DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERRA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS E INFORMÁTICA

TEMA: FRAMEWORK DE SEGURIDAD PARA DISPOSITIVOS IOT, BASADO EN EL PROTOCOLO OAUTH

AUTOR: VALDIVIESO ARIAS, EDISON DAVID

DIRECTOR: ING. GUALOTUÑA ÁLVAREZ, TATIANA MARISOL

SANGOLQUÍ

2019

CERTIFICADO DEL DIRECTOR

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

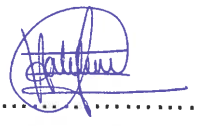
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERRA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, "***FRAMEWORK DE SEGURIDAD PARA DISPOSITIVOS IOT, BASADO EN EL PROTOCOLO OAUTH***" fue realizado por el señor ***Valdivieso Arias, Edison David*** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 10 de julio del 2019

Firma



.....

Tatiana Marisol Gualotuña Álvarez

C. C. 1711498418

AUTORÍA DE RESPONSABILIDAD

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERRA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Yo, ***Valdivieso Arias, Edison David***, declaro que el contenido, ideas y criterios del trabajo de titulación***: Framework de seguridad para dispositivos IoT, basado en el protocolo OAuth*** es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 10 de julio del 2019

Firma

.....

Edison David Valdivieso Arias

C.C.: 1720308194

AUTORIZACIÓN

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERRA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, ***Valdivieso Arias, Edison David*** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: ***Framework de seguridad para dispositivos IoT, basado en el protocolo OAuth*** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 10 de julio del 2019

Firma

.....

Edison David Valdivieso Arias

C.C.:1720308194

DEDICATORIA

El presente trabajo de titulación es dedicado principalmente a Dios y a mis hijos, quienes se convirtieron en mi fuente de energía para no desmayar en este proceso, a mi padrastro Giancarlo Mingozzi, a mi madre Mónica Arias, a mi abuela Dolores Morales que con amor, esfuerzo y paciencia me apoyaron en todo momento de mi etapa universitaria, permitiéndome alcanzar este tan deseado logro.

AGRADECIMIENTO

A todos los ingenieros de la Universidad de las Fuerzas Armadas ESPE, quienes a través de sus enseñanzas me permitieron adquirir grandiosos y valiosos conocimientos, en especial a la Ing. Tatiana Gualotuña mi directora que con su entrega y dedicación hizo posible la culminación de este trabajo de titulación.

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

En los últimos años la implementación del Internet of Things se ha ido incrementando considerablemente, logrando evidenciar ecosistemas IoT en diferentes ámbitos como hogares, compañías, salud, educación, entre otros. Al ser una tecnología reciente las limitaciones de procesamiento, memoria, potencia y la falta de prioridad en la seguridad de algunos dispositivos IoT por parte de los fabricantes, ha dado lugar a una brecha en la seguridad de estos, lo que ha ocasionado problemas a los usuarios y compañías que han decidido utilizar esta tecnología. Una de las principales ventanas para el robo de la información se da en el proceso de autenticación del usuario en las diferentes aplicaciones, ya que este proporciona sus credenciales en internet sin generarse una seguridad adecuada, lo que los vuelve sujetos vulnerables dentro de esta gigantesca red. La propuesta de una metodología de investigación Ad hoc, que como fase inicial contempla tareas de revisiones de literatura, ha permitido identificar y analizar estudios relacionados con la autenticación de ecosistemas IoT, dando apertura a la implementación de un framework de seguridad basado en el protocolo OAuth 2.0. El desarrollo del framework mitigará las vulnerabilidades causadas por ataques como: ataque de fuerza bruta, ataque de suplantación o ataque de hombre en el medio, garantizando de esta manera la seguridad en la autenticación en un ecosistema IoT.

PALABRAS CLAVE:

- **INTERNET OF THINGS**
- **AUTENTICACIÓN**
- **OAUTH 2.0**

ABSTRACT

In recent years, the implementation of the Internet of Things has been increasing, making it possible to demonstrate IoT systems in different areas such as homes, companies, health, education, and others. As a recent technology, the limitations of processing, memory, power and the lack of priority in the safety of some IoT devices by manufacturers has led to a breach in the safety of these devices, which has caused problems for users and companies that have decided to use this technology. One of the main windows for information theft occurs in the process of user authentication in different applications, as this provides their credentials on the Internet without generating adequate security, which makes them vulnerable subjects within this gigantic network. The proposal of an Ad hoc research methodology, that as the initial phase contemplating the literature review tasks, has allowed to identify and analyze the studies related to the authentication of the IoT ecosystems, giving rise to the implementation of a security framework based on the OAuth 2.0 protocol. The development of the framework will mitigate vulnerabilities caused by attacks such as brute force attack, impersonation attack or man attack in the middle, thus guaranteeing authentication security in an IoT ecosystem.

KEY WORDS:

- **INTERNET OF THINGS**
- **AUTHENTICATION**
- **OAUTH 2.0**

CAPÍTULO I

INTRODUCCIÓN

1.1. Antecedentes

Una definición relevante sobre la seguridad de la información es la que proporciona la ISO 27001, donde dice que “el termino CIA (Confidencialidad, Integridad, Disponibilidad), que presenta los principios básicos de la seguridad de la información. Realizar una correcta gestión de la seguridad de la información establece como principio básico que sin los tres elementos mencionados no existe nada seguro, con que solo falle uno de los componentes nos encontramos ante un peligro para la seguridad de la información. Tenemos que recordar que ningún sistema de seguridad es completamente seguro, siempre debemos tener claro que un sistema es mucho más vulnerable de lo que pensamos” (ISO 27001:2013, 2017).

La Internet es fundamentalmente un lugar inseguro. Para cada servicio, cada API, hay usuarios que no amarían más que romper las distintas capas de seguridad que ha erigido. Las consideraciones con respecto a la seguridad han sido por el momento para los servicios web modernos y las API; rara vez, si alguna, se habla sobre la próxima ola de pequeños dispositivos IoT conectados y no conectados que pronto harán de esto una preocupación aún mayor. Desde la nevera conectada al smartwatch, IoT abarca muchos nuevos dispositivos habilitados para la web que llegan al mercado. Mientras estamos diseñando nuevas infraestructuras API, Jacob Ideskog cree que “la IoT nos va a afectar mucho si no hacemos nada al respecto” (Sandoval, 2017).

De acuerdo con la definición de Albors “La inseguridad del Internet de las Cosas es algo que, por desgracia, está a la orden del día. Con mayor frecuencia vemos que se producen ataques que afectan a dispositivos IoT o se detectan fallas que las compañías tecnológicas intentan rápidamente solucionar con un parche. Y es que centrados en incorporar nuevas funcionalidades y hacer sus dispositivos más fáciles de usar y conectables, los fabricantes han descuidado casi por completo un apartado crucial como es la seguridad de sus dispositivos. Los usuarios son conscientes de los problemas de seguridad que afectan a los dispositivos IoT. Y así lo demuestra una encuesta realizada por ESET donde el 70% de los participantes consideró que este tipo de dispositivos no son seguros, fundamentalmente en términos de privacidad; que es donde radica la principal preocupación. Sin embargo, el 62% consideró que no dejaría de comprar este tipo de tecnología por esta razón” (Albors, 2018).

A medida que las implementaciones de Internet de las cosas (IoT) aumentan constantemente, aumenta la necesidad de una mejor experiencia de usuario para manejar las tareas de autenticación y autorización en entornos restringidos. Si bien ya se han desarrollado varias tecnologías que permiten el acceso a recursos protegidos, la naturaleza de las implementaciones de IoT requiere atención con los recursos limitados disponibles en muchos de estos dispositivos (Tschofenig, 2016).

En los últimos años, el estándar de autorización abierta OAuth2 se ha hecho popular en el uso sobre servicios web y APIs. Compañías como Google, Facebook, Microsoft y Twitter permiten a sus usuarios delegar de forma segura el acceso de los clientes a los recursos del servidor sin compartir sus credenciales de usuario con esos clientes. En

parte debido a este éxito, Oauth2 se está adaptando para ayudar a garantizar el acceso a los dispositivos de IoT (Hovsmith, 2017).

1.2. Planteamiento del problema

El paso de los años, el avance tecnológico y desde la llegada Cloud Computing ha permitido que el desarrollo de dispositivos IoT sea una realidad, ha comenzado a formar parte de la sociedad, en la actualidad puede ser utilizado en la vida cotidiana, actualmente lo encontramos en hogares, educación, salud, vehículos y en varias otras importantes industrias. Cada vez es más frecuente que las organizaciones se vean en la necesidad del uso de estos dispositivos inteligentes, sin embargo, la innovación implica el incremento de ataque a las empresas, la transmisión de datos vía internet abre una brecha de inseguridad de la información, el uso de estos dispositivos permite que el usuario se encuentre expuesto al robo de información por la falta de seguridad en la autenticación (Rivas, 2018).

La mayoría de las aplicaciones utilizadas por los dispositivos IoT, solicitan al usuario como paso inicial él envió de sus credenciales y su autenticación para acceder al sistema, convirtiéndose este paso en el primer problema de inseguridad por las diversas amenazas que existen en la Internet. En caso de no contar con las debidas seguridades será inevitable que la información sea robada.

1.3. Justificación

El constante incremento de ecosistemas IoT, hace que la seguridad de los dispositivos que los conforman se muestren vulnerables, un ejemplo sencillo y claro de comprender sobre la vulnerabilidad que presenta la información, es el ataque que sufrió

Uber, el cual debió pagar a piratas informáticos $100.000 para recuperar 57 millones de datos, de conductores y usuarios (York, 2017).

El autor Núñez hace referencia a que “según la firma de investigación Gartner dice que los dispositivos de IoT han aumentado un 31% de 2016 a 2017, alcanzando 8.400 millones de “cosas” conectadas en el 2017 y que el número aumentará a 20.400 millones para 2020” (Núñez, 2017).

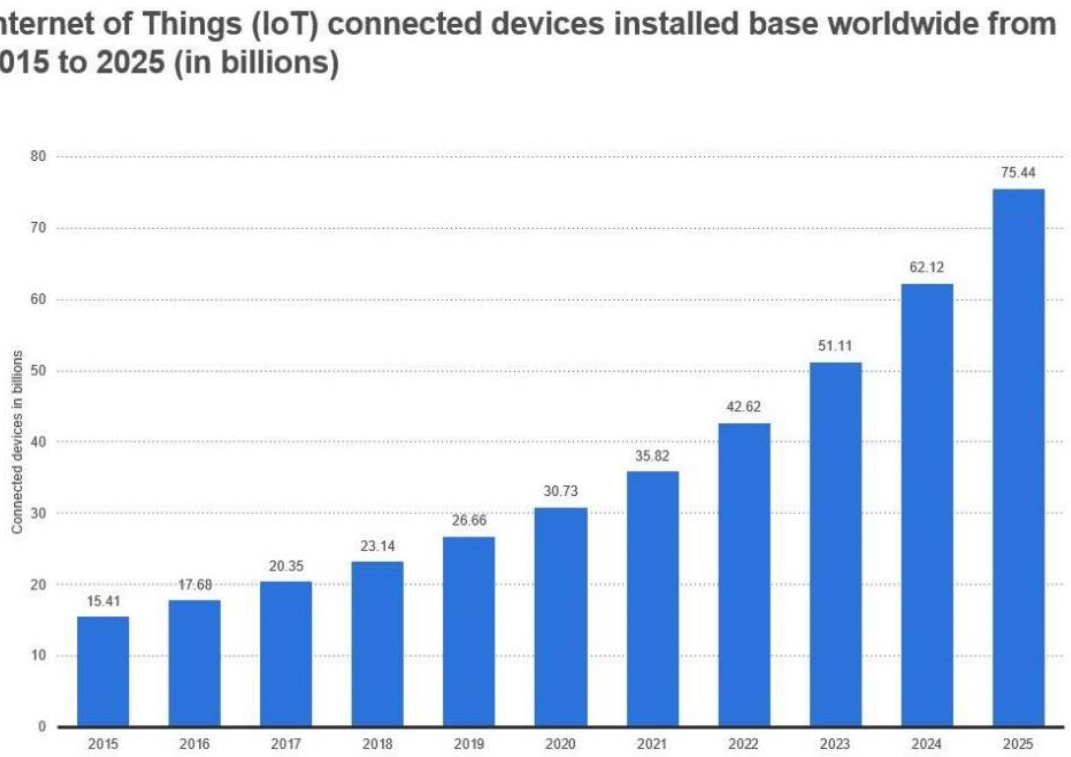


Figura 1. Dispositivos IoT conectados para el año 2025
Fuente: (IoT World Online, 2018)

Solo en EEUU, las brechas de seguridad cuestan a las empresas más de $ 445 mil millones de dólares anuales. A medida que la Internet de las cosas (IoT) crece, este número solo aumentará (Sandoval, 2017).

1.4. Objetivos

1.4.1. Objetivo General

Implementar un framework de seguridad basado en protocolo OAuth 2.0, que permita garantizar la autenticación en un ecosistema IoT.

1.4.1. Objetivos Específicos

- Identificar las posibles soluciones de autenticación existentes y modelos de seguridad para dispositivos IoT mediante una revisión de literatura preliminar.
- Desarrollar un framework de seguridad para la autenticación de dispositivos IoT, utilizando el modelo Proof of Possession y el protocolo OAuth 2.0.
- Validar el framework de seguridad implementado mediante pruebas de concepto, donde se evidencie la seguridad de autenticación en un ecosistema IoT.

1.5. Hipótesis de trabajo

La implementación de un framework de seguridad basado en protocolo OAuth 2.0, garantizará la autenticación en un ecosistema IoT.

Variable independiente:

- Framework de seguridad basado en protocolo OAuth 2.0

Variable dependiente:

- Autenticación en un ecosistema IoT

CAPÍTULO II

MARCO TEÓRICO

2.1. Internet of Things (IoT)

Figura 2. Internet of Things
Fuente: (Peña, 2016)

Con la evolución tecnológica y el uso del internet siendo una necesidad, ha dado lugar al denominado Internet de las Cosas (IoT) que representa la interconexión de objetos cotidianos como teléfonos, lavadora, y otros, que se conectan entre sí a través de una Red, convirtiéndose así en objetos inteligentes que permiten enviar y recibir datos en tiempo real, uniendo tanto el mundo digital con el físico. Para la interconexión de objetos entre sí a través de una red se debe tomar en cuenta tres factores, los dispositivos, la red y el sistema de control (Valois, 2018).

En la actualidad es posible que casi cualquier cosa forme parte del IoT, sin embargo los beneficios no llegan sin riesgos, porque en gran cantidad los dispositivos IoT no tienen instalado un software que les brinde seguridad y el mismo hecho de estar conectados a una red mediante la cual se proporciona todo tipo de información personal, como por ejemplo contraseñas, vuelve vulnerables a los usuarios a ataques existentes en el Red (Peña, 2016).

2.1.1. Arquitectura de referencia

Existen varias arquitecturas que brindan servicios para el manejo de dispositivos IoT, una de ellas es la de CISCO, la cual permite conectar las cosas de forma segura, analizar datos y entregar los resultados a compañías en plataformas inteligentes de IoT. Permite a los clientes automatizar el proceso de conexión de dispositivos a través de una red global con altas medidas de seguridad (Cisco, 2016).

En la siguiente imagen se puede muestra la arquitectura, donde se puede evidenciar que existe una capa principal que se encarga de la seguridad de IoT y otra capa transversal que da seguridad a los dispositivos IoT.

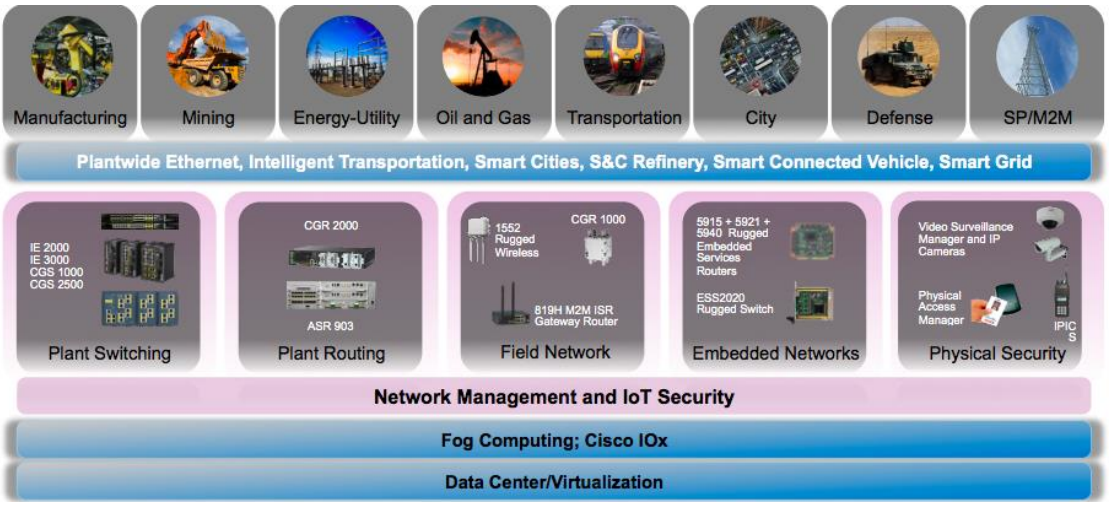


Figura 3. Cisco Internet of Things
Fuente: (Allam, 2015)

2.1.2. Plataformas y dispositivos

Dentro del mundo de IoT existen dos terminologías importantes como son plataformas y dispositivos. La plataforma IoT es la base para que los dispositivos estén interconectados y se genere un ecosistema IoT, es decir, representa al software que conecta el hardware, puntos de acceso y redes de datos, en palabras más simples la aplicación que utiliza el usuario (Cárdenas, 2016).

