Estefania Acosta & Anthony Sánchez

A continuación, se detalla el proceso a seguir en cada dispositivo de la topología, para la obtención de resultados.

6.3.1 Creación de Certificado en OpenSSL

En el bróker Raspberry pi, el cual utiliza el sistema operativo Rasbian creado por Linux, se instala Open SSL, ya que por medio de este programa se va a crear los certificados de encriptación. Para la creación del certificado se requiere contestar ciertas preguntas de seguridad como:

- País en 2 letras
- Provincia
- Ciudad
- Empresa
- Organización
- Nombre
- Correo electrónico

Este certificado es único ya que la única persona que puede tener esta información es quien creo el certificado, a continuación, se muestra en la Figura 11 la creación del certificado.

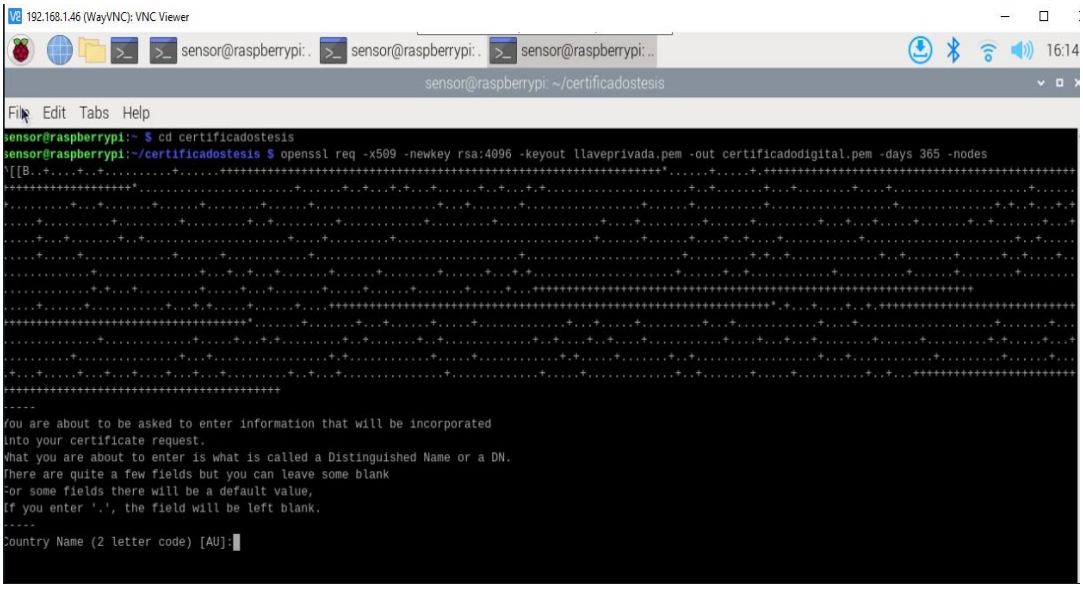

Figura 11. Creación de certificado en Open SSL Elaborado por: Estefania Acosta & Anthony Sánchez

6.3.2 Creación de llave privada en Open SSL

A continuación, en la Figura 12, se crea la llave privada la cual se va a colocar en el servidor como se muestra en la Figura 17. Esta llave es generada con 4096 bits por la autoridad certificadora.

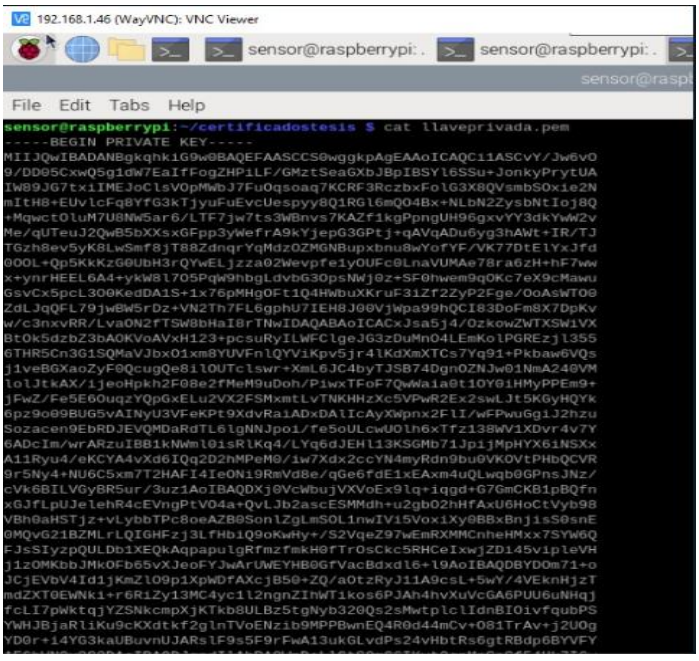

Figura 12. Llave privada configurada en la Raspberry Pi Elaborado por: Estefania Acosta & Anthony Sánchez

6.3.3 Creación de certificado privado en Open SSL

En la configuración del Esp32 después de seleccionar la librería , colocar credenciales de acceso del wifi del hogar y configurar el sensor Sharp, se crea un certificado privado en Open SSL como se puede observar en la Figura 13, este se va a colocar en la programación del ESP32, el cual tiene configurado el sensor Sharp para el coteo del embotellamiento diario, como se muestra en la Figura 14, Este certificado también se coloca en el servidor como se muestra en la Figura 16, este certificado es creado por la autoridad certificadora.

