ANÁLISIS DE LA SEGURIDAD DEL PROTOCOLO DE TRANSPORTE MQTT EN DISPOSITIVOS PARA INTERNET DE LAS COSAS.

PAULITA FLOR SALAZAR

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA – UNAD
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA - ECBTI
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
POPAYÁN
2022

ANÁLISIS DE LA SEGURIDAD DEL PROTOCOLO DE TRANSPORTE MQTT EN DISPOSITIVOS PARA INTERNET DE LAS COSAS.

PAULITA FLOR SALAZAR

Proyecto de Grado – Monografía presentado para optar por el título de
ESPECIALISTA EN SEGURIDAD INFORMATICA

JOEL CARROLL VARGAS
DIRECTOR

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA – UNAD
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA - ECBTI
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
POPAYÁN
2022

NOTA DE ACEPTACIÒN

Firma del Presidente de Jurado

Firma del Jurado

Firma del Jurado

Popayán, 8 de marzo de 2022

DEDICATORIA

Dedico este trabajo a Dios quien me ha regalado todo lo que tengo, él llevará a cabo los planes que tiene para mi vida, pues tu fiel amor, oh, Señor, permanece para siempre.

AGRADECIMIENTOS

Agradezco a la Universidad Nacional Abierta y a Distancia UNAD, y a los docentes del programa de Especialización en Seguridad Informática por brindarme su experiencia y conocimiento para formarme en un área que se requiere actualmente para que la tecnología pueda ser utilizada en beneficio de la sociedad.

CONTENIDO

LISTA DE FIGURAS *8*

GLOSARIO *9*

RESUMEN *11*

ABSTRACT *12*

INTRODUCCIÓN *13*

1. DEFINICIÓN DEL PROBLEMA *14*

1.1 ANTECEDENTES DEL PROBLEMA **14**

1.2 FORMULACIÓN DEL PROBLEMA **14**

2 JUSTIFICACIÓN *16*

3 OBJETIVOS *18*

3.1 OBJETIVOS GENERAL **18**

3.2 OBJETIVOS ESPECÍFICOS **18**

4 MARCO REFEReNCIAL *19*

4.1 MARCO CONCEPTUAL **19**

4.1.1 Internet de las cosas 19

4.1.2 Aplicaciones del IoT 21

4.1.3 Protocolos de la capa de transporte 21

4.1.4 Seguridad informática 22

4.1.5 Pilares de la seguridad de la información 23

4.2 MARCO TEÓRICO **24**

5. DISEÑO METODOLÓGICO *31*

5.1. Metodología tipo Exploratorio **31**

5.2. TECNICAS PARA LA RECOLECCIÓN DE INFORMACIÓN **31**

5.2.1. Análisis documental 32

6. DESARROLLO DE LOS OBJETIVOS *33*

6.1. DETERMINAR LAS VULNERABILIDADES DEL PROTOCOLO DE TRANSPORTE MQTT EN COMUNICACIONES BAJO LA TECNOLOGÍA DE INTERNET DE LAS COSAS, MEDIANTE REVISIÓN DE LITERATURA. **33**

6.2. ESTABLECER A PARTIR DE REVISIÓN DE LA LITERATURA MECANISMOS Y HERRAMIENTAS PARA MITIGAR LAS VULNERABILIDADES ENCONTRADAS EN EL PROTOCOLO DE TRANSPORTE MQTT **45**

6.3. EVALUAR LOS MECANISMOS Y HERRAMIENTAS DE MITIGACIÓN A TRAVÉS DE SU IMPLEMENTACIÓN Y ANÁLISIS EN LA COMUNICACIÓN DE DISPOSITIVOS IOT DENTRO DE UN ESCENARIO CONTROLADO. 53

CONCLUSIONES 62

RECOMENDACIONES 63

BIBLIOGRAFÍA 64

ANEXOS 69

LISTA DE FIGURAS

Figura 1 Componentes MQTT 25
Figura 2 Modelo de comunicación MQTT 26
Figura 3 QoS mensajes MQTT 27
Figura 4 Ejemplo arquitectura MQTT 33
Figura 5 flujo de mensajes en MQTT 34
Figura 6 Escenario de contraseñas en texto plano 37
Figura 7 Ejemplo Niveles de Jerarquía MQTT 38
Figura 8 Resultados Shodan MQTT 39
Figura 9 Control MQTT mediante Botnet 41
Figura 10 Interfaces de modelo de referencia Intel IoT 48
Figura 11 Relación de modelo ITU 50
Figura 12 Arquitectura de pruebas 56
Figura 13 instalación cliente mosquitto 70
Figura 14 Verificación activación mosquitto 71
Figura 15 Verificación estado mosquitto 71
Figura 16 Resultado escaneo de puertos 74
Figura 17 Resultado dispositivos públicos MQTT 75

GLOSARIO

AUTENTICIDAD: es uno de los principios de la seguridad de la información que establece que la información recibida es autentica y legítima, ya que no ha sido modificada de su formato original durante los procesos de comunicación, se conserva tal y como la envió el emisor.[1]

CONFIDENCIALIDAD: hace referencia a la protección de la información para que solo puedan acceder a ella el personal autorizado dentro de la organización, y no puede ser revelada a terceros que dentro de sus responsabilidades no requieran de dicha información para desarrollar sus funciones. [2]

DISPONIBILIDAD: define que la información siempre debe estar disponible dentro de cualquier medio digital o físico para que pueda ser procesada y que aporte al correcto funcionamiento de la organización, garantizando el acceso ininterrumpido de los diferentes usuarios que cuentan con autorización para acceder a ella.[3]

INTEGRIDAD: La información generada debe ser conservarse exactamente igual y no puede ser sometida a modificaciones no autorizadas, durante procesos de comunicación, para llegar sin ser copiada, borrada o modificada a su destino. La integridad de la información puede verse comprometida cuando los datos no son cifrados que pueden encontrarse en diferentes sitios como bases de datos, registros, documentos, entre otros.[4]

---

[1] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2, p. 193-197.
[2] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2, p. 193-197.
[3] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2, p. 193-197.
[4] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2, p. 193-197.

INTERNET DE LAS COSAS: red de dispositivos físicos conectados a través de sensores y aplicaciones que permite el intercambio de datos a través de internet. [5]

PROTOCOLO DE COMUNICACIÓN: conjunto de normas que deben seguir los equipos hardware y el software que hacen parte de un proceso de comunicación de datos entre dos estaciones para que el proceso pueda ser de calidad y se lleve a cabo con éxito. [6]

SEGURIDAD INFORMÁTICA: hace referencia a la disciplina que se encarga de proteger la información en los pilares de disponibilidad, integridad y confidencialidad, cuando dicha información se encuentra almacenada dentro de un sistema informático, para evitar los diferentes daños y riesgos que pueden presentarse en diferentes contextos que se pueden dar de manera intencional o circunstancial. [7]

VULNERABILIDAD: falla de los sistemas de información que genera un riesgo en la seguridad de la información en sus tres principios confidencialidad, disponibilidad e integridad. [8]

---

[5] ALCARAZ, Marcelo. Internet de las cosas. Universidad Católica Nuestra Señora de la Asunción, 2014, no 1, p. 2-3.
[6] BLACK, Uyless. Redes de computadores: Protocolos, normas e interfaces. Alfaomega, 1997.
[7] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2, p. 193-197.
[8] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2, p. 193-197.

RESUMEN

Internet de las cosas ha crecido rápidamente y su implementación dentro de todos los campos es una realidad al igual que las amenazas de seguridad por lo que implica tener conectado a internet los dispositivos cotidianos que utiliza la sociedad en general por lo que se ha hecho necesario establecer normatividad y estandarizar la implementación de esta tecnología en todos sus aspectos como los protocolos de comunicación, es así que la organización internacional de estándares, ISO ha estandarizado el protocolo de transporte de mensajes MQTT (Message Queue Telemetry Transport) como protocolo aplicable a los entornos donde se realiza comunicación de dispositivos de Internet de las cosas para contribuir a las buenas prácticas de seguridad se presenta este estudio que pretende determinar las vulnerabilidades de la implementación del protocolo MQTT para obtener mecanismos y herramientas de mitigación de amenazas mediante la identificación de las amenazas y evaluación de los mecanismos de mitigación del riesgo.

Esto permite determinar los mejores campos de aplicación del protocolo MQTT teniendo en cuenta en qué tipo de comunicación se hace más o menos vulnerable determinando las mejores prácticas de seguridad con un protocolo de transporte estandarizado para el uso de comunicaciones en internet de las cosas.

Palabras clave: MQTT, Internet de las cosas, seguridad de la información, protocolos de comunicación, falla de seguridad, estándar.

ABSTRACT

The internet of things technology has grown rapidly and its implementation within all fields is a reality as well as security threats, so it means having the daily devices used by society in general connected to the internet, so it has been It is necessary to establish regulations and standardize the implementation of this technology in all its aspects such as communication protocols, so the international organization of standards, ISO has standardized the message transport protocol MQTT (Message Queue Telemetry Transport) as a protocol applicable to The environments where communication of Internet devices of things is carried out to contribute to good security practices presents this study that aims to determine the vulnerabilities of the implementation of the MQTT protocol to obtain mechanisms and tools for mitigating threats by identifying threats and assess tion of risk mitigation mechanisms.

This makes it possible to determine the best fields of application of the MQTT protocol, taking into account the type of communication that is more or less vulnerable, determining the best security practices with a standardized transport protocol for the use of communications on the Internet of Things.

Keywords: MQTT, Internet of things, information security, communication protocols, security flaw, standard.

INTRODUCCIÓN

En el presente trabajo se realiza una consulta bibliográfica que permita analizar el nivel de seguridad que ofrece el protocolo de la capa de transporte MQTT estandarizado por la ISO en las comunicaciones de dispositivos para internet de las cosas. Se ha realizado un análisis de las características del protocolo y los requisitos para su implementación con el fin de determinar su campo de aplicación y el nivel de confiabilidad del protocolo identificando sus diferentes vulnerabilidades para determinar los mejores mecanismos de mitigación del riesgo generado por dichas vulnerabilidades.

1. DEFINICIÓN DEL PROBLEMA

1.1 ANTECEDENTES DEL PROBLEMA

Los dispositivos IoT generan gran cantidad de datos, todos esos datos que se generan excesivamente son los que dan lugar a los ataques informáticos principalmente de DDoS, para la mitigación de estos ataques se hace necesario la implementación de sistemas de detección y protección de intrusiones, y una de las principales causas de estos ataques son los niveles altos de QoS, un nivel inferior de QoS puede disminuir el efecto de denegación del servicio, el manejo de varios niveles de QoS es una de las principales características del protocolo de transporte MQTT por esta razón ha sido estandarizado como un protocolo de comunicación para comunicaciones de dispositivos IoT. Sin embargo, algunas características del protocolo como la autenticación, los niveles de QoS, el uso de wildcards, entre otras hacen que presente varias vulnerabilidades y fallas de seguridad.

Se han realizado pruebas con la plataforma Shodan que posee una API de Python para realizar el rastreo de dispositivos de acceso público, esto se hace mediante filtros como protocolos y puertos, y aquí se ha encontrado más de cincuenta mil dispositivos accesibles que utilizan el protocolo MQTT. [9]

1.2 FORMULACIÓN DEL PROBLEMA

El constante crecimiento del Internet de las cosas es muy evidente, todos los objetos cuentan con la conexión y configuración necesaria para controlar y monitorear las diferentes actividades cotidianas, y las diferentes tareas de la industria; por lo que dichos dispositivos generan una gran cantidad de flujo de datos, lo que en términos de seguridad se convierte en una actividad compleja para lograr su gestión, debido

[9] HARSHA MS, BHAVANI BM, KR KUNDHAVAI, Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs, 2018.

a que los protocolos de comunicación disponibles para esta tecnología no poseen las medidas y controles adecuados para garantizar los pilares de la seguridad de la información además de que muchos usuarios confían en las configuraciones predeterminadas de estos, lo que aumenta los riesgos de la seguridad de la información.[10]

Siendo MQTT un protocolo ya estandarizado por la ISO para ser implementado en las comunicaciones para IoT y uno de los más utilizados, ya que ofrece las siguientes ventajas: posee 3 niveles de calidad del servicio, realiza transferencia de mensajes cortos, tiene opción de cifrado y autenticación de los datos y no requiere demasiado software para su implementación; se selecciona para realizar un análisis de la seguridad del protocolo en dispositivos IoT y así determinar ¿Cuáles son las vulnerabilidades del protocolo MQTT y cómo pueden ser mitigadas?

[10] Gonzalez, Carlos, Faluzac, Olivier, Nolot, Florent. Evolution and Contribution for the Internet of Things by the Emerging Software - defined networking (2016)

2 JUSTIFICACIÓN

Internet de las cosas (IoT) permite la extensión de capacidades de todo tipo de dispositivos permitiéndoles conectar e intercambiar datos con otros dispositivos en la red. La cantidad de dispositivos conectados ha crecido exponencialmente por lo que es necesario asegurar que todos los dispositivos que estén conectados a internet estén protegidos contra las diferentes vulnerabilidades de seguridad de la información, por esta razón las organizaciones internacionales han buscado estandarizar la implementación de esta tecnología.

IoT utiliza diferentes protocolos para la comunicación, pero el protocolo de transporte de mensajes MQTT (Message Queue Telemetry Transport) es muy utilizado porque se caracteriza por su tamaño de encabezado y requisitos bajos de ancho de banda, por esta razón el protocolo MQTT ya ha sido estandarizado por la organización internacional de estándares, ISO, para las comunicaciones en internet de las cosas, IoT, posee ventajas en cuanto a la seguridad como el cifrado y autenticación de los datos pero también presenta limitaciones y un amplio conjunto de vulnerabilidades que pueden ser explotadas por los diferentes atacantes.[11]

La identificación de estas vulnerabilidades permitirá encontrar los mecanismos y herramientas que ayuden a mitigar el nivel de riesgo y disminuir los ataques informáticos que se llevan a cabo con éxito a través de la gran cantidad de dispositivos conectados a internet que utilizan este protocolo para la comunicación,

[11] HARSHA MS, BHAVANI BM, KR KUNDHAVAI, Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs, 2018.

[12]Syaiful, Andy, afirma que: “El mecanismo de seguridad del protocolo MQTT, especialmente para dispositivos con recursos limitados, aún necesita desarrollo porque cada investigación que se ha realizado todavía tiene un enfoque específico que aún no ha sido integrado.” Esto dado que los diferentes estudios realizados sobre el análisis de la seguridad del protocolo se enfocan en un solo tipo de ataque, y por tanto las medidas de mitigación de las vulnerabilidades también son limitadas, por eso es necesario integrar esa información para determinar las diferentes vulnerabilidades de manera integral y así determinar las mejores medidas para salvaguardar la información en las comunicaciones que usan el protocolo MQTT.

