UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE INGENIERÍA DE SISTEMAS

COMPARACIÓN EXPERIMENTAL DE PROTOCOLOS DE COMUNICACIÓN DE IOT EN CUANTO A LA SEGURIDAD DE TRANSMISIÓN DE DATOS EN DISPOSITIVOS IOT DE COBERTURA WLAN

Trabajo de Titulación previo a la obtención del

Título de Ingenieros de Sistemas

AUTORES: RICHARD GEOVANNY BRACHO GUAMANÍ

DIEGO EDUARDO GUAÑUNA ANDI

TUTOR: MANUEL RAFAEL JAYA DUCHE

Quito – Ecuador

2023

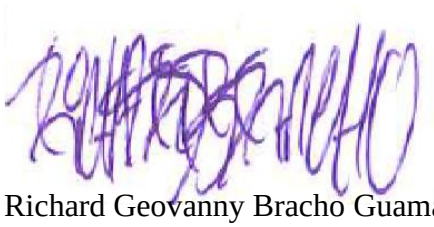
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Richard Geovanny Bracho Guamaní con documento de identificación N° 1723439079 y Diego Eduardo Guañuna Andi, y N° 1723110761; manifestamos que:

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

Quito, 01 febrero de 2023

Atentamente



Richard Geovanny Bracho Guamaní
1723439079

Diego Eduardo Guañuna Andi
1723110761

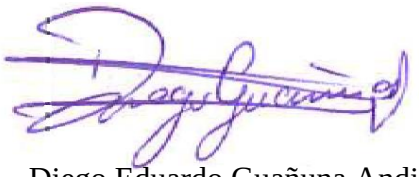
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Richard Geovanny Bracho Guamaní con documento de identificación N° 1723439079 y Diego Eduardo Guañuna Andi, y N° 1723110761, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Artículo Académico: "Comparación experimental de protocolos de comunicación de IOT en cuanto a la seguridad de transmisión de datos en dispositivos IOT de cobertura WLAN", el cual ha sido desarrollado para optar por el título de: Ingenieros de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Quito, 01 febrero de 2023

Atentamente,



Richard Geovanny Bracho Guamaní
1723439079

Diego Eduardo Guañuna Andi
1723110761

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Manuel Rafael Jaya Duche con documento de identificación N.° 1710631035, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: COMPARACIÓN EXPERIMENTAL DE PROTOCOLOS DE COMUNICACIÓN DE IOT EN CUANTO A LA SEGURIDAD DE TRANSMISIÓN DE DATOS EN DISPOSITIVOS IOT DE COBERTURA WLAN, realizado por Richard Geovanny Bracho Guamaní con documento de identificación N.° 1723439079 y Diego Eduardo Guañuna Andi, y N.° 1723110761, obteniendo como resultado final el trabajo de titulación bajo la opción de Artículo Académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 01 de febrero de 2023

Atentamente,

Ing. Manuel Rafael Jaya Duche, Msc.

1710631035

DEDICATORIA

A mis padres, quien con sacrificio y esfuerzo lograron ayudarme a seguir creciendo profesionalmente y darme todo lo necesario para seguir adelante y no permitirme estancarme, gracias a ellos soy quien soy y seguiré dando lo mejor de mi cada día para que estén orgullosos de mí. Gracias por todo Jhonson Bracho y Lucia Guamaní los amo con todo mi corazón.

A mi esposa e hijo quien con su amor me ayudan a querer superarme cada día para darles lo mejor de mí, gracias por estar siempre a mi lado y seguir creciendo juntos, así como lo hemos hecho desde el Colegio con mi esposa ahora veremos juntos los logros que nuestro hijo realizara y cosechara, los amo infinitamente Daniela Cevallos y Mathias Bracho.

Este título es para ustedes.

Richard Geovanny Bracho Guamaní

El presenta trabajo de titulación está dedicado principalmente a mis padres Teresa y Vicente, ya que con sus consejos y su apoyo incondicional me permitieron culminar mis estudios universitarios, dándoles gracias por toda la confianza que me brindaron en esta etapa de mi vida, así también sin olvidar a mis hermanos Fernando, Juan y Manuel quien con sus palabras de aliento me ayudaron a no rendirme a seguir a delante sin importar lo que cueste, porque al final siempre habrá una gran recompensa. A mis sobrinos porque siempre creyeron en mi siendo yo un gran ejemplo para ellos.

Y por último doy gracias a Dios por haberme otorgado la fuerza, el sacrificio y sabiduría que se necesita para asumir este camino que está culminando de forma exitosa.

Diego Eduardo Guañuna Andi

AGRADECIMIENTOS

A mis padres quien sin ellos no estaría donde estoy gracias por su apoyo incondicional que me han brindado no solo a mi si no a mi hijo y esposa para seguir creciendo.

A Daniela, mi esposa y mejor amiga, gracias por tu amor, por acompañarme durante todo este proceso no solo mío si no de ambos al crecer profesionalmente como lo hemos hecho desde el colegio.

A Mathias, mi hijo por darme fuerza e inspiración para seguir adelante cada día de mi vida.

A mis hermanos Sebastián, Caro, Ana, por su ayuda, por molestarlos cada que necesitaba de su ayuda gracias por todo gracias por seguir ahí y ayudarme como siempre lo han hecho.