A continuación se presentan algunos ejemplos de plataformas:

- Google Cloud IoT
- Microsoft Azure IoT Suite
- SAP Cloud Platform for the IoT
- Salesforce IoT
- Oracle IoT
- Cisco IoT Cloud Connect
- Bosch IoT Suite

Los dispositivos son objetos o el medio que permite al usuario conectarse a una Red como por ejemplo Internet, con el objetivo de recoger e intercambiar información. Existen un sin número de dispositivos inteligentes que permiten la conexión a través de la red como por ejemplo (Apiumhub, 2018).

- Sensores que permiten tener sistemas de alerta temprana de terremoto o tsunami.
- Dispositivos que consumen energía, como interruptores, bombillas, televisores.
- El termostato Nest, permite a los usuarios controlar la temperatura de su hogar desde su teléfono inteligente o tableta
- Philips Hue, permite controlar las luces desde tu habitación hasta tu jardín.

2.1.3. Comunicaciones

Las tecnologías existentes en el momento que inicio IoT no eran las óptimas para la comunicación entre dispositivos, sin embargo, con el paso de los años el mercado se ha innovado mejorando de esta manera el tema de la comunicación (Cendón, 2017).

A continuación se describe las diferentes tecnologías utilizadas para la comunicación entre dispositivos IoT.

- **Tecnologías de acceso**

  Existen tecnologías tradicionales de conectividad inalámbrica como Wifi, Conectividad celular (2G y 4G), también existen tecnologías de corto alcance como ZigBee, Z-Wave, 6LowPAN, Bluethooth, NFC.

- **Tecnologías de core**

Existen nuevas tecnologías nativas de comunicación para el IoT, de largo alcance, bajo consumo y con bajo coste de dispositivos.

- **Sigfox,** un operador que gestiona su propia red basada en su propia tecnología.
- **LoRa,** da la opción de desplegar redes privadas, o bien ser usada por operadores para sus propias redes IoT, es una tecnología alternativa a Sigfox.

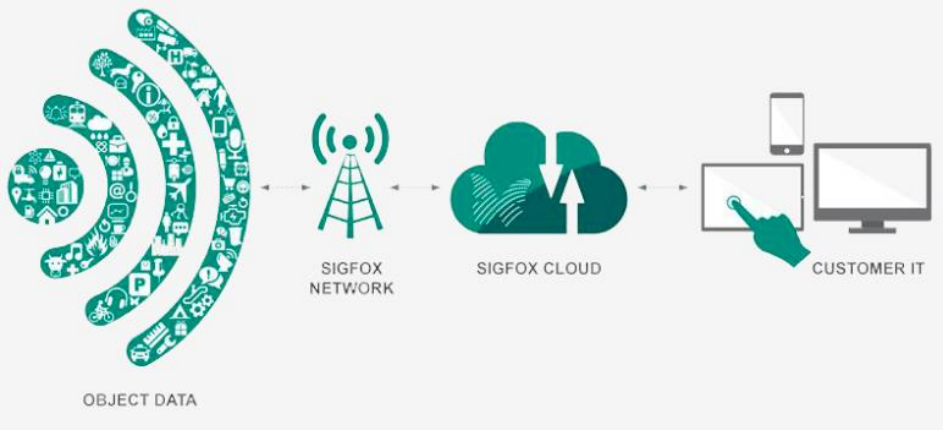


Figura 4. Arquitectura de red Sigfox
Fuente: (Sigfox, 2019)

2.2. Framework para IoT

Un framework es un entorno que agrega funcionalidad a una implementación y desarrollo de un tipo de software, productos o solución. En el caso de dispositivos IoT proporciona que el camino a las aplicaciones, desarrolladores y creadores sea más fácil, permite que los dispositivos conectados se integren de manera transparente y dinámica y que los desarrollares tengan un código base que puede ser reutilizado acorde a la necesidad de este (Martínez, Camacho, & Gutierrez, 2010). Para este caso la construcción de un marco de seguridad para dispositivos IoT.

2.3. Modelo de referencia para seguridad en IoT

Entre los principales protocolos de gestión de acceso podemos definir Oauth y OpenID.

OAuth vs OpenID

OpenID (Open Identification) es un protocolo abierto, si un usuario se autentica con este protocolo puede tener acceso a servicios y aplicaciones que soportan OpenID, esto se realiza a través de un token. Un buen ejemplo se lo encuentra en el botón “Iniciar sesión con Google”, que permite llevar a cabo el procedimiento de autenticación única utilizando la cuenta de Google del usuario (Ionos, 2019).

OAuth hace referencia a un framework de autorización y OpenID hace referencia a un protocolo de autenticación. Los dos protocolos guardan relación: OAuth 2.0 es la base sobre la que se construyó la nueva versión de OpenID, llamada OpenID Connect (OIDC). OAuth2 permite que un usuario se autentique en una aplicación mediante un servidor de autorización. Así mismo, OpenID Connect puede ampliarse con funciones opcionales como la gestión de sesiones y el encriptado de las credenciales personales (Inza, 2017).

2.4. OAuth 2.0

OAuth 2.0 es un framework de autorización que permite al usuario compartir la información de un punto A (proveedor de servicio) a un punto B (consumidor), el usuario otorga acceso limitado a sus recursos sin exponer sus credenciales, se trata de interactuar con datos protegidos sin compartir datos de contraseñas, utiliza tokens de autorización para probar una identidad entre los consumidores y proveedores de servicios (Sobers, 2018).

Proporciona flujos de autorización para aplicaciones web, de escritorio y dispositivos móviles, es un mecanismo utilizado por grandes empresas para permitir a los usuarios compartir información sobre sus cuentas con aplicaciones de terceros o sitios web (auth0, 2018).

En las figuras presentadas a continuación se muestra los roles y flujo del framework OAuth 2.0.

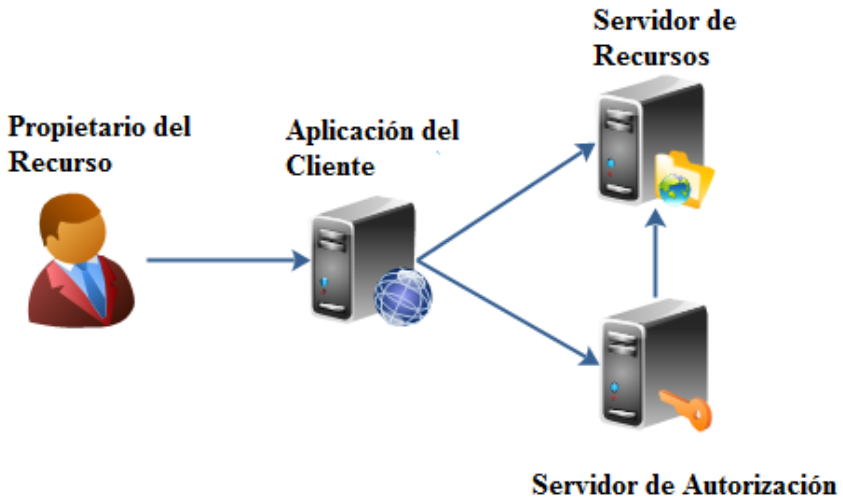


Figura 5. Roles y Flujo de OAuth 2.0
Fuente: (Jenkov, 2014)

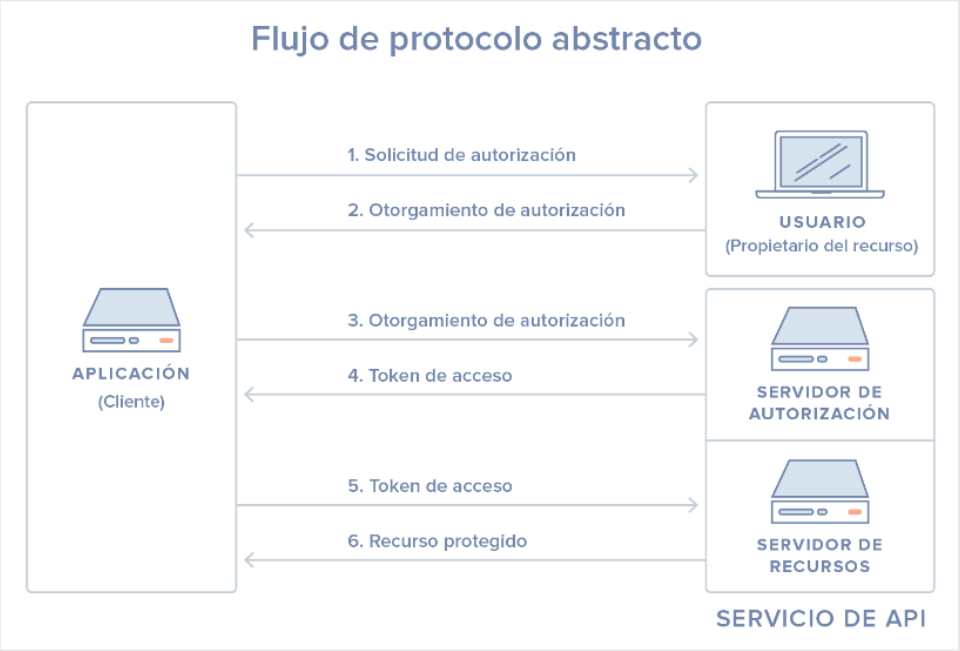


Figura 6. Flujo general de OAuth 2.0
Fuente: (Anicas, 2018)

De acuerdo con el autor Anicas, los pasos que se realizan en Oauth 2.0 son los siguientes (Anicas, 2018).

1. El Cliente solicita autorización para acceder a los recursos de servicio del usuario
2. Solo cuando el propietario autorice la solicitud, la aplicación recibe la autorización.
3. La aplicación solicita al servidor de autorización un token de acceso, presentando la autenticación de su identidad y la autorización otorgada.
4. Si la identidad de la aplicación se autentica con éxito y la autorización se válida, el servidor de autorización (API) emite un token de acceso y lo envía a la aplicación.
5. La aplicación solicita acceso al recurso protegido por el servidor de recursos (API) y presenta el token de acceso para autenticarse.

6. Si el token de acceso es válido, el servidor de recursos (API) atiende la solicitud de la aplicación.

2.4.1. OAuth Proof of Possesion (PoP)

Tiene como objetivo demostrar que el presentador de JWT (Json Web Token) tiene en su posesión la clave de Prueba de Posesión (PoP) y que el destinatario puede confirmar criptográficamente la prueba de posesión de la clave emitida por el presentador. Dentro del proceso de prueba de posesión intervienen los actores presentados a continuación: (Jones, 2016).

- **Editor (Issuer):** Es quien crea el JWT y enlaza la clave de prueba de posesión.
- **Presentador (Presenter):** Es quien debe demostrar la posesión de una clave privada o clave secreta, de una firma asimétrica o clave simétrica, respectivamente, a un destinatario.
- **Receptor (Recipient):** Es el destinatario que recibe el JWT y quien contiene la clave del comprobante de posesión del presentador.

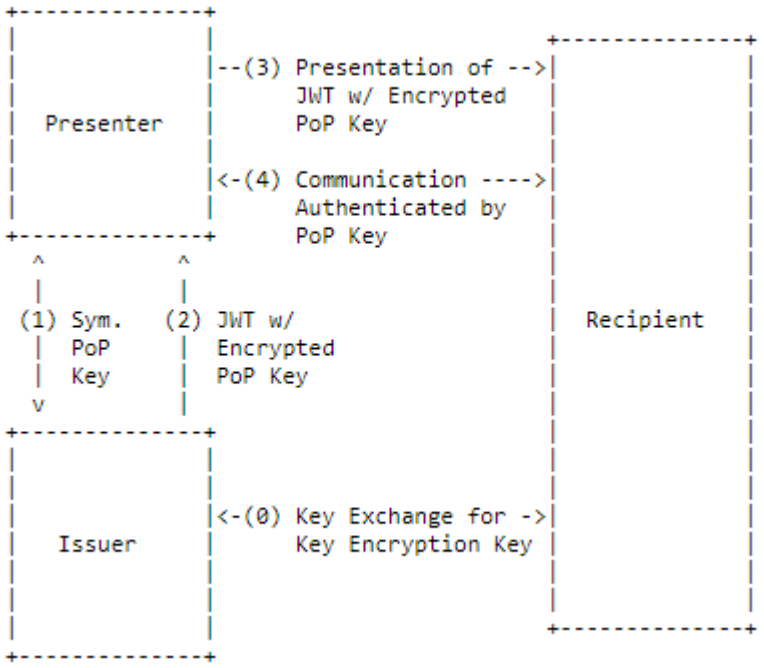


Figura 7. Proof of Possession
Fuente: (Jones, 2016)

2.4.2. Autenticación y autorización

- **Autenticación:** Es demostrar que una persona es quien dice ser, validar sus credenciales, una vez identificada como válida puede acceder a recursos definidos (Iglesias, 2014).
- **Autorización:** Es solicitar permiso para realizar cosas específicas, tiene acceso limitado. La autorización está ligada a perfiles y roles de usuario (Iglesias, 2014).

2.4.3. Json Web Tokens (JWT)

Es un estándar basado en Json que sirve para crear tokens de acceso entre dos aplicaciones (Cliente y Servidor). Los JWT son firmados digitalmente con un algoritmo de cifrado, pero no está encriptado, por lo tanto el uso de HTTPS es obligatorio al almacenar datos de usuario, el objetivo de un JWT no es cifrar los datos para que no puedan ser

leídos sino para que la parte receptora pueda confiar que los datos recibidos no se modificaron por ningún intermediario durante el transporte. Un JWT se compone de tres partes Cabecera, Payload y Firma separadas por un punto (Jones M. , 2015).

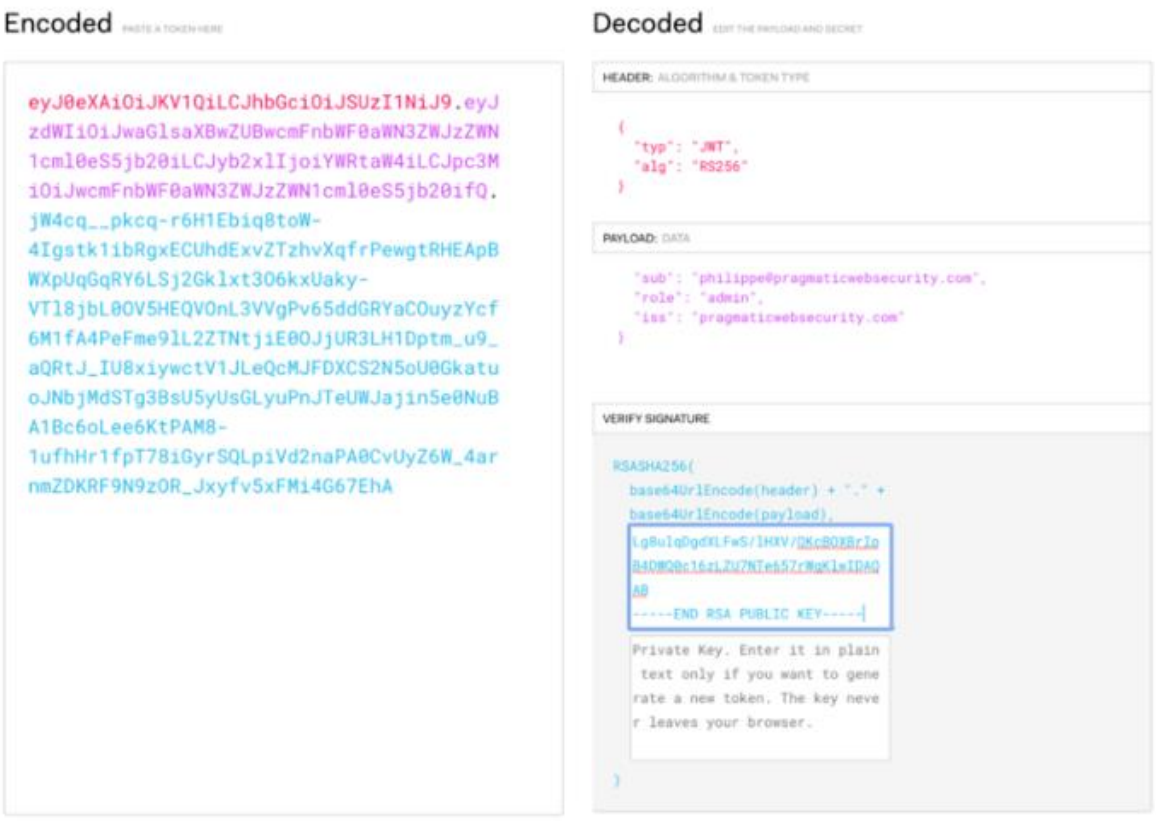


Figura 8. Json Web Token (JWT)
Fuente: (De Ryck, 2019)

- **Header:** Se define el algoritmo utilizado y el tipo de JWT
- **Payload:** Se compone por Claims que representan los atributos del token como por ejemplo: fecha de creación del token, fecha de expiración, emisor, asunto del token, entre otros.