[12] Syaiful, Andy, Budi, Rahardjo, Bunis, Hanindhito, Attack Scenarios and Security Analysis of MQTT Communication Protocol in IoT System, 2017.

3 OBJETIVOS

3.1 OBJETIVOS GENERAL

Analizar a partir de la literatura la seguridad y el comportamiento del protocolo de transporte MQTT en dispositivos para Internet de las cosas los cuales se encuentran en constante riesgo informático, mediante la implementación de un entorno simulado.

3.2 OBJETIVOS ESPECÍFICOS

- Determinar las vulnerabilidades del protocolo de transporte MQTT en comunicaciones bajo la tecnología de internet de las cosas, mediante revisión de literatura.
- Establecer a partir de revisión de la literatura mecanismos y herramientas para mitigar las vulnerabilidades encontradas en el protocolo de transporte MQTT
- Evaluar los mecanismos y herramientas de mitigación a través de su implementación y análisis en la comunicación de dispositivos IoT dentro de un escenario controlado.

4 MARCO REFERENCIAL

4.1 MARCO CONCEPTUAL

4.1.1 Internet de las cosas

Hace referencia a escenarios en los que la conectividad de red y la capacidad de cómputo se extienden a objetos, sensores y artículos de uso diario que habitualmente no se consideran computadoras, permitiendo que estos dispositivos generen, intercambien y consuman datos con una mínima intervención humana. El despliegue de internet de las cosas implica cambios generales como: la forma de usar la tecnología, las comunicaciones, y el manejo que se dan a los datos e información generada. Se puede considerar una evolución de internet hacia una red más extensa que provee servicios inteligentes e integrales para una mejor percepción de la información. [13]

Características del Internet de las Cosas [14]

- Uso de mecanismos para monitorización teniendo en cuenta el entorno externo.
- Permite la personalización de la experiencia de usuario para controlar las funciones de los diferentes dispositivos conectados.
- Optimización de tareas mediante la detección de variables y diagnóstico predictivo de los dispositivos conectados.
- Autonomía de los dispositivos conectados para procesos de personalización.
- Entrega de información y datos en tiempo real que permita la toma de decisiones.

---

[13] Somayya Madakam, R. Ramaswamy, Siddharth Tripathi, Internet of Things (IoT): A Literature Review, 2015.
[14] Salazar, Jordi, Silvestre, Santiago. Internet de las Cosas, 2016.

Principios de Internet de las Cosas

- La seguridad de las aplicaciones tanto a nivel de información como de los mecanismos habilitadores.
- Asequibilidad, relacionada con los costos de infraestructura, las posibilidades de competencia, estándares abiertos y aspectos de Propiedad Intelectual definidos.
- Previsibilidad tanto como elemento de confianza de la red como de privacidad de la información y el contexto.
- Resiliencia, entendida como la capacidad de recuperación y adaptación ya sea ante fallo o modificaciones del contexto.
- Escalabilidad, entendida como la capacidad de aumentar sus capacidades ya sea en objetos o mecanismos habilitadores.

¿Qué es y cómo funciona el IoT?

El internet de las cosas se refiere a la interconexión digital de una variedad de objetos cotidianos, basados en el principio de que cada objeto posee conexión directa a Internet, en términos técnicos consiste en integrar una serie de componentes electrónicos que permitan la conectividad a internet. Los objetos inteligentes funcionan bajo tres pilares fundamentales:

- Componentes computacionales que permitan procesar la información
- Sensores que permitan obtener la información física del entorno y convertirla en información procesable digitalmente.
- Actuadores que son dispositivos electrónicos que permiten modificar o generar un efecto sobre la información física del entorno.

4.1.2 Aplicaciones del IoT[15]

- **Infraestructuras inteligentes conectadas:** gestión de seguridad electrónica y eficiencia energética, control y monitorización de servicios de salud, educación y domésticos, y en ambientes de construcción, así como en servicios de gestión de la energía.
- **Ciudades inteligentes y transporte:** integración de los diferentes servicios públicos con gestión inteligente del tráfico en tiempo real, redes energéticas, seguridad, administración de los servicios de acueducto, monitorización de los niveles de contaminación, y control de basuras, entre otras.
- **Educación:** integración de aulas físicas y virtuales para el aprendizaje, portables educativos virtuales, nuevos aprendizajes, gestión de asistencia y sistemas de calificaciones en tiempo real.
- **Electrónica de consumo:** diferentes dispositivos inteligentes como laptop, smartphone, tabletas y electrodomésticos.
- **Salud:** monitoreo de signos vitales permanente para pacientes con enfermedades crónicas, mejoramiento de la calidad de atención a los pacientes, teleconsulta y telediagnóstico, monitoreo de actividad física, seguimiento al uso de medicamentos, entre otros.
- **Automoción:** Control de tráfico, monitoreo del funcionamiento de los elementos de vehículos, auto diagnóstico, sensores para proximidad, seguridad y posicionamiento, vehículos inteligentes.
- **Medio Ambiente:** monitoreo de niveles de contaminación, pronóstico de cambios climáticos, monitoreo en tiempo real de variables climáticas.

4.1.3 Protocolos de la capa de transporte

[15] Salazar, Jordi, Silvestre, Santiago. Internet de las Cosas, 2016.

[16]Hoy en día, muchos protocolos se utilizan como protocolo de comunicación en los dispositivos IoT. Cinco de los protocolos más destacados utilizados para IoT son el Protocolo de transferencia de hipertexto (HTTP), el Protocolo de aplicación restringida (CoAP), el Protocolo extensible de mensajería y presencia (XMPP), el Protocolo avanzado de envío de mensajes (AMQP) y el Protocolo de telemetría MQ (MQTT). Algunas consideraciones que deben tenerse en cuenta cuando se elige el protocolo de comunicación a implementar son la eficiencia energética (energía total consumida durante el tiempo de ejecución dado), el rendimiento (tiempo total de transmisión que se tarda en enviar mensajes y recibir sus acuses de recibo), el uso de recursos (CPU, RAM, y uso de ROM), y confiabilidad (capacidad de evitar la pérdida de paquetes, es decir, QoS). Además, cuando las funcionalidades avanzadas, la confiabilidad y la capacidad de asegurar el mensaje de multidifusión son altamente consideradas, el protocolo MQTT es una de las mejores opciones.

4.1.4 Seguridad informática

La seguridad informática es una disciplina encargada de la protección de la información y de los datos almacenados dentro de un sistema informático independientemente del área en el que se generen, debe permitir la protección los datos de los posibles riesgos y/o daños a los que puedan estar expuestos y que los generen personas de manera voluntaria o involuntaria, que pueden ser parte de la organización o ajenas a la misma. También debe garantizar la protección de la información en cuanto a los daños por parte de desastres naturales u otro tipo de incidentes externos.[17]

---

[16] Syaiful, Andy, Budi, Rahardjo, Bunis, Hanindhito, Attack Scenarios and Security Analysis of MQTT Communication Protocol in IoT System, 2017

[17] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2, p. 193-197

4.1.5 Pilares de la seguridad de la información[18]

AUTENTICIDAD: es uno de los principios de la seguridad de la información que establece que la información recibida es auténtica y legítima, ya que no ha sido modificada de su formato original durante los procesos de comunicación, se conserva tal y como la envió el emisor.[19]

CONFIDENCIALIDAD: hace referencia a la protección de la información para que solo puedan acceder a ella el personal autorizado dentro de la organización, y no puede ser revelada a terceros que dentro de sus responsabilidades no requieran de dicha información para desarrollar sus funciones. [20]

DISPONIBILIDAD: define que la información siempre debe estar disponible dentro de cualquier medio digital o físico para que pueda ser procesada y que aporte al correcto funcionamiento de la organización, garantizando el acceso ininterrumpido de los diferentes usuarios que cuentan con autorización para acceder a ella.[21]

INTEGRIDAD: La información generada debe ser conservarse exactamente igual y no puede ser sometida a modificaciones no autorizadas, durante procesos de comunicación, para llegar sin ser copiada, borrada o modificada a su destino. La integridad de la información puede verse comprometida cuando los datos no son cifrados, que pueden encontrarse en diferentes sitios como bases de datos, registros, documentos, entre otros.[22]

---

[18] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2.
[19] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2, p. 193-197.
[20] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2, p. 193-197.
[21] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2, p. 193-197.
[22] VERA, Víctor Daniel Gil; VERA, Juan Carlos Gil. Seguridad informática organizacional: un modelo de simulación basado en dinámica de sistemas. Scientia et technica, 2017, vol. 22, no 2, p. 193-197.

4.2 MARCO TEÓRICO

4.1.1. Protocolo MQTT

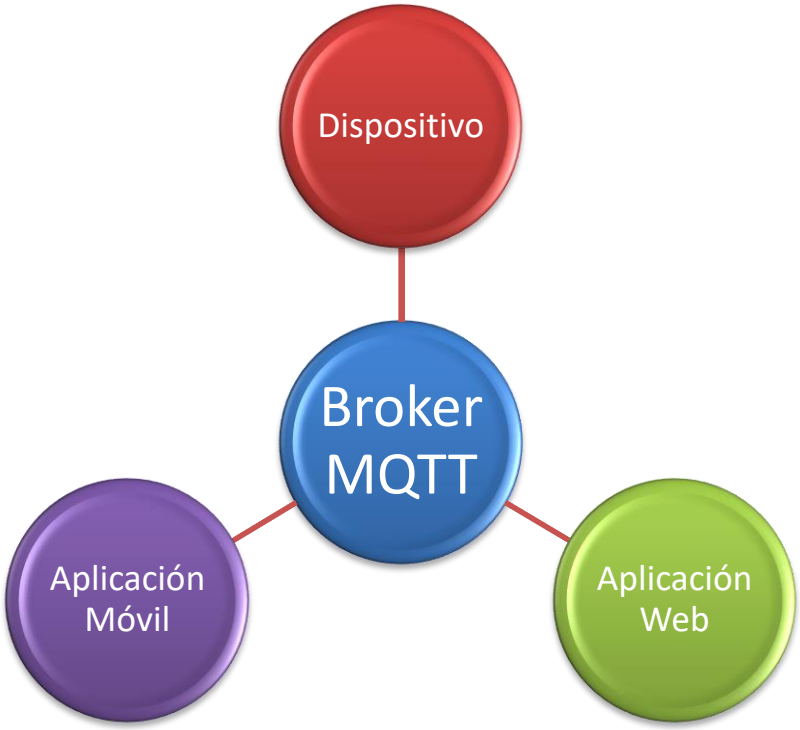
El protocolo de transporte MQTT (Message Queue Telemetry Transport) es utilizado para la comunicación de entornos bajo la tecnología de internet de las cosas y funciona dentro del protocolo de control de transporte TCP[23]. Fue creado por IBM como un protocolo máquina a máquina y estandarizado por ISO/IEC 20922, es un protocolo de mensajes que no requiere actualizaciones minimizando el uso de recurso y permitiendo que se trabaje en un ambiente con bajo ancho de banda.

4.1.2. Componentes MQTT

El protocolo MQTT trabaja bajo el modelo de publicación-suscripción y tiene 3 componentes básicos: El Broker MQTT que es el servidor de las comunicaciones dentro del protocolo, los dispositivos, las aplicaciones móviles o aplicaciones web, estos últimos 3 son considerados los clientes, como lo muestra la figura 1:

[23] Dinculeana, Dan, Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices, 2018.

Figura 1 Componentes MQTT



Fuente: elaboración propia

La principal característica del protocolo MQTT es el desacople tridimensional que se da bajo 3 aspectos:

Espacio: Significa que el publicador y el suscriptor no se conocen.

Tiempo: El proceso de publicación y suscripción no coinciden temporalmente, es decir que no es requisito que la comunicación se de en tiempo real.

Sincronización: No es necesario que el publicador y suscriptor estén sincronizados pues pueden realizar otras tareas sin recibir ni emitir mensajes.

Los dispositivos dentro de las comunicaciones pueden realizar las siguientes acciones básicas:

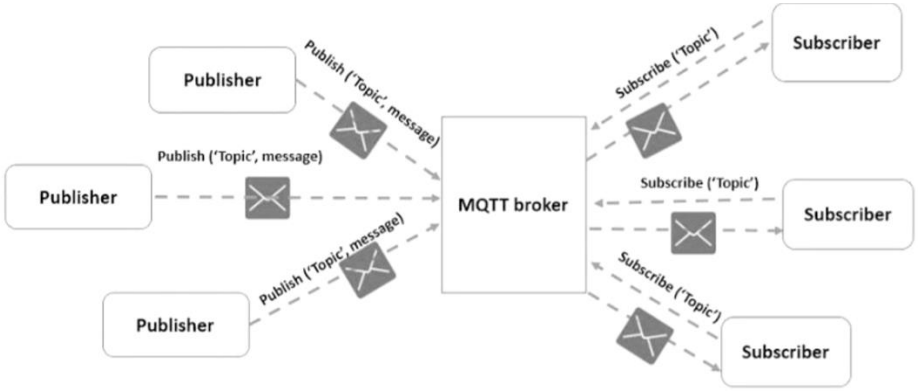
- Conectar
- Publicar

- Suscribirse
- Desuscribirse

4.1.3. Modelo de comunicación Publicar/Suscribir

El modelo de publicación/suscripción tiene 3 componentes principales, como lo muestra la figura 2:

Figura 2 Modelo de comunicación MQTT



Fuente: Mathews, Suja y Gondkar, Raju. Protocol Recommendation for Message Encryption in MQTT. 2017.

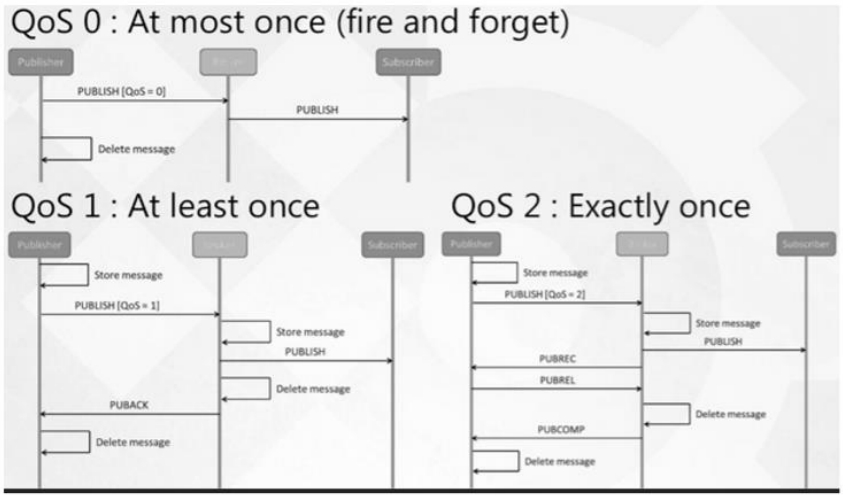
Los componentes son:

- **Editor:** producen o publican la información
- **Suscriptor:** son los que consumen la información generada por los editores.
- **Corredor:** controla y coordina todo el proceso de suscripción

4.1.4. Calidad de servicio de los mensajes en MQTT

De acuerdo con la especificación del protocolo MQTT, Se proporcionan 3 modos de calidad de servicio (QoS) para entrega del mensaje. Los cuales se muestran en la figura 3.[24]:

Figura 3 QoS mensajes MQTT



Fuente: Mathews, Suja y Gondkar, Raju. Protocol Recommendation for Message Encryption in MQTT. 2017.