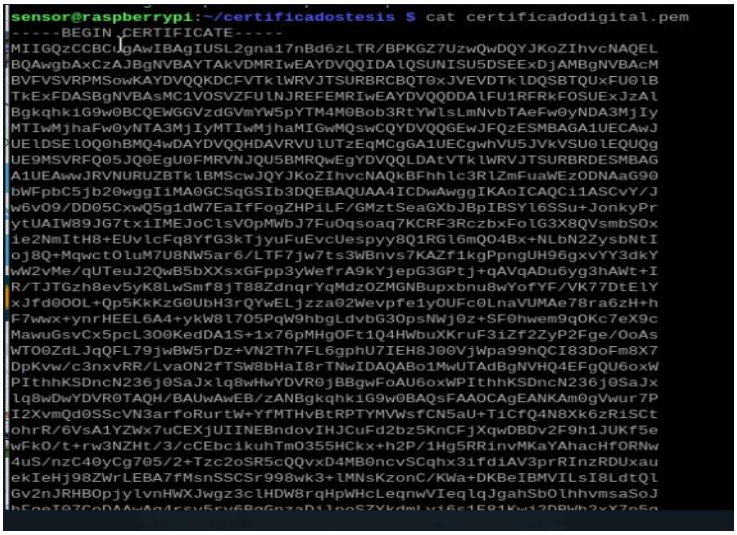

Figura 13. Certificado privado creado en Open SSL Elaborado por: Estefania Acosta & Anthony Sánchez

```
#ifndef SECRET
  const char ssid[] = "CELERITY SOFY";
  const char pass[] = [redacted]

  #define HOSTNAME "esp2[redacted]

  const char *MQTT_HOST = "192.168.1.46";
  const int MQTT_PORT = 8883;
  const char *MQTT_USER = ""; // leave blank if no credentials used
  const char *MQTT_PASS = ""; // leave blank if no credentials used

  const char* local_root_ca = R"EOF(
    -----BEGIN CERTIFICATE-----
MIIDgTCCAmmgAwIBAgIUXA94QzEG86iAmt+9XNWvgJJxOQIwDQYJKoZIhvcNAQEL
BQAwUDELMAkGA1UEBhMCRUMxEzARBgNVBAgMClNvbWUtU3RhdGUxDDAKBgNVBAoM
A01PVDEPMA0GA1UECwwGY2xpZW50MQ0wCwYDVQQDDAR0ZXN0MB4XDTIzMDMyMzIz
NTQwNloXDTI4MDMyOTIzNTQwNlowUDELMAkGA1UEBhMCRUMxEzARBgNVBAgMClNv
bWUtU3RhdGUxDDAKBgNVBAoMA01PVDEPMA0GA1UECwwGY2xpZW50MQ0wCwYDVQQD
DAR0ZXN0MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0SgfV2OWhohn
SeujTZADRjqRo09+S9FclVnT3Gows4Am+IWzg03j/eZrrJPjRZvEHpSzTOb0c/lX
b+avhuOSvc6howyVAle7Uymbj8wO/i7wbVU9M3KLaW73jXw9HrzkDn9ce2r5gKZq
T8qNwbe8F4ZSDJkxJiMawJaPlk6LOLaTmd3uzIlbzy0sMFbMV9wzx0bDDw32mC6v
PgBphmmSh+mzv262GCzHRp2F9ER9WklWd4OOZt7Mk/ghrCrJUfIHFI/xx7Kou6Qk
84d+5Ux2jppeTseyGYyQ33E4d0WBwDE8vZjZWqScuAIb1u+6o2letYnxZRmKBS4S
RDgCIki6pQIDAQABo1MwUTAdBgNVHQ4EFgQUCwKvVsU28P/SJycChTZqs7KLawkw
HwYDVR0jBBgwFoAUCwKvVsU28P/SJycChTZqs7KLawkwDwYDVR0TAQH/BAUwAwEB
/zANBgkqhkiG9w0BAQsFAAOCAQEAdeJ1bsJbkQed0jflssHlFOMN7CzNbSHvHl1h
l5ACLqZPlIU+Kmxna8npAcTA/+lOfbZ6qDVoIsJlNsbKYiKwG+TvfWm7TKrNeJDM
ssq6StIPmkXrpg89xDvufQMIRldzNFZF7YGZbqugDcI69JWTS49aJ7jdndDFOTfS
```

Figura 14. Certificado privado colocado en cliente. Elaborado por: Estefania Acosta & Anthony Sánchez

Se crea una carpeta en la Raspberry pi donde se coloca todos los certificados como se muestra en la Figura 15 y en Figura 16 se puede observar la colocación de todos los certificados, utilizando la dirección de la carpeta contenedora de los certificados.

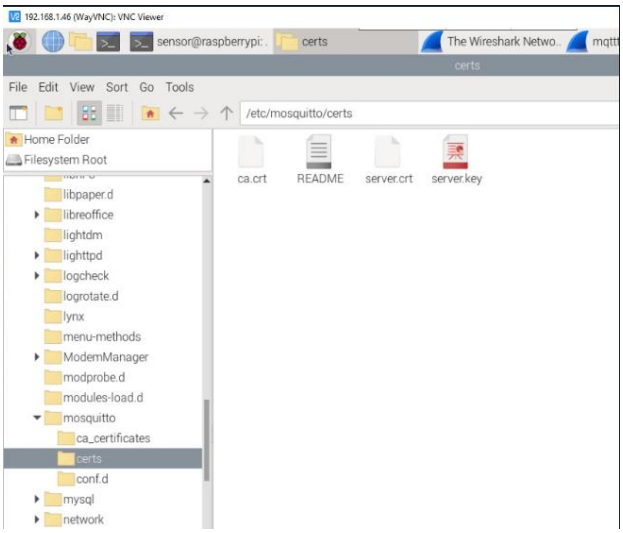

Figura 15. Carpeta de certificados Elaborado por: Estefania Acosta & Anthony Sánchez

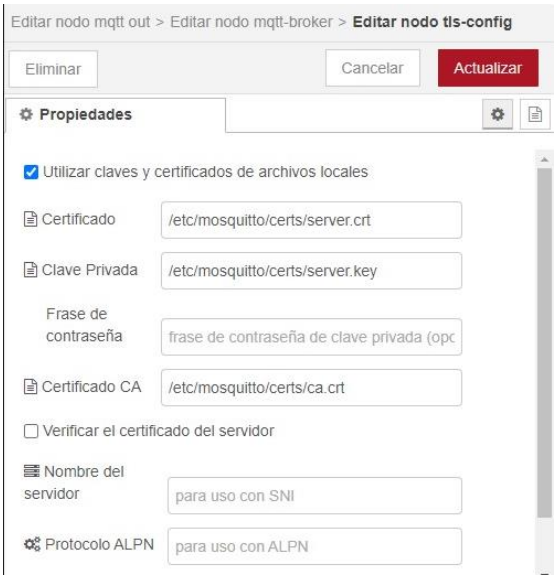

Figura 16. Certificados colocados en Node-RED Elaborado por: Estefania Acosta & Anthony Sánchez

6.3.4 Configuración de MQTT TLS en Raspberry pi

Para esta programación se utiliza el programa Node-RED en el cual se realiza el protocolo MQTT TLS como se muestra en la Figura 17, en la Figura 18 se puede observar que se coloca el puerto 8883, ya que este él es puerto seguro que utiliza MQTT TLS y se configura la dirección ip del servidor.

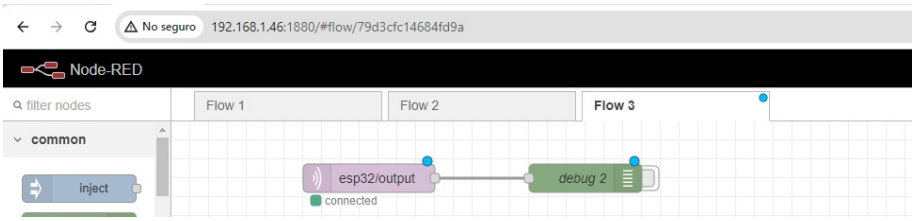

Figura 17. Node-RED protocolo MQTT TLS Elaborado por: Estefania Acosta & Anthony Sánchez

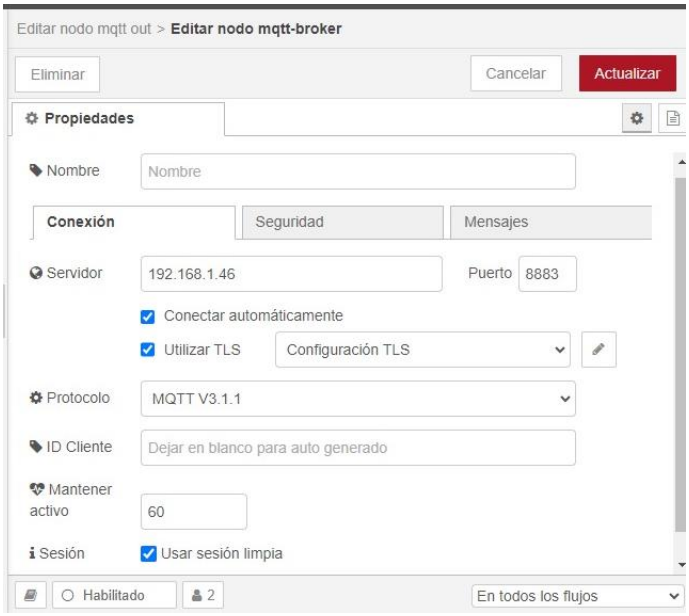

Figura 18. Configuración de puerto seguro en Node-RED Elaborado por: Estefania Acosta & Anthony Sánchez

A continuación, se visualiza en la Figura 19, el envío de datos del Esp32 a las Raspberry pi por medio del protocolo MQTT, para colocarle la seguridad y convertirse en MQTT TLS y realizar la prueba con el programa Wireshark.