- **Signature:** Es la más importante del token, sirve para detectar la manipulación no autorizada de un token, es la parte que hace que sea seguro, se genera en base a la combinación del header y payload cifrados con una clave secreta.

2.4.4. Json Web Key (JWK)

Una clave web JWK es una estructura de datos de JavaScript (JSON) que representa una clave criptográfica. La representación del conjunto de JWKs se denomina JWK Set JSON. La construcción de una clave JWK se realiza mediante la definición de un conjunto de parámetros que son declarados acorde la necesidad del usuario, a continuación se presenta algunas de las propiedades que se pueden establecer en un JWK (Jones, 2015).

- **“kty” (Key Type) Parameter:** Identifica el algoritmo criptográfico utilizado, existen 2 tipos RSA o EC.
- **"use" (Public Key Use) Parameter:** Indica si la clave pública es utiliza para cifrar datos o verificar la firma de los datos. Existe dos tipos: "sig" (firma) o "enc" (cifrado).
- **“key_ops” (Key Operations) Parameter:** Identifica la operación para lo cúal va a ser utilizada la clave.
- **“alg” (Algorithm) Parameter:** Es el algoritmo de cadena ASCII que se define para usar con la llave.
- **"kid" (Key ID) Parameter:** Representa el ID de la clave y se utiliza para coincidir con una clave específica. Este parámetro es utilizado en Json Web Sign (JWS.)

En la siguiente imagen se presenta un ejemplo de la construcción de un JWK:

```
{
  "keys":
  [
   {"kty":"oct",
    "alg":"A128KW",
    "k":"GawgguFyGrWKav7AX4VKUg"},

   {"kty":"oct",
    "k":"AyM1SysPpbyDfgZld3umj1qzKObwVMkoqQ-EstJQLr_T-1qS0gZH75
  aKtMN3Yj0iPS4hcgUuTwjAzZr1Z9CAow",
    "kid":"HMAC key used in JWS spec Appendix A.1 example"}
  ]
}
```


Figura 9. Json Web Key (JWK)
Fuente: (Ory, 2019)

Existen dos tipos de firmas: Las firmas Simétricas que hacen uso de claves compartidas privadas entre cliente y servidor, las partes deben conocer la clave para envió y lectura de comprobación del token. Las firmas Asimétricas son firmadas con una clave privada por el remitente, y para que el receptor valide la firma recibida utiliza una clave pública (De Ryck, 2019).

Figura 10. Firma simétrica
Fuente: (De Ryck, 2019)

Figura 11. Firma asimétrica
Fuente: (De Ryck, 2019)

2.4.5. Json Web Signature (JWS)

La firma web JSON es la parte más importante de un JWT ya que representa la seguridad y contenido protegido de firmas digitales o códigos de autenticación de mensajes, contiene la prueba que la información no haya sido alterada desde la firma del emisor, proporcionan protección de integridad. JWS representa contenido firmado utilizando estructuras de datos JSON y codificación base64url. Un Json Web Signature se compone de tres partes (Jones, 2012).

- **Encabezado JWS:** Describe el método de firma y los parámetros empleados.
- **Carga útil JWS**: Contenido del mensaje que debe protegerse.
- **Firma JWS:** Garantiza la integridad tanto del encabezado y carga útil JWS.

La siguiente figura presenta un ejemplo de un JWS, donde, typ: representa el tipo de token, alg: representa el tipo de algoritmo empleado y finalmente se muestra la firma digital.

Encabezado JWS

```
{"typ":"JWT",
 "alg":"HS256"}
```

Firma Digital

```
eyJ0eXAiOiJKV1QiLA0KICJhbGciOiJIUzI1NiJ9
```

Figura 12. Json Web Signature (JWS)
Fuente: (Jones, 2012)

2.4.6. HTTP vs HTTPS

Para conocer la diferencia de los términos HTTP y HTTPS es importante primero conocer el significado de las siglas HTTP y el objetivo del mismo, del inglés Hyper Text Transport Protocol, es un sistema orientado al funcionamiento de petición-respuesta y está diseñado para asegurarse que los datos transportados entre distintos equipos (Cliente y Servidor) que conforman una red, lleguen de manera correcta durante el proceso de petición–respuesta. En este proceso el Cliente es quien emite la petición y el Servidor la respuesta. Una vez identificado el termino HTTP, se puede continuar con las siglas HTTPS del inglés, Hyper Text Transport Protocol Secure, este se encuentra basado en los protocolos HTTPS y SSL/TLS, cuyo objetivo es mantener segura y cifrada la información que se transporta entre cliente y servidor, de esta manera si un tercer actor intentara descifrar la información enviada no le será posible. Para que la información permanezca segura es necesario implementar el esquema de “Certificado” que debe estar firmado por una autoridad (Losada Perez, 2015). A continuación se presenta un cuadro de las diferencias entre los protocolos:

Tabla 1
HTTP vs HTTPS

http	HTTPS
URL comienza con http://	URL comienza con https://
Sin Garantía	Asegurado
Funciona a nivel de aplicación	Funciona a nivel de transporte
Sin cifrado	Con certificado
No hay certificados requeridos	Certificado prescrito

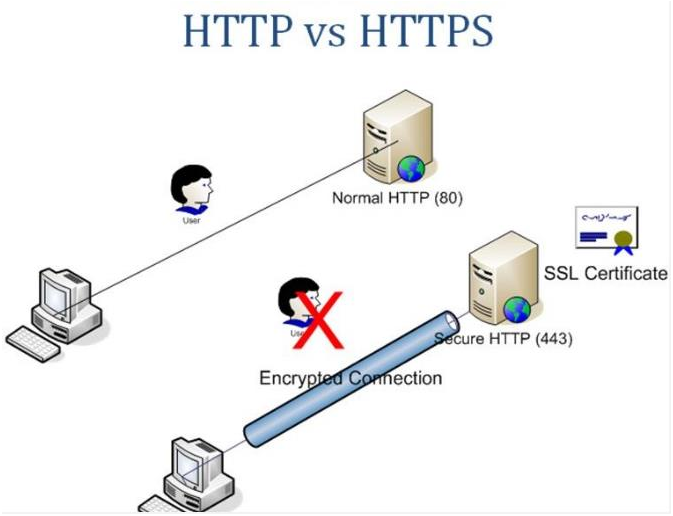


Figura 13. HTTP vs HTTPS
Fuente: (Losada Perez, 2015)

2.4.7. Transmission Control Protocol (TCP)

Es un protocolo orientado a la conexión, que permite la transmisión de datos entre dos máquinas y que cada una de ellas controle el estado de la transmisión. En el nivel de aplicación, habilita la administración de datos que vienen en el nivel bajo del modelo, mediante el Internet Protocol (IP). Cuando se usa este protocolo, las aplicaciones pueden establecer conexión de forma segura, la máquina emisora (cliente) es la que solicita la

conexión, y la máquina receptora (servidor) es quien recibe la petición. Por eso se dice que es en un entorno Cliente-Servidor. Una de las funciones que permite el protocolo es la multiplexación, que es la transferencia de datos a diversas aplicaciones por la misma línea. En la imagen adjunta se presenta un ejemplo de la función (Villagómez, 2017).

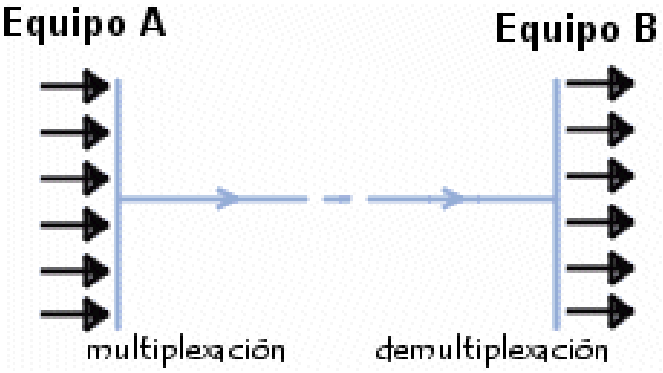


Figura 14. Función de multiplexación
Fuente: (Villagómez, 2017)

2.4.8. Transport Layer Security (TLS)

Es un protocolo que permite la comunicación segura entre dos aplicaciones Cliente-Servidor a través de internet, impide que durante la comunicación los datos sean manipulados, falsificados o que atacantes vean los datos. Para que un canal de comunicación sea seguro debe presentar las siguientes propiedades y componentes principales (Rescorla, 2018).

Propiedades

- Autenticación: El Servidor debe estar autenticado, el Cliente es opcional.
- Confidencialidad: Los datos enviados son solo visibles para los puntos finales.
- Integridad: Los datos no pueden ser manipulados por atacantes.

Componentes

- Protocolo de Enlace: Sirve para autenticar, negociar, comunicar y establecer parámetros criptográficos y claves compartidas entre las dos partes Cliente-Servidor, en caso de un ataque no se puede forzar la comunicación entre las partes.
- Protocolo de Registro: Usa parámetros para proteger el tráfico entre ambas partes; divide la información en varias partes durante el tráfico y cada registro se encuentra protegido de manera independiente utilizando claves.

2.4.9. Secure Socket Layer (SSL)

SSL por sus siglas en inglés (Secure Socket Layer) es un protocolo de seguridad que garantiza que el transporte de datos entre un cliente y servidor web se realice de forma segura e integra y lleguen al servidor correcto. Los datos enviados son cifrados o encriptados, de manera que se asegura no puedan ser leídos por un tercer actor. En la ejecución de SSL se crean caminos de cifrado para las sesiones privadas y la clave pública puede ser compartida con cualquier actor, por esta razón se utilizan dos claves: la clave pública para encriptar la información y la clave privada para desencriptar la información. Es importante conocer que para utilizar un certificado SSL en una página web, el servidor de Internet debe soportar SSL (certsuperior, 2016).

2.5. Herramientas de desarrollo y pruebas

2.5.1. Node.js

Es un entorno de ejecución del lenguaje de programación Javascript, el cuál es la razón de la terminología JS, utiliza un modelo asíncrono y dirigido a eventos. Diseñado para construir aplicaciones en red escalables. El modelo de eventos no es una librería en su contrario se presenta un bucle de eventos como entorno (Node js Foundation, 2018).

2.5.2. Express

Es un framework web flexible para Node.js, escrito en JavaScript. Es robusto, rápido y muy simple. Proporciona varios mecanismos.

- Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
- Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
- Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.
- Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.

2.5.3. BCrypt

Es un algoritmo de funciones hash para contraseñas, se encuentra basado en el cifrado de Blowfish. BCrypt tiene un fragmento aleatorio denominado salt, que añade un grado de complejidad y seguridad que evita que el hash de una contraseña sea único.

Este fragmento, impide un ataque por fuerza bruta a todas las contraseñas que forman parte del sistema (Vicente, 2017).

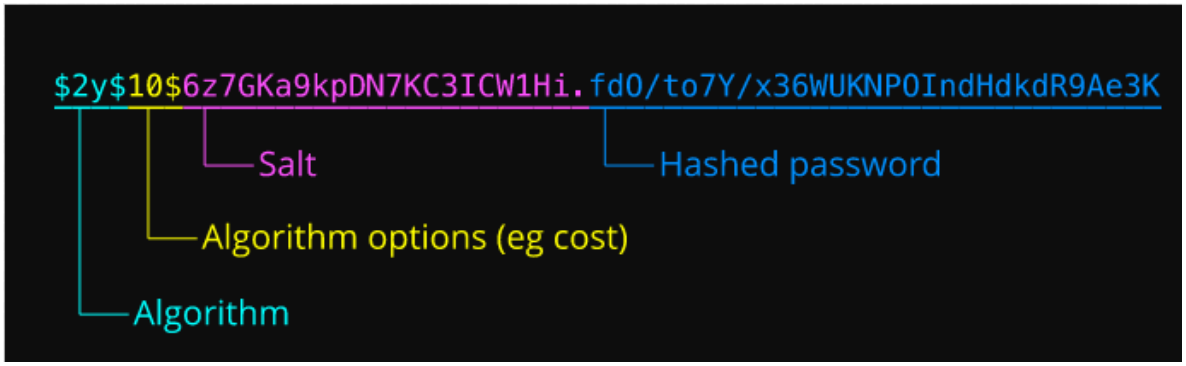


Figura 15. Ejemplo Hash BCrypt
Fuente: (Davis, 2016)

2.5.4. WireShark

Es un analizador de protocolos que permite realizar un análisis y ver lo que está sucediendo en la red a nivel detallado. Entre sus características comprende (Wireshark, 2019).

- Inspección profunda de cientos de protocolos
- Captura en vivo y análisis fuera de línea
- Multiplataforma
- Los filtros de visualización más potentes de la industria.
- Lee / escribe muchos formatos de archivo de captura
- Compatibilidad con descifrado para muchos protocolos, incluidos Kerberos, SSL / TLS, y WPA / WPA2, entre otros.

2.5.5. Hydra

Es una de las herramientas más conocidas en hacking ético y lógicamente por atacantes informáticos, cuyo objetivo es descifrar las contraseñas que se han almacenado en una base de datos o se encuentran en tránsito dentro de un sistema; para de esta manera conseguir acceso no autorizado a una red o sistemas (Velasco, 2019).

Hydra trabaja con un diccionario de palabras que le permite al atacante obtener una lista de usuarios y contraseñas.

2.5.6. Lenguaje Unificado Modelado (UML)

Es un estándar que se utiliza para crear diagramas, esquemas y documentación de un sistema de software, no es un lenguaje de programación, sino es un conjunto de normas y estándares gráficos que permiten representar algo. UML puede realizar distintos tipos de diagramas acorde a la necesidad del usuario, define normas para construir muchos tipos de esquema, no un solo tipo. Existen diferentes tipos de Diagrama, listados a continuación (Krall, 2019).

- Diagramas de casos de uso
- Diagramas de clases
- Diagramas de secuencia
- Diagramas de colaboración
- Diagramas de estado
- Diagramas de actividad
- Diagramas de paquetes
- Diagramas de arquitectura software

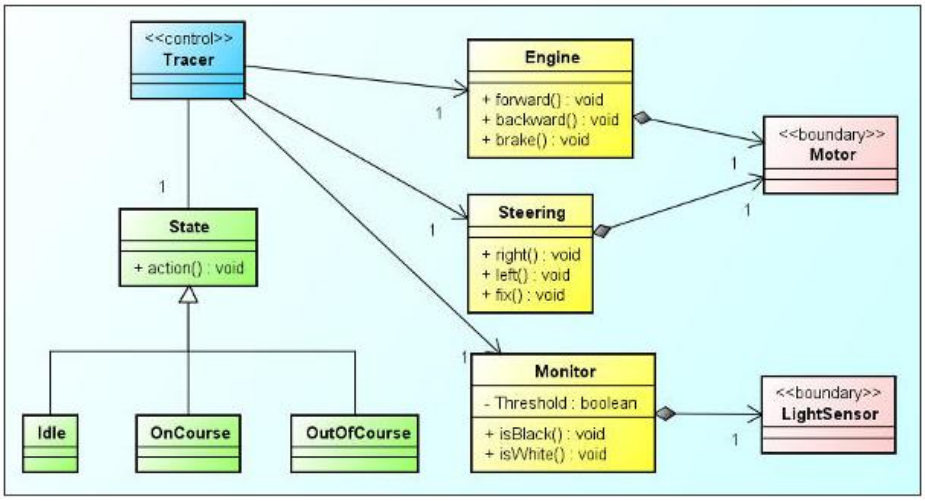


Figura 16. Ejemplo de diagrama UML
Fuente: (Krall, 2019)

2.6. Metodologías

2.6.1. Metodología de la investigación

Debido a que la investigación está orientada al desarrollo de un framework para asegurar la autenticación en un ecosistema IoT, se propone una metodología propia (Ad hoc), la cual tienen como finalidad el cumplimiento de los objetivos específicos planteados.