- QoS=0: Modo "disparar y olvidar", también conocido como "como máximo una vez": en este caso, la pérdida de mensajes puede ocurrir; por lo tanto, se puede usar en entornos donde la medición individual no es de gran importancia, ya que luego se publicaría una próxima.
- QoS=1: "Entrega confirmada" o "al menos una vez": pueden enviarse mensajes duplicados, sin embargo, todos los mensajes llegan.
- QoS=2: "Entrega asegurada" o "exactamente una vez": en este modo, todos los mensajes llegan a su destino exactamente una vez. Esta QoS es útil en aplicaciones donde faltan o los mensajes duplicados conducen a resultados no deseados, ya que pueden suceder en un servicio de pago, por ejemplo.

[24] Nastase, Lavinia, Security in the Internet of Things: A Survey on Application Layer Protocols, 2017.

4.1.5. Aplicaciones MQTT

El protocolo MQTT se utiliza dentro de las tecnologías de internet de las cosas dado que es un protocolo que requiere bajo ancho de banda y donde se envía información con una tasa máxima de 256 Mbps reduciendo el tráfico de la red por eso se utiliza principalmente en:

- Telemetría
- Hogares inteligentes
- Ciudades inteligentes
- Aplicaciones de mensajería instantánea
- Aplicaciones de comunicación
- Aplicaciones de notificaciones

4.1.6. Seguridad en MQTT

[25]La parte interesante de la especificación MQTT es el hecho de que no ha impuesto mecanismos de seguridad, porque está diseñado para operar en redes seguras, desarrollado para necesidades específicas. Por lo tanto, no es una buena idea crear una red global MQTT, porque como el tamaño del árbol crece, la complejidad aumenta.

[26]Sin embargo, el entorno de IoT, hoy en día requiere un estándar para la autenticación y por lo tanto, MQTT se basa en Cifrado SSL/TLS; de lo contrario, el nombre de usuario y la contraseña se enviaría en texto plano. Durante el saludo, el cliente valida el certificado del servidor, que significa que verifica su identidad para autenticarla.

En cuanto a seguridad, el protocolo cuenta con mecanismos bastante simples entre los que se encuentran: un sistema de autenticación mediante usuario y contraseña,

[25] Nastase, Lavinia, Security in the Internet of Things: A Survey on Application Layer Protocols, 2017.
[26] Nastase, Lavinia, Security in the Internet of Things: A Survey on Application Layer Protocols, 2017.

el modelo de seguridad MQTT se dividen en diferentes capas con el objetivo de prevenir diferentes ataques específicos:

- *Capa de red:* garantiza una red física segura entre el cliente y el bróker para lo cual se puede implementar redes privadas virtuales.
- *Capa de transporte:* utiliza TLS como mecanismo ideal para cifrar la información lo que permite tener un túnel cifrado para intercambiar información entre el servidor y el cliente.
- *Capa de aplicación:* utiliza el mecanismo de autenticación con usuario y contraseña para asegurar la información transmitida.

4.1.7. Vulnerabilidades del protocolo MQTT

Algunas de las vulnerabilidades del protocolo se presentan en:

- *Privacidad de los datos:* no existe cifrado de los datos en los mensajes MQTT, aunque haya un sistema de autenticación eso no evita que la información pueda ser interceptada.
- *Autenticación:* si no se proporciona el nombre de usuario contraseña, o si se proporciona un nombre de usuario o contraseña incorrectos. Un atacante se puede encontrar en la misma red que el editor. Por lo tanto, el atacante puede detectar el tráfico en la red mientras espera que un paquete "Conectar" del editor esté en tránsito para que se pueda revelar el nombre de usuario y la contraseña que se utilizan para conectarse al bróker.
- *Integridad de los datos:* una vez que el atacante puede ingresar y ver el tráfico no cifrado puede rastrear los paquetes y modificar los datos que contienen.
- *Oscuridad portuaria:* el puerto oficial del protocolo MQTT es el 1883 para MQTT regular y para MQTT sobre SSL/TLS es el 8883, en este caso un administrador intermediario puede configurar el puerto no estándar en el sistema si solo se cuenta con mecanismos de seguridad únicamente del protocolo MQTT, un atacante puede continuar monitoreando los paquetes que pasan a través de la red.

- *Botnet sobre MQTT:* se ha evidenciado que una botmaster es capaz de enviar órdenes a los bots a través del protocolo MQTT convirtiendo al atacante en un servidor intermediario que reciba la información de los dispositivos clientes. [27]

4.1.8. Ataques DoS en MQTT

- *Ataque de inundación SYN:* se envía gran cantidad de solicitudes SYN al bróker para que comience el saludo de tres vías hasta agotar los recursos de la comunicación del momento, y así se acabe el tiempo y el servidor sea incapaz de responder.
- *Ataque de inundación CONNECT:* Se envían varios mensajes de connect para iniciar una sesión MQTT, estos mensajes ya contienen credenciales de acceso por lo que se envía gran cantidad de mensajes hasta lograr la conexión con el bróker.
- *Ataques de mensajes con QoS alto:* Se envía gran cantidad de mensajes QoS hasta dejar sin recursos la comunicación dado que los mensajes con alto QoS requieren más recurso de lo normal.

[27] Mathews, Suja y Gondkar, Raju. Protocol Recommendation for Message Encryption in MQTT. 2017.

5. DISEÑO METODOLÓGICO

5.1. METODOLOGÍA TIPO EXPLORATORIO

La metodología aplicada a la monografía, teniendo en cuenta que se busca medir el nivel de seguridad de un protocolo de comunicaciones y de esta manera dimensionar cuales son las mayores vulnerabilidades de este, se elige la metodología de tipo exploratorio, ya que se ha formulado el problema para poder obtener información relevante que aporte a su solución a través de la consulta bibliográfica.

[28]La metodología de tipo exploratorio contempla esencialmente 2 tipos de acciones:

- El estudio de la documentación, que se refiere a la construcción del trabajo realizado por otros: revisión de archivos, informes, estudios y otro tipo de documentos o publicaciones.
- Los contactos directos con la problemática a estudiar que se pueden realizar después o simultáneamente con la revisión de la documentación. Probablemente, solo una pequeña parte del conocimiento y la experiencia existente se encuentre en forma escrita.

La presente monografía utiliza la investigación basada en literatura, haciendo una revisión y consulta bibliográfica de datos, estadísticas y diferentes análisis sobre la problemática planteada para poder generar resultados orientados a la solución de dicha problemática,

5.2. TECNICAS PARA LA RECOLECCIÓN DE INFORMACIÓN

[28] Cauas, Daniel, Definición de las variables, enfoque y tipo de investigación, 2015.

5.2.1. Análisis documental

Permite obtener la información de fuentes secundarias como libros, artículos, revistas y documentos de sitios web, para poder recolectar los datos sobre las variables o hipótesis de interés, para seleccionar la información a analizar y determinar fenómenos, o dimensionar los objetivos de la investigación. Para el caso de la monografía se consultan estos diferentes tipos de fuentes con el fin de recolectar todos los datos relevantes respecto a la seguridad del protocolo MQTT

6. DESARROLLO DE LOS OBJETIVOS

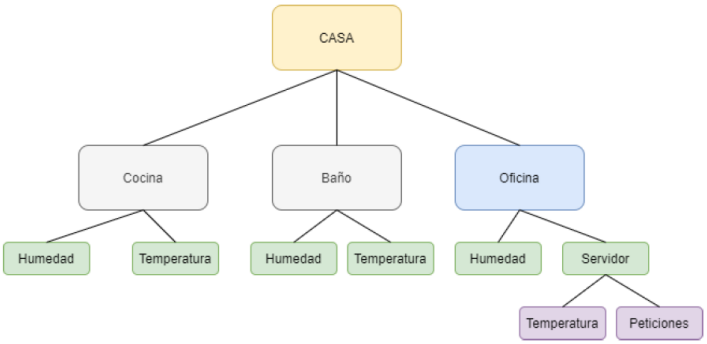
6.1. DETERMINAR LAS VULNERABILIDADES DEL PROTOCOLO DE TRANSPORTE MQTT EN COMUNICACIONES BAJO LA TECNOLOGÍA DE INTERNET DE LAS COSAS, MEDIANTE REVISIÓN DE LITERATURA.

6.1.1. Arquitectura de un sistema MQTT

Teniendo en cuenta que el protocolo MQTT trabaja bajo el modelo de publicación/suscripción, su arquitectura se basa en una topología de tipo estrella ya que los publicadores y subscriptores se conectan a un mismo bróker, pero no hay ningún tipo de comunicación entre los clientes (subscriptores). Esto quiere decir que a los brokers les puede llegar mensajes de todos los publicadores, pero cada uno de estos trabaja de manera independiente y no poseen relación entre los mismos.

En la figura 4 se puede observar que el elemento casa posee varios publicadores de temperatura y humedad, y un cliente que esté suscrito puede recibir información de todos sin inconveniente ya que no existe relación entre dichos publicadores porque cada uno trabaja independientemente.

Figura 4 Ejemplo arquitectura MQTT

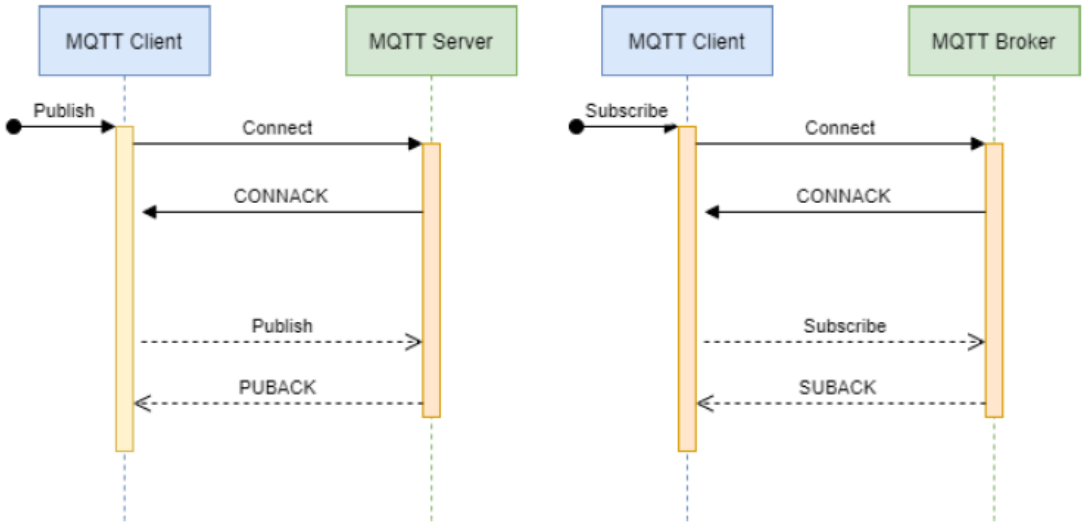


Fuente: Mathews, Suja y Gondkar, Raju. Protocol Recommendation for Message Encryption in MQTT. 2017.

6.1.2. Operaciones MQTT

Como se observa en la figura 5, para el proceso de comunicaciones en MQTT, lo que sucede inicialmente es que el cliente MQTT se conecta al bróker, a través del protocolo TCP/IP, utilizando un paquete CONNECT, este es confirmado cuando se genera un mensaje de reconocimiento CONNACK que genera el mismo cliente, dado que no puede publicar o recibir mensajes, si se ha establecido una conexión previa. Para poder que se establezca la conexión cada uno de los clientes necesita tener contar con un identificador clientID único, ya que, si un cliente intenta conectarse con un identificar que ya hace parte de la red, se desconectará al cliente legítimo que ya actualmente se encuentra conectado.

Figura 5 flujo de mensajes en MQTT



Fuente: Mathews, Suja y Gondkar, Raju. Protocol Recommendation for Message Encryption in MQTT. 2017.

Luego de este proceso, cada que se envía un mensaje al bróker, dicho mensaje va como un paquete PUBLISH y se recibe un mensaje de reconocimiento PUBACK que genera el bróker y, y cada que el cliente recibe un mensaje del bróker, este será

un paquete SUBSCRIBE para que el cliente finalmente envíe un mensaje de reconocimiento ACK SUBACK.

Se debe tener en cuenta que si un cliente no envía mensajes de manera continua, entonces este puede enviar un paquete tipo Keep alive, con PINGREQ para que el bróker envíe un mensaje de respuesta con un PINGRESP, para de esta manera poder conservar la conexión activa, esto debido a que si un cliente no envía ningún mensaje en un periodo de tiempo mayor a 60 segundos perderá la conexión, sin embargo dicho intervalo de tiempo puede ser modificado y configurado de acuerdo con los requerimientos que sean necesario, y en caso de que la intención del cliente sea desconectarse del bróker, entonces deberá enviar un paquete tipo DISCONNECT.[29]

6.1.3. Seguridad del protocolo MQTT

6.1.3.1. Conexión autenticada

La comunicación MQTT establece una conexión TCP la cual no está cifrada, lo que permite que se pueda acceder a la información de autenticación fácilmente, y un atacante podría obtener las credenciales de acceso de cualquier dispositivo. Por tal razón se recomienda utilizar TLS para cifrar las conexiones, pero aplicar dicha práctica sobrecarga el CPU del bróker y representa también una sobrecarga a los clientes, por eso en la mayoría de los procesos de implementación se descarta esta técnica ya que en las comunicaciones IoT se prioriza la optimización de los recursos porque son muy limitados en los dispositivos.

[29] Mathews, Suja y Gondkar, Raju. Protocol Recommendation for Message Encryption in MQTT. 2017

Teniendo en cuenta la limitación de recursos de los brokers, estos permiten la conexión de clientes sin autenticación, lo que desencadena una vulnerabilidad y riesgo de suplantación de identidad o modificación de la información afectando la integridad de los datos.