Richard Geovanny Bracho Guamaní

A mis padres por haberme forjado como la persona que soy, este logro es para ellos quienes me motivaron constantemente a alcanzar mis metas.

A mis hermanos y a mis sobrinos que con sus voces de aliento me ayudaron a seguir adelante cada día permitiéndome culminar una etapa importante en mi vida profesional.

Diego Eduardo Guañuna Andi

COMPARACIÓN EXPERIMENTAL DE PROTOCOLOS DE COMUNICACIÓN DE IOT EN CUANTO A LA SEGURIDAD DE TRANSMISIÓN DE DATOS EN DISPOSITIVOS IOT DE COBERTURA WLAN

EXPERIMENTAL COMPARISON OF IOT COMMUNICATION PROTOCOLS IN TERMS OF DATA TRANSMISSION SECURITY IN WLAN COVERAGE IOT DEVICES

Richard Bracho[1], Diego Guañuna[2], Rafael Jaya[3]

Resumen:
Este artículo tiene como objetivo realizar un análisis de los protocolos de seguridad IoT dentro de una red WAN, mediante una metodología experimental y sistemática, basada en la recolección de información para la selección de los protocolos. Estos protocolos serán sometidos a un ataque ARP Spoffing para visualizar las vulnerabilidades que se pueden presentarse en cada protocolo, la información recolectada se realiza mediante el sniffer wireshark que permitió la captura de paquetes para su análisis. Al realizar el procedimiento se determina que en los protocolos MQTT, COAP, WEBSOCKET se pudo ejecutar un envenenamiento ARP el cual capturo información en texto plano como la interacción que se realiza mediante métodos GET y POST. Se pudo visualizar los puertos de comunicación sobre los cuales está interactuando el dispositivo IoT y la plataforma con el usuario. De acuerdo con las pruebas realizadas se determinó que el protocolo más seguro es MQTT, debido a que presentó menores vulnerabilidades en las pruebas realizadas, es decir, que la interacción que se realizó entre este protocolo y la plataforma Thinger.io, mostro datos encriptados por el protocolo TLS/SSL a pesar de ser capturados por el sniffer.

Palabras Clave: Sniffer, Protocolo, Internet de las cosas, degradación, interfaz.

Abstract:
This article aims to perform an analysis of IoT security protocols within a WAN network, through an experimental and systematic methodology, based on the collection of information for the selection of protocols. These protocols will be subjected to an ARP Spoofing attack to visualize the vulnerabilities that may occur in each protocol, the information collected is done using the wireshark sniffer that allowed the capture of packets for analysis. By performing the procedure, it was determined that in the MQTT, COAP, WEBSOCKET protocols it was possible to execute an ARP poisoning which captured information in plain text as the interaction that is performed by GET and POST methods. It was possible to visualize the communication ports on which the IoT device and the platform are interacting with the user. According to the tests performed, it was determined that the most secure protocol is MQTT, since it presented fewer vulnerabilities in the tests performed, i.e., the interaction between this protocol and the Thinger.io platform showed data encrypted by the TLS/SSL protocol despite being captured by the sniffer.

Keywords*:* Sniffer, Protocol, Internet of things, degradation, interface.

[1] Estudiante de Ingeniería de Sistemas – Universidad Politécnica Salesiana, Egresado – UPS – sede Quito
[2] Estudiante de Ingeniería de Sistemas – Universidad Politécnica Salesiana, Egresado – UPS – sede Quito
[3] Magister en Redes de Información y Conectividad, Ingeniero en Electrónica y Telecomunicaciones, Profesor Ingeniería en Sistemas, Ciencias de la Computación – UPS – sede Quito– Email: mjaya@ups.edu.ec

1. Introducción

El uso de dispositivos Internet de las Cosas (IOT), ha aumentado en los últimos años debido a la gran facilidad de uso que ofrece esta tecnología debido a que son artefactos comunes para el uso cotidiano o empresarial que se encuentran conectados al internet.

En los últimos años, la tecnología IoT ha experimentado un enorme crecimiento y se prevé que se expanda más rápidamente en el futuro aumentando a más de 16.000 millones de dispositivos conectados al IoT, sin contar los smartphones, las tabletas y los ordenadores. Esta rápida expansión de la IO es posible gracias a algunas tecnologías clave, como las redes inalámbricas de alta velocidad, los servicios en la nube de rápido crecimiento y los dispositivos inalámbricos de bajo coste. Por el lado de la demanda, está impulsada por las empresas que se esfuerzan por lograr un proceso de producción más eficiente, un tiempo de respuesta más rápido y una mayor satisfacción de los clientes, que son los beneficios potenciales que ofrece la tecnología de la IO [1].

Las tecnologías IoT permiten que las cosas escuchen, vean, piensen, actúen, se comuniquen y se coordinen con otras cosas, y tomen decisiones tan importantes como salvar vidas[9].

Para el uso de estos dispositivos los fabricantes utilizan distintos protocolos de comunicación para el intercambio de datos los más comunes son MQTT, HTTP, COAP, Websocket, estos pueden variar de acuerdo con el fabricante de los dispositivos referente a implementación.