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4
Attempting to connect to SSID: [redacted]
...
Connected to [redacted]
Setting time using SNTP.
Current time: Thu Jul 25 22:59:41 2024
Time:Thu Jul 25 17:59:41 2024
MQTT connectingconnected
135.00
136.00
```

Figura 19. Visualización de datos enviados de cliente a servidor Elaborado por: Estefania Acosta & Anthony Sánchez

6.3.5 Cambio de puerto inseguro a puerto seguro

Como se visualiza en la Figura 20, por medio del protocolo MQTT se tiene el puerto 1883 y para que sea seguro se debe colocar el puerto 8883, para ello se procede a bloquear el puerto 1883 y activar el puerto 8883.

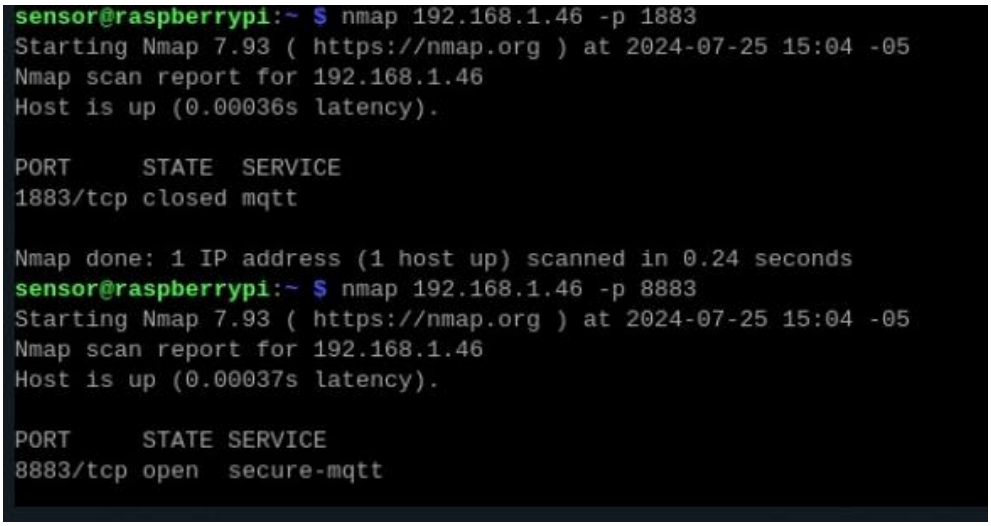

Figura 20. Bloqueo de puerto 1883 Elaborado por: Estefania Acosta & Anthony Sánchez

6.3.6 Prueba de viabilidad del protocolo MQTT TLS

Para realizar una prueba de este protocolo se utilizó el software Wireshark el cual permite validar los paquetes que se envían desde el Esps32 y la Raspberry pi, como se muestra en la Figura 21, se visualiza que se está utilizando el protocolo MQTT TLSv1.2 el cual encripta los mensajes.

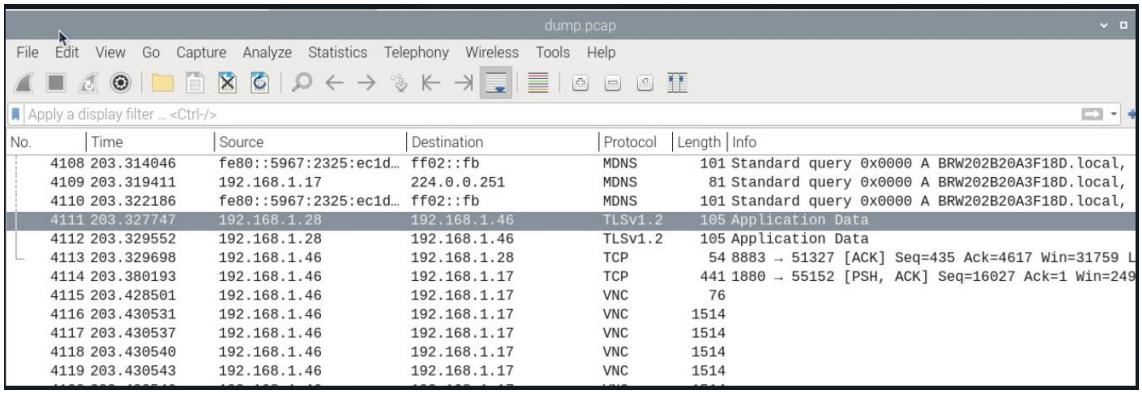

Figura 21. Validación de protocolo MQTT TLS Elaborado por: Estefania Acosta & Anthony Sánchez

6.4 Topología HTTP

A continuación, se muestra en Figura 22 el protocolo de transferencia de hipertexto (HTTP), el cual funciona de la misma manera que el protocolo MQTT mencionado con anterioridad, los clientes envían una solicitud http al servidor que en este caso es la Raspberry Pi, la cual está programada con Node-Red que es un programa para crear una interfaz de visualización de resultados entre dispositivos, este protocolo utiliza en Post HTTP para publicar lectura del sensor Sharp en el servidor y también este conformado por el HTTP Get que se utiliza para solicitar los datos obtenidos por es Esp32, en este protocolo se debe considera que al utilizar HTTP Get los valores recolectados serán visibles en la solicitud de URL y cuando se utiliza el HTTP Post los valores no son visibles en la solicitud URL, pero si no se encuentra con la encriptación los valores se podrían visualizar en el cuerpo de la solicitud es decir en la misma URL.

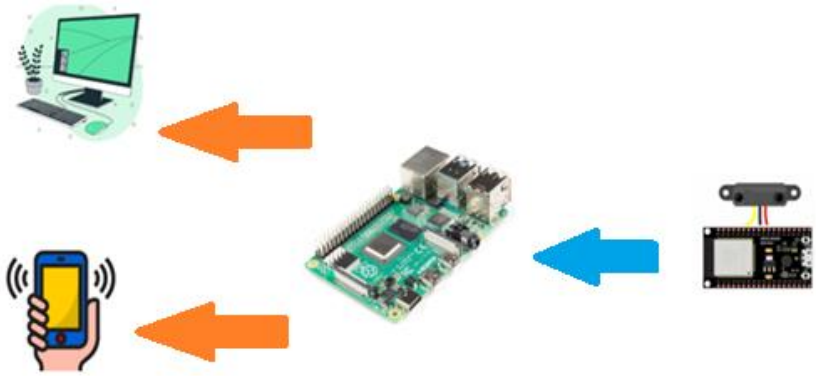

Figura 22. Topología HTTP Elaborado por: Estefania Acosta & Anthony Sánchez

A continuación, se detalla el proceso a seguir en cada dispositivo de la topología, para la obtención de resultados.

6.4.1 Programación del módulo esp32

A continuación, en la Figura 23 se puede visualizar la configuración del cliente o Esp32 en el cual se selección la librería de http, se coloca credenciales de acceso al wifi del hogar y se configura el sensor Sharp.

http_v1

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <Arduino_JSON.h>

const char* ssid = "CELERITY_SOFY";
const char* password = "$%Thepolice";

//Your Domain name with URL path or IP address with path
const char* serverName = "http://192.168.1.46:1880/get-sensor";

// the following variables are unsigned longs because the time, measured in
// milliseconds, will quickly become a bigger number than can be stored in an int.
unsigned long lastTime = 0;
// Timer set to 10 minutes (600000)
//unsigned long timerDelay = 600000;
// Set timer to 5 seconds (5000)
unsigned long timerDelay = 5000;

String sensorReadings;
float sensorReadingsArr[3];

void setup() {
  Serial.begin(115200);

  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
```

Figura 23. Configuración del cliente Elaborado por: Estefania Acosta & Anthony Sánchez

6.4.2 Programación del servidor o Raspberry pi

En la Figura 24, se puede visualizar la programación del protocolo http, en Node-RED y en el Figura 26 se valida el funcionamiento de envío de datos desde el cliente al servidor.