6.1.3.2. Seguridad de los mensajes

Los mensajes dentro de la comunicación bajo el protocolo MQTT no están cifrados por lo que pueden ser manipulados afectando la integridad de la información para evitar que terceros puedan modificar los mensajes se agregan elementos como firma digital, MAC o ckecksum, teniendo en cuenta los requerimientos se selecciona alguno de estos métodos.

Los niveles de QoS del protocolo también permiten que los mensajes sean entregados a los clientes garantizando su integridad y evitando la modificación de la información.

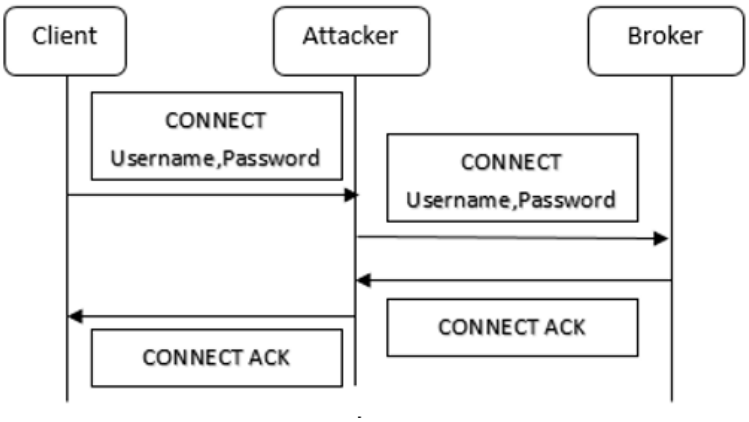
6.1.4. Escenarios de ataque

Existen diferentes escenarios en los que se pueden evaluar las vulnerabilidades del protocolo de transporte MQTT, para analizarlo desde el aspecto de comunicaciones IoT, se pueden implementar escenarios de pruebas con Raspberry Pi y lenguaje Python y plataformas como Mosquito, adicional utilizando el analizador de paquetes wireshark.
Con esta sencilla implementación se pueden identificar las siguientes fallas de seguridad:

- **No hay mecanismo de autenticación:** teniendo en cuenta los procesos dentro de las comunicaciones MQTT y que no existen mecanismos de autenticación, cualquier usuario puede publicar o suscribirse a cualquier tema generando un riesgo para la confidencialidad de los datos.

- **Credenciales de usuario enviadas como texto plano:** aunque se implemente un sistema de autenticación las credenciales de acceso de los datos de las credenciales de usuarios se encuentran en texto plano es decir que si alguien logra obtener estos datos los va a poder utilizar fácilmente para el ingreso a los sistemas, como lo indica la Figura 6.

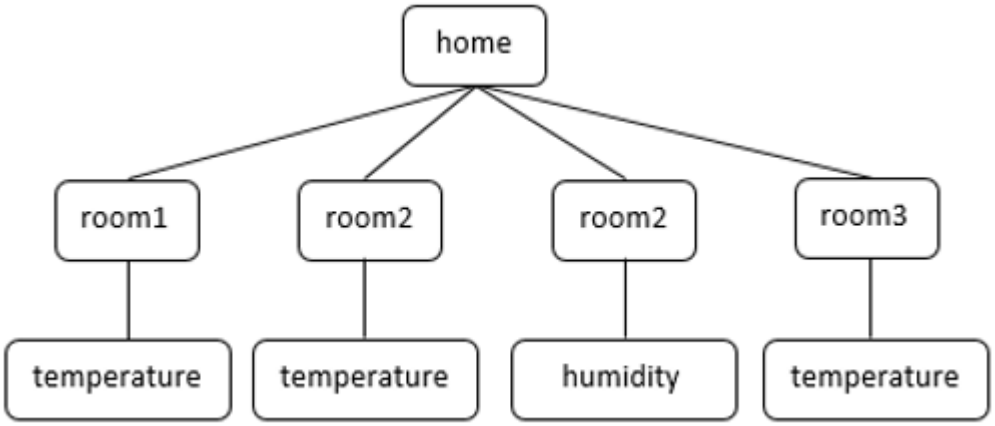
Figura 6 Escenario de contraseñas en texto plano



Fuente: Mathews, Suja y Gondkar, Raju. Protocol Recommendation for Message Encryption in MQTT. 2017.

- **Uso de wildcards en los nombres de los temas:** esto consolida una amenaza de seguridad ya que, si un bróker mantiene los diversos temas de la jerarquía, cualquier usuario puede utilizar las wildcard de cualquiera de los temas en la jerarquía para publicar o suscribir datos sin estar autorizado, si se tienen niveles de jerarquía como lo muestra el ejemplo de la figura 7.

Figura 7 Ejemplo Niveles de Jerarquía MQTT



Fuente: Mathews, Suja y Gondkar, Raju. Protocol Recommendation for Message Encryption in MQTT. 2017.

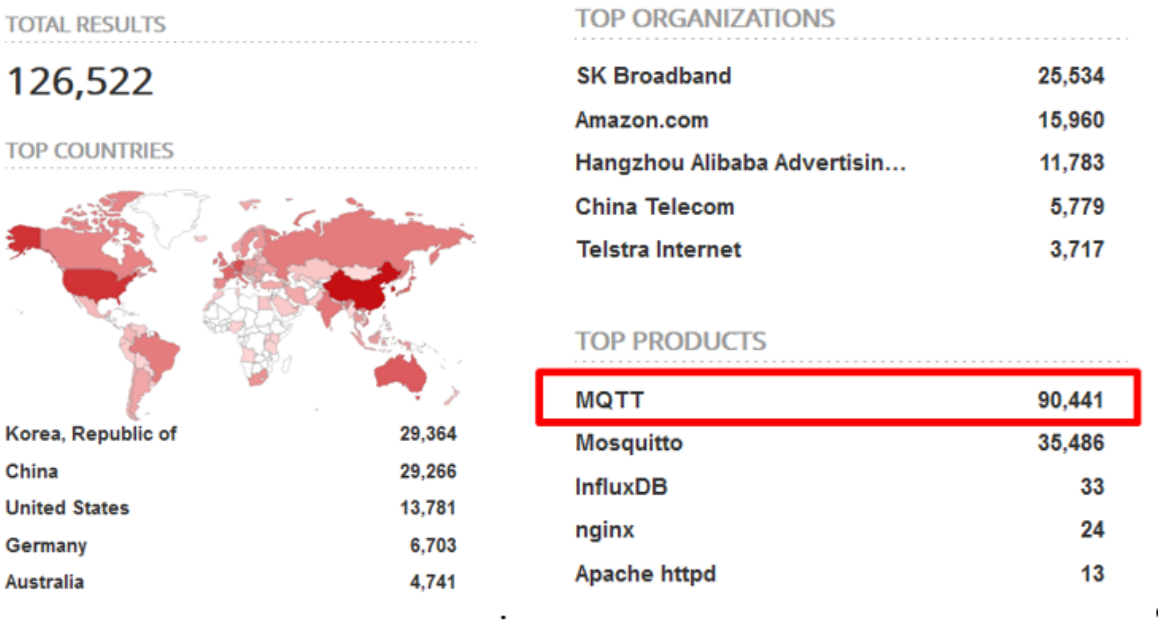
6.1.5. Búsqueda de vulnerabilidades

6.1.5.1. Uso de Shodan

Shodan es un motor de búsqueda que permite encontrar sistemas y servicios conectados a internet y obtener información de estos que puede ser utilizada para procesos de auditoria como también puede ser utilizada por los ciberdelincuentes para determinar la ubicación y direcciones IP de dispositivos y de esta manera poder generar ataques contra dichos dispositivos.

Shodan hace uso de una API de Python para la búsqueda de dispositivos conectados a internet de acceso público a través de los cuales podemos acceder a sistemas informáticos o servidores, en la figura 8 se puede observar que hay más de noventa mil dispositivos accesibles que utilizan el protocolo MQTT.

Figura 8 Resultados Shodan MQTT



Fuente: Shodan. Search Engine for the Internet of Everything. [En línea] 2021. Recuperado en 2021-05-30. Disponible en: https://www.shodan.io/

6.1.5.2. Vulnerabilidad de autenticación

La causa principal de que estos dispositivos que trabajan bajo el protocolo MQTT sean de acceso público es que el bróker MQTT no cuenta con un sistema de autenticación establecido.

Como el protocolo trabaja bajo el modelo publicación/suscripción; al no haber mecanismo de autenticación, cualquier usuario puede publicar o suscribirse a cualquier tema lo que representa una amenaza para la confidencialidad de los datos. Los datos que el Bróker está tratando podrían contener información sensible y esta información se puede obtener simplemente proporcionando el nombre del tema. Implementar un sistema de autenticación mitiga el riesgo de este tipo de ataque al permitir que solo las entidades registradas publiquen y suscriban datos.

Además existen códigos de respuesta del protocolo que permiten identificar cuando un dispositivo no ha implementado métodos de autenticación y es accesible, estos datos se pueden obtener haciendo uso de la herramienta shodan y un código simple para obtener dichos códigos, de esta manera se vulnera la integridad de los datos pues el atacante que logre acceder al tráfico puede modificar los datos en tránsito filtrando los datos de los paquetes, la autenticación puede realizarse mediante diferentes métodos, para determinar el método más adecuado es necesario revisar la infraestructura del sistema y así determinar ¿cuál método se adecua más a los requerimientos?

6.1.5.3. Vulnerabilidad de obstrucción de puerto

El número de puerto oficial de IANA utilizado por MQTT es 1883 para MQTT normal y 8883 para MQTT utilizando SSL/TLS. Sin embargo, un administrador de bróker puede configurar para usar el puerto no estándar en el sistema. Si el mecanismo de seguridad solo depende del protocolo MQTT en sí, el atacante aún puede observar fácilmente los paquetes que pasan por la red haciendo uso de un sniffer como Wireshark para rastrear el paquete y aplicar el filtrado de datos.

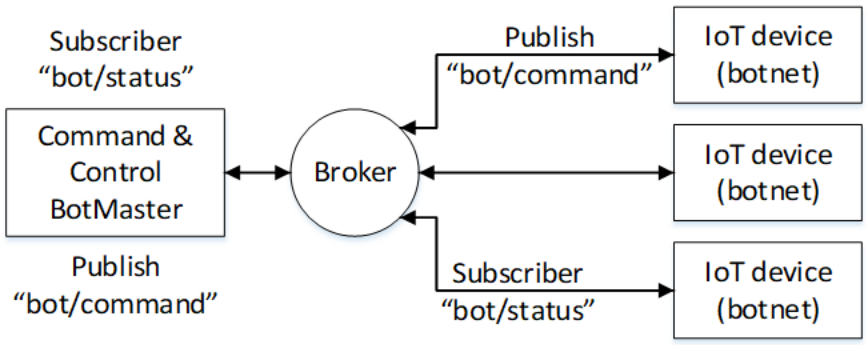
6.1.5.4. Botnet en MQTT

En una de las convenciones DefCon se demostró que BotMaster envió un comando a los bots sobre el protocolo MQTT, una botnet es una red que consta de muchos bots, un nuevo tipo de malware instalado en una computadora comprometida, que luego puede ser controlado por BotMaster. Con esto a través de herramientas como Shodan se puede lograr un corredor que se convierte en un servidor gratuito que conecta al atacante con el dispositivo de la víctima, al usar este escenario, el

atacante puede esconderse de cualquier investigación porque usa al corredor no seguro como árbitro para comunicarse con la botnet.

La figura 9 muestra que BotMaster actúa como comandante de una botnet y usa un determinado intermediario para controlar muchos dispositivos IoT (botnet) a la vez con solo un mensaje publicado en un tema específico. BotMaster también puede recibir el estado de la víctima y suscribirse al estado de cada dispositivo de IoT (botnet). Este escenario es muy eficiente, especialmente si BotMaster quiere dar un comando a todas las botnet a la vez, por ejemplo, lanzar un ataque DDoS, enviar una gran cantidad de correos electrónicos no deseados o de phishing.

Figura 9 Control MQTT mediante Botnet



Fuente: Mathews, Suja y Gondkar, Raju. Protocol Recommendation for Message Encryption in MQTT. 2017.

6.1.5.5. Denegación de servicio

El politécnico Di Milano en un estudio realizado en conjunto con TrendMicro encontró varias fallas de seguridad del protocolo MQTT porque permite que se filtren mensajes de los agentes colocando al descubierto información sensible sobre datos que pueden ser utilizados para el robo de información y ataques de denegación de

servicio, ya que los atacantes pueden controlar de manera remota cada end point de IoT y al lograr el acceso poder estar dentro de la red sin ser identificado.[30]

Para las comunicaciones bajo el protocolo MQTT se tienen distintos dispositivos interconectados, esto permite que se pueda desplegar ataques de tipo denegación de servicio, que tienen como objetivo inhabilitar el bróker y de esta manera dejar fuera de funcionamiento toda la red de comunicación MQTT, alguno de los ataques que se pueden ejecutar son:

- Inundación por CONNECT: en este tipo de ataque lo que sucede es que el bróker recibe gran cantidad de solicitudes de conexión hasta que es saturado quedando inhabilitado para poder procesar las nuevas solicitudes de conexiones.
- Inundación por CONNECT retrasados: en este tipo de ataque se logra que haya el máximo número de conexiones posibles al bróker para saturarlo y que quede inhabilitado para recibir nuevas conexiones legítimas.
- Inundación por CONNECT grandes: este tipo de ataque consiste en que se agrega un payload dentro de los paquetes CONNECT, el payload debe ser de un gran tamaño para que se encargue de consumir todo el ancho de banda y todo el recurso CPU del bróker.
- Inundación por Subscripciones Inválidas: este tipo de ataque consiste en que dentro de los topics se intenta publicar mensajes que no contengan los permisos necesarios para el uso de recursos por parte del servidor MQTT, pero que sean consumidos al verificar los permisos para cada una de las peticiones.[31]

[30] Digital security, Encontrados importantes fallos en los principales protocolos de IoT. (2018) Recuperado en: 2021-06-02. Disponible en: https://www.itdigitalsecurity.es/actualidad/2018/12/encontrados-importantes-fallos-en-los-principales-protocolos-de-iot

[31] Portas. L, TFM, (2019). Recuperado en 2021-06-19.

6.1.5.6. Suplantación de identidad

Obtener las credenciales de acceso de algún usuario de la red para suplantarlo y de esta manera comenzar a publicar mensajes con información falsa. La obtención de las credenciales se puede lograr a través de ataques por fuerza bruta o por ingeniería social, además de publicar mensajes el atacante también puede suscribirse a cualquier elemento de la red obteniendo acceso a información en tiempo real para la cual no está autorizado. También se puede suplantar el bróker, con esto el atacante tendría control total de la red MQTT.