Figura 17. Fases de la metodología propuesta

- **Situación actual:** Realizar revisiones de literatura sobre la seguridad de autenticación para dispositivos IoT.
- **Viabilidad de la propuesta:** Mediante una revisión inicial de literatura, se identificará y analizará estudios o soluciones que se hayan realizado con respecto a la autenticación en un ecosistema IoT.
- **Análisis y diseño:** Aplicar la metodología SCRUM para el análisis y diseño del framework de seguridad.
- **Construcción y pruebas:** Desarrollar y validar el framework, mediante ataques de penetración los cuales ayudarán a medir la vulnerabilidad de seguridad de autenticación en un ecosistema IoT.

2.6.2. Metodologías ágiles de desarrollo

Las metodologías ágiles de desarrollo permiten adaptar la formar de trabajo a las condiciones del proyecto, ayudan a diseñar y crear el producto deseado por cliente. Mediante una metodología ágil se puede dar seguimiento al desarrollo del proyecto durante todo su ciclo, el equipo de desarrollo puede emitir respuestas rápidas e impredecibles sobre su proyecto. Los equipos evalúan sus proyectos mediante reuniones continuas lo que les permite analizar y realizar mejoras durante el desarrollo del proyecto, dando como como resultado la capacidad de generar un producto valioso (Gonzalves, 2019).

A continuación se detallan algunas de las metodologías ágiles (Rosselló, 2019).

- **Scrum**

  Se basa en una estructura de desarrollo incremental, contiene las etapas de análisis, desarrollo y pruebas. La etapa de desarrollo es conocida como iteraciones del proceso sprint, es decir presenta entregas parciales del proyecto.

  Las reuniones son el pilar fundamental de esta metodología ya que se realizan procesos de planificación, revisiones, y retrospectiva.

- **Extreme Programming XP**

  Herramienta utilizada cuando las empresas se encuentran en proceso de consolidación, su principal objetivo es ayudar a las relaciones de los clientes con los empleados.

- **Kaban**

  Conocida como Tarjeta Visual es útil para responsables de los proyectos. Elabora un diagrama de tres columnas pendientes, en proceso y terminales. Ayuda a mejorar la productividad y eficiencia del equipo de desarrollo.

- **Desing Sprint**

  Se trata de un proceso que dura cinco días donde el negocio debe resolver diseño, prototipo y testeo de clientes. Elabora etapas de sprints, permite reducir tiempos de proyecto y errores mediante el prototipo, en vez de esperar a lanzar un producto para entender si la idea es buena, el prototipo proporciona antes la información para evitar posibles errores.

Si bien es cierto existen varias metodologías acordes a las necesidades de cada usuario, después de realizar un análisis, se pudo concluir que Scrum es la metodología adecuada para el desarrollo del presente proyecto, al ser un proyecto de investigación Scrum permitirá controlar los avances del proyecto y corregir a tiempo los errores presentados, disminuyendo de esta manera el impacto que pueda presentarse por problemas mayores.

2.6.3. Metodología SCRUM

Es una metodología ágil que permite trabajar en una serie de iteraciones (sprints) para gestionar el desarrollo del software. Se gestiona los sprints mediante reuniones frecuentes, el mismo que es un punto fundamental dentro de esta metodología. Se presenta a continuación el flujo de la metodología (Ali, y otros, 2019).

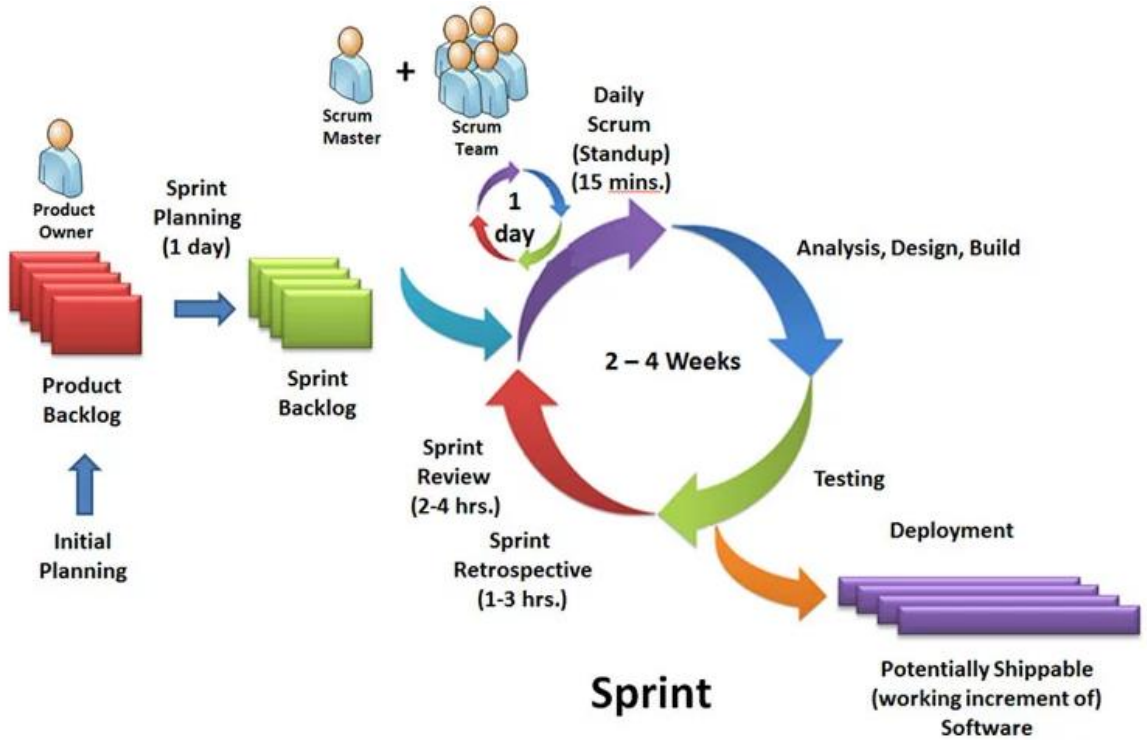


Figura 18. Metodología SCRUM
Fuente: (Lara, 2017)

Roles

Product Owner: Representa la voz del cliente y recopila los objetivos del proyecto, se encarga de garantizar que el equipo de trabajo cumpla con dichos objetivos.

Scrum Master: Se encuentra encargado del manejo del equipo que desarrollará el proyecto, ayuda al equipo a mantenerse activo.

Scrum Team: Son las personas encargadas del desarrollo y entregar el producto.

Stakeholders: Son las personas interesadas en el producto final.

Eventos

Planeación de sprint: Los involucrados del equipo se reúnen, se realiza al inicio de cada sprint es aquí donde se determina cuáles son los requerimientos y que tareas se asigna cada integrante con los objetivos a cumplir dentro de una iteración.

Reunión Diaria: Tiene una duración de máximo 15 minutos, se revisa el trabajo realizado hasta la fecha y la previsión para el siguiente día.

Refinamiento: El dueño del producto revisa los elementos dentro de la Pila de Productos, con el fin de esclarecer cualquier duda dentro del equipo desarrollador.

Revisión de Sprint: Análisis y revisión del incremento generado.

Retrospectiva: El dueño del producto se reúne con el Scrum master para conocer los detalles del Sprint.

CAPÍTULO III

ESTADO DEL ARTE

3.1. Planificación del estado del arte

3.1.1. Problemática

La seguridad en la autenticación es una de las ventanas más trascendentales para el robo de información, ya que los usuarios al proporcionar las credenciales en la internet y al no tener una seguridad adecuada de los dispositivos IoT, pueden convertirse en recursos vulnerables.

3.1.2. Objetivo

Identificar las diferentes técnicas para vulnerar la autenticación de los dispositivos IoT, con la finalidad de determinar las técnicas para combatir dichas vulnerabilidades.

3.1.3. Preguntas de investigación

RQ1: ¿Qué características poseen los protocolos de seguridad para dispositivos IoT?

RQ2: ¿Qué tipos de técnicas existen para vulnerar la autenticación de los dispositivos IoT?

RQ3: ¿Qué características poseen las técnicas para vulnerar la autenticación de los dispositivos IoT?

RQ4: ¿Bajo qué criterios pueden clasificarse las técnicas para vulnerar la autenticación de los dispositivos IoT?

RQ5: ¿Qué tipos de técnicas existen para mitigar la vulnerabilidad de la autenticación de los dispositivos IoT?

RQ6: ¿Qué características poseen las técnicas para mitigar la vulnerabilidad la autenticación de los dispositivos IoT?

RQ7: ¿Bajo qué criterios pueden clasificarse las técnicas para mitigar la vulnerabilidad de la autenticación de los dispositivos IoT?

RQ8: ¿Es posible desarrollar un framework que asocie las características más importantes de las técnicas existentes para mitigar la vulnerabilidad de la autenticación de los dispositivos IoT, contemplando para su diseño las técnicas existentes de vulneración?

3.2. Método de investigación

Para el análisis del estado del arte se aplicó el proceso de Revisión Inicial de Literatura (Kitchenham, 2009). Con la problemática planteada en la planificación del estado del arte, procederemos con la ejecución de las fases a continuación descritas.

- Criterios de inclusión y exclusión
- Grupo de control
- Cadena de búsqueda
- Proceso de selección
- Síntesis de resultados

3.2.1. Criterios de inclusión y exclusión

La definición de los criterios de inclusión y exclusión permitieron facultar artículos que estén enmarcados de manera explícita con el propósito del trabajo de investigación.

Inclusión

- Artículos que se hayan publicado desde el año 2014 hasta la fecha, con el propósito de englobar temas de investigaciones que vayan de acuerdo con la problemática planteada.
- Artículos publicados en idioma inglés en reconocidas bibliotecas digitales.
- Artículos que contengan técnicas para vulnerar y técnicas para mitigar la seguridad de autenticación para dispositivos IoT.

Exclusión

- Artículos que tengan relación con la seguridad de autenticación de aplicaciones web o aplicaciones móviles.
- Artículos que su contenido se base en la seguridad de las contraseñas.

3.2.2. Grupo de control

La realización la RIL tiene como objetivo determinar el estado del arte de la problemática presentada en la vulnerabilidad de seguridad de autenticación para dispositivos IoT en concordancia con el alcance definido para el proyecto de investigación. A continuación de describe los pasos utilizados para esta etapa.

Revisión inicial

La búsqueda inicial se la hizo en diferentes bibliotecas digitales, de donde se seleccionaron artículos preliminares, los cuales conformarán el grupo de control.

Validación cruzada

Con la ayuda del docente evaluador se analizaron los artículos preseleccionados, constatando que estos cumplan con los criterios de inclusión y exclusión propuestos.

Integración del grupo de control

Con la ejecución de las actividades anteriores y por medio de un consenso, se conformó el grupo de control con los siguientes artículos.

Tabla 2
Palabras claves para la investigación

Código	Título	Cita	Palabras claves
EC1	Study of authentication with IoT testbed	(Crossman & Liu, 2015)	“Internet of Things”, “IoT”, “vulnerabilities”, “risks”, “cyber security”, “attack”, “authorization”, “authentication”, “network”, “internet”, “protocols”
EC2	Botnets and Internet of Things Security	(Bertino & Islam, 2017)	“Internet of Things”, “IoT systems”, “IoT devices”,

CONTINÚA

			“security”, “attacks”, “IoT ecosystem”, “network”
EC3	Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks	(Abomhara & Koien, 2015)	“Internet of Things devices”, “IoT”, “vulnerabilities”, “threats”, “protocols”, “attacks”, IoT devices”, “IoT services”, “cyber attack”, “internet”
EC4	Authentication Techniques for the Internet of Things: A Survey	(Saadeh, Sleit, Qatawneh, & Almobaideen, 2016)	“Internet of Things”, “IoT”, “authentication techniques, “evaluation models”, “security attacks”, “IoT architectures”, “authorization”
EC5	IoT Security Techniques Based on Machine Learning	(Xiao, Wan, Lu, Zhang, & Wu, 2018)	“Internet of Things”, “IoT”, “authentication”, “authorization”, “access control”, “IoT devices”, “network”, “attack model”, “IoT systems”, “IoT security solutions”

CONTINÚA

EC6	A lightweight authentication protocol for Internet of Things	(Lee, Lin, & Huang, 2014)	“Internet of Things”, “IoT”, “security”, “authentication”, “privacy”, “security protocol”, “encryption”
EC7	IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios	(Cirani, Picone, Gonizzi, Veltri, & Ferrari, 2014)	“Open authorization”, “Oauth”, “Internet of Things”, “IoT”, “authorization”, “authentication”, “open protocol”, “access control”, “internet”
EC8	Authentication and Authorization for the Internet of Things	(Kim & Lee, 2017)	“Internet of Things”, “IoT”, “authentication”, “authorization”, “security”, “access control”, “security”

3.2.3. Cadena de búsqueda

Para esta fase se fijaron los términos más representativos que servirán como base para formación de la cadena de búsqueda, a continuación, se detalla los términos encontrados ordenados por frecuencia de apariciones.

Tabla 3

Frecuencia de apariciones de palabras claves

Término	EC1	EC2	EC3	EC4	EC5	EC6	EC7	EC8	No. De apariciones
Internet of Things	X	X	X	X	X	X	X	X	8
IoT	X	X	X	X	X	X	X	X	8
Authentication	X			X	X	X	X	X	6
Authorization	X			X	X		X	X	5
Security	X	X		X	X	X			5
Attacks	X	X	X	X	X				5
protocols - protocol	X		X			X	X		4
39ccess control					X		X	X	3
Vulnerabilities	X		X						2

Para determinar la cadena de búsqueda se usaron lo términos representativos del grupo de control, para la relación de los términos se utilizaron los operadores lógicos AND y OR. Mediante la ejecución de la opción búsqueda de comandos en la búsqueda avanzada de la biblioteca digital IEEE Xplore Digital Library se fijó la siguiente cadena.

(((internet of things OR iot) AND authentication) OR ((internet of things OR iot) authorization)) AND (security OR access control OR protocol OR protocols) AND

((protocols AND security) OR (protocols AND authorization) OR (protocols AND authentication)) AND (((internet of things OR iot) AND attacks) OR ((internet of things OR iot) and vulnerabilities))

3.2.4. Proceso de selección

Como ya se mencionó en la fase anterior, la cadena de búsqueda se construyó con la ayuda de la IEEE Xplore Digital Library, por lo que, para la selección de los artículos se hizo uso de la misma biblioteca. Utilizando la cadena de búsqueda construida y agregando un filtro de rango de fechas desde el año 2017 al año 2019, se obtuvieron los artículos más recientes, de los cuales, 50 artículos fueron sometidos a una lectura a nivel de title y abstract considerados como los más relevantes para el objetivo de la investigación. Se debe indicar que las referencias bibliográficas de cada artículo valorado se encuentran disponible al final del documento.