El aspecto que debe tomar en cuenta un fabricante es la seguridad que puedan proporcionar estos protocolos, para de esta manera poder realizar una selección de acuerdo con las necesidades que fabricante y a su vez puedan cumplir con los diferentes estándares, normativas de seguridad que cada uno de estos protocolos pueda proporcionarnos y con ello garantizar la integridad, confidencialidad al usuario y/o consumidores finales.

La diversidad en software, hardware y protocolos de comunicación significa un análisis más completo y tiene un mayor impacto en la seguridad de los sistemas y dispositivos de IoT. Muchos dispositivos IoT vienen con un sistema operativo portable no configurado, a menudo con un conjunto de utilidades a nivel de desarrollo, que no deberían poder usarse en un sistema de producción. Muchos desarrolladores de sistemas IoT utilizan componentes de software de terceros como por ejemplo librerias que pueden contener actualizaciones sobre nuevas vulnerabilidades encontradas para su corrección [2].

Es importante tener claro que protocolo de transmisión de datos es el más seguro para su uso, debido a esto se plantea realizar una comparación de los tres protocolos de comunicación más utilizados en el mercado, los cuales serán sometidos a un ataque Poisoning, permitiendo determinar mediante un análisis de paquetes las vulnerabilidades y fugas de información encontradas en los protocolos.

2. Materiales y Métodos

2.1 Materiales

2.1.1 Protocolos de transmisión de datos y escenarios de experimentación

Para la selección de protocolos IoT se basó en la descripción realizada en el artículo [10], en ella se realiza una comparación porcentual del uso de los protocolos IoT.

En el presente artículo se tomará tres protocolos los cuales superan el 10% de uso como se muestra en la Tabla1.

Tabla 1 Protocolos de Comunicación más utilizados porcentualmente en el mercado.

PROTOCOLO	*PORCENTAJE*
MQTT	70%
HTTP	35.5%
CoAP	33.3%
WebSocket	19%
LWM2M	12%

De estos como lo menciona el artículo [10], son 5 protocolos que superan el 10% de uso en el mercado. De los cuales utilizaremos los protocolos MQTT, COAP, WEBSOCKET los mismos que estarán programados en un módulo NodeMCU ESP8266 (ver Figura 1), cada uno de estos protocolos tendrán una interfaz de interacción con el usuario, así obteniendo los escenarios de prueba (ver Tabla 2).

Tabla 2. Interfaz de usuario para la interacción de protocolos de comunicación.

Protocolo	***Plataforma***
MQTT	Thinger.io
COAP	Copper (Cu4Cr)
WEBSOCKET	Cliente web

.

Figura 11. NodeMCU ESP8266 módulo para la programación de protocolos.

Para la captura de datos se realizará con el sniffer wireshark, obteniendo los paquetes transmitidos desde el módulo NodeMCU ESP8266 hasta el Access Point que se encuentra dentro de una red WAN, con el fin de validar la interacción y ejecución del ataque.

2.1.2 Ataques

El ataque que se utilizó es el ARP Poisoning con el fin de detectar la red de enlace entre el dispositivo IoT y el Access Point. Para ello se utilizó el sistema operativo Wifislax, que es un SO de software libre basado en Linux.

Dentro de sus apartados nos permite utilizar la herramienta Ettercap con la versión 0.8.3.1 permitiendo ejecutar un ataque, suplantando la MAC de destino dentro de una red WAN.

2.2 Métodos

Para la elaboración del artículo académico se va a usar una metodología cuantitativa y enfoque sistemático, experimental, dado que se procederá con la recolección de información de diferentes fuentes bibliográficas para seleccionar los protocolos de comunicación más relevantes que se consideraron para este artículo, con el fin de obtener información que permita determinar el protocolo de trasmisión de datos más seguro.

2.2.1 Protocolos de comunicación

Los protocolos y estándares de red son una especificación de varias reglas para un tipo particular de comunicación entre dos o más dispositivos en la red.

Los protocolos IoT son uno de esos sistemas que transferirán datos de forma segura a través de Internet. Por un lado, están los protocolos generales utilizados por los dispositivos personales que no responden a los requisitos específicos de la aplicación IoT. Hay algunas versiones mejoradas de los protocolos existentes y se han desarrollado nuevos protocolos IoT para satisfacer los requisitos de los dispositivos IoT [3].

Se seleccionaron los protocolos de comunicación más comunes, sobre los cuales se trabajó un análisis de datos.

A continuación, se describe brevemente cada uno de ellos:

2.2.1.1 Message Queue Telemetry Transport (MQTT)

MQTT es un protocolo de comunicación de código abierto que se distingue por su ligereza y por su fácil implementación, gracias a lo cual se ha logrado implementar con éxito en pequeños microcontroladores, con recursos de hardware limitado, permitiendo encajar perfectamente con IoT.

Hoy en día se utiliza en diversas industrias como la de telecomunicaciones, automotriz y petrolera.

MQTT está estandarizado por la norma ISO (Organización Internacional de Normalización) (ISO/IEC PRF 20922) y puede utilizarse junto con TCP/IP [4].

2.2.1.2 Constrained Application Protocol (COAP):

El protocolo de aplicación restringida (CoAP) se especializa en el uso de nodos inalámbricos y limitados de baja potencia permitiendo comunicarse a través de internet de forma interactiva, su modelo es similar al de HTTP con la diferencia que CoAP realiza el intercambio de mensajes de forma asíncrona.