La suplantación de identidad puede permitir diferentes acciones como:

- Elevación de privilegios: el atacante puede publicar mensajes o suscribirse a uno o varios temas, teniendo en cuenta que MQTT permite el uso de wildccards.
- Revelación de información: el atacante puede acceder al contenido de los mensajes tanto de los clientes como del bróker, y si accede a la información del bróker puede obtener otro tipo de datos como credenciales de acceso para ingresar a todos los dispositivos, esto se puede generar a través de un ataque de hombre en el medio que se puede mitigar implementando el cifrado de los datos.
- Modificación del contenido: al modificar el contenido de los mensajes de la comunicación MQTT se puede afectar el funcionamiento de la red pues las acciones de los dispositivos dependen del contenido de los mensajes.

6.1.5.7. Malaria MQTT

El ataque de Malaria consiste en propagar diferentes ataques, mediante sistemas basado en mosquito en contra del protocolo MQTT, lo que se hace es que se simula gran cantidad de clientes que publican mensajes que poseen un tamaño especifico y que tienen configurada una velocidad de envío mayor. Los ataques que se pueden

realizar son de tipo pasivo y activo, realizando todas las publicaciones en contra de las víctimas o como un ataque de hombre en el medio simplemente visibilizando el tráfico, de esta manera se obtiene un monitoreo y análisis permanente de cada uno de los mensajes enviados y recibidos durante un periodo de tiempo específico. [32]

[32] Multiclass Classification Procedure for Detecting Attacks on MQTT-IoT Protocol. Complexity. Alaiz Moreton, Hector & Aveleira, Jose & Ondicol-Garcia, Jorge & Muñoz-Castañeda, Angel & García, Isaías & Benavides, Carmen. (2019).

6.2. ESTABLECER A PARTIR DE REVISIÓN DE LA LITERATURA MECANISMOS Y HERRAMIENTAS PARA MITIGAR LAS VULNERABILIDADES ENCONTRADAS EN EL PROTOCOLO DE TRANSPORTE MQTT

Teniendo en cuenta que el protocolo MQTT es uno de los más utilizados para las comunicaciones en IoT y que al ser utilizado por una gran cantidad de dispositivos debido a sus fallas de seguridad se encuentran expuestos muchos datos que permiten diferentes tipos de ataques a las comunicaciones, teniendo en cuenta esto se pueden determinar algunas alternativas de control de la seguridad de las comunicaciones que utilizan el protocolo.

6.2.1. Modelos de referencia

Los modelos de referencia dentro de las comunicaciones IoT establecen como debe realizarse la implementación de las conexiones y la comunicación de los componentes IoT de manera estandarizada y las características de cada modelo permiten que se garantice en un mayor nivel la seguridad de los datos:

6.2.1.1. Modelo de referencia IoTWF

Trabaja bajo 7 capas, lo que permite una mayor interacción de los datos, para que estos tengan un mejor tratamiento y análisis antes de ser almacenados en la base de datos, comprendiendo la función de cada capa y su importancia para el correcto funcionamiento de la red.

En este modelo el principio fundamental es el flujo de información abarcando aspectos importantes de los dispositivos conectados como: integración, interoperabilidad y desacoplamiento.

[33]Las capas del modelo son:

- **Capa1:** esta capa hace referencia a los dispositivos conectados y los controladores, básicamente es donde se encuentran todo lo que llamamos objetos o cosas dentro de IoT, la función principal de esta capa es la generación de datos para poder ser analizados y/o controlados dentro de una red.
- **Capa2:** en esta capa se hace referencia a la conectividad, donde se debe garantizar una transmisión de datos oportuna y confiable para cada uno de los dispositivos de la capa 1 que generan los datos que luego serán procesados por la capa 3.
- **Capa3:** hace referencia a la computación de borde, su función principal está enfocada al tratamiento de los datos, filtrado, evaluación, almacenamiento, alertas y notificaciones, esto con el fin de garantizar que el flujo de datos no exceda cada uno de los límites establecidos para cada uno de los parámetros de la comunicación, y de esta manera la información pueda ser manejada por las capas superiores sin inconvenientes.
- **Capa4:** esta capa hace referencia a la acumulación de datos, es donde se capturan y almacenan los datos para que puedan ser consultados por las diferentes aplicaciones en el momento en que lo requieran, y dichos datos se convierten en información importante que puede ser procesada por las capas superiores del modelo.
- **Capa5:** hace referencia a la abstracción de datos es donde se realiza la consolidación de los datos provenientes de las capas inferiores, y los agrega a diferentes centros de datos, dicho proceso se realiza a través de métodos de virtualización.

---

[33] ANÁLISIS COMPARATIVO DE TRÁFICO ENTRE LA RED WIFI Y LA RED IOT-WIFI EN EL CAMPUS SUR DE LA UNIVERSIDAD POLITÉCNICA SALESIANA. Hidalgo, Edward (2019). Universidad Politécnica Salesiana, https://dspace.ups.edu.ec/bitstream/123456789/16811/1/UPS-ST003902.pdf

- **Capa6:** hace referencia a las aplicaciones software donde son interpretados cada uno de los datos generados, es aquí donde se generan diferentes resultados de análisis de acuerdo con los datos obtenidos, dicha información debe estar disponible para ser consultada desde cualquier dispositivo móvil.
- **Capa7**: hace referencia a la colaboración y procesos, es donde se realiza la interacción con el usuario final, que puede decidir cualquier tipo de cambio de configuración o implementación de la red IoT basándose en las necesidades y requerimientos de su organización.

6.2.1.2. **Modelo de referencia Intel IoT**

Trabaja con 6 capas y un componente de seguridad adicional que trabaja de manera integral a las 6 capas con el objetivo de garantizar seguridad en cada uno de los dispositivos IoT que sean agregados a la red para garantizar que los datos que serán entregados a la nube son legítimos. [34]. A continuación, se realiza una descripción de las capas más importantes.

- Capa1: es la capa encargada de la conectividad y las comunicaciones se encarga de hacer uso de diferentes tipos de protocolos entre los distintos dispositivos que se conectan a través de una red PAN/LAN o WAN.
- Capa 2: es la capa encargada del análisis de los datos, su función es llevar control de los datos y para lograr esto hace uso de computación de borde.
- Capa 3: es la capa de administración se encarga de supervisar cada una de las operaciones sobre los dispositivos conectados.

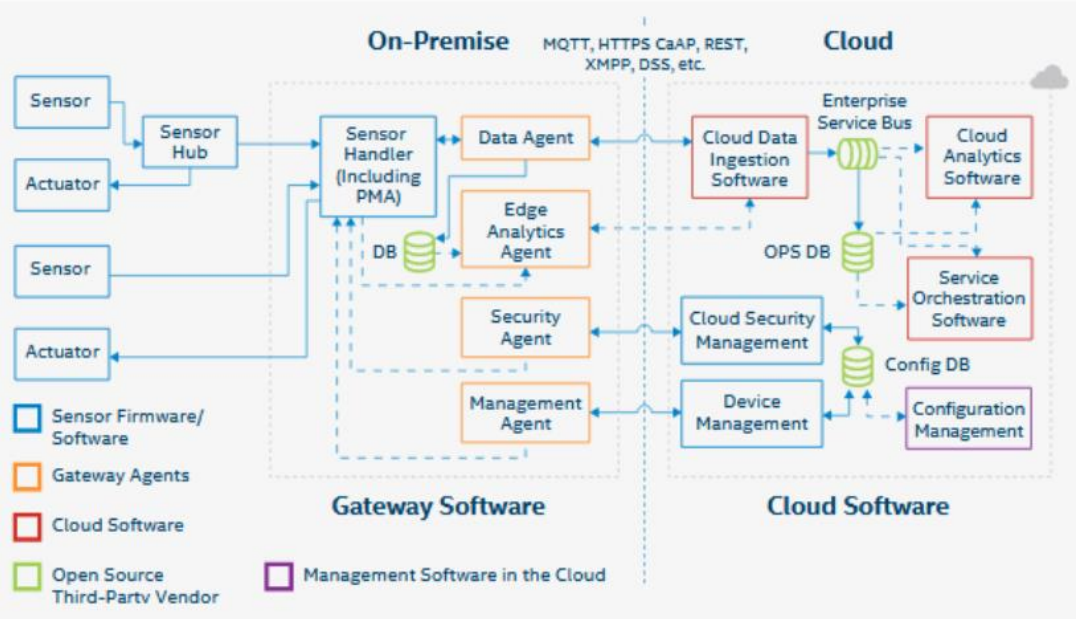
---

[34] ARQUITECTURAS DE REFERENCIA PARA IOT CON TRANSFERENCIA SEGURA DE INFORMACIÓN. Velez, Andres (2019). Universidad Nacional Abierta y a Distancia. https://repository.unad.edu.co/bitstream/handle/10596/27648/avelezpe.pdf?sequence=4

- Capa 4: es la capa de control, sus funciones son separadas de la capa de gestión y se encarga del control de acceso y de las políticas de seguridad para los dispositivos conectados a la red.
- Capa 5: es la capa de la seguridad, se encarga de brindar protección integral a cada una de las capas, y las herramientas y técnicas utilizadas depende de en qué capa se trabaje.

En la figura 10 se describen, los dos conjuntos de componentes del modelo, en el primer grupo de elementos están todos los dispositivos IoT que son sensores, así como también se encuentran los dispositivos Gateway, en el segundo grupo se encuentran todos los componentes encargados de la computación en la nube, que se encargan de toda la gestión, almacenamiento, análisis y seguridad de los datos que generan cada uno de los dispositivos conectados a la red.

Figura 10 Interfaces de modelo de referencia Intel IoT



Fuente: Plataforma unificada para IoT Intel, Ranchal, Juan (2014). Internet of Things My Computer. https://www.muycomputer.com/2014/12/10/internet-de-las-cosas-intel/amp/

6.2.1.3. **Modelo de referencia IoT simple**

[35]Este modelo está compuesto por 5 capas y con un componente de seguridad incluido en todas las capas, este componente utiliza mecanismos como: autenticación, filtrado, cifrado y protección de datos, de manera integral.

- Capa 1: en esta capa se encuentran todos los dispositivos que son actuadores o sensores.
- Capa 2: en esta capa se encuentran los dispositivos Gateway, que dan soporte a cada uno de los dispositivos de la red que no poseen ninguna conexión de tipo TCP/IP.
- Capa 3: esta capa es la encargada del control y gestión de la red de conectividad para los dispositivos conectados.
- Capa 4: es la capa de administración y análisis, se encarga de la gestión de todos los datos recolectados que generan los dispositivos IoT conectados.
- Capa 5: es la capa donde se manejan y almacenan los datos, por tanto debe poseer gran capacidad de almacenamiento y también suficientes recursos para poder realizar el procesamiento de la información.

El componente de seguridad integra algunos principios de la seguridad de la información como son: Cifrado de datos, filtrado de datos, autenticación y protección de la información, con un enfoque es tanto físico como lógico, y que es integral a cada una de las cinco capas del modelo.

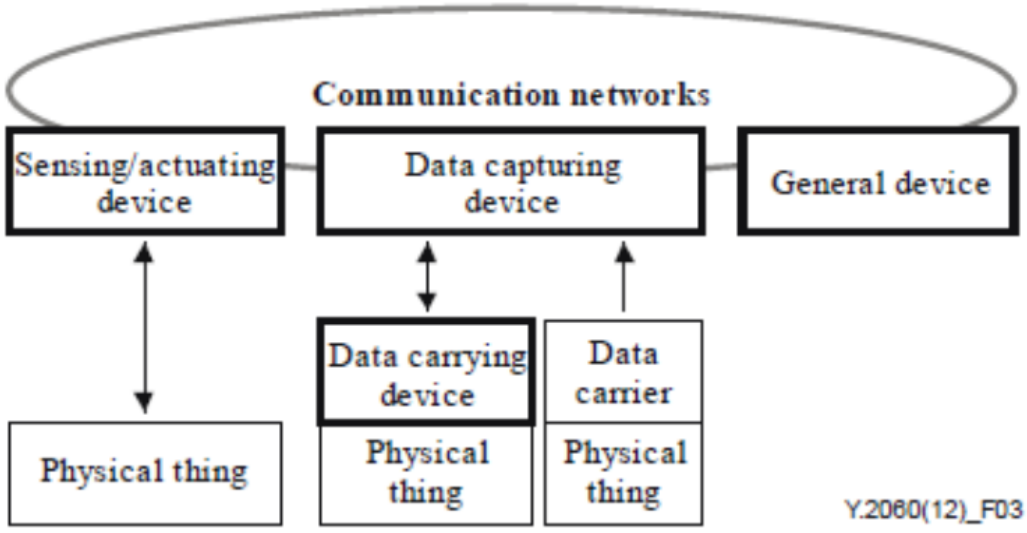
---

[35] ARQUITECTURAS DE REFERENCIA PARA IOT CON TRANSFERENCIA SEGURA DE INFORMACIÓN. Velez, Andres (2019). Universidad Nacional Abierta y a Distancia. https://repository.unad.edu.co/bitstream/handle/10596/27648/avelezpe.pdf?sequence=4

6.2.1.4. Modelo de referencia ITU

Este modelo posee 4 capas y el componente de seguridad integral a todas las capas y adicional posee un componente de gestión para diagnóstico y control remoto de los dispositivos y el componente de seguridad enfocado al acceso, autenticación y autorización para la protección de los datos, como muestra la figura 11

Figura 11 Relación de modelo ITU



Fuente: Internet de las cosas y sus aplicaciones, incluidas las ciudades y comunidades inteligentes. ITU (2016). Asamblea mundial de normalización de las telecomunicaciones AMNT-16

Respecto a gestión, este modelo garantiza gestión integral de toda la topología tanto física como lógica de la red, así mismo del tráfico de red generado, en cuanto a los dispositivos permite acciones de actualización, diagnostico, activación y desactivación de los dispositivos conectados de manera remota.

A diferencia de los modelos anteriores que poseen un componente de seguridad integral a las diferentes capas, este ofrece diferentes herramientas según la capa trabajada, respecto a la capa de aplicación se garantizan procesos de autenticación,

autorización, privacidad, e integridad de los datos, adicional se deben realizar procesos de evaluación mediante auditorías, y protección con software antimalware. Respecto a la capa de red se garantizan los principios de autorización, autenticación y confidencialidad, al igual que para la capa de dispositivos que adicionalmente debe ofrecer control de acceso y herramientas para la protección de los datos.