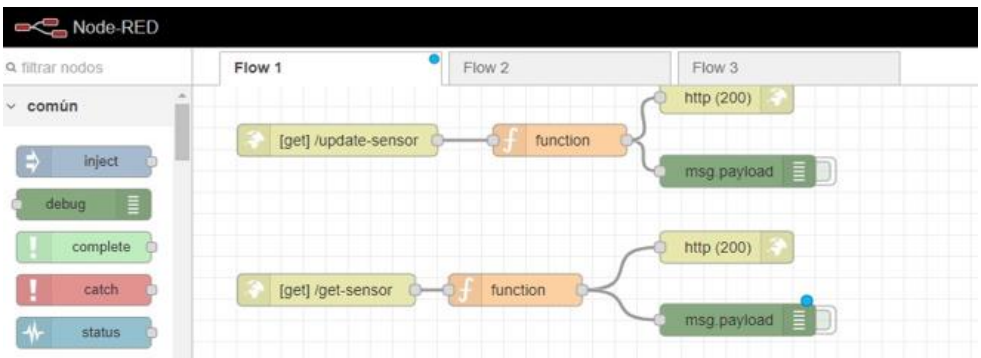

Figura 24. Configuración del cliente en Node-RED Elaborado por: Estefania Acosta & Anthony Sánchez

Figura 25. Visualización de envío de datos Elaborado por: Estefania Acosta & Anthony Sánchez

6.4.3 Prueba de viabilidad del protocolo http

A continuación, se realiza un ataque MITM, por medio del software Wireshark, el cual permite la validación de envío de paquetes como se muestra en la Figura 26, utilizando el protocolo http

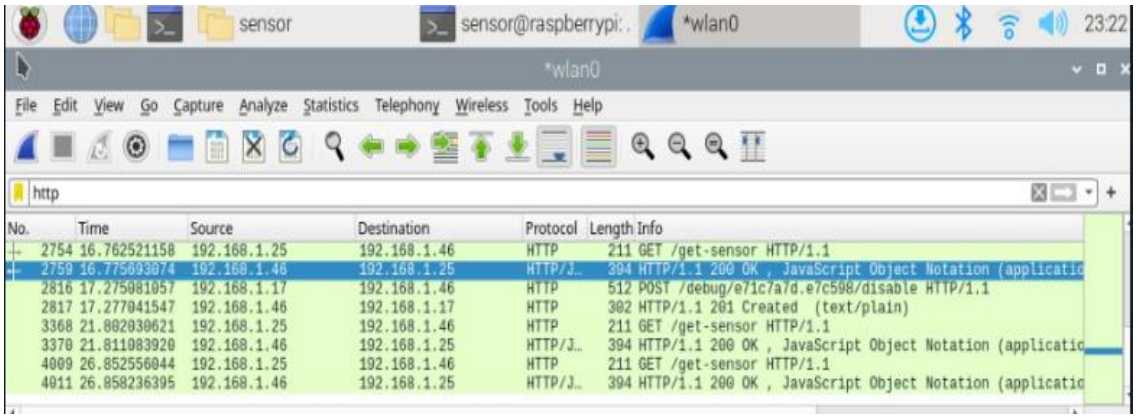

Figura 26. Validación de puertos abiertos y direcciones ip Elaborado por: Estefania Acosta & Anthony Sánchez

6.5 Topología COAP

El Protocolo COAP como se observa en la Figura 27 es la evolución del protocolo http, los dos protocolos tienen mucho en común cuando en estructura se refiere, la capa de trasporte TCP es usada por el protocolo Http, mientras que el protocolo COAP lo realiza por medio de UDP y se puede tener comunicación con varios sensores, ya que es escalable, el código es liviano lo cual le hace apto para aplicaciones IIoT, el tamaño de los paquetes son menores comparado con HTTP.

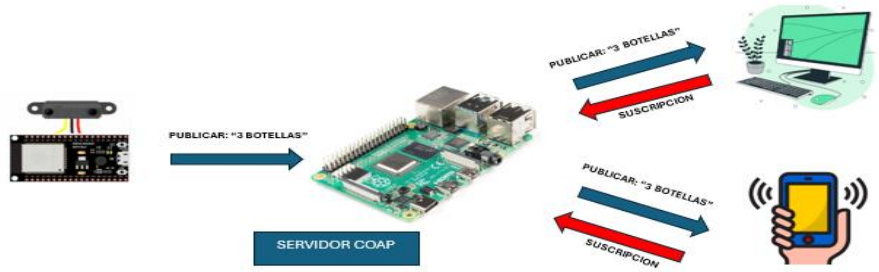

Figura 27. Topología COAP Elaborado por: Estefania Acosta & Anthony Sánchez

A continuación, se detalla el proceso a seguir en cada dispositivo de la topología, para la obtención de resultados.

6.5.1 Programación del módulo esp32

A continuación, en la Figura 28 se puede visualizar la configuración del cliente o Esp32, se utiliza la librería que tiene Arduino de COAP, para poder realizar la programación, se coloca credenciales de wifi del hogar y se configura el sensor Sharp.

```
esp32coap §
#include <WiFi.h>
#include <WiFiUdp.h>
#include <coap-simple.h>

const char* ssid     = "CE[redacted]
const char* password = "$%[redacted]

#define LED 2

// CoAP client response callback
void callback_response(CoapPacket &packet, IPAddress ip, int port);

// CoAP server endpoint url callback
void callback_light(CoapPacket &packet, IPAddress ip, int port);

WiFiUDP udp;
Coap coap(udp);

// LED STATE
bool LEDSTATE;
// CoAP server endpoint URL
void callback_light(CoapPacket &packet, IPAddress ip, int port) {
  Serial.println("[Light] ON/OFF");

  // send response
  char p[packet.payloadlen + 1];
  memcpy(p, packet.payload, packet.payloadlen);
  p[packet.payloadlen] = NULL;
```

Figura 28. Configuración del cliente con protocolo COAP Elaborado por: Estefania Acosta & Anthony Sánchez

6.5.2 Programación del servidor o Raspberry pi

Como se muestra a continuación en la Figura 29, se realiza la configuración del protocolo COAP en Node-RED y se realiza la programación del puerto como se visualiza en la Figura 30 y en la Figura 31, se puede validar el envío de datos del cliente al servidor.