Utiliza UDP en lugar de TCP para mantener la ligereza. Este protocolo ha sido especialmente diseñado para ser utilizado en dispositivos con poca capacidad de procesamiento y poca memoria RAM.

Sigue el modelo petición-respuesta para la interacción entre el cliente y el servidor. CoAP cumple la mayoría de los requisitos necesarios para la comunicación M2M [5].

2.2.1.3 WebSocket:

El protocolo WebSocket fue desarrollado para cumplir con el intercambio constante de datos entre el cliente y el servidor que no se soportaba antes desde HTTP.

El protocolo consiste en un completo canal de comunicación bidireccional que funciona a través de un único socket, además de tener una comunicación asíncrona (a diferencia del protocolo HTTP), es decir, ambas partes pueden enviar datos en cualquier momento durante la configuración de la conexión.

El protocolo se divide en dos partes: handshake y transferencia de datos. En el handshake, básicamente el cliente y el servidor establecen una comunicación inicial utilizando HTTP y un puerto, el 80 que es el predeterminado. [6]

2.2.2 Análisis de tráfico de datos.

En la actualidad existen diferentes herramientas para el análisis y monitoreo de tráfico más conocidos como sniffer.

Un sniffer es un dispositivo con un software especial que puede capturar y analizar el tráfico de la red. Puede realizar análisis de rendimiento de red, detección de interferencias de red y análisis de paquetes sin afectar el rendimiento de la red. Los sniffers son muy útiles para desarrollar y probar nuevos estándares y protocolos, diseñar nuevas redes e implementar sistemas en aplicaciones del mundo real [7].

2.2.3 Uso de los sniffers

Técnicos experimentados los despliegan en redes para interceptar el tráfico y rastrear lo que se envía. Probablemente habrá visto la película donde un detective interviene la línea telefónica de un sospechoso y escucha mientras discuten todo tipo de cosas. Un sniffer es esencialmente lo mismo, pero aplicado a Internet [8].

2.2.4 Criterios de análisis

El Sniffer será ejecutado en cada uno de los protocolos IoT, se analizará el comportamiento de ellos frente a un ataque ARP Spoofing o más conocido como *Man In the Middle*, se revisará el tamaño de paquetes que se puede capturar en un corto periodo de tiempo, con el fin de obtener la

información necesaria para realizar la comparación y determinar que protocolo es el más confiable para su uso.

2.3 Experimentación

2.3.1 Escenario de prueba MQTT

Para la ejecución de este protocolo se ejecuta un script de Arduino para el uso de un led y un potenciómetro en el módulo NodeMCU ESP8266 y como parte de interfaz de interacción estará publicado en la plataforma Thinger.io obteniendo así el primer escenario para la comparación (ver Figura 2).

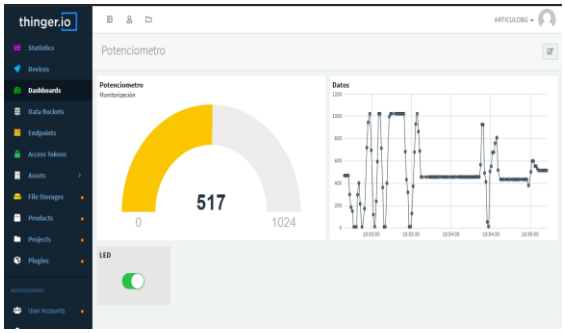


Figura 2. Plataforma de interacción Thinguer.io con tráfico de datos (Escenario 1).

2.3.2 Escenario de prueba CoAP

Para la ejecución de este protocolo se utilizó la plataforma Copper con la extensión Copper (Cu4Cr) instalada en Google Chrome para su acceso o administración, en el módulo se encuentra programado un script de Arduino con el protocolo para la interacción con un led, así obteniendo el segundo escenario (ver Figura 3).

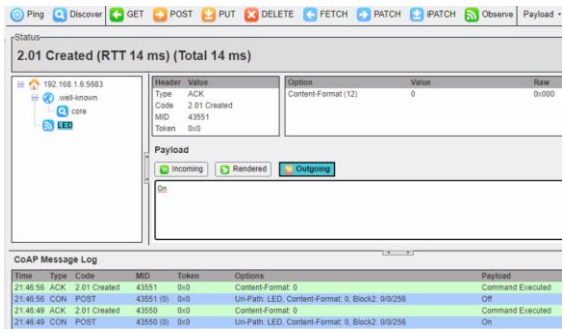


Figura 23. Plataforma de interacción Cu4Cr (CoAP) con intercambio de datos (Escenario 2).

2.3.3 Escenario de prueba WebSocket

Para este protocolo se ejecuta un script de Arduino para el envió y la recepción de datos de un led conectado a el módulo NodeMCU ESP8266, en el script se crea un servidor WebSocket el cual está definido por el puerto 80 y 81, este no servirá para la administración y visualización de datos, es decir este protocolo no necesita de una plataforma adicional para su funcionamiento el mismo puede estar alojado dentro de un dispositivo IoT. (ver Figura 4).

Figura 4. Plataforma de interacción WebSocket Cliente (Escenario 3).