Se debe tener en cuenta que a pesar de que cada capa del modelo tiene unas configuraciones de seguridad sugeridas específicas, la efectividad de dichas técnicas depende del tipo de datos que se manejen según la aplicación y sus requisitos, es decir que depende del tipo de tráfico y la sensibilidad de los datos.

6.2.2. Certificados digitales

Son métodos de seguridad que pueden ser implementados para autenticación dentro de las comunicaciones IoT agregando un cifrado a través de TLS, así los dispositivos IoT conectados se identifican por medio de los certificados.

6.2.3. Criptografía

Implementar cifrado de datos para conservar la integridad de los datos mediante funciones hash agregadas a los datos para verificar su correcto origen y contenido de los datos.

6.2.4. Implementación de políticas para las comunicaciones

Es necesario la creación de políticas de seguridad de la información que establezcan cuales son las responsabilidades y obligaciones de cada uno de los usuarios de la red IoT, así mismo se establezcan los parámetros de seguridad con los que debe cumplir cada dispositivo conectado independiente del tipo de infraestructura que se esté implementando. Las políticas serán el manual para poder determinar si los dispositivos conectados cumplen o no con los requerimientos de la red empresarial, esto para garantizar el cumplimiento de los modelos de arquitectura implementados, sin importar el seleccionado y así mismo que se

garantice el cumplimiento de las normas y estándares establecidos para este tipo de comunicaciones.

[36]Greg Young recomienda revisar toda la infraestructura operativa para implementar políticas adecuadas con el fin de eliminar todos los servicios M2M que no sean utilizados, realizar monitoreo permanente de la red, cumplir con estándares internacionales, así como mantener todas las actualizaciones de los servicios, ya que rápidamente se generan nuevas vulnerabilidades.

[36] Digital security, Encontrados importantes fallos en los principales protocolos de IoT. (2018) Recuperado en: 2021-06-02. Disponible en: https://www.itdigitalsecurity.es/actualidad/2018/12/encontrados-importantes-fallos-en-los-principales-protocolos-de-iot

6.3. EVALUAR LOS MECANISMOS Y HERRAMIENTAS DE MITIGACIÓN A TRAVÉS DE SU IMPLEMENTACIÓN Y ANÁLISIS EN LA COMUNICACIÓN DE DISPOSITIVOS IOT DENTRO DE UN ESCENARIO CONTROLADO.

Se establece una metodología para aplicar sobre un escenario controlado, dicha metodología debe tener en cuenta las fases que se realizan en un proceso de pentesting, pues es un proceso muy ligado a las herramientas de pentesting con el fin de identificar vulnerabilidades y sus posteriores herramientas de mitigación.

- **Identificar:** determinar cuáles son los dispositivos que trabajan bajo el protocolo MQTT dentro del escenario controlado y que tipo de elemento MQTT son y los servicios que utilizan con el fin de investigar que vulnerabilidades pueden ser explotadas.
- **Planear:** determinar los tipos de ataques que se realizaran sobre los dispositivos previamente identificados con el fin de analizar las vulnerabilidades asociadas y los exploits necesarios para explotarlas, y evaluar la posibilidad de ejecutar los ataques que se determinaron en objetivos anteriores.
- **Ejecutar:** determinar el paso a paso para ejecutar los ataques previamente seleccionados y la forma en que serán ejecutados si integralmente o de manera individual desde el escenario controlado.
- **Evaluar:** análisis de los resultados de los ataques ejecutados para determinar las medidas de mitigación que permitan eliminar las vulnerabilidades encontradas y las consecuencias de estas, luego de implementar las medidas de mitigación, se ejecutarán nuevamente los mismos ataques con el fin de evaluar el éxito de la implementación de estas medidas.

6.3.1. Métodos de implementación de comunicación MQTT

6.3.1.1. Instalación de mosquito

Es una herramienta de código abierto que permite la instalación de un bróker con sus suscriptores y publicadores para conformar una red MQTT, es muy útil para las comunicaciones IoT ya que permite la optimización de recursos porque es de muy bajo consumo de recursos. La instalación de mosquito se puede realizar en el sistema operativo Linux, distribución Ubuntu

- **Configuración del bróker**

Se realiza a través de un fichero de configuración que ubica el bróker en el puerto 1883, establece las métricas del bróker, para conocer la cantidad de dispositivos conectados, la cantidad de mensajes publicados; debe contener las configuraciones que permitan conocer el número de clientes conectados, la cantidad de mensajes publicados por minuto, las características del bróker, entre otros datos.

- **Configuración de publicadores y subscriptores**

Se puede publicar cualquier tipo de mensaje pues no se tiene un formato especifico, para ello es necesario conocer la IP del bróker, el puerto, el topic y el mensaje deseado, la suscripción es como la publicación de un mensaje, con eso podemos conocer todos los mensajes que van dirigidos a un topic especifico.

6.3.1.2. Paho-mqtt

Es una aplicación de Python que sirve para lograr la comunicación entre los elementos MQTT como son bróker, publicador y suscriptor, consta de un script que contiene cada uno de los mensajes que se deben publicar y las diferentes acciones según el contenido de esos mensajes.

Para la configuración del publicador, solo se debe conocer la dirección IP y puerto del bróker, así como el topic y el tipo de mensaje a publicar.

Para la configuración del suscriptor, se requiere algunas funciones adicionales para definir cuál es el comportamiento para conectarse y para recibir un mensaje

6.3.1.3. Influx db

Es una base de datos para el almacenamiento de series de datos temporales, es de código abierto y logra soportar una alta carga de escrituras y lecturas. Lo que la convierte en una gran herramienta para almacenar los datos de los diferentes dispositivos IoT conectados ya que dichos dispositivos siempre están enviando gran cantidad de datos en un intervalo de tiempo breve.

6.3.1.4. Grafana

Grafana es una herramienta de código abierto usada para componer gráficas a partir de distintos orígenes de datos. Es una gran herramienta que permite unificar métricas de distinto origen como InfluxDB, ElasticSearch o MySQL, entre otros. Se utiliza para poder crear gráficas en tiempo real de los mensajes obtenidos e insertados posteriormente en InfluxDB.

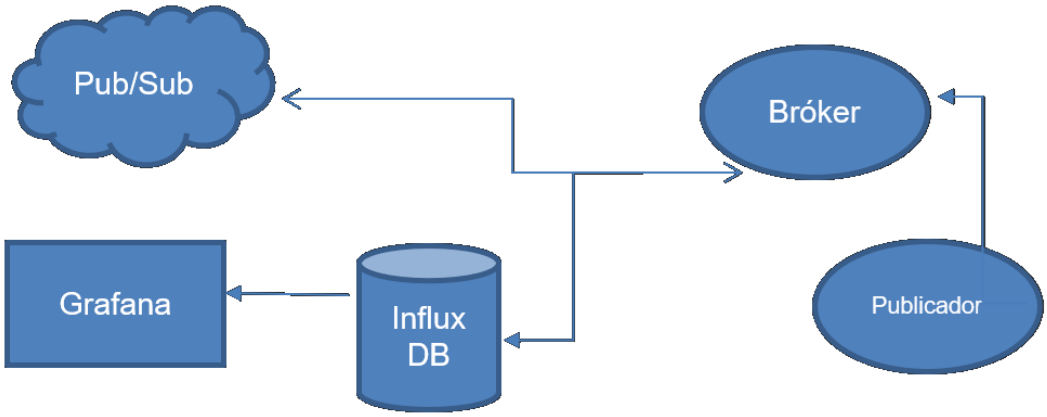
6.3.2. Arquitectura de red de pruebas

Clientes: se simularán mediante una máquina virtual con sistema operativo Linux que contendrá un script de Python de la librería Paho-mqtt donde se configurará los mensajes y acciones de los clientes, todos los datos serán procesados en influxdb y proyectados en grafana.

Bróker: Se utilizará un controlador RaspberryPi3 para simularlo y a donde se dirijan diferentes tipos de ataques para identificar y evaluar las diferentes vulnerabilidades.

Dicho diseño de arquitectura de pruebas para implementar se muestra en la figura 12

Figura 12 Arquitectura de pruebas



Fuente: elaboración propia

6.3.3. Herramientas para el análisis de seguridad

- **Nmap:** permite descubrir dispositivos conectados a la red para encontrar sus posibles vulnerabilidades que pueden ser explotadas para generar un ataque, contiene scripts específicos para comunicaciones MQTT para pruebas de los subscriptores y el bróker.
- **Wireshark:** permite analizar el tráfico para encontrar los paquetes MQTT y descubrir su contenido obteniendo información que permita generar un ataque exitoso.
- **Shodan:** permite encontrar los diferentes dispositivos conectados a internet y analizar las características de conexión por lo que permite descubrir todos los dispositivos MQTT y sus vulnerabilidades.
- **Ethercap:** conjunto de herramientas que permite realizar ataques de hombre en el medio, para interceptar comunicaciones y analizar el tráfico de esta sin ser detectado.

6.3.4. Metodología de análisis de seguridad

Se tienen en cuenta las fases de un ataque de penetración, pero enfocando las acciones sobre el protocolo MQTT

- **Reconocimiento**

Fase de descubrimiento de todos los dispositivos conectados a la red bajo la comunicación MQTT y determinar que componentes de la arquitectura MQTT son, bróker, publicador, suscriptor, y los servicios implementados en la red. [37]

En esta fase inicial, lo primero que se realiza es la identificación y análisis del bróker para poder identificar todos los dispositivos conectados que están realizando procesos de comunicación con el protocolo MQTT, con esto se obtendrán las direcciones IP de cada uno de los dispositivos conectados, para ello se puede hacer uso la herramienta nmap para poder realizar el escaneo de los dispositivos clientes, teniendo en cuenta el puerto de conexión y los servicios que están utilizando para poder descartar cualquier dispositivo que no esté usando el protocolo MQTT.

El siguiente paso es determinar cada una de las características de los dispositivos clientes conectados, apoyándose en herramientas como ethercap que permiten realizar ataques de envenenamiento ARP para que a través del análisis de tráfico se puede generar un ataque de hombre en el medio e identificar los mensajes MQTT, con esto se pueden identificar los servicios utilizados y las versiones correspondientes, para que con dicha información se proceda a la búsqueda de las vulnerabilidades que poseen estos dispositivos.

---

[37] DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MQTT SIN BRÓKER BASADO EN SDN. Arco, Alvaro (2019) Universidad de Granada.

Con ello nuevamente se realiza un escaneo de puertos y de sistemas operativos para encontrar posibles vulnerabilidades que puedan ser explotadas.

- **Planificación**

Luego del reconocimiento de la red se planean los diferentes ataques a realizar a partir de las vulnerabilidades identificadas en la fase inicial, y también determinar los exploits a utilizar. [38]

Teniendo en cuenta las características del protocolo MQTT, estudiadas en los capítulos anteriores, se determina que se deben evaluar las vulnerabilidades relacionadas con la autenticación ya que es la principal falla de seguridad del protocolo, teniendo en cuenta esto se determina realizar ataques de suplantación de identidad y control de acceso.

- **Ejecución**

Se ejecutan los ataques establecidos en la fase anterior y se documentaran los resultados obtenidos. [39]

Para realizar el ataque de suplantación de identidad se obtiene el identificador del cliente para publicar mensajes, el Id del cliente se obtiene con cualquiera de los métodos descritos anteriormente en las fases previas, una vez se obtiene esta información, se procede a realizar la conexión con el identificador obtenido para que

---

[38] Modelling and Evaluation of Malicious Attacks against the IoT MQTT Protocol. Syed Naeem, Firdous, y otros. 2017, IEEE International Conference on Internet of Things .

[39] Scanning for vulnerable devices in the internet of things . Markowsky, Linda y Markowsky, George. 2015, 8th IEE International conference of intelligent data acquisition anda advanced computing systems.

el dispositivo legítimo quede desconectado, y el dispositivo infiltrado pueda comenzar a publicar todos los mensajes que desee.

- **Mantenimiento**

Determinar las acciones para mitigar las vulnerabilidades encontradas y verificar que se hayan eliminado los riesgos, realizando nuevamente cada uno de los procesos descritos en las primeras fases luego de implementar las medidas de mitigación.

7. RESULTADOS

Realizando un análisis inicial la principal característica de las comunicaciones IoT que utilizan el protocolo MQTT es que no hay autenticación, por tanto, se puede realizar un ataque de suplantación de Identidad de manera sencilla, ya que solo se requiere el Id de cliente para publicar mensajes en su nombre. Para conseguir el Id de cliente se requiere escuchar el paquete connect con cualquier técnica mencionada en la fase de reconocimiento y después se realiza la conexión con ese Id de cliente. Se identifica alguno de los ID de clientes y se selecciona como víctima utilizando sus datos para realizar la conexión y este como dispositivo original pierde la conexión.

Una vez se realizada la suplantación se puede publicar todos los mensajes que se deseen; se analiza que se pierde la conexión y que los datos que se empiezan a publicar no son legítimos y que por tanto las variables de medición no corresponden al dispositivo original, generando la lista de datos ilegítima. Teniendo en cuenta que los parámetros de configuración y conexión de los dispositivos son muy similares se puede desconectar a usuarios legítimos suplantando su identidad y se podría llegar a realizar un ataque de denegación de servicio donde se desconecte a todos los clientes que se intenten conectar. Este tipo de ataque se puede modificar dependiendo del objetivo que se desee conseguir.

Se puede realizar otro ataque de denegación de servicio conectando muchos clientes para consumir los recursos del broker y evitar que puedan entrar conexiones nuevas legítimas.

Otra posible falla de seguridad es la suplantación de identidad del broker para ello se puede realizar un ataque de MitM y en conjunto con una redirección de tráfico con Iptables, logrando redirigir el tráfico generado a un puerto diferente, donde se encuentra el bróker ilegítimo.

Teniendo en cuenta los procedimientos realizados se evidencia que la configuración inicial presenta muchas carencias de seguridad, ya que hay muchas vulnerabilidades que pueden ser explotadas con facilidad, pero la vulnerabilidad de suplantación de identidad siempre está presente.

Sobre la arquitectura de pruebas, se pueden realizar las pruebas con las herramientas de monitorización definidas en la fase de reconocimiento, bastará con desconectar el subscriptor de para provocar que no haya datos generados.

CONCLUSIONES

Como se vio en el capítulo 6.2, el protocolo MQTT, posee varias vulnerabilidades que los atacantes pueden aprovechar; por ejemplo, utilizando el motor de búsqueda Shodan se puede evidenciar que la mayoría de los servidores que contienen el MQTT Broker no poseen mecanismo de autenticación lo que puede ser aprovechado para atentar contra la confidencialidad e integridad de la información.