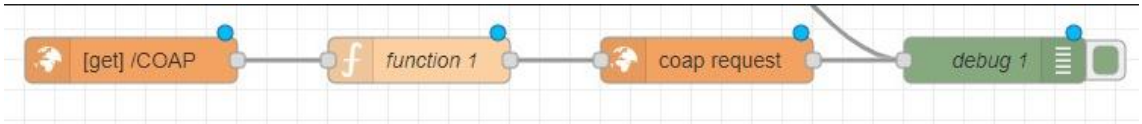

Figura 29. Configuración de protocolo COAP en Node-RED Elaborado por: Estefania Acosta & Anthony Sánchez

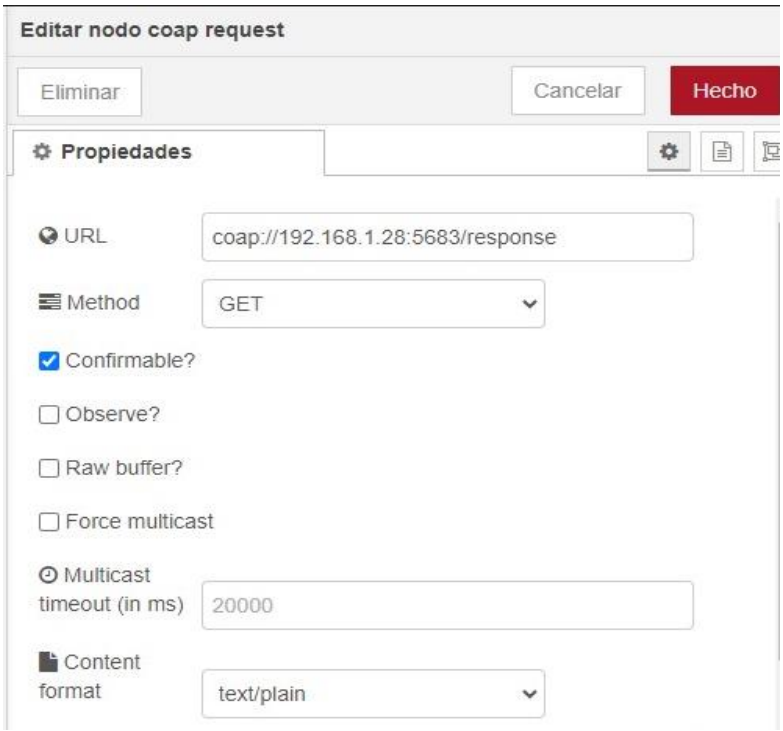

Figura 30. Programación de puerto de Node-RED

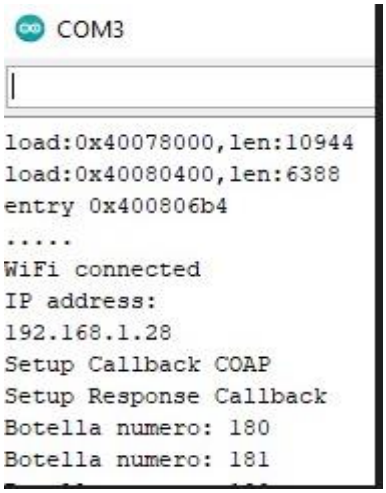

Figura 31. Envió de datos por medio de protocolo COAP Elaborado por: Estefania Acosta & Anthony Sánchez

6.5.3 Prueba de viabilidad del protocolo COAP

Se realiza un ataque con el uso de del software Wireshark con el método del hombre en el medio, para realizar la captura de datos como se muestra en la Figura 32 se puede validar el protocolo que se usa en este caso es el protocolo COAP

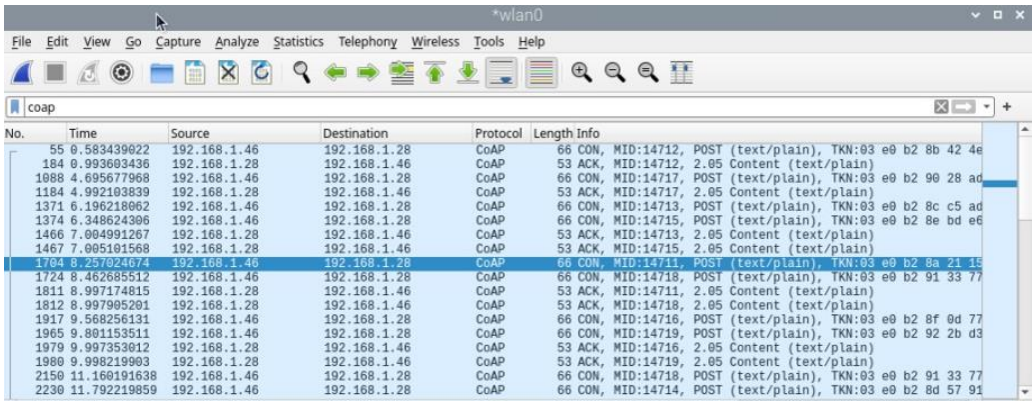

Figura 32. Ataque con software Wireshark Elaborado por: Estefania Acosta & Anthony Sánchez

7. Resultados

Después de haber creado un ambiente de prueba de la empresa embotelladora de agua, los datos se obtuvieron por medio del sensor Sharp, este sensor envía señales análogas al dispositivo Esp32, el cual mantiene un ADC incorporado que convierte las señales análogas en digitales, el valor digital es enviado hacia la Raspberry pi por medio de los protocolos que se van a mostrar a continuación:

El primer protocolo que se analizo es el MQTT, después de que se envió los datos de cliente a servidor, se realizó un ataque llamado hombre en el medio, con el uso del software Wireshark el cual captura los paquetes enviados desde el Esp32 a la Raspberry pi como se muestra en la Figura 33. Se pudo observar que los datos que fueron enviados de dispositivo a dispositivo fue el 135, el topic que se coloco es el esp32/output, el protocolo por el cual se envió el mensaje es el MQTT en capa TCP, la dirección ip del cliente fue 192.168.1.28, la dirección ip del servidor es 192.168.1.46, la versión, el puerto de origen que es el 1883, puerto de destino 61253, toda esta información se puede validar en el ataque realizado.

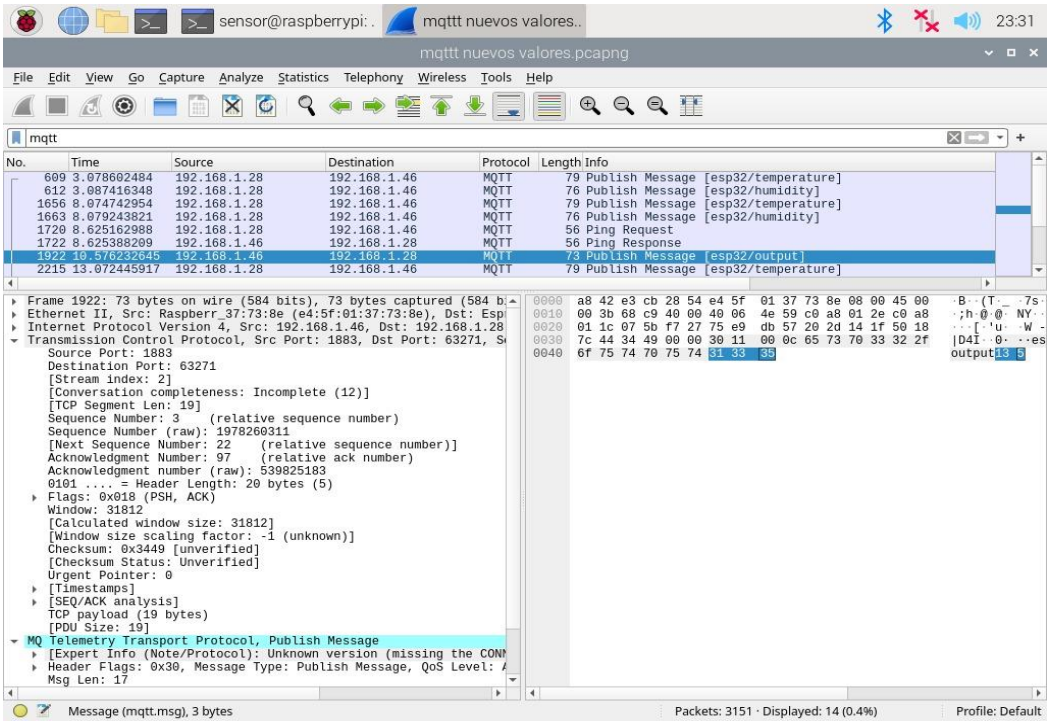

Figura 33. Captura de paquetes con el uso de Wireshak Elaborado por: Estefania Acosta & Anthony Sánchez