Tabla 4

Artículos preseleccionados

No.	Title	Abstract
1	**Analysis of authentication techniques in Internet of Things (IoT)**	Internet of Things (IoT) is slowly but steadily becoming part of different aspects of our lives, with its applications ranging from smart homes, to wearable devices, to healthcare, etc. This wide spectrum of applications results in shared data containing large amount of users' private information. The security of such information becomes a paramount concern. The IoT security requirements include

CONTINÚA

		data confidentiality, data integrity, authentication, access control, privacy, etc. In particular, authentication of IoT devices has a particular importance given the variety of attacks which might result from its breach [1]. This paper provides an analysis of the different authentication schemes proposed in the literature. Through a multi-criteria classification, it compares and analyzes the existing authentication protocols, showing their advantages and disadvantages.
2	**Authentication of IoT Device and IoT Server Using Secure Vaults**	Internet of Things is a topic of much interest and, in last few years, security of the IoT systems is a field of tremendous research activities. Mutual authentication between IoT devices and IoT servers is an important part of secure IoT systems. Single password-based authentication mechanisms, which are widely used, are vulnerable to side-channel and dictionary attacks. In this paper, we present a multi-key (or multi-password) based mutual authentication mechanism. In our approach, the shared secret between the IoT server and the IoT device is called secure vault, which is a collection of equal sized keys. Initial contents of the secure vault are shared between the server and the IoT device and contents of the

CONTINÚA

		secure vault change after every successful communication session. We have implemented this mechanism on an Arduino device to prove our algorithm is feasible on IoT devices with memory and computational power constraints.
3	**Security of a new lightweight authentication and key agreement protocol for internet of things**	The Internet of Things (IoT) consists of all kinds of sensors which collect much information. The privacy and anonymity in IoT is fundamental to the system and must be resolved. Sima proposed a protocol about lightweight authentication and key agreement and claimed to be secure and robust against all existing methods of attack. In this paper, the author points out the vulnerability of Sima's protocol and makes improvements to it which can resist the malicious attack.
4	**(WIP) Authenticated Key Management Protocols for Internet of Things**	The Internet of Things (IoT) provides transparent and seamless incorporation of heterogeneous and different end systems. It has been widely used in many applications such as smart homes. However, people may resist the IOT as long as there is no public confidence that it will not cause any serious threats to their privacy. Effective secure key management for things authentication is the prerequisite of security operations. In this paper, we

CONTINÚA

		present an interactive key management protocol and a non-interactive key management protocol to minimize the communication cost of the things. The security analysis show that the proposed schemes are resilient to various types of attacks.
5	**Security Vulnerabilities of Internet of Things: A Case Study of the Smart Plug System**	With the rapid development of the Internet of Things, more and more small devices are connected into the Internet for monitoring and control purposes. One such type of devices, smart plugs, have been extensively deployed worldwide in millions of homes for home automation. These smart plugs, however, would pose serious security problems if their vulnerabilities were not carefully investigated. Indeed, we discovered that some popular smart home plugs have severe security vulnerabilities which could be fixed but unfortunately are left open. In this paper, we case study a smart plug system of a known brand by exploiting its communication protocols and successfully launching four attacks: 1) device scanning attack; 2) brute force attack; 3) spoofing attack; and 4) firmware attack. Our real-world experimental results show that we can obtain the authentication credentials from the

CONTINÚA

		users by performing these attacks. We also present guidelines for securing smart plugs.
6	**A Collaborative PHY-Aided Technique for End-to-End IoT Device Authentication**	Nowadays, Internet of Things (IoT) devices are rapidly proliferating to support a vast number of end-to-end (E2E) services and applications, which require reliable device authentication for E2E data security. However, most low-cost IoT end devices with limited computing resources have difficulties in executing the increasingly complicated cryptographic security protocols, resulting in increased vulnerability of the virtual authentication credentials to malicious cryptanalysis. An attacker possessing compromised credentials could be deemed legitimate by the conventional cryptography-based authentication. Although inherently robust to upper-layer unauthorized cryptanalysis, the device-to-device physical-layer (PHY) authentication is practically difficult to be applied to the E2E IoT scenario and to be integrated with the existing, well-established cryptography primitives without any conflict. This paper proposes an enhanced E2E IoT device authentication that achieves seamless integration of PHY security into traditional asymmetric cryptography-based authentication schemes. Exploiting the collaboration of

CONTINÚA

		several intermediate nodes (e.g., edge gateway, access point, and full-function device), multiple radio-frequency features of an IoT device can be estimated, quantized, and used in the proposed PHY identity-based cryptography for key protection. A closed-form expression of the generated PHY entropy is derived for measuring the security enhancement. The evaluation results of our cross-layer authentication demonstrate an elevated resistance to various computation-based impersonation attacks. Furthermore, the proposed method does not impose any extra implementation overhead on resource-constrained IoT devices.
7	**An Authentication Technique Based on Oauth 2.0 Protocol for Internet of Things (IoT) Network**	Internet of Things (IoT)is the fast-growing emerging technology and it is making our daily life (activities)easier and better. The IoT is connecting millions of sensors and heterogeneous devices having different computation, communication, and storage capabilities which are increasing rapidly. The IoT network is dissimilar and larger from the conventional network. However, different natures of the connected devices raising security concerns and challenges which affect the importance of IoT technology. In this paper, we proposed an authentication scheme

CONTINÚA

		based on the OAuth 2.0 protocol to secure access IoT network by providing authentication service. An OAuth 2.0 protocol is used to propose authentication mechanism to allow only authorized and authentic users by comparing user information and access tokens in the security manager local database and denies the access to the IoT network. It also keeps safe IoT network from different types of attacks like impersonation and replays attacks etc.
8	**An Improved User Authentication Protocol for IoT**	Rapid development in the field of internet of things (IoT) has increased numbers of applications. Meanwhile, security and privacy threats are also introduced. Various authentication protocols are devised to resist the malicious attacks. Li et al. proposed a remote user authentication protocol using smart cards and they claimed their protocol was secure. However, we find that it cannot resist DoS attack, stolen-verifier attack and replay attack. Then we propose a three-factor remote authentication protocol using smart card based on biometric. The proposed protocol can resist DoS attack effectively by increasing local verification of user identity and password.
9	**A Survey: Authentication**	Internet of things (IoT) is a new technology that enables things in a specific environment to communicate with each

CONTINÚA

Protocols for Wireless Sensor Network in the Internet of Things; Keys and Attacks	other over the internet. Objects need to communicate wirelessly with each other, so IoT includes wireless sensor networks (WSN), radio frequency identification (RFID), near field communication (NFC), and many to facilitate the communication process. WSN is a network of connected nodes use sensors to collect and share data between each other. In order to build heterogeneity IoT environments, Wireless Sensor Networks should be established to monitor and record activities of connected things. Standard security protocols are used to provide a seamless and secure connection in Wireless Sensor Networks, especially authentication and access control of nodes. In this paper, the importance of security issues within IoT-WSN environment highlighted, by providing a description of some existing authentication protocols used in wireless sensor network. Additionally, classifies the chosen protocols according to types of the secret key used to achieve their goals and types of attack that each protocol can prohibit in the network. The result presents common attacks in WSN resisted by some chosen protocols. This research considered a base for other researchers in Wireless Sensor Network and Internet of Things field.

CONTINÚA

10	**Taxonomy of authentication techniques in Internet of Things (IoT)**	Internet of Things (IoT) is slowly but steadily becoming part of different aspects of our lives, with applications ranging from smart homes, to wearable devices, to healthcare, etc. This wide spectrum of applications results in shared data containing large amount of users' private information. The security of such information becomes a paramount concern. The IoT security requirements include data confidentiality, data integrity, authentication, access control, privacy, etc. In particular, authentication of IoT devices has a particular importance given the variety of attacks which might result from its breach [1]. This paper provides a survey of the different authentication schemes proposed in the literature. Through a multi-criteria classification, it compares and analyzes the existing authentication protocols, showing their advantages and disadvantages.
11	**Analysis of vulnerabilities in MQTT security using Shodan API and implementation**	Among the technologies evolved in the recent years, a remarkable one is the IoT (Internet of Things), wherein the `thing' in IoT could be smart phones, tablets, PCs and almost anything with a sensor on it like cars, people, machines in production plants, jet engines, oil drills, wearable devices and many more objects. A standardized,

CONTINÚA

	of its countermeasures via authentication and ACLs	light-weight, session layer protocol with publish/subscribe architecture widely used for messaging and information exchange among IoT devices is the MQTT (MQ Telemetry Transport) protocol. In this paper, we identify various security loopholes in MQTT, using Shodan API and implementing an experimental setup on a Raspberry Pi as an MQTT Broker and python programs as publisher/subscriber clients. The experimental results with respect to the security issues in this protocol at packet and topic levels were studied and the corresponding security measures, consisting of authentication and authorization techniques (ACLs) were implemented. As a result, the Broker was then found to be immune to such attacks. This paper is a concise study of security inconsistencies in MQTT and its countermeasures.
12	**A Survey of Key Bootstrapping Protocols Based on Public Key Cryptography in the Internet of Things**	The Internet of Things envisages connecting all physical objects or things to the Internet, using devices as diverse as smartphones, coffee makers, washing machines, automobiles, lamps, and wearable devices, among many others. The explosive growth of Internet-connected sensing and actuating devices has bridged the gap between the physical and the digital world, with new

CONTINÚA

solutions bringing benefits to people, processes, and businesses. However, security will be a major challenge in enabling most of such applications. The lack of secure links exposes data exchanged by devices to theft and attacks, with hackers already showing a keen interest in this area. Secure communication in the IoT will require a multifaceted approach, in particular, targeting aspects as relevant as the communications' protocols and data that need to be secured. One of the major aspects among these is how keys are bootstrapped in devices, for the purpose of supporting secure communications. In this paper, we survey the state of the art in key bootstrapping protocols based on public-key cryptography in the Internet of Things. Due to its inherent scalability, such protocols are particularly relevant for the implementation of distributed identity and trust management mechanisms on the IoT, in the context of which devices may be authenticated and trusted. The reviewed proposals are analyzed and classified on the basis of the key delivery method, the underlying cryptographic primitive, and the authentication mechanism supported. We also identify and discuss the main challenges of implementing such methods in the

CONTINÚA

		context of IoT applications and devices, together with the main avenues for conducting further research in the area.
13	**Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices**	Device authentication is an essential security feature for Internet of Things (IoT). Many IoT devices are deployed in the open and public places, which makes them vulnerable to physical and cloning attacks. Therefore, any authentication protocol designed for IoT devices should be robust even in cases when an IoT device is captured by an adversary. Moreover, many of the IoT devices have limited storage and computational capabilities. Hence, it is desirable that the security solutions for IoT devices should be computationally efficient. To address all these requirements, in this paper, we present a lightweight and privacy-preserving two-factor authentication scheme for IoT devices, where physically uncloneable functions have been considered as one of the authentication factors. Security and performance analysis show that our proposed scheme is not only robust against several attacks, but also very efficient in terms of computational efficiently.
14	**Boot-IoT: A Privacy-Aware Authentication**	The Internet of Things (IoT) devices perform security-critical operations and deal with sensitive information in the IoT-based systems. Therefore, the increased deployment

CONTINÚA

Scheme for Secure Bootstrapping of IoT Nodes	of smart devices will make them targets for cyber attacks. Adversaries can perform malicious actions, leak private information, and track devices' and their owners' location by gaining unauthorized access to IoT devices and networks. However, conventional security protocols are not primarily designed for resource constrained devices and therefore cannot be applied directly to IoT systems. In this paper, we propose Boot-IoT - a privacy-preserving, lightweight, and scalable security scheme for limited resource devices. Boot-IoT prevents a malicious device from joining an IoT network. Boot-IoT enables a device to compute a unique identity for authentication each time the device enters a network. Moreover, during device to device communication, Boot-IoT provides a lightweight mutual authentication scheme that ensures privacy-preserving identity usages. We present a detailed analysis of the security strength of BootIoT. We implemented a prototype of Boot-IoT on IoT devices powered by Contiki OS and provided an extensive comparative analysis of Boot-IoT with contemporary authentication methods. Our results show that Boot-IoT is resource efficient and provides better scalability compared to current solutions.

CONTINÚA

15	**Dynamically Configurable Architecture for User Identification and Authentication for Internet of Things Platform**	With the significant improvement of the deployment of Internet of Things (IoT) into the smart grid infrastructure and the increasing number of connected devices as well as users through the internet, the demand for cyber security is rapidly growing. Which is at the same time producing an acute threat of identity theft, forgery and breach of information security. In order to provide security to every user and device within IoT platform, advance identification and authentication techniques have been investigated by information security users for decades. In this paper, we propose a dynamically configurable connected framework and supporting algorithm which is capable to identify threat and malicious attack through authenticating user or device's identity, controlling service access and establishing trust between object and cloud services. The major contribution of this research work includes a dynamically configurable system framework that is capable of ensuring identification and authentication to every different type of devices within an IoT platform, a scalable system architecture that supports both limited-resource and ensemble devices. Another significant contribution of this work is multi-category security checking

CONTINÚA

		that address different aspects of IoT identification and authentication problem. Our proposed system is an application-layer based technology hence, easy to implement and deploy on edge devices and finally we also have presented a comparative discussion with existing approaches.
16	**Security of message queue telemetry transport protocol**	In this study, the Internet of things is defined and lightweight protocols that the devices can use to communicate with each other are specified, and message queue telemetry transport (MQTT) protocol, which is designed to be used for restricted devices, is detailed. In the scope of this paper, the studies about the authentication of the objects and the confidentiality of the data have been examined, some suggestions have been made against the identified security vulnerabilities. In the scope of this paper, as the test environment, Mosquitto, the server of the MQTT protocol, and MongoDB in order to store the data were installed. In the prepared test environment, the objects were communicated with each other via the MQTT protocol and identity verification and security vulnerabilities were tried to be detected.

CONTINÚA

17	**NFC Secure Element-Based Mutual Authentication and Attestation for IoT Access**	Certain resourceful and powered Internet of Things (IoT) can become victims to launch cyber attacks. Near field communication (NFC) can be used for their secure on-demand access. In this paper, we present a novel framework for the NFC secure element (SE)-based mutual authentication and attestation for IoT access with a user device such as a mobile device using NFC-based Host Card Emulation (HCE) mode for the first time. HCE is robust as compared to the other NFC modes. A cloud-based Trusted Certified Authority (TCA) manages all cryptographic credentials and stores them in the tamper-resistant SE and Trusted Platform Module (TPM)-based attestation modules on the devices. It uses a newly proposed NFC SE-based mutual authentication and attestation (NSE-AA) protocol for proof-of-locality, end-to-end anonymous mutual authentication between the SEs and an associated remote attestation for trust. The protocol is robust and lightweight as compared to the existing schemes. We provide its informal and formal security analysis using the Real-Or-Random (ROR) model. A simulation on the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool proves

CONTINÚA

		its safety. We also briefly present the details of a prototype with a commercial mid-range priced mobile device and Single Board Computer (SBC)-based IoT device.
18	**Hybrid approach for securing IoT communication using authentication and data confidentiality**	Internet of Things (IoT) is a new emerging technology of the Internet. Traditionally, Internet has been connecting people and sharing information through human intervention. IoT is an extension of internet, whose aim is to provide machine to machine (M 2M) or object to object (O2O) communication without human intervention. It independently has technical knowledge of the system in existing environment which is leading towards “communication era of entities” but there are various security issues and challenges such as privacy, data protection, DOS attacks, dealing with the bulky data collected by IoT devices, increasing transparency, access control, authentication and many more whose solution need to be addressed. This work acquainted the hybrid approach for securing IoT communication using authentication and data confidentiality to address the vulnerabilities in IoT network.
19	**Authentication and**	Internet of things (IoT) allows devices especially with low process capability and power consumption, to transmit

CONTINÚA

Authorization Mechanism on Message Queue Telemetry Transport Protocol	data to each other using various communication technologies such as wired, wireless network and radio frequency. It also makes easier to access these data from anywhere by storing them in the cloud or database. Nowadays, IoT applications in diverse areas including sensitive personal information such as especially health, financial, industry have become widespread. When huge number of devices with limited resources are connected to IoT application, provided security gains significant importance to ensure the integrity, confidentiality, accessibility of these data. In addition, the availability of a variety of specialized devices and communication technologies demonstrate the hassle of providing a standard security mechanism. When device features with their low process capability and power consumption are taken into account, message queue telemetry transport (MQTT) is the most appropriate lightweight communication protocol. In this study, the MQTT security is defined, and preliminary work related to MQTT on basic security issues such as privacy, authentication and access control is examined. This study is based on the previous work that is interested in open authorization (OAuth 2.0) protocol,

CONTINÚA

		which is recommended to gain authorization. In this study, in addition to the OAuth token, authentication is performed in two steps using a HMAC-based one-time password (HOTP) due to its short life span. Since MQTT protocol does not have bidirectional authentication other than using transport layer security (TLS), mutual authentication is provided by using one-time password (OTP) with hash chain. Advanced encryption standard (AES) is used for providing confidentiality to prevent against potential security vulnerabilities. Finally, security analysis has been discussed by giving an alternative solution by using these methods against selected attacks.
20	**New Security Level of Authentication and Key Agreement Protocol for the IoT on LTE Mobile Networks**	Nowadays, systems communications have evolved greatly and creating a new ecosystem with appearance of a high variety of objects, which are able to communicate with each other. The convergence of the cellular networks is raising a new concept called Machine-to-Machine (M2M) communication, which are the most popular application for the Internet of Things (IoT). The future IoT causes different challenges among them we can cite the security of communication and privacy of users. The standard authentication and key agreement protocol (EPS-AKA)

CONTINÚA

		normalized by the 3rd Generation Partnership Project (3GPP) in Long Term Evolution (LTE) system is proposed to provide the authentication service between the device and the network. However, this protocol does not support IoT objects and has some issues, including security attack vulnerabilities and traffic overload due to control message. In this article, we propose a new security level of authentication protocol that satisfies the security requirements by providing protection against several attacks, reducing bandwidth consumption by optimizing the authentication messages and secures an efficient communications among various IoT devices. Simulation results showed that the proposed when compared to existing protocols offers better performances in terms of security, overhead and computations delay.
21	**Framework for authentication and access control in IoT**	The framework of Internet of Things (IoT) helps to establish connection between people and things. Regarding the heterogenous technologies that characterizes the nature of IoT, the internet of things embraces many emerging products and services in several areas. According to the innovative products and services that IoT provides, it is required to ensure its