Se estableció a partir de la revisión de la literatura que a pesar de que el protocolo MQTT ya es un protocolo estandarizado para comunicaciones IoT no posee técnicas y procesos de seguridad ya establecidos que permitan mitigar las vulnerabilidades que se generan en las redes de comunicaciones implementadas bajo este protocolo, se hace necesario estandarizar también los métodos de seguridad que permitan disminuir los riesgos de ataques.

Existen diferentes técnicas para la mitigación de las diferentes vulnerabilidades del protocolo MQTT sin afectar la optimización de los recursos de los dispositivos que es una de las principales razones por la que no se emplean métodos de seguridad, por eso es necesario estudiar el tipo de red, tráfico y dispositivos, antes de seleccionar algún método de seguridad.

RECOMENDACIONES

La seguridad es importante para todo tipo de dispositivos, sin importar el tipo de tráfico e información que maneje; si bien los dispositivos IoT que trabajan el protocolo MQTT no manejan gran volumen de información contienen también datos sensibles que al ser modificados puede afectar el funcionamiento de diferentes infraestructuras o servicios, por eso es importante tener en cuenta la seguridad de los mismos a la hora de diseñar e implementar una red de comunicaciones de IoT que trabaje bajo este protocolo ya que en sí posee varias fallas de seguridad.

Es necesario tener en cuenta el rendimiento y optimización de los recursos a la hora de seleccionar los métodos de seguridad a implementar en la red, porque se debe mantener un equilibrio entre ambos aspectos optimizar el rendimiento de la red y garantizar una disminución de las vulnerabilidades a las que está expuesto el uso del protocolo.

Establecer un plan de respuesta a incidentes con este tipo de dispositivos y redes que trabajan bajo este protocolo ya que como se puede evidenciar el ataque a estos dispositivos puede generar consecuencias graves en el funcionamiento de la red y la integridad de la información.

BIBLIOGRAFÍA

ARCO, Álvaro. Diseño e implementación de un sistema MQTT sin bróker basado en SDN. {En línea} 2019 {Marzo 19 de 2021} Disponible en: https://wpd.ugr.es/~jorgenavarro/thesis/2019_TFG_AlvaroArcoCastillo.pdf

CHENG, Xiaochun - DINCULEANA, Dan. Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices. {En línea} 2019 {Marzo 17 de 2021} Disponible en: https://www.researchgate.net/publication/331392480_Vulnerabilities_and_Limitations_of_MQTT_Protocol_Used_between_IoT_Devices

GÓMEZ, Rubén. Diseño e implementación de una red de sensores basada en protocolos IoT para monitorización de mercancías. {En línea} 2020 {Octubre 12 de 2021} Disponible en: https://repositorio.uam.es/bitstream/handle/10486/690544/gomez_moreno_ruben_tfm.pdf?sequence=1

GONZALEZ, Carlos - FLAUZAC, Olivier - NOLOT, Florent. Evolution and Contribution for the Internet of Things by the Emerging Software - defined networking. {En línea} 2016. {Octubre 12 de 2021} II Congreso Internacional en Inteligencia Ambiental, Ingeniería de Software y Salud Electrónica y Móvil. Disponible en: https://revistas.utp.ac.pa/index.php/memoutp/article/view/1842/2688

HARSHA, Ms - BHAVANI, Bm – Khundavai, Krk. Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs. {En línea}. 2018 {Julio 22 de 2021} Disponible en: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8554472

HERNÁNDEZ, Santiago - VILLALBA, Teresa - LACUESTA, Raquel. MQTT Security: A Novel Fuzzing Approach. {En línea} 2018 {Octubre 12 de 2021} Disponible en: https://www.researchgate.net/publication/322820969_MQTT_security_A_novel_fuzzing_approach

HIDALGO, Edward. Análisis comparativo de tráfico entre la red wifi y la red iot-wifi en el campus sur de la universidad politécnica salesiana. {En línea} 2019 {Julio 22 de 2021} Disponible en: https://dspace.ups.edu.ec/bitstream/123456789/16811/1/UPS-ST003902.pdf

IBRAHIM, Ahmed – VALLI, Craig – BAIG, Zubair. Modelling and Evaluation of Malicious Attacks against the IoT MQTT Protocol. {En línea} 2017 {Octubre 17 de 2021} IEEE International Conference on Internet of Things. Disponible en: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8276834

ITU. Asamblea mundial de normalización de las telecomunicaciones AMNT-16. Internet de las cosas y sus aplicaciones, incluidas las ciudades y comunidades inteligentes. {En línea} 2016 {Marzo 17 de 2021} Disponible en: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjh5ZTdwL30AhVWTTABHcSjAWsQFnoECBUQAQ&url=https%3A%2F%2Fwww.itu.int%2Fdms_pub%2Fitu-t%2Fmd%2F13%2Fwtsa.16%2Fc%2FT13-WTSA.16-C-0022!!MSW-S.docx&usg=AOvVaw2ODLgwLl5ZEIv0nUw8XRwg

LAVINIA, Nastase. Security in the Internet of Things: A Survey on Application Layer Protocols. {En línea} 2017 {Marzo 17 de 2021} 21st International Conference on Control Systems and Computer Science. Disponible en: https://ieeexplore.ieee.org/document/7968629

MARKOWSKY, Linda - MARKOWSKY, George. Scanning for vulnerable devices in the internet of things. {En línea} 2015 {Julio 17 de 2021} 8th IEEE International conference of intelligent data acquisition anda advanced computing systems. Disponible en: https://ieeexplore.ieee.org/abstract/document/7340779

MATHEWS, Suja - GONDKAR, Raju. Protocol Recommendation for Message Encryption in MQTT. {En línea} 2017 {Marzo 17 de 2021} Disponible en: https://ieeexplore.ieee.org/document/8817043

MENDEZ, Diego – PAPAPANAGIOTOU, Loannis - YANG, Baijian. Internet of Things: Survey on Security and Privacy. {En línea} 2017 {Marzo 17 de 2021} Disponible en: https://www.researchgate.net/publication/318259049_Internet_of_Things_Survey_on_Security_and_Privacy

MOHAN, Kimar. A Forensic Analysis on the Availability of MQTT Network Traffic. {En línea}. 2021. {Octubre 19 de 2021} Disponible en: https://www.researchgate.net/publication/349187990_A_Forensic_Analysis_on_the_Availability_of_MQTT_Network_Traffic/citations

MORETON, Hector - AVELEIRA, Jose - CASTAÑEDA, Angel. Multiclass Classification Procedure for Detecting Attacks on MQTT-IoT Protocol. {En línea} 2019 {Marzo 17 de 2021} Disponible en: https://www.researchgate.net/publication/332261994_Multiclass_Classification_Procedure_for_Detecting_Attacks_on_MQTT-IoT_Protocol

PACHAR, Ever. Desarrollo y evaluación de un gateway móvil IoT para redes 4G LTE. {En línea} 2020 {Marzo 17 de 2021} Disponible en: http://oaji.net/articles/2020/1783-1601569305.pdf

PALMIERI, Andrea. MQTTSA: A Tool for Automatically Assisting the Secure Deployments of MQTT brokers. {En línea} 2019 {Julio 22 de 2021} IEEE World Congress on Services. Disponible en: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8817102

POTRINO, Giuseppe - DE RANGO, Floriano - SANTAMARIA, Amilcare. Modeling and evaluation of a new IoT security system for mitigating DoS attacks to the MQTT broker. {En línea} 2019 {Septiembre 12 de 2021} IEEE Wireless Communications and Networking Conference (WCNC). Disponible en: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8885553

RANCHAL, Juan. Plataforma unificada para IoT Intel. {En línea} 2014 {Marzo 17 de 2021} Internet of Things My Computer. Disponible en: https://www.muycomputer.com/2014/12/10/internet-de-las-cosas-intel/amp/

RUIZ, Diana. Diseño de un sistema en cloud para controlar dispositivos IoT vía internet. {En línea} 2016 {Marzo 17 de 2021} Disponible en: https://repository.unimilitar.edu.co/bitstream/handle/10654/15343/RuizMartinezDianaMarcela2016.pdf?sequence=1&isAllowed=y

SALAZAR, Jordi, SILVESTRE, Santiago. Internet de las Cosas. {En línea} 2016 {Octubre 12 de 2021} European Virtual Learning Platform for Electrical and Information Engineering. Disponible en: https://www.academia.edu/download/64747844/1.6_industria_4.pdf

SEGARRA, Carlos – DELGADO, Ricard – SCHIAVONI, Valerio. MQT-TZ: Secure MQTT Broker for Biomedical Signal Processing on the Edge. {En línea} 2020 {Septiembre 17 de 2021} Disponible en: https://www.researchgate.net/publication/342706855_MQT-

TZ_Secure_MQTT_Broker_for_Biomedical_Signal_Processing_on_the_Edge

SOMAYYA, Madakam – SIDDHARTH, Tripathi. Internet of Things (IoT): A Literature Review. {En línea} 2015 {Marzo 17 de 2021} Journal of Computer and Communications. Disponible en: https://www.scirp.org/html/56616_56616.htm?pagespeed=noscript

SWAPNIL, Naik - VIKAS, Maral. Cyber Security - IoT. 2017 {Marzo 13 de 2021} Disponible en: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8256700

SYAIFUL, Andy - BUDI, Rahardjo - BAGUS, Hanindhito. Attack Scenarios and Security Analysis of MQTT Communication Protocol in IoT System. {En línea} 2017 {Julio 22 de 2021} Disponible en: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8239179

VELEZ, Andrés. Arquitecturas de referencia para iot con transferencia segura de información. {En línea} (2019). {Julio 22 de 2021} Disponible en: https://repository.unad.edu.co/bitstream/handle/10596/27648/avelezpe.pdf?sequence=4

ANEXOS

ANEXO A – CONFIGURACIÓN DE ARQUITECTURA DE PRUEBAS

En la arquitectura de pruebas los dispositivos se encontrarán dentro de una categoría llamada Dispositivos, donde se encontrará toda la información relacionada con la publicación de datos, y el momento en que debe iniciar dicha publicación.

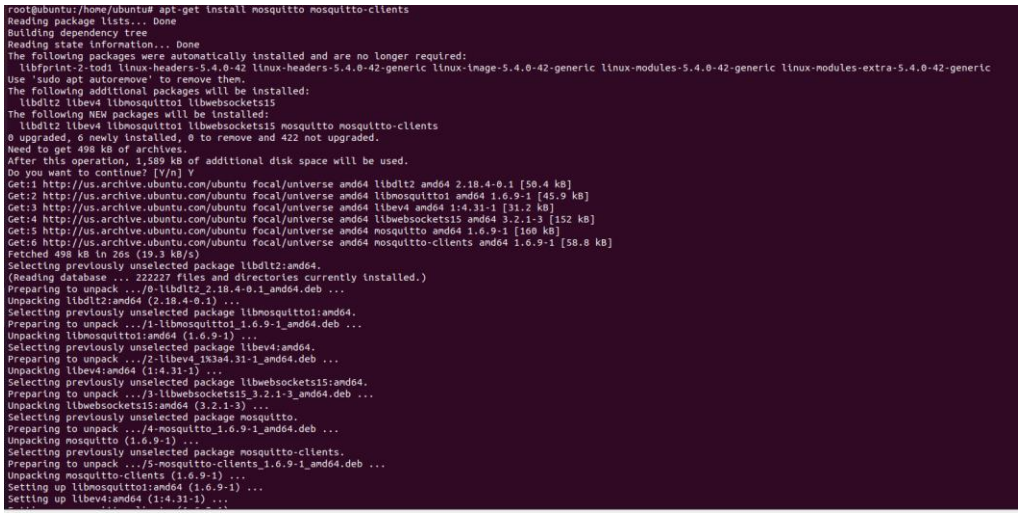
Cada dispositivo debe poseer las variables necesarias según el tipo y cantidad de datos a publicar, los valores de estos datos serán enviados mediante el protocolo MQTT, de esta manera se podrá agregar dispositivos nuevos que puedan publicar datos de manera sencilla y rápida.

Una vez establecidas las categorías de los dispositivos, se debe determinar el intervalo de tiempo en que publicarán los datos.

Configuración broker

La instalación de mosquitto consume pocos recursos, se realiza sobre Linux en distribución Ubuntu, de esta manera se pueden editar los archivos de configuración mediante los diferentes comandos propios para la configuración de mosquitto pero que también se aplican a Linux, la descarga de la herramienta se realiza mediante el comando: *apt-get install mosquitto mosquitto-clients* como se muestra en la figura 13.

Figura 13 instalación cliente mosquitto



Fuente: autor

- Edición del archivo de configuración de mosquitto

En este archivo de configuración se establece diferentes características como: el puerto para el bróker y el intervalo de tiempo en que se enviará la información de dispositivos conectados y cuantos mensajes han publicado por minutos, la cantidad de bytes que comprenden dichos datos y hasta la versión del bróker. La configuración del fichero es la siguiente:

autosave_interval 60

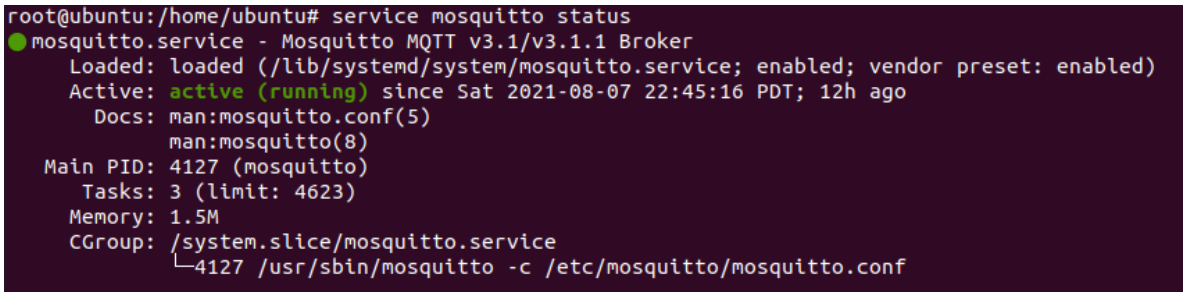
persistence true persistence_file mosquitto.db

persistence_location / var / lib / mosquitto

- Inicio del servidor mosquitto

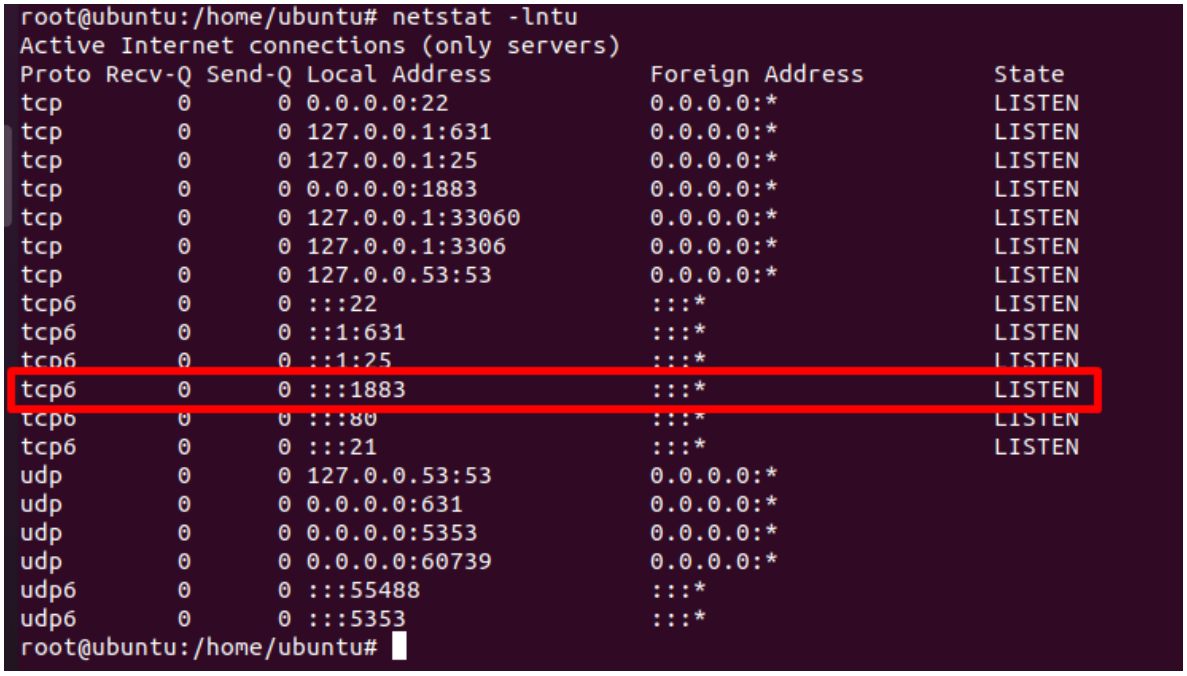
Una vez configurado el servicio de mosquitto se procede a activar el servicio y a verificar su estado y el estado del puerto, ambos activos como se muestra en las figuras 14 y 15.