3. Resultados y Discusiones

3.1 Resultados

3.1.1 Resultados de pruebas MQTT

En el escenario de pruebas con respecto al protocolo MQTT se puede observar que al ejecutar el escaneo de red mediante un ARP Poisoning, se logra identificar los hosts que se encuentra dentro de la red WAN, identificando el dispositivo IoT con IP 192.168.1.6 y el Access Point 192.168.1.1, las direcciones encontradas muestran en la Figura 5 y a su vez se realiza el envenenamiento ARP entre la IP del módulo y el Access Point dándonos como resultado nuestras direcciones MAC que estarán dentro de nuestra red WAN.

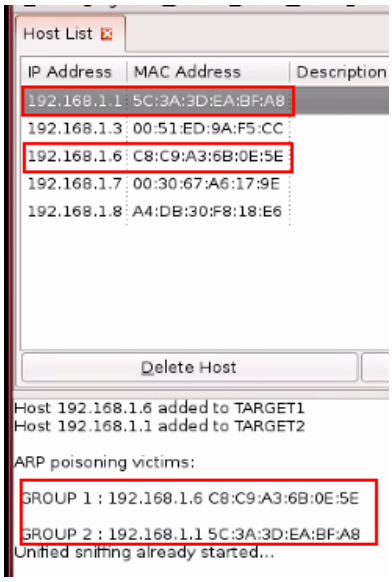


Figura 35. Escaneo ARP Poisoning y generación de direcciones MAC.

Mientras el ataque es realizado se ejecuta el sniffer para capturar el tráfico que se envía desde el módulo hacia la plataforma Thinger.io obteniendo los paquetes como se muestra en la Figura 6, como se puede observar la herramienta está realizando un intercambio de información pero al momento de analizar uno de los paquetes se visualiza que la información esta encriptada por el protocolo TSL como se muestra en la Figura 7, este protocolo de transmisión de datos seguro se encarga de encriptar la información tanto de emisión como de recepción.

Figura 46. Captura de paquetes wireshark, con información enviada a la plataforma Thinger.io.

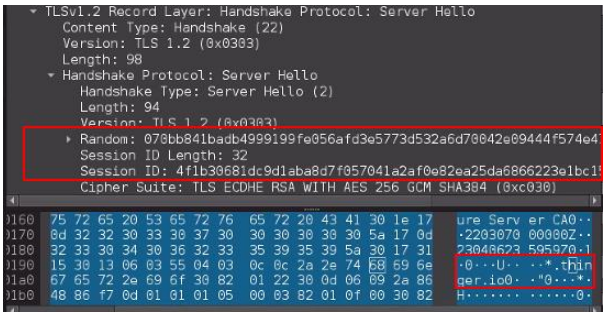


Figura 57. Datos encriptados por TLS que se envían a la plataforma Thinger.io.

De igual manera al ejecutar el ataque se encuentra un problema anómalo dentro del tihnger.io a pesar de ser solo un ataque de envenenamiento de ARP, para ver el intercambio de paquetes, la plataforma presenta una inestabilidad como se muestran en la Figura 8 y 9.

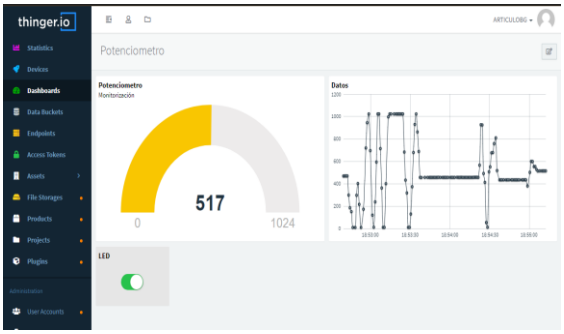


Figura 68. Thinger.io sin ataque ARP Spoofing.

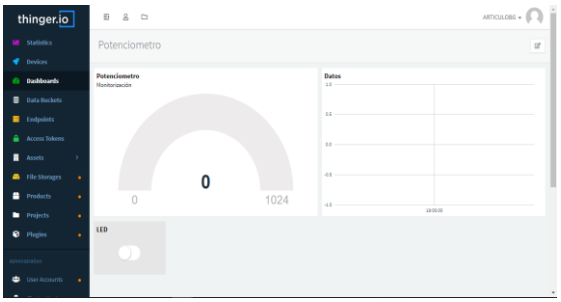


Figura 79. Thinger.io con ataque ARP Spoofing.

En la Figura 8 se visualiza un comportamiento normal referente a los datos mostrados en cada dashboard que permite ver la interacción realizada con el módulo, En la Figura 9 se visualiza que al ejecutar el Spoofing está ya no permite visualizar datos ni interactuar con el módulo de manera correcta esto a pesar de que el

ataque enviando no es de denegación de servicio o DDoS.

3.1.2 Resultados de pruebas COAP

Al ejecutar el escaneo de red mediante un ARP Poisoning, se logra identificar los hosts que se encuentra dentro de la red WAN, identificando el dispositivo IoT con IP 192.168.1.6 y el Access Point 192.168.1.1, los datos encontrados del escaneo se muestran en la Figura 10 y a su vez se realiza el envenenamiento ARP entre la IP del módulo y el Access Point.