El segundo protocolo que se analizo fue el http de igual manera como se mencionó anteriormente, primero se envió el dato desde el esp32 a la Raspberry pi y se realizó un ataque por medio del uso del software Wireshark como se visualiza en la Figura 34. La información que se recolecta de este ataque es la visualización del protocolo que se usó en este caso es el http, con la capa de transmisión TCP, su versión es la 4, la ip de origen 192.168.1.25, ip de destino 192.168.1.46, el puerto de origen 1880, puerto de destino 57872 y los valores enviados que fueron el 150, 151, 152.

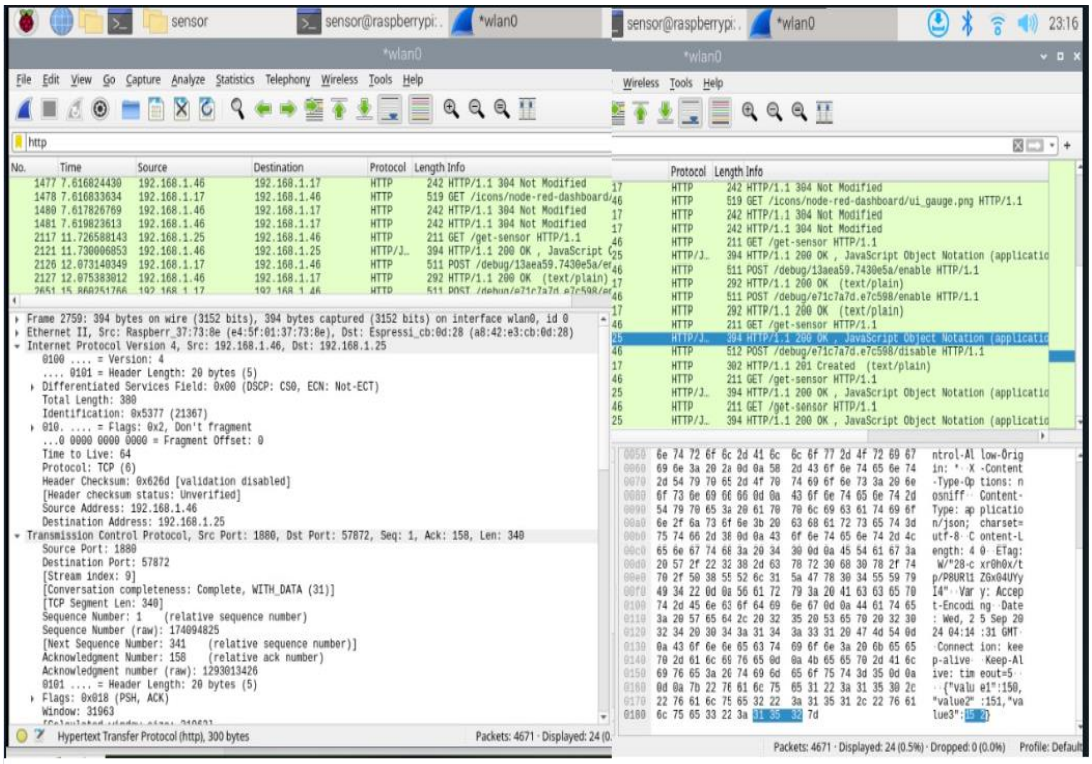

Figura 34. Captura de información en protocolo http Elaborado por: Estefania Acosta & Anthony Sánchez

El tercer protocolo que se analizo fue COAP, como hemos venido mencionando anteriormente primero se ejecuta el envío de datos desde el cliente al servidor y por medio del software Wireshark se realiza el ataque de obtención de datos como se muestra en la Figura 35 y Figura 36 , de este ataque la información que se puede recolectar es la versión del protocolo que es 4, la ip del cliente 192.168.1.46, la ip del servidor 192.168.1.28, puerto de origen 40529 , puerto de destino 5683 y para finalizar el valor que se envió fue 180.

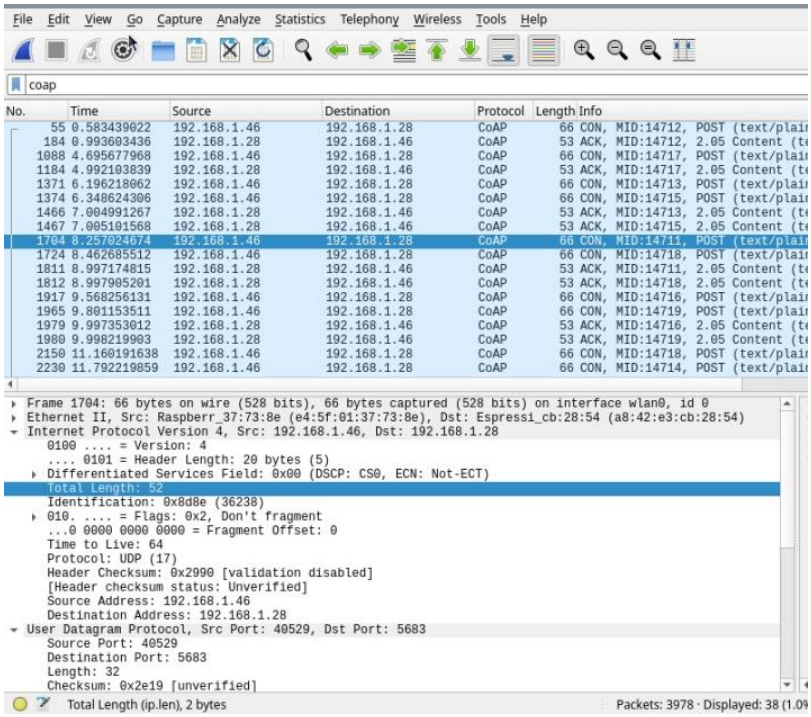

Figura 35. Captura de datos con Wireshark en protocolo COAP Elaborado por: Estefania Acosta & Anthony Sánchez