CONTINÚA

		security that includes authentication and access control due to its interconnectivity of things such as people, devices and organizations. This scenario of interconnectivity may present many constraints and limitations according to people, devices, components and power resources. About the vulnerabilities of IoT, a well-defined framework and techniques need to be designed to establish the security of IoT. We have proposed in this paper an authentication and access control framework in IoT.
22	**Study on access control approaches in the context of Internet of Things: A survey**	Internet of things (IoT) has brought new challenges to traditional access control models and protocols, since it exposes information that could be private or sensitive. This survey paper demonstrates a number of attacks that threaten that information in the context of internet of things as well as some of the access control approaches that try to improve security, authentication, and trust between internet of things devices and any internet host. It will provid some proposed delegation methods of authority techniques and presents the methods which are designed to address problems in web-based services based on IT. This survey will help the researchers in finding an

CONTINÚA

		improved models or systems for access control in internet of things based on the provided information which concentrates on a number of approaches that try to solve the challenges of access control in IoT context from different aspects.
23	**Detection and Prevention of Routing Attacks in Internet of Things**	Internet of things (IoT) is the smart network which connects smart objects over the Internet. The Internet is untrusted and unreliable network and thus IoT network is vulnerable to different kind of attacks. Conventional encryption and authentication techniques sometimes fail on IoT based network and intrusion may succeed to destroy the network. So, it is necessary to design intrusion detection system for such network. In our paper, we detect routing attacks such as sinkhole and selective forwarding. We have also tried to prevent our network from these attacks. We designed detection and prevention algorithm, i.e., KMA (Key Match Algorithm) and CBA (Cluster- Based Algorithm) in MatLab simulation environment. We gave two intrusion detection mechanisms and compared their results as well. True positive intrusion detection rate for our work is between

CONTINÚA

		50% to 80% with KMA and 76% to 96% with CBA algorithm.
24	**Improving Security on IoT Applications Based on the FIWARE Platform**	Internet of Things (IoT) has increased its presence in many environments. However, this means a greater exposure of sensitive data, rising potential security threats. Thus, security becomes a key requirement for the protection and prevention of cyber attacks to IoT applications and devices. FIWARE Platform, for example, has an architecture of components responsible for interconnecting devices to IoT applications, decreasing complexity and providing a standard set of services to developers. The IDAS 5 version of the platform presents several security gaps, so this work aims to solve some of them by incorporating end-to-end security services using encryption and access control in all NGSI requests (RESTFul API). The main contribution of this paper is the implementation of the DTLS 1.2 protocol in NodeJs to support the LWM2M/CoAP protocol and its addition to the IoT Agent. This paper also allowed the IoT Agent of the FIWARE Platform support TLS communication to the MQTT protocol. Through an experimental evaluation, it was possible to validate the implementation. Our

CONTINÚA

		preliminary results show that the encrypted requests had a small increase regarding latency, but this cost is compensated by the increase of security in FIWARE based IoT Applications. The source code of this work is open on the GitHub and it can be used to support security services in other IoT communication protocols.
25	**Attack scenarios and security analysis of MQTT communication protocol in IoT system**	Various communication protocols are currently used in the Internet of Things (IoT) devices. One of the protocols that are already standardized by ISO is MQTT protocol (ISO / IEC 20922: 2016). Many IoT developers use this protocol because of its minimal bandwidth requirement and low memory consumption. Sometimes, IoT device sends confidential data that should only be accessed by authorized people or devices. Unfortunately, the MQTT protocol only provides authentication for the security mechanism which, by default, does not encrypt the data in transit thus data privacy, authentication, and data integrity become problems in MQTT implementation. This paper discusses several reasons on why there are many IoT system that does not implement adequate security mechanism. Next, it also demonstrates and analyzes how we can attack this protocol easily using several attack

CONTINÚA

		scenarios. Finally, after the vulnerabilities of this protocol have been examined, we can improve our security awareness especially in MQTT protocol and then implement security mechanism in our MQTT system to prevent such attack.
26	**Intelligent security framework for iot devices cryptography based end-to-end security architecture**	Internet of Thing (IoT) provide services by linking the different platform devices. They have the limitation in providing intelligent service. The IoT devices are heterogeneous which includes wireless sensors to less resource constrained devices. These devices are prone to hardware/software and network attacks. If not properly secured, it may lead to security issues like privacy and confidentiality. To resolve the above problem, an Intelligent Security Framework for IoT Devices is proposed in this paper. The proposed method is made up of (1) the light weight Asymmetric cryptography for securing the End-To-End devices which protects the IoT service gateway and the low power sensor nodes and (2) implements Lattice-based cryptography for securing the Broker devices/Gateway and the cloud services. The proposed architecture implements Asymmetric Key Encryption to share session key between the nodes and then uses this

CONTINÚA

		session key for message transfer This protects the system from Distributed Denial of Service Attacks, eavesdropping and Quantum algorithm attacks. The proposed protocol uses the unique Device ID of the sensors to generate key pair to establish mutual authentication between Devices and Services. Finally, the Mutual authentication mechanism is implemented in the gateway.
27	**Dynamic Authentication Protocol Using Self-Powered Timers for Passive Internet of Things**	Passive Internet of Things (IoT) like radio frequency identification (RFID) tags can be used to offer a wide range of services, such as object tracking or classification, marking ownership, noting boundaries, and indicating identities. While the communication link between a reader of the tag and the authentication server is generally assumed to be secure, the communication link between the reader and participating tags is mostly vulnerable to malicious acts. Many authentication protocols have been proposed in literature, however, they either are vulnerable to certain types of attacks or require prohibitively a large amount of computational resources to be implemented on a passive tag. In this paper, we present variants of a novel authentication protocol that can overcome the security flaws of previous protocols while being well suited to the

CONTINÚA

		computational capability of the tags. At the core of the proposed approach is our recently demonstrated self-powered timing devices that can be used for robust time-keeping and synchronization without the need for any external powering. The outputs of the timers are processed using a single hash function on the tag to produce tokens that continuously change with time, while being synchronized to tokens generated by the authentication server. The proposed protocol also incorporates margins of tolerance that make the authentication process robust to any deviations in the timer responses due to fabrication artifacts.
28	**Node-to-Node Secure Data Transmission Protocol for Low-power IoT Devices**	Through the internet and local networks, IoT devices exchange data. Most of the IoT devices are low-power devices, meaning that they are designed to use less electric power. To secure data transmission, it is required to encrypt the messages. Encryption and decryption of messages are computationally expensive activities, thus require considerable amount of processing and memory power which is not affordable to low-power IoT devices. Therefore, not all secure transmission protocols are low-power IoT devices friendly. This study proposes a secure

CONTINÚA

		data transmission protocol for low-power IoT devices. The design inherits some features in Kerberos and onetime password concepts. The protocol is designed for devices which are connected to each other, as in a fully connected network topology. The protocol uses symmetric key cryptography under the assumption of that the device specific keys are never being transmitted over the network. It resists DoS, message replay and Man-of-the-middle attacks while facilitating the key security concepts such as Authenticity, Confidentiality and Integrity. The designed protocol uses less number of encryption/decryption cycles and maintain session based strong authentication to facilitate secure data transmission among nodes.
29	**A certificate based authorization and protected application layer protocol for IoT**	Safety communication is very important in emerging intelligent constrained world. In order to enhance secure communication, it is essential to enhance the unreliable Constrained Application Protocol (CoAP) by adding compressed Datagram Transport Layer Security (DTLS) Protocol based on 6LoWPAN standard. The purpose of Compressed DTLS is to minimize the length of packet and to prevent from disjunction. Along with this, the certificate based authorization encryption mechanism is added for

CONTINÚA

		enforcing authorized access to keep away from Denial of Service (DoS) Attacks. Finding report illustrate that CoAPs with certification afford secure message communication between the dissimilar multi vendor surroundings in intelligent world and the impact of Denial of Service Attack is reduced.
30	**Key Distribution Protocol for Industrial Internet of Things Without Implicit Certificates**	The deployment of the Internet of Things (IoT) in industry, called the Industrial IoT (IIoT), is supporting the introduction of very desirable improvements, such as increasing production flexibility, self-organization, and real-time and quick response to events. However, security and privacy challenges are still to be well addressed. The IIoT requires different properties to achieve secure and reliable systems and these requirements create extra challenges considering the limited processing and communication power available to IIoT field devices. In this research article, we present a key distribution protocol for IIoT that is computationally and communicationally lightweight (requires a single message exchange) and handles node addition and revocation, as well as fast rekeying. The scheme can also resist the consequences of node capture attacks (we assume that captured nodes

CONTINÚA

		can be detected by the gateway and previous works have shown this assumption to be acceptable in practice), server impersonation attacks and provides forward/backward secrecy. We show formally the correctness of our protocol and evaluate its energy consumption under realistic scenarios using a real embedded platform compared to previous state-of-the-art key-exchange protocols, to show our protocol reliability for IoT.
31	**Mutual Authentication in IoT Systems Using Physical Unclonable Functions**	The Internet of Things (IoT) represents a great opportunity to connect people, information, and things, which will in turn cause a paradigm shift in the way we work, interact, and think. IoT devices are usually small, low cost, and have limited resources, which makes them vulnerable to physical, side-channel, and cloning attacks. Therefore, any protocol designed for IoT systems should not only be secure but also efficient in terms of usage of chip area, energy, storage, and processing. To address this issue, we present light-weight mutual authentication protocols for IoT systems based on physical unclonable functions. Protocols for two scenarios are presented, one when an IoT device and server wish to communicate and the other

CONTINÚA

		when two IoT devices want to establish a session. A security and performance analysis of the protocols shows that they are not only robust against different types of attacks, but are also very efficient in terms of computation, memory, energy, and communication overhead. The proposed protocols are suitable for real time applications and are an attractive choice for implementing mutual authentication in IoT systems.
32	**Survey on Internet of Things (IoT) security issues & solutions**	Internet of things (IOT) is a rising wireless technology that connects different things to the internet. Things can be "anything" that is around us. `Over 50 billion one-of-a-kind gadgets can be related to the IoT through 2020' said through Cisco. Contemporary security technologies are very constrained as they may be unable to fulfil safety requirements of IoT devices. As Cisco said, in coming years the growth of IoT can be progressive but at the equal time we need to offer a strong provision for security of user's data. In this paper, we can consciousness on specific vulnerabilities to the security of IoT devices and provide possible solution to them
33	**Secure Authentication**	In recent years, Internet of Things(IoT) gained popularity due to its enormous applications in many fields. IoT

CONTINÚA

	Protocol for IoT Architecture	network comprises heterogeneous devices to a great scale, which creates numerous security threats. In this paper, a smart home based on IoT architecture is considered, where the IoT smart Hub(ISH) communicate with the cloud in one hand, and home appliances and smart devices on the other. The IoT smart Hub(ISH) receives commands from the smart phone which is connected to the cloud through the internet, where lies the possibility of an external attack. This paper proposes a secure authentication protocol between smart phone and ISH, which supports ISH for ensuring security in the smart home scenario.
34	**Lightweight zero knowledge authentication for Internet of things**	Importance of Internet of Things technologies increased in recent years. However, these technologies carry some security vulnerabilities because of their network communication layer. The root cause of these vulnerabilities is usually the Authentication problem. Zero knowledge proof method is a strong cryptographic solution for Authentication problem that is proving of having a knowledge to another party without revealing anything other than the veracity of the statement. Zero knowledge proof method is consist of two-way complex mathematical

CONTINÚA

		algorithms for both parties. In this work, a new method that uses zero knowledge proofs has been proposed to provide efficient solution for Internet of Things technologies. New method was implemented and tested, then compared with existing proposed zero knowledge proof based authentication methods.
35	**Token-Based Lightweight Authentication to Secure IoT Networks**	The rapid growth of Internet of Things (IoT) technology offers huge opportunities and also brings many new challenges related to the authentication in (IoT) devices. Using passwords or pre-defined keys have drawbacks that limit their use for different (IoT) applications like smart hotel and smart office. In fact, they didn't provide temporary access to data in such reservation systems. Thus, authenticating users basing on password mechanism is not feasible. In this paper, we propose a new Token-Based Lightweight User Authentication (TBL UA) for (IoT) devices, which is based on token technique in order to enhance the robustness of authentication. Security analysis shows the security strength of the proposed scheme such as token security, Perfect Forward Secrecy (PFS), etc. In addition, the presented performance

CONTINÚA

		analysis shows that it is a strong competitor among existing ones for user authentication in (IoT) environments.
36	**IoT Security: ZWave and Thread**	This paper reviews the security aspects of two Internet-of-Things (IoT) protocols, Z-Wave and Thread. Z-Wave is one of the oldest and most commercially successful IoT protocol, while Thread is one of the most recent protocols. As millions of IoT systems are installed for home and industrial automation, the security of these IoT systems is a concern. There is a potential for these IoT systems to be misused. This paper looks at security challenges for an IoT system. The security features of the IoT systems are spread across many parts of the IoT protocols. Paper discusses different attacks types on the IoT systems and the manners in which the protocols handle them. One of the most venerable times for an IoT system is when a new device is added to the IoT system. To avoid an intruder from getting access, the new device must be authenticated. Authentication of new network devices is discussed for both the protocols. IoT protocols are complex and the security aspects are also very complex, this paper should serve as a starting point to further study these and other IoT protocols.

CONTINÚA

37	**Lightweight and Secure Authentication Protocol for the Internet of Things in Vehicular Systems**	Vehicular systems are an application of the internet of things (IoT) in which vehicles are equipped with sensors. In this system, sensors collect traffic information and send data to the nearest sink node. By analyzing this information special users, including police officers, can make better decisions. In this scenario, confidentiality and integrity of the information against active and passive attacks are vital. To provide these important security requirements in vehicular systems, researchers have proposed numerous authentication protocols. Recently, Mohit et al. proposed a new authentication protocol in vehicular systems and claimed that their protocol is secure against smartcard stolen attack, traceability attack and session key attack. However, in this paper we prove that their protocol is not only vulnerable against the aforementioned attacks but also it cannot preserve sensor node anonymity. Finally, we propose a new improved authentication protocol with better security; our experimental results show that our protocol not only is secure against the above attacks but also is still enough lightweight.
38	**Token-Based Security for the**	In this paper, token-based security protocols with dynamic energy-security level tradeoff for Internet of Things (IoT)

CONTINÚA

Internet of Things With Dynamic Energy-Quality Tradeoff	devices are explored. To assure scalability in the mechanism to authenticate devices in large-sized networks, the proposed protocol is based on the OAuth 2.0 framework, and on secrets generated by on-chip physically unclonable functions. This eliminates the need to share the credentials of the protected resource (e.g., server) with all connected devices, thus overcoming the weaknesses of conventional client–server authentication. To reduce the energy consumption associated with secure data transfers, dynamic energy-quality tradeoff is introduced to save energy when lower security level (or, equivalently, quality in the security subsystem) is acceptable. Energy-quality scaling is introduced at several levels of abstraction, from the individual components in the security subsystem to the network protocol level. The analysis on an MICA 2 mote platform shows that the proposed scheme is robust against different types of attacks and reduces the energy consumption of IoT devices by up to 69% for authentication and authorization, and up to 45% during data transfer, compared to a conventional IoT device with fixed key size.

CONTINÚA

39	**A review on security problems and measures of Internet of Things**	Internet of Things (IoT) plays a significant role as it includes traditional appliances as well as everyday household objects. The security problems of IoT are related to spacious applications of its system. There come several challenges, with tremendous potential of IoT. By presenting the security architecture and components of IoT, this paper analyzes the security problems that occur in the layers: perception layer, network layer and application layer, including the security measures for each layer separately. This paper also explains the affair of IoT security as whole and tries to find answer for them.
40	**SlimIoT: Scalable Lightweight Attestation Protocol for the Internet of Things**	The Internet of Things (IoT) is increasingly intertwined with critical industrial processes, yet contemporary IoT devices offer limited security features, creating a large new attack surface. Remote attestation is a well-known technique to detect cyber threats by remotely verifying the internal state of a networked embedded device through a trusted entity. Multi-device attestation has received little attention although current single-device approaches show limited scalability in IoT applications. Though recent work has yielded some proposals for scalable attestation, several aspects remain unexplored, and thus more research is

CONTINÚA

		required. This paper presents slimIoT, a scalable lightweight attestation protocol that is suitable for all IoT devices. slimIoT depends on an efficient broadcast authentication scheme along with symmetric key cryptography. It is resilient against a strong adversary with physical access to the IoT device. Our protocol is informative in the sense that it identifies the precise status of every device in the network. We implement and evaluate slimIoT considering many factors. On the one hand, our evaluation results show a low overhead in terms of memory footprint and runtime. On the other hand, simulations demonstrate that slimIoT is scalable, robust and highly efficient to be used in static and dynamic networks consisting of thousands of heterogenous IoT devices.
41	**Secure authentication on the Internet of Things**	This paper describes biometric-based cryptographic techniques for providing confidential communications and strong, mutual and multifactor authentication on the Internet of Things. The described security techniques support the goals of universal access when users are allowed to select from multiple choice alternatives to authenticate their identities. By using a Biometric

CONTINÚA 

		Authenticated Key Exchange (BAKE) protocol, user credentials are protected against phishing and Man-in-the-Middle attacks. Forward secrecy is achieved using a Diffie-Hellman key establishment scheme with fresh random values each time the BAKE protocol is operated. Confidentiality is achieved using lightweight cryptographic algorithms that are well suited for implementation in resource constrained environments, those limited by processing speed, limited memory and power availability. Lightweight cryptography can offer strong confidentiality solutions that are practical to implement in Internet of Things systems, where efficient execution, and small memory requirements and code size are required.
42	**An Unlinkable Authentication Scheme for Distributed IoT Application**	The Internet of Things (IoT) is an enormous ubiquitous-network, which connects the objects through various sensors. The IoT technology promotes the interconnection and fusion between the physical world and information space, and it facilitates the day-to-day life of people. However, since a lot of equipped sensors are unattended and open, the IoT must face and overcome the main problems of security and privacy. Authentication is one of the paramount security concerns in the IoT environment,

CONTINÚA

		in which a user could directly access data from the sensors. Therefore, we propose an authentication and key agreement scheme providing unlinkability for the IoT environment based on bilinear pairings. The formal security proof demonstrates that the proposed protocol is unforgeable under the adaptively chosen message attack, and the session key exchange is semantic secure under the eCK model. In addition, the computation and communication costs of the proposed scheme are evaluated and compared with some existing similar schemes, which exhibits that it pleasantly addresses the needs of the IoT as far as security properties and computation expenses.
43	**Securing the Internet of Things: A Meta-Study of Challenges, Approaches, and Open Problems**	The Internet of Things (IoT) is becoming a key infrastructure for the development of smart ecosystems. However, the increased deployment of IoT devices with poor security has already rendered them increasingly vulnerable to cyber attacks. In some cases, they can be used as a tool for committing serious crimes. Although some researchers have already explored such issues in the IoT domain and provided solutions for them, there remains the need for a thorough analysis of the

CONTINÚA

		challenges, solutions, and open problems in this domain. In this paper, we consider this research gap and provide a systematic analysis of security issues of IoT-based systems. Then, we discuss certain existing research projects to resolve the security issues. Finally, we highlight a set of open problems and provide a detailed description for each. We posit that our systematic approach for understanding the nature and challenges in IoT security will motivate researchers to addressing and solving these problems.
44	**Provably Secure ECC-Based Device Access Control and Key Agreement Protocol for IoT Environment**	For secure communication between any two neighboring sensing devices on the Internet of Things (IoT) environment, it is essential to design a secure device access control and key agreement protocol, in which the two phases, namely, “node authentication” and “key agreement” are involved. While the node authentication allows two sensing devices to authenticate each other using their own pre-loaded secret credentials in memory, the key agreement phase permits to establish a secret key between them if the mutual authentication is successful. In this paper, we propose a new certificate-based “lightweight access control and key agreement protocol in the IoT

CONTINÚA

		environment, called LACKA-IoT," that utilizes the elliptic curve cryptography (ECC) along with the "collision-resistant one-way cryptographic hash function." Through a detailed security analysis using the formal security under the "Real-Or-Random (ROR) model," informal (non-mathematical) security analysis, and formal security verification using the broadly used "Automated Validation of Internet Security Protocols and Applications (AVISPA)" tool, we show that the LACKA-IoT can protect various known attacks that are needed for a secure device access control mechanism in the IoT. Furthermore, through a comparative study of the LACKA-IoT and other relevant schemes, we show that there is a better tradeoff among the security and functionality features and communication and computational costs of the LACKA-IoT as compared to other schemes. Finally, the "practical demonstration using the NS2 simulation" has been carried out on the LACKA-IoT to measure various network parameters.
45	**Hardware based Two-Factor User Authentication**	In the distributed Internet of Things (IoT) architecture, sensors collect data from vehicles, home appliances and office equipment and other environments. Various objects contain the sensor which process data, cooperate and

CONTINÚA

	for the Internet of Things	exchange information with other embedded devices and end users in a distributed network. It is important to provide end-to-end communication security and an authentication system to guarantee the security and reliability of the data in such a distributed system. Two-factor authentication is a solution to improve the security level of password-based authentication processes and immunized the system against many attacks. At the same time, the computational and storage overhead of an authentication method also needs to be considered in IoT scenarios. For this reason, many cryptographic schemes are designed especially for the IoT; however, we observe a lack of laboratory hardware test beds and modules, and universal authentication hardware modules. This paper proposes a design and analysis for a hardware module in the IoT which allows the use of two-factor authentication based on smart cards, while taking into consideration the limited processing power and energy reserves of nodes, as well as designing the system with scalability in mind.
46	**A Robust ECC-Based Provable Secure**	Wireless sensor networks (WSNs) play an important role in the industrial Internet of Things (IIoT) and have been widely used in many industrial fields to gather data of

CONTINÚA

	Authentication Protocol With Privacy Preserving for Industrial Internet of Things	monitoring area. However, due to the open nature of wireless channel and resource-constrained feature of sensor nodes, how to guarantee that the sensitive sensor data can only be accessed by a valid user becomes a key challenge in IIoT environment. Some user authentication protocols for WSNs have been proposed to address this issue. However, previous works more or less have their own weaknesses, such as not providing user anonymity and other ideal functions or being vulnerable to some attacks. To provide secure communication for IIoT, a user authentication protocol scheme with privacy protection for IIoT has been proposed. The security of the proposed scheme is proved under a random oracle model, and other security discussions show that the proposed protocol is robust to various attacks. Furthermore, the comparison results with other related protocols and the simulation by NS-3 show that the proposed protocol is secure and efficient for IIoT.
47	**Security with IP Address Assignment and Spoofing for**	The widespread use of internet is causing more cyber-attacks using IP spoofing. The security for IoT devices to prevent IP spoofing involves validating the source address of received IP packets at the gateway. This is required to

CONTINÚA

	Smart IOT Devices	prevent a unauthorized user from using IP address as source address and flooding packets to the gateway there by using the bandwidth allocated to authorized users. This paper proposes a scheme for IP address assignment to smart IoT devices (which communicate using TCP/IP) and validation of source IP address in the received IP packets from the IoT Device at the Gateway device.
48	**Challenges in security in Internet of Things**	Nowadays, Internet of Things is one of the future developing directions. Some of the most important challenges it faces are related to security. Several massive distributed denial of service attacks have been deployed in the last months using “smart” devices, such as baby monitors or refrigerators. Connecting electronic devices without considering security can lead to severe problems. Using encryption and authentication, the devices can check if they are allowed to trust a remote system. The code that runs should be authorized, to avoid malicious applications and the devices should be permanently updated with security patches. Traffic analyzers are also part of IoT security, for monitoring patterns and be able to find abnormal traffic.

CONTINÚA

49	**An End-to-End View of IoT Security and Privacy**	In this paper, we present an end-to-end view of IoT security and privacy and a case study. Our contribution is twofold. First, we present our end-to-end view of an IoT system and this view can guide risk assessment and design of an IoT system. We identify 10 basic IoT functionalities that are related to security and privacy. Based on this view, we systematically present security and privacy requirements in terms of IoT system, software, networking and big data analytics in the cloud. Second, using the end-to-end view of IoT security and privacy, we present a vulnerability analysis of the Edimax IP camera system. We are the first to exploit this system and have identified various attacks that can fully control all the cameras from the manufacturer. Our real- world experiments demonstrate the effectiveness of the discovered attacks and raise the alarms again for the IoT manufacturers.
50	**Mitigating DoS attacks in publish-subscribe IoT networks**	Internet of Things (IoT) is an emerging subject which enables multiple applications and requires robust security solutions. IoT architectures contribute to critical aspects of our society like transportation, health-care, industry, telecommunications and many others. Security is difficult to be implemented in the IoT context because the

CONTINÚA

	embedded devices are resource constrained and deployed in uncontrolled areas, thus being targeted by various security attacks. In scenarios like smart city, IoT networks are populated by both trusted and untrusted transient devices, thus mechanisms for mitigating complex security attacks must be implemented. Embedded networks with near real-time constraints are subject to denial-of-service attacks which can easily affect the applications availability by injecting malicious packets into the network. This paper proposes a lightweight security mechanism which addresses DoS attacks in IoT publish-subscribe applications.

3.2.5. Síntesis de resultados

Una vez realizado el proceso de selección de artículos y considerando que estos constituyen la base para responder las preguntas de investigación planteadas, se seleccionaron los siguientes artículos primarios.

EP1: Analysis of authentication techniques in Internet of Things (IoT) (Chamoun, El-hajj, Fadlallah, & Serhrouchni, 2017).

Este artículo se centra en la seguridad de la información IoT tales como la integridad de los datos, la autenticación, el control de acceso, la privacidad, etc…, Su contenido está enfocado más a la autenticación de los dispositivos IoT, por lo que el

documento es una fuente importante en lo que respecta al análisis y comparación de los protocolos de autenticación existentes.

EP2: Security Vulnerabilities of Internet of Things: A Case Study of the Smart Plug System (Zhen, et al., 2017).

En este documento los autores hablan sobre la vulnerabilidad de los dispositivos inteligentes, mediante ataques de escaneo de dispositivos, ataque de fuerza bruta, ataque de suplantación de identidad y ataque de firmware, lo que se intenta demostrar es que al realizar dichos ataques se pueden obtener las credenciales de los usuarios. Además, proponen normas para asegurar los dispositivos inteligentes.

EP3: An Authentication Technique 87ase don Oauth 2.0 Protocol for Internet of Things (IoT) Network (Jalaluddin, et al., 2018).

El contenido del artículo propone un esquema de autenticación basado en el protocolo OAuth 2.0 donde se busca asegurar el acceso a la red mediante la comparación y autorización del usuario y los tokens de acceso almacenados en la base de datos, este artículo también hace referencia a que mantiene la red IoT segura de ataques como: suplantación de identidad y repetición de ataques.

EP4: An Improved User Authentication Protocol for IoT (Jianming, Zuowen, Hengzhong, & Rongquan, 2018).

Este estudio está basado en los protocolos de autenticación diseñados para combatir amenazas de seguridad y privacidad en IoT, sin embargo, en un protocolo de autenticación propuesto por Li et al se encontró que no puede resistir al ataque de DoS,

al ataque de verificado robado y al ataque de repetición. Los autores de este estudio proponen un protocolo donde se aumenta la verificación de la identidad del usuario y password permitiendo de esta manera resistir el ataque DoS.

EP5: Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices (Gope & Sikdar, 2018).

En este artículo se aborda temas referentes a la implementación de dispositivos IoT en lugares públicos y privados, convirtiéndose en fuentes vulnerables para ataques físicos y de clonación. Para resolver dichos ataques se presenta un esquema de autenticación mediante dos factores, que sea ligero y que preserve la privacidad de los dispositivos IoT.

EP6: Authentication and Authorization Mechanism on Message Queue Telemetry Transport Protocol (Yerlikaya & Dalkilic, 2018).

Lo que se pretende en este artículo es garantizar la seguridad de integridad, confidencialidad y accesibilidad a los datos de las diversas áreas en la que se puede implementar un ecosistema IoT. En este artículo también se propone utilizar MQTT para la comunicación, Oauth 2.0, tokens y HMAC para la autorización, y AES para proporcionar confidencialidad y prevenir posibles vulneraciones de seguridad.

EP7: Framework for authentication and access control in IoT (Bate, Kumar, & Khatri, 2017).

Debido a la interconectividad que existe entre personas, dispositivos y organizaciones se requiere la garantizar la seguridad de autenticación y control de acceso

en productos y varias áreas que abarca el IoT. Los autores en este documento proponen un framework de autenticación y control de acceso en IoT.

EP8: Study on access control approaches in the context of Internet of Things: A survey (Al-Halabi, Raeq, & Abu-Dabaseh, 2017).

Este artículo hace referencia a que debido al constante crecimiento del IoT, los protocolos y modelos de acceso tradicionales ya no son suficientes para asegurar la privacidad de la información, se mencionan también varios ataques y las posibles mejoras que intentan combatir la seguridad de autenticación y la confianza entres dispositivos IoT mediante métodos diseñados para abordar problemas basados en la web.

EP9: Mutual Authentication in IoT Systems Using Physical Unclonable Functions (Aman, Chua, & Sikdar, 2017).

Al ser algunos dispositivos IoT pequeños, de bajo costo y de recursos limitados se convierten en recurso vulnerables para ataques físicos y de clonación, por lo tanto, cualquier protocolo diseñado para ecosistema IoT debe brindar seguridad. En este estudio se presentan protocolos de autenticación mutua es decir cuando un dispositivo desea comunicarse con un servidor de IoT y también cuando un dispositivo desea comunicarse con otro dispositivo IoT, estos protocolos propuestos son adecuados para implementar la autenticación en sistemas IoT.

EP10: A review on security problems and measures of Internet of Things (Verma & Chahal, 2017).

En este documento se realiza un análisis de los problemas y medidas de seguridad que se presentan en la capa de percepción, capa de red, y capa de aplicación que deben tomarse en cuenta en los sistemas IoT. Aquí también se busca una respuesta a la explicación de la seguridad de IoT como un todo.

EP11: Securing the Internet of Things: A Meta-Study of Challenges, Approaches, and Open Problems (Hossain, Hasan, & Skjellum, 2017).

La mayoría de las implementaciones de dispositivos IoT con poca seguridad hace que un ecosistema inteligente se convierta en una fuente vulnerable para ataques cibernéticos. En este artículo se proporciona un análisis sistemático de los problemas de seguridad de los ecosistemas IoT. Haciendo referencia a ciertos proyectos de investigación existentes se busca también resolver ciertos problemas de seguridad.

EP12: Challenges in security in Internet of Things (Florea, Ruse, & Rughinis, 2017).

En este artículo se menciona que dentro de los desafíos que se enfrenta el IoT está inmerso la seguridad, especialmente la vulnerabilidad a los ataques de denegación de servicio (DoS), que pueden causar graves problemas. Lo que se propone es que mediante la autenticación y autorización a un ecosistema IoT se reduzca el ataque de aplicaciones maliciosas.

3.3. Respuestas a las preguntas de investigación

RQ1: ¿Qué características poseen los protocolos de seguridad para dispositivos IoT?

A continuación, se describe las características de los protocolos, a los que los autores hacen referencia en los artículos analizados.

Tabla 5
Características protocolos de seguridad para dispositivos IoT

Protocolo	Descripción
Datagram Transport Layer Security (DTLS)	Este protocolo proporciona privacidad en la comunicación, evita los accesos no autorizados, previene la interrupción y manipulación de información, este protocolo usa User Datagram Protocol (UDP) para la transmisión.
Constrained Application Protocol (CoAP)	Es un protocolo de transferencia de datos, diseñado para aplicaciones máquina a máquina (M2M), este protocolo usa User Datagram Protocol (UDP).
Message Queue Telemetry Transport (MQTT)	MQTT es un protocolo de comunicación de máquina a máquina (M2M), debido a que el consumo de ancho de banda es bajo este puede ser utilizado en dispositivos IoT con bajos recursos.
Oauth 2.0	Su principal objetivo es la autorización para acceder a un recurso, este protocolo propone varios flujos para cumplir con su propósito.
Secure Network Encryption Protocol (SNEP)	Este protocolo garantiza la integridad de la información que se transmite a través de una red, su

CONTINÚA

	finalidad es evitar que cualquier actor no autorizado pueda acceder a la red.
Smart Space Access Protocol (SSAP)	Es un protocolo de mensajería que permite una comunicación entre le cliente (KP) y el IoT Broker (SIB), este protocolo está basado en json por lo que se convierte en un protocolo ligero.
Transport Layer Security (TLS)	Este protocolo es similar a DTLS, mejor dicho, es la base del DTSL, la diferencia radica en que este protocolo usa Transmission Control Protocol (TCP) para la transmisión.

RQ2: ¿Qué tipos de técnicas existen para vulnerar la autenticación de los dispositivos IoT?

En los artículos analizados, los autores detallan las siguientes técnicas para vulnerar la autenticación de los dispositivos IoT.

- Brute Forcé attack
- DoS / DdoS
- Device scanning attack
- Impersonation
- Man in the Middle attack
- Node capture

- Phishing
- Replay attack
- Side-channel attack
- Stolen-verifier attack
- Timing attack

RQ3: ¿Qué características poseen las técnicas para vulnerar la autenticación de los dispositivos IoT?