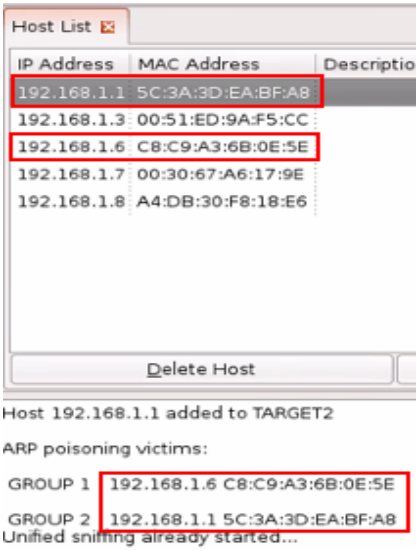


Figura 810. Escaneo ARP Poisoning y generación de direcciones MAC.

Como se puede observar en avenamiento de ARP se puede ejecutar de manera satisfactoria, al saber la dirección de publicación de la plataforma Cu4Cr remplazando la MAC de destino.

Mientras se ejecuta el ataque, se procede con la captura de paquetes con el sniffer wireshark.

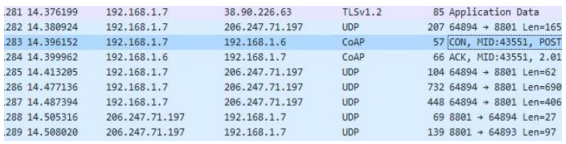


Figura 911. Paquetes CoAP capturados con wireshark.

Al realizar el análisis se evidencia que los recibidos desde la plataforma Cu4Cr son evidenciados en forma de texto plano como se muestra en la Figura 12.

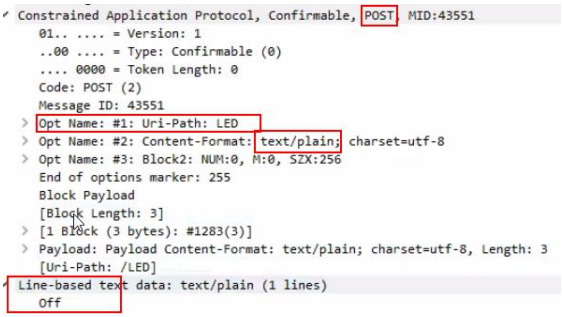


Figura 1012. Paquete CoAP wireshark, con información delicada para el usuario representada en texto plano.

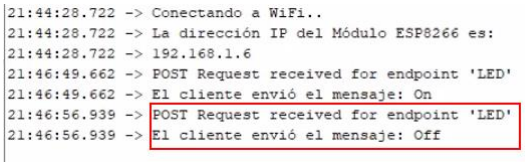


Figura 1113. Recepción de datos en módulo con el protocolo CoAP.

En la Figura 13 se evidencia que los datos que se reciben en el módulo son los mismos datos que son enviados a la plataforma CoAP, es decir que los datos del usuario pueden estar expuestos y ser interpretados fácilmente dado que no utilizan algún método de encriptación de información.

3.1.3 Resultados de Pruebas WebSocket

Al ejecutar el escaneo de red mediante un ARP Poisoning, se logra identificar los hosts que se encuentra dentro de la red WAN, identificando el dispositivo IoT con IP 192.168.1.7 y el Access Point 192.168.1.1, los datos encontrados se muestran en la Figura 14 y a su vez se realiza el envenenamiento ARP entre la IP del módulo y el Access Point.

Se realiza el envenenamiento ARP entre la IP del módulo y el Access Point dándonos como resultado nuestras direcciones MAC que estarán dentro de nuestra red WAN (ver Figura 13).

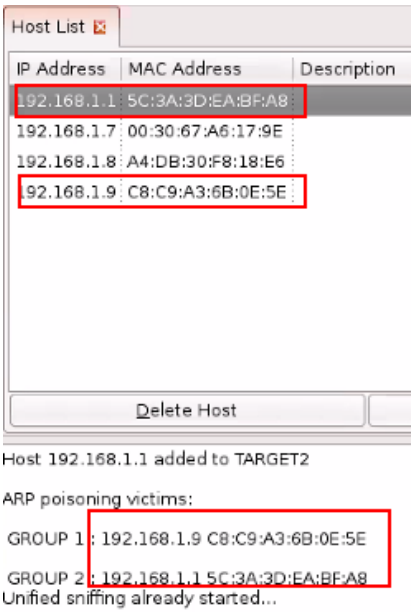


Figura 1214. Escaneo ARP Poisoning Escaneo ARP Poisoning y generación de direcciones MAC.

Solo con conocer la IP del servidor de interacción de este protocolo ya que se puede acceder a los datos del cliente y se puede realizar la interacción directa sin necesidad de ejecutar el sniffer como se muestra en la Figura 15.

Figura 1315. Publicación de cliente sin seguridad se visualiza la IP sobre la cual se estará enviando y recibiendo datos.

Se ejecuta el Sniffer capturando los paquetes transmitidos hacia la IP 192.168.1.6.