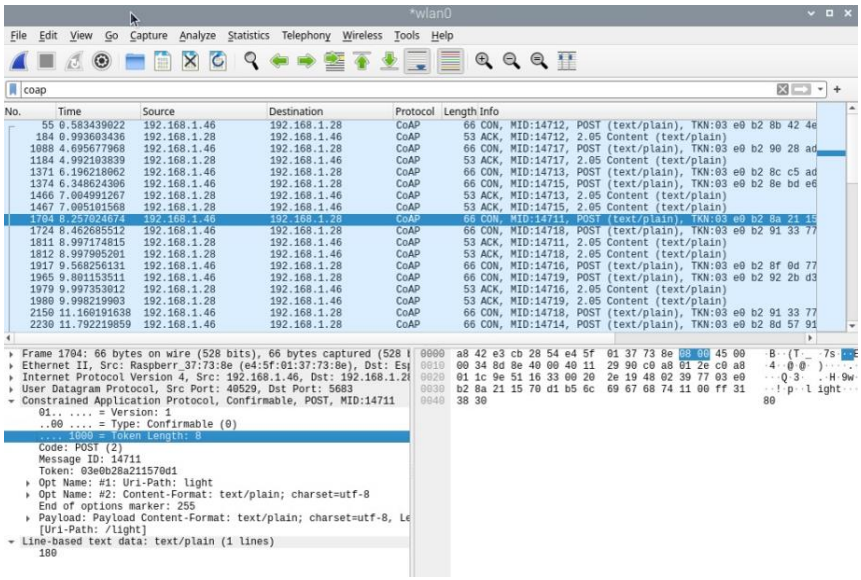

Figura 36. Captura de información enviada de cliente a servidor Elaborado por: Estefania Acosta & Anthony Sánchez

Como último protocolo a analizar fue el protocolo MQTT TLS, de la misma manera que los protocolos anteriores, primero se debe enviar el dato de cliente a servidor y por medio de un ataque MITM, se procede a recolectar la información enviada, con el uso del software Wireshark, como se muestra en la Figura 37, gracias a este ataque se puede visualizar que el protocolo que se está utilizando es el MQTT con versión

incorporada de TLS versión 1.2, Se visualiza que la ip del cliente es 192.168.1.28, la ip de servidor es 192.168.1.46, se visualiza que el puerto de origen es 51327 y el puerto de destino es 8883 el cual es el puerto seguro ya que es el de encriptación y acerca de los paquetes que fueron enviados, se valida que toda esta información esta encriptada y tiene los certificados validados por una autoridad certificadora, estos certificados contienen una llave publica la cual se genera con datos personales del individuo , algoritmos establecidos y archivos firmados por la autoridad certificadora o CA, con todos los requisitos mencionados anteriormente, se puede asegurar que estos certificados se generan de una manera que sean únicos y seguros, al momento de colocar la llave en el ESP32 o cliente, y este quiera saber alguna información, primero se debe validar con el del servidor o Raspberry pi , los certificados y tiene que existir una compatibilidad, caso contrario no permite el acceso de visualización a la información, de esta forma se asegura de mantener la confidencialidad, integridad, disponibilidad de los datos.

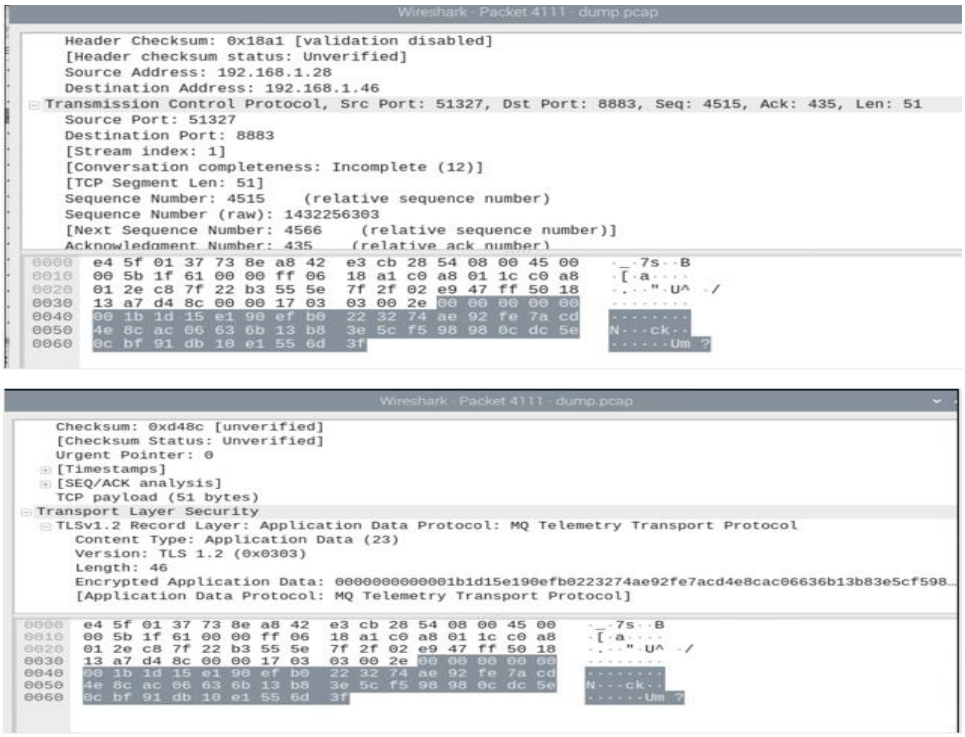

Figura 37. Captura de paquetes por medio del protocolo MQTT TLS Elaborado por: Estefania Acosta & Anthony Sánchez

Como parte final, en base a las pruebas realizadas con Wireshark, a continuación, se indican los hallazgos principales a manera de comparación entre cada protocolo tomando en cuenta algunos aspectos o métricas en términos de ciberseguridad, como,

por ejemplo: escalabilidad, calidad de servicio o fiabilidad, latencia, autenticación entre clientes con el uso de software libre, como se muestra en la siguiente Figura 38 y Figura 39.

MQTT	MQTT-TLS
- Utiliza el protocolo TCP - El proceso de comunicación es del tipo publicación-suscripción - Es eficiente y tiene baja sobrecarga - Es altamente escalable - En fiabilidad y calidad de servicio admite niveles de Qos es decir QoS, QoS 1, QoS2 - Mantiene conectividad uno a uno, de uno a muchos y de muchos a muchos. - Tiene baja latencia, aunque los mensajes sean largos - Consume ancho de banda ya que los mensajes que se envía son largos - Se puede automatizar hogares e Industria	- Utiliza el protocolo TCP - La comunicación es publicación-suscripción - Es eficiente y tiene baja sobrecarga - Es altamente escalable - En seguridad garantiza la autentificación, integridad y encriptación por medio de una autoridad certificadora TLS - En fiabilidad y calidad de servicio admite niveles de QoS, es decir QoS, QoS 1, QoS2 - Mantiene conectividad uno a uno, de uno a muchos y de muchos a muchos. - Tiene baja latencia - Consume ancho de banda ya que los mensajes que se envía son largos - Se puede automatizar hogares e Industria

Figura 38. Comparación entre MQTT y MQTT TLS Elaborado por: Estefania Acosta & Anthony Sánchez

HTTP	COAP
- Utiliza protocolo TCP - La comunicación es Solicitud- Respuesta - Es menos eficiente y mantiene una mayor sobrecarga - Es escalable - En fiabilidad y calidad de servicio admite opciones de fiabilidad, pero son limitados ya que usa el protocolo TCP - Mantiene conectividad uno a uno - Tiene alta latencia - Necesita mayor ancho de banda - Es una aplicación web	- Utiliza protocolo UDP - La comunicación es pregunta- respuesta - Es eficiente y tiene una baja sobrecarga - Es escalable - En fiabilidad y calidad de servicio admite opciones de fiabilidad que son mensajes confirmable, mensaje no confirmable. - Mantiene conectividad uno a uno y de muchos a muchos. - Presenta tiempos de latencia bajos, sin embargo, este tiempo aumenta si existen mensajes largos - Tiene ancho de banda baja ya que los mensajes son cortos - Permite automatizar diferentes entornos