En la pregunta anterior se menciona varias técnicas de vulneración de la autenticación de dispositivos IoT, a continuación, se describe las características de dichas técnicas.

Tabla *6*
Características de las técnicas para vulnerar la autenticación

Técnica	Descripción
Brute Forcé attack	En esta técnica de ataque, se pretende vulnerar la seguridad enviando varias contraseñas con el propósito de descubrir cuál es la contraseña real.
DoS / DdoS	El objetivo principal de esta técnica es inhabilitar cualquier recurso dentro de una red, es decir, da de baja al recurso para que no se pueda hacerse ningún tipo de solicitud, la principal fuente que alimenta esta técnica es los replayes attacks.

CONTINÚA

Device scanning attack	Este tipo de ataque esta centralizado al escaneo de puertos a los cuales los dispositivos se encuentran conectados.
Impersonation	Es un ataque en el que un oponente busca robar una parte de información legitima dentro de un sistema o protocolo, con la finalidad de tener acceso mediante la información robada.
Man-in-the-Middle attack	Acción donde el atacante actúa como un proxy entre dos actores (persona-persona, persona-dispositivo, dispositivo-dispositivo) con el objetivo de interceptar y obtener información enviada entre los actores.
Node capture	Esta técnica está orientada a los dispositivos físicos, debido a que el atacante puede manipular un dispositivo (nodo) causando problemas de seguridad.
Phishing	En esta técnica el objetivo es engañar a la víctima, para que, a través de medios digitales o llamadas telefónicas, esta pueda otorgar sus credenciales.
Replay attacks	Está técnica de ataque ocurre cuando el atacante logra robar parte de la información válida y esta misma información es enviada varias veces dentro de otras partes de la información, causando una DoS.

CONTINÚA

Side-channel attack	El atacante intenta descubrir cierta información, analizando los dispositivos físicos implementados en un ecosistema IoT.
Stolen-verifier attack	Con esta técnica el atacante trata de obtener la llave secreta (simétrica o asimétrica) con que cierta información fue encriptada para ser transmitida entre un usuario y un servidor.
Timing Attack	Técnica en la que se trata de obtener datos clave de acuerdo al cálculo de tiempo que se utilizó para el cifrado de dichos datos.

RQ4: ¿Bajo qué criterios pueden clasificarse las técnicas para vulnerar la autenticación de los dispositivos IoT?

En el análisis de los artículos, la clasificación de las técnicas para vulnerar la autenticación de los dispositivos IoT están dadas de la siguiente forma.

Tabla 7
Clasificación de las técnicas para vulnerar la autenticación

Criterio de clasificación	Descripción
Daño de la información	**Interrupción:** Afecta a la disponibilidad de un servicio, degradando la calidad de los servicios o hace que los mismos no se encuentren disponibles.

CONTINÚA

	Escuchas: El atacante entra sin autorización a un canal de comunicación y puede captar la información que se está transmitiendo. **Modificación:** Se crea una confusión en la información que viaja a través de un canal de comunicación debido a que el atacante desea alterar esta información.
Estrategia	**Físicos:** Se produce cuando las propiedades o configuraciones de un dispositivo son cambiados. **Lógicos:** Estos no causan ningún daño físico al dispositivo, pero de alguna manera hacen que su funcionamiento no sea el adecuado.
Nivel de acceso	**Activo:** Se interrumpe la funcionalidad de un dispositivo o de la red, debido a la DoS o la interferencia entre recursos. **Pasivo:** El dispositivo autorizado pretende realizar actividades ilegales para recopilar información de un canal de comunicación.
Severidad	**Alta:** Son los ataques que pueden comprometer a un ecosistema IoT completamente. **Media:** Son ataques que tiene un compromiso parcial en un ecosistema IoT, está destinado a modificar los permisos, pero nunca obtendrá el control completo.

CONTINÚA

	Baja: Son ataques que no afectan a la disponibilidad de los servicios dentro de un ecosistema IoT.
Ubicación de adversario	**Interna:** El dispositivo víctima y el adversario autorizado se encuentran en una misma red. **Externa:** El dispositivo víctima y el adversario se encuentran en redes distintas que al violentar el sistema de autorización puede causar daño a los dispositivos IoT.

RQ5: ¿Qué tipos de técnicas existen para mitigar la vulnerabilidad de la autenticación de los dispositivos IoT?

En los artículos estudiados, los autores citan las siguientes técnicas para mitigar la vulnerabilidad de la autenticación de los dispositivos IoT.

- Basada en tokens
- Código de autorización
- Credenciales
- Hashing
- Symmetric key
- Asymmetric key
- One Time Password

RQ6: ¿Qué características poseen las técnicas para mitigar la vulnerabilidad la autenticación de los dispositivos IoT?

De las técnicas mencionadas en la pregunta anterior, a continuación, se describe las características.

Tabla 8

Características de las técnicas para mitigar las vulnerabilidades de la autenticación

Técnica	Descripción
Basada en tokens	Esta técnica se basa en encriptar cierta información con la que se busca obtener una autorización para acceder a determinados recursos.
Código de autorización	Esta técnica busca acceder a un recurso después de que un actor le envió un código de autorización.
Credenciales	Podemos considerar a esta técnica como la más básica, ya que, la autenticación se la realiza mediante las credenciales del usuario.
Hashing	Técnica basada en algoritmos matemáticos para proteger la información, en este caso sería utilizada para proteger el password.
Symmetric key	Son algoritmos criptográficos que utilizan una misma clave para la encriptación y desencriptación de la información.
Asymmetric key	Son algoritmos criptográficos que mediante una clave privada se encripta la información y la desencriptación se la realiza mediante la clave pública. Es decir, una clave asimétrica es una clave compuesta por una parte publica y una privada.

CONTINÚA

One Time Password	OTP, es una técnica donde, el password es utilizado una única vez para la autenticación, es decir, la autenticación con el mismo password no se logrará hacer nuevamente.

RQ7: ¿Bajo qué criterios pueden clasificarse las técnicas para mitigar la vulnerabilidad de la autenticación de los dispositivos IoT?

La clasificación de los artículos, están dados de acuerdo con los siguientes criterios propuestos por los autores.

Tabla 9

Clasificación de las técnicas para mitigar las vulnerabilidades de la autenticación

Criterio de clasificación	Descripción
Capas IoT	Se puede decir que aquí se concentran todas las técnicas, debido a que, estás técnicas se encuentran distribuidas en la capa de percepción, la capa de red y la capa de aplicación, cubriendo por completo un ecosistema IoT.
Entidad Certificada	Este tipo de técnicas hace referencia al tipo de comunicación y de autenticación que existe entre los actores, un ejemplo puede ser one-way, two-way.
Factor	Es el tipo de seguridad que se va a utilizar para acceder al recurso, un ejemplo de lo que se menciona puede ser, encriptar cierta información utilizando claves simétricas o asimétricas.

CONTINÚA

Metodología	Las técnicas están definidas de acuerdo con la arquitectura que se utilice para la autenticación como pueden ser distribuida y centralizada. Decimos que es distribuida cuando la autenticación se la hace con alguna aplicación de terceros, y centralizada cuando la autenticación se la realiza directamente con la base de datos.
Token	En este tipo de técnicas el acceso a ciertos recursos está dado por un algoritmo de encriptación, la verificación del token es el punto principal para que exista una autorización y se pueda acceder a dichos recursos.

RQ8: ¿Es posible desarrollar un framework que asocie las características más importantes de las técnicas existente para mitigar la vulnerabilidad de la autenticación de los dispositivos IoT, contemplando para su diseño las técnicas existentes de vulneración?

La posibilidad de desarrollar un framework que enlace las características más relevantes de las técnicas encontradas y considere los problemas de las técnicas de vulneración es viable, pero el framework depende del ambiente al cual se lo plantee. Se logró evidenciar que existen varias técnicas para vulnerar la autenticación de los dispositivos IoT y son efectivas en diferentes entornos, cada una de estas técnicas pueden ser tomadas en cuenta al momento de desarrollar el framework. Al igual que las

vulnerabilidades, para el desarrollo de framework también se tomará en cuenta las técnicas de mitigación más importantes que se adapten con el tema del proyecto propuesto.

CAPÍTULO IV

ANÁLISIS Y DISEÑO

4.1. Análisis de la situación actual

4.1.1. Requisitos no funcionales

A continuación se realiza una descripción de los requisitos no funcionales para el framework de seguridad.

Tabla 10
Requisitos no funcionales

No	Requisito no funcional
1	Se requiere de una conexión a internet para poder realizar el proceso de autenticación.
2	Garantizar la confiabilidad y el buen desempeño en el manejo de la concurrencia para todos los usuarios del framework.
3	Capacitación a los usuarios para la buena gestión de sus credenciales.

4.1.2. Diagrama de casos de uso

Para definir las acciones o funcionalidad del framework de seguridad, se va a utilizar uno de los diagramas del Leguaje Unificado de Modelado, en este caso el diagrama de casos de uso.

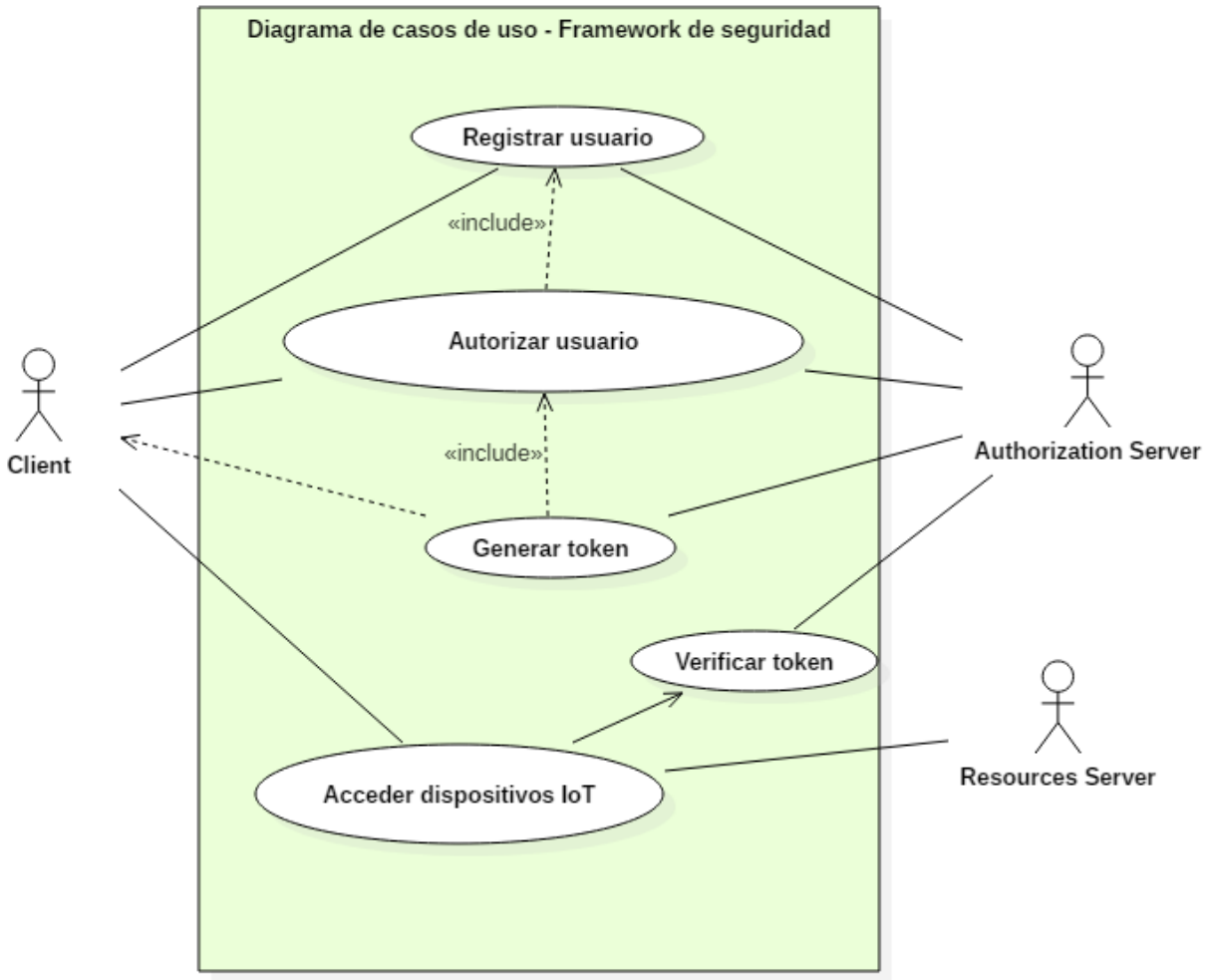


Figura 19. Diagrama casos de uso framework de seguridad

4.1.3. Descripción de casos de uso

Tabla 11

Caso de uso registrar usuario

Nombre:	Registrar usuario
Actores:	Client, Authorization Server
Descripción:	Registra las credenciales de usuario, en este caso el email y la contraseña.
Precondiciones:	

CONTINÚA

Flujo:	• Ingresar credenciales de usuario. • Enviar credenciales de usuario al servidor de autorización desde el cliente • Validar existencia de usuario. • Registrar usuario.
Postcondiciones:	Notificar el registro de usuario
Observaciones:	La contraseña es almacenada criptográficamente.

Tabla 12

Caso de uso autorizar usuario

Nombre:	Autorizar usuario
Actores:	Client, Authorization Server
Descripción:	Validación de la existencia del usuario para poder generar un código de autorización.
Precondiciones:	Registrar credenciales de usuario
Flujo:	• Ingresar credenciales de usuario • Enviar credenciales de usuario al servidor de autorización desde el cliente • Validar existencia del email de usuario • Validar contraseña de usuario • Generar código de autorización si los datos son correctos, caso contrario notificar datos incorrectos

CONTINÚA

Postcondiciones:	Enviar al cliente el código de autorización o la notificación de los datos incorrectos
Observaciones:	El código de autorización será enviado al callback proporcionado por el cliente.

Tabla 13

Caso de uso generar token

Nombre:	Generar token
Actores:	Client, Authorization Server
Descripción:	Validación de la existencia de un código de autorización para la generación de un token de acceso y un token de actualización que serán enviados al cliente.
Precondiciones:	Obtener el código de autorización
Flujo	• Enviar el código de autorización al servidor de autorización desde el cliente • Verificar que el código de autorización sea correcto • Generar el token de acceso y token de actualización si el código de autorización es correcto, caso contrario notificar invalidez del código de autorización
Postcondiciones:	Enviar al cliente el token de acceso y el token de actualización o la notificación del código de autorización incorrecto.

CONTINÚA

Observaciones:	Tanto el token de autorización como el token de actualización serán enviados al callback proporcionado por el cliente.

Tabla 14

Caso de uso verificar token

Nombre:	Verificar token
Actores:	Authorization Server, Resources Server
Descripción:	A través de introspección el token de acceso que fue enviado por el cliente al servidor de recursos será verificado en el servidor de autorización si es válido o no.
Precondiciones:	Obtener token de acceso
Flujo	• Enviar el token de acceso al servidor de autorización desde el servidor de recursos • Verificar la validez del token de acceso • Si el token de acceso es correcto, se devolverá la información de los dispositivos IoT del usuario, si el token ha expirado o es erróneo se devolverá una advertencia indicando lo sucedido.
Postcondiciones:	Enviar al servidor de recursos los datos obtenidos
Observaciones:	

Tabla 15

Caso de uso acceder dispositivos IoT

Nombre:	Acceder dispositivos IoT
Actores:	Client, Resources Server
Descripción:	Una vez generado el token de acceso, se va a mostrar los dispositivos IoT a los cuales el usuario tiene acceso.
Precondiciones:	Verificación de token
Flujo	• Enviar el token de acceso al servidor de recursos desde el cliente • Obtener la validez del token de acceso a través del servidor de autorización • Obtener la información de los dispositivos IoT del usuario si el token es correcto, si el token ha expirado se enviará una notificación para que el usuario pueda hacer uso del token de actualización, caso contrario se notificará la invalidez del token.
Postcondiciones:	Enviar al cliente los dispositivos IoT a los cuales tiene acceso o las notificaciones por token expirado o token inválido.
Observaciones:	La generación de tokens de acceso se volverá un ciclo repetitivo ya que cada vez que un token expire se generará uno nuevo.

4.2. Desarrollo mediante metodología SCRUM

4.2.1. Roles

- **Product Owner:** Ing. Tatiana Gualotuña
- **Scrum Master:** Ing. Tatiana Gualotuña
- **Scrum Team:** David Valdivieso

4.2.2. Product Backlog

La priorización de las tareas dentro del product backlog están dadas en base a colores donde: el color naranja es prioridad alta, el color amarillo es prioridad media y el color verde es prioridad baja.