	Time	Source	Destination	Protocol	Length
1544…	1528.327049	fe80::526b:96ee:b69…	fe80::1	DNS	9
1544…	1528.336507	fe80::1	fe80::526b:96ee:b69…	DNS	11
1544…	1528.336507	fe80::1	fe80::526b:96ee:b69…	DNS	12
1544…	1528.430984	2800:370:11c:8b30:b…	2407:30c0:182::aa72…	TCP	8
1544…	1528.473991	192.168.1.7	192.168.1.6	WebSocket	6
1544…	1528.481570	192.168.1.6	192.168.1.7	WebSocket	6
1544…	1528.491700	2407:30c0:182::aa72…	2800:370:11c:8b30:b…	TCP	8
1544…	1528.491922	2800:370:11c:8b30:b…	2407:30c0:182::aa72…	TCP	7
1544…	1528.522092	192.168.1.7	192.168.1.6	TCP	5
1544…	1528.538909	2800:370:11c:8b30:b…	2407:30c0:182::aa72…	TLSv1.3	59

Figura 1416. Captura de paquetes wireshark, protocolo WebSocket.

Analizando uno de estos paquetes se puede ver la interacción que se está realizando y la información que se está recibiendo desde la plataforma hacia el dispositivo IoT como se puede ver en la Figura 17.

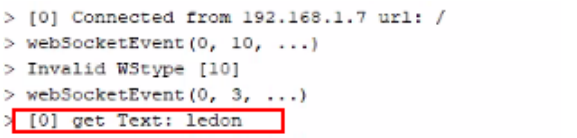


Figura 1517. Recepción de información Arduino WebSocket.

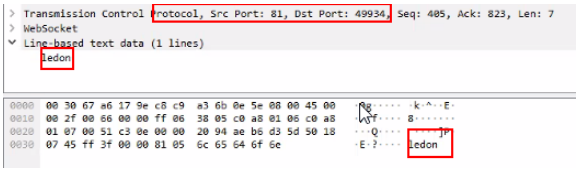


Figura 1618 Captura de información wireshark

Como se puede observar en la Figura 18 la información es capturada en texto plano sobre la interacción que se está realizando de igual manera se visualiza los puertos sobre el cual está trabajando este protocolo en este caso el puerto 81 y 49934.

3.2 Discusión

Para el protocolo MQTT en el escenario de pruebas en tiempo real con la plataforma Thinger.io, se logra realizar un envenenamiento ARP entre el módulo y el Access Point, en este caso se logra una suplantación entre el receptor de los paquetes de envió de datos por parte del módulo NodeMCU ESP8266 suplantando la dirección MAC con la dirección del equipo, y así lograr recolectar la información enviada a la plataforma Thinger.io.

Se puede observar que los datos enviandos y recibidos por parte del módulo se encuentran encriptados, esto debido a que MQTT utiliza el protocolo TLS (Transport Layer Security) para el intercambio de datos, este protocolo al utilizar TLS garantiza que los datos sean seguros al momento de ser enviados.

Sin embargo, al realizar el envenenamiento ARP para la suplantaciòn, la plataforma tuvo un comportamiento anomalo ante el ataque, esta no permitia realizar ninguna interacción desde la

plataforma hacia el módulo, esto puede ser un comportamiento que ayude al usuario a identificar que algo esta sucediendo con el dispositivo IoT y a su vez pueda ejecutar un plan de mitigaciòn como un reset de claves o simplemente desconectar el dispositivo de la red para evitar fugas de información.

En el protocolo CoAP se puede obsevar que para obtener el escenario de pruebas se tuvo que incluir una extensión dentro de Google Chrome, limitandonos a un navegador para el uso de este protocolo, a pesar de esto se logra realizar el envenenamiento ARP y la suplantación de dirección MAC como se menciono anteriormente, solo que este captaria la informaciòn enviada a la plataforma CoAP.

Al momento de ejecutar el envenenamiento ARP la plataforma no presenta ninguna novedad en cuento a la interaccion de datos, pero una vez ejecutado el sniffer se puede validar que captura toda la informaciòn recibida desde la plataforma, dandonos informaciòn sobre el texto enviado al dispositivo los puertos de comunicación del mismo, teniendo asi una brecha de seguridad importante dentro de nuestra red WAN.

El protocolo WebSocket para ser ejecutado se encuentra el problema de las plataformas de trabajo, encontrando que el servidor donde se va a ejecutar la interacción con el usuario es dentro del mismo dispositivo IoT (Cliente WEB).

Esta al ser integrada dentro del mismo dispositivo se podría considerar que una empresa pequeña que fabrique algún dispositivo con este protocolo debe estar sujeta a controles de calidad de seguridad altos dado que si no crea bien el cliente web este podría tener varias brechas de seguridad como las encontradas en la ejecución de pruebas en el escenario 3, donde se puede observar que solo con la ejecución del escaneo de host con nuestra herramienta Ettercap. se logró identificar la IP del cliente web y de manera fácil se pudo entrar a la interfaz de interacción con el usuario.

Esto ya depende mucho de la empresa que trabaje con este protocolo, el cual debería tener un Login o un doble factor de autenticación para mitigar esta vulnerabilidad.

Posterior a esto se identifica que al ejecutar el Sniffer la aplicación no tomo ninguna acción de seguridad, es decir, que se puede ingresar y visualizar cada interacción que realice un usuario en un dispositivo IOT hacia el cliente WEB.

Se puede validar que el protocolo más seguro para la trasmisión de datos es el protocolo MQTT quien no presento problemas o eventos de seguridad en las pruebas realizadas, si bien presento un poco de intermitencias al realizar un ataque no fuerte, es una buena señal dado que no permitió el intercambio de información entre el dispositivo IoT y la plataforma, es por ello que se lo encuentra entre los protocolos más utilizados.