Figura 39. Comparación de HTTP y COAP Elaborado por: Estefania Acosta & Anthony Sánchez

Luego de indicar los hallazgos principales y realizadas las pruebas por medio del software Wireshark, a continuación, se indica una comparación final de cada protocolo en términos de ciberseguridad: Disponibilidad, Confidencialidad e Integridad, como se muestra en la Figura 40, finalizando la investigación y puesta a prueba de cada protocolo de comunicación utilizado dentro de un entorno IIoT de la Industria 4.0

Protocolos / Términos	MQTT	HTTP	COAP	MQTT TLS
Disponibilidad	X	X	X	X
Integridad	X	X	X	X
Confidencialidad				X

Figura 40. Comparación de los protocolos en términos de ciberseguridad

8. CONCLUSIONES

1. Como se pudo comprobar en esta investigación es importante mantener medidas de seguridad en los dispositivos IIot en la Industria, ya que los desafíos que enfrentan estos entornos frente a ataques cibernéticos, cada día son más desafiantes, porque se crean nuevas técnicas más sofisticadas que usan los ciberdelincuentes para robar la información que se transmite entre los dispositivos o incluso alterarla, por lo cual se necesita que exista un protocolo de comunicación que cumpla con los tres pilares de seguridad que son disponibilidad, integridad y confidencialidad en la transmisión de datos.
2. Las vulnerabilidades más recurrentes que se encontraron en esta investigación fueron: puertos abiertos, versiones de software desactualizadas, credenciales de autentificación débiles, estas vulnerabilidades pueden ser detectadas por los ciberdelincuentes, con esta información ellos lograrían acceder a los dispositivos e infiltrarse a los sistemas , para obtener los datos enviados entre dispositivos o incluso detener un proceso crítico dentro de la infraestructura industrial, por ello es importante mantener encriptada esta información, con el uso de puertos seguros, software actualizados y seguir políticas de seguridad apropiadas acorde al entorno de la Industria 4.0.
3. En IIot en la industria 4.0, se utiliza 2 técnicas de cifrado que son: SSL y TLS, siendo TLS la más usada con respecto a SSL, debido a que SSL es una versión antigua que tiene algunas falencias en comparación a TLS, cabe destacar que TLS utiliza certificados de encriptación y desencriptación de datos, que son creados por una autoridad certificadora CA, por lo tanto, TLS es la técnica de cifrado más viable y eficaz para combinarla con los protocolos de comunicación estudiados en esta investigación como: MQTT, HTTP y COAP, ya que TLS utiliza certificados únicos, creados con información personal del usuario, lo que permite mantener el nivel adecuado de seguridad en la comunicación de los dispositivos y así evitar ataques cibernéticos.
4. Al realizar las pruebas de viabilidad de los protocolos: MQTT, MQTT TLS, HTTP y COAP y analizar todos los resultados obtenidos, se pudo concluir que el

protocolo MQTT, HTTP , COAP y MQTT TLS cumplen con los términos de seguridad como Disponibilidad e Integridad pero el protocolo MQTT TLS es el único protocolo que cumplió con confidencialidad de los datos y se determinó en los hallazgos más importantes que MQTT TLS posee mejores características y ventajas, en cuanto a seguridad, escalabilidad, fiabilidad, calidad de servicio, latencia, etc. Por lo cual se puede decir, que en términos de seguridad el protocolo MQTT TLS es el más seguro, eficiente y viable para ser aplicado dentro de una Industria 4.0 con IIoT, con el uso de software libre.

Referencias

Ávila-Camacho, F. J., & Moreno-Villalba, L. M. (2023). Internet de las Cosas (IoT) Retos para las Empresas en la era de la Industria 4.0. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías Del ICBI*, *10*(20), 10–16. https://doi.org/10.29057/icbi.v10i20.9516

Ayerbe, A. I. (2018). *La Ciberseguridad de la Industria 4.0: Un Medio para la continuidad del negocio*.

Caiza, G., Alvarez-M, E., Remache, E., Ortiz, A., & Garcia, M. V. (2019). *Comparación de AMQP y CoAP para la integración de las comunicaciones en el área de producción*.

Castro Heredia, J., & Losilla López, F. (2014). *Uso del protocolo CoAP para la implementación de una aplicación domótica con redes de sensores inalámbricas*.

Colazo Ornella, & Fabbri Lucia. (2023). *Analisis del Estado de la Ciberseguridad del Internet Industrial de las Cosas IIoT CameraReady*.

Cruz Lucas, G. I., Galarza Espinoza, R. E., Delgado De La Cruz, R. S., & Marcillo Merino, M. J. (2022). Aplicación de protocolos SSL y TSL para él envió de información. *Journal TechInnovation*, *1*(2), 4–9. https://doi.org/10.47230/journal.techinnovation.v1.n2.2022.4-9

Digicert. (2024). *Digicert*. Https://Www.Digicert.Com/Es.

Hoffman, F. (2019). INDUSTRIAL INTERNET OF THINGS VULNERABILITIES AND THREATS: WHAT STAKEHOLDERS NEED TO CONSIDER. *Issues in Information Systems*, *20*(1), 119–133. https://doi.org/10.48009/1_iis_2019_119-133

Joyanes Aguilar, L. (2018). *Capítulo primero Ciberseguridad: la colaboración público-privada en la era de la cuarta revolución industrial (Industria 4.0 versus ciberseguridad 4.0)*. http://tecnologia.elpais.com/

Kabachinski Jeff. (n.d.). *An Introduction To RFID*.

Let´s Encrypt. (2024). *Let´s Encrypt*. Https://Letsencrypt.Org/Es/.

Muralles Eduardo. (2020). *Diseño de Protocolo de IoT basado en Http para el control de actuadores y sensores en Domótica*.

Neco Villegas Saiz. (2021). *Comportamiento de Protocolos de Transporte en entornos IIoT*.

Open SSL Corporation. (2024). *Open SSL*. Https://Openssl.Org/.

Rodríguez Llerena, A. E. (n.d.). Herramientas fundamentales para el hacking ético Fundamental Tools for Ethical Hacking. In *Revista Cubana de Informática Médica* (Vol. 2020, Issue 1). http://scielo.sld.cu

Sotolani, R. S., Menezes, I. D. A. C., Galegale, N. V., & Feitosa, M. D. (2022). Vulnerabilidades de Segurança da Informação na Indústria 4.0: Proposição de Critérios para o uso de Análise Multicritério. *Exacta*. https://doi.org/10.5585/exactaep.2022.21683

Steve Gómez-Meza, J., Vanessa, S., Teddy, M.-A., & Negrete-Peña, J. (n.d.). *Diseño de un prototipo IoT para el monitoreo de material particulado*. https://orcid.org/0000-

Tavares Jonathas. (2024). *DESENVOLVIMENTO DE UMA MICRO-BLOCKCHAIN PRIVADA PARA COLETA DE DADOS DE DISPOSITIVOS IIOT*.

Valencia, A., & Portilla, P. (2019). *Internet Industrial de las Cosas (IIOT): Nueva Forma de Fabricación Inteligente*.

Valeske, B., Osman, A., Römer, F., & Tschuncky, R. (2020). Next Generation NDE Sensor Systems as IIoT Elements of Industry 4.0. In *Research in Nondestructive Evaluation* (Vol. 31, Issues 5–6, pp. 340–369). Bellwether Publishing, Ltd. https://doi.org/10.1080/09349847.2020.1841862