Figura 14 Verificación activación mosquitto



Fuente: autor

Figura 15 Verificación estado mosquitto



Fuente: autor

5.1.2. **Configuración publicadores y suscriptores**

Una vez configurado el bróker se procede con la configuración de los dispositivos, estableciendo las diferentes variables y tipos de datos a publicar, para la publicación de datos se configura teniendo en cuenta el siguiente script:

- Agente de publicación, script de configuración

```
git clone https://github.com/obgm/libcoap.git /
cd libcoap /
./autogen.sh /
./configure --disable-documentation /Config
make
sudo make install
import paho.mqtt.client
import sys
def main():
client = paho.mqtt.client.Client('89172934567')
client.connect(host='127.0.0.1', port=1883)
client.publish("home/sensor", "{temp: 60}")
if __name__=='__main__':
main()
sys.exit()
```

Para la suscripción de datos se aplicará el siguiente script:

- Agente de suscripción, script de configuración

```
import paho.mqtt.client
import sys
def on_connect(client, userdata, flags, rc):
print('connected '+str(client._client_id))
client.subscribe(topic='Home/sensor')
def on_message(client, userdata, message):
print('message '+str(message.payload))
def main():
client = paho.mqtt.client.Client(client_id='836136274')
client.on_connect = on_connect
```

```
client.on_message = on_message
client.connect(host='127.0.0.1', port=1883)
client.loop_forever()
```

También se hace necesario la configuración de la base de datos de influx DB que tendrá un puente con el bróker de mosquitto, para que todos los mensajes que lleguen de los diferentes dispositivos conectados sean almacenados, el puente básicamente será un nuevo suscriptor que podrá observar todos los mensajes que se publican y luego almacenarlos, para la configuración se tendrá en cuenta el siguiente script:

- Configuración influx DB

```
influx
CREATE DATABASE sensor
CREATE USER prueba WITH PASSWORD 'mqtt'
GRANT ALL ON sensor  TO MQTT
```

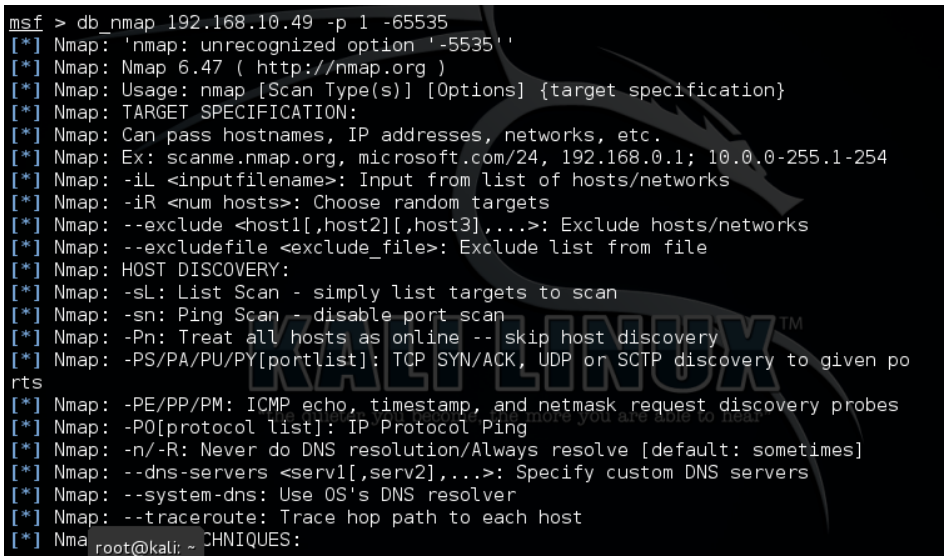
Anexo B- Evaluación arquitectura de pruebas MQTT

Se desarrollan las diferentes fases que hacen parte de los ataques de penetración, pero se enfocan solo en el protocolo MQTT

- Escaneo de puertos

Con el uso de la herramienta nmap se pretende conocer cuales son los dispositivos conectados mediante el protocolo MQTT, para ello se realiza un escaneo de puertos para identificar si hay servicios MQTT que se estén ejecutando, con esta herramienta podemos definir la versión del servicio y así encontrar sus diferentes vulnerabilidades. La información obtenida del escaneo de puertos se muestra en la figura 16.

Figura 16 Resultado escaneo de puertos



Fuente: autor

- Búsqueda de dispositivos públicos con shodan

Con las diferentes herramientas de penetración es posible identificar dispositivos conectados que utilicen el protocolo MQTT dentro de una red local, pero hay herramientas que permiten descubrir estos dispositivos que tienen acceso público,

se puede hacer con el uso de la herramienta Shodan, aplicando el filtro MQTT, como lo muestra la figura 17.

Figura 17 Resultado dispositivos públicos MQTT

Fuente: www.shodan.io

ANEXO C	
RESUMEN ANALÍTICO ESPECIALIZADO - RAE	
Fecha de realización	8 de marzo de 2022
Programa	Especialización en seguridad informática
Línea de investigación	Gestión de Sistemas
Título	ANÁLISIS DE LA SEGURIDAD DEL PROTOCOLO DE TRANSPORTE MQTT EN DISPOSITIVOS PARA INTERNET DE LAS COSAS
Autor(es)	Flor Salazar Paulita
Palabras Claves	MQTT, Internet de las cosas, seguridad de la información, protocolos de comunicación, falla de seguridad, estándar.
Descripción	Internet de las cosas ha crecido rápidamente y su implementación dentro de todos los campos es una realidad al igual que las amenazas de seguridad por lo que implica tener conectado a internet los dispositivos cotidianos que utiliza la sociedad en general por lo que se ha hecho necesario establecer normatividad y estandarizar la implementación de esta tecnología en todos sus aspectos como los protocolos de comunicación, es así que la organización internacional de estándares, ISO ha estandarizado el protocolo de transporte de mensajes MQTT (Message Queue Telemetry Transport) como protocolo aplicable a los entornos donde se realiza comunicación de dispositivos de Internet de las cosas para contribuir a las buenas prácticas de seguridad se presenta este estudio que pretende determinar las vulnerabilidades de la implementación del protocolo MQTT para obtener mecanismos y herramientas de mitigación de amenazas mediante la identificación de las amenazas y evaluación de los mecanismos de mitigación del riesgo. Esto permite determinar los mejores campos de aplicación del protocolo MQTT teniendo en cuenta en que tipo de comunicación se hace más o menos vulnerable determinando las mejores prácticas de seguridad con un protocolo de transporte estandarizado para el uso de comunicaciones en internet de las cosas.

Fuentes Bibliográficas Destacadas	ARCO, Álvaro. Diseño e implementación de un sistema MQTT sin bróker basado en SDN. {En línea} 2019 {Marzo 19 de 2021} Disponible en: https://wpd.ugr.es/~jorgenavarro/thesis/2019_TFG_AlvaroArcoCastillo.pdf CHENG, Xiaochun - DINCULEANA, Dan. Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices. {En línea} 2019 {Marzo 17 de 2021} Disponible en: https://www.researchgate.net/publication/331392480_Vulnerabilities_and_Limitations_of_MQTT_Protocol_Used_between_IoT_Devices GÓMEZ, Rubén. Diseño e implementación de una red de sensores basada en protocolos IoT para monitorización de mercancías. {En línea} 2020 {Octubre 12 de 2021} Disponible en: https://repositorio.uam.es/bitstream/handle/10486/690544/gomez_moreno_ruben_tfm.pdf?sequence=1 GONZALEZ, Carlos - FLAUZAC, Olivier - NOLOT, Florent. Evolution and Contribution for the Internet of Things by the Emerging Software - defined networking. {En línea} 2016. {Octubre 12 de 2021} II Congreso Internacional en Inteligencia Ambiental, Ingeniería de Software y Salud Electrónica y Móvil. Disponible en: https://revistas.utp.ac.pa/index.php/memoutp/article/view/1842/2688 HARSHA, Ms - BHAVANI, Bm – Khundavai, Krk. Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs. {En línea}. 2018 {Julio 22 de 2021} Disponible en: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8554472 HERNÁNDEZ, Santiago - VILLALBA, Teresa - LACUESTA, Raquel. MQTT Security: A Novel Fuzzing Approach. {En línea} 2018 {Octubre 12 de 2021} Disponible en: https://www.researchgate.net/publication/322820969_MQTT_security_A_novel_fuzzing_approach HIDALGO, Edward. Análisis comparativo de tráfico entre la red wifi y la red iot-wifi en el campus sur de la universidad politécnica salesiana. {En línea} 2019 {Julio 22 de 2021} Disponible en:

https://dspace.ups.edu.ec/bitstream/123456789/16811/1/UPS-ST003902.pdf

IBRAHIM, Ahmed – VALLI, Craig – BAIG, Zubair. Modelling and Evaluation of Malicious Attacks against the IoT MQTT Protocol. {En línea} 2017 {Octubre 17 de 2021} IEEE International Conference on Internet of Things. Disponible en: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8276834

MENDEZ, Diego – PAPAPANAGIOTOU, Loannis - YANG, Baijian. Internet of Things: Survey on Security and Privacy. {En línea} 2017 {Marzo 17 de 2021} Disponible en: https://www.researchgate.net/publication/318259049_Internet_of_Things_Survey_on_Security_and_Privacy

PACHAR, Ever. Desarrollo y evaluación de un gateway móvil IoT para redes 4G LTE. {En línea} 2020 {Marzo 17 de 2021} Disponible en: http://oaji.net/articles/2020/1783-1601569305.pdf

POTRINO, Giuseppe - DE RANGO, Floriano - SANTAMARIA, Amilcare. Modeling and evaluation of a new IoT security system for mitigating DoS attacks to the MQTT broker. {En línea} 2019 {Septiembre 12 de 2021} IEEE Wireless Communications and Networking Conference (WCNC). Disponible en: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8885553

RANCHAL, Juan. Plataforma unificada para IoT Intel. {En línea} 2014 {Marzo 17 de 2021} Internet of Things My Computer. Disponible en: https://www.muycomputer.com/2014/12/10/internet-de-las-cosas-intel/amp/

RUIZ, Diana. Diseño de un sistema en cloud para controlar dispositivos IoT vía internet. {En línea} 2016 {Marzo 17 de 2021} Disponible en: https://repository.unimilitar.edu.co/bitstream/handle/10654/15343/RuizMartinezDianaMarcela2016.pdf?sequence=1&isAllowed=y

Contenido del documento	GLOSARIO RESUMEN ABSTRACT INTRODUCCIÓN 1. DEFINICIÓN DEL PROBLEMA 1.1. ANTECEDENTES DEL PROBLEMA 1.2. FORMULACIÓN DEL PROBLEMA 2. JUSTIFICACIÓN 3. OBJETIVOS 3.1. OBJETIVO GENERAL 3.2. OBJETIVOS ESPECÍFICOS 4. MARCO REFERENCIAL 4.1. MARCO CONCEPTUAL 4.2. MARCO TEÓRICO 5. DISEÑO METODOLÓGICO 6. DESARROLLO DE LOS OBJETIVOS 7. CONCLUSIONES 8. RECOMENDACIONES 9. BIBLIOGRAFIA 10. ANEXOS
Diseño metodológico	Metodología de tipo exploratorio
Conceptos adquiridos	Funcionamiento del protocolo MQTT, aplicaciones del protocolo MQTT, Vulnerabilidades del protocolo MQTT, evaluación de nivel de seguridad.
Conclusiones	Como se vio en el capítulo 6.2, el protocolo MQTT, posee varias vulnerabilidades que los atacantes pueden aprovechar; por ejemplo, utilizando el motor de búsqueda Shodan se puede evidenciar que la mayoría de los servidores que contienen el MQTT Broker no poseen mecanismo de autenticación lo que puede ser aprovechado para atentar contra la confidencialidad e integridad de la información. Se estableció a partir de la revisión de la literatura que a pesar de que el protocolo MQTT ya es un protocolo estandarizado para comunicaciones IoT no posee técnicas y procesos de seguridad ya establecidos que permitan mitigar las vulnerabilidades que se generan en las redes de comunicaciones implementadas bajo este protocolo, se hace necesario estandarizar también los métodos de seguridad que permitan disminuir los riesgos de ataques. Existen diferentes técnicas para la mitigación de las diferentes

	vulnerabilidades del protocolo MQTT sin afectar la optimización de los recursos de los dispositivos que es una de las principales razones por la que no se emplean métodos de seguridad, por eso es necesario estudiar el tipo de red, tráfico y dispositivos, antes de seleccionar algún método de seguridad.