Los protocolos CoAP y Websoket, si bien se encuentran entre los protocolos más utilizados como se muestra en la Figura 1, deberían mejorar las vulnerabilidades encontradas para que puedan seguir usando sin ninguna complicación.

Si bien el envenenamiento por ARP fue ejecutado de manera satisfactoria en los tres protocolos con diferentes escenarios de interacción, encontrando que se puede acceder de manera fácil a la conexión entre el módulo ESP8266 y el Access Point, existiendo una brecha de seguridad en las conexiones, dado que al integrar el dispositivo IoT este no requiere de alguna conexión segura.

Al no tener una conexión segura, el atacante puede ingresar a la red y realizar la captura de paquetes como se la realizo en cada uno de los escenarios, los dispositivos IoT deberían tener mejor calidad en la seguridad de la información y basarse en normas de seguridad.

4. Conclusiones

Se presenta un análisis de los aspectos de seguridad y vulnerabilidades encontradas en cada uno de los tres protocolos, se puede determinar mediante las pruebas realizadas que el protocolo MQTT es uno de los más seguros, funciona con el protocolo TLS/SLL, cifrando así los datos para que no puedan ser interpretados.

Dentro del análisis se encontró degradación e intermitencias en la plataforma de Thinger.io esta no permite realizar la interacción con el dispositivo, esto debe ser un punto importante para el usuario final, tomar conciencia en que dentro de su red está existiendo algún problema ya sea que este no pueda identificar que es un ataque como tal, el reflejo del usuario va a ser reiniciarlo en algún punto y con esto el dispositivo al encontrarse dentro de una red WAN por DHCP puede asumir una nueva IP y así obligando al atacante a realizar de nuevo el proceso de escaneo cada que ocurra este evento.

El protocolo más utilizado en el mercado es el protocolo MQTT con un 60% de aceptación, sin embargo, nos encontramos ante un cambio constante en las Tecnologías de la Información (TI) por lo que no se descarta que la implementación de otros protocolos como AMQP, Zigbee con el tiempo lleguen a ser los más utilizados, dado que cada día se van corrigiendo errores tanto de hardware como de software en el cual está incluido el tema de seguridad.

De igual manera se logró visualizar los puertos sobre los cuales el protocolo está realizando la comunicación. Esto puede ser una de las vulnerabilidades más buscada por los atacantes con el fin de determinar qué servicio está consumiendo ese puerto para la comunicación.

5. Referencia

[1] H. Arjadi, H. Candra, H. D. Prananto y T. A. W. Wijanarko, «RSSI Comparison of ESP8266 Modules,» IEEE Explore, 2020.

[2] A. Castro, E. Casanova y V. Gil-Costa, «Aspectos de Seguridad de Internet de las Cosas,» XXIII Congreso Argentino de Ciencias de la Computacion, 13 Octubre 2017.

[3] E. Sakina, M. Abdeaziz y S. Nawal, «Operating models of Network protocols IoT:Short-range protocols,» IEEE Explore, 2020.

[4] M. Dave, D. J. Doshi y D. H. Arolkar, «MQTT- CoAP Interconnector: IoT Interoperability Solution for Application Layer Protocols,» IEEE Xplore, 2020.

[5] D. B. Momin y D. Seetharam, «An Empirical Study of Application Layer Protocols for IoT,» IEEE Xplore, 2017.

[6] G. Oliveira, D. Costa, R. Cavalcanti, J. Oliveira, D. Silva, M. Nogueira y M. Rodrigues, «Comparison Between MQTT and WebSocket Protocols for IoT Applications Using ESP8266,» IEEE, p. 3, 2018.

[7] J. Crnogorac, J. Kovač, E. Kočan y M. Vučinić, «d-Argus: a Distributed IEEE 802.15.4 Sniffer,» IEEE Explore, 27 11 2019. s-2/.

[8] I. Belcic, «AVAST,» avast, 13 01 2022. [En línea]. Available: https://www.avast.com/es-es/c-sniffer#:~:text=Un%20sniffer%20es%20una%20herramienta,y%20salen%20de%20su%20equipo.. [Último acceso: 14 05 2020].

[9] G. Gardašević, M. Veletić y D. Vasiljević, «The IoT Architectural Framework, Design Issues and Application

Domains,» Wireless Pers Commun, vol. 92, nº 1, pp. 127-148, 01 2017.

[10] Á. M. M. Castillo, «EVALUACIÓN DE LAS PLATAFORMAS MASIVAS DE INTERNET DE LAS COSAS Y TEST DE UNA APLICACIÓN PRÁCTICA EN UNA PLATAFORMA SELECCIONADA,» Escuela Técnica Superior de Ingenieros Industriales (UPM), p. 49, Febrero 2019.

[11] K. Yang y J. Zhang, «Design of Remote Control Inverter Based on MQTT Communication Protocol,» IEEE Explore, 2021.

[12] N. Nikolov, «Research of MQTT, CoAP, HTTP and XMPP IoT Communication protocols for Embedded Systems,» IEEE Explore, 2020.

[13] A. Semle, «Protocolos IoT para considerar,» AADECA REVISTA, 2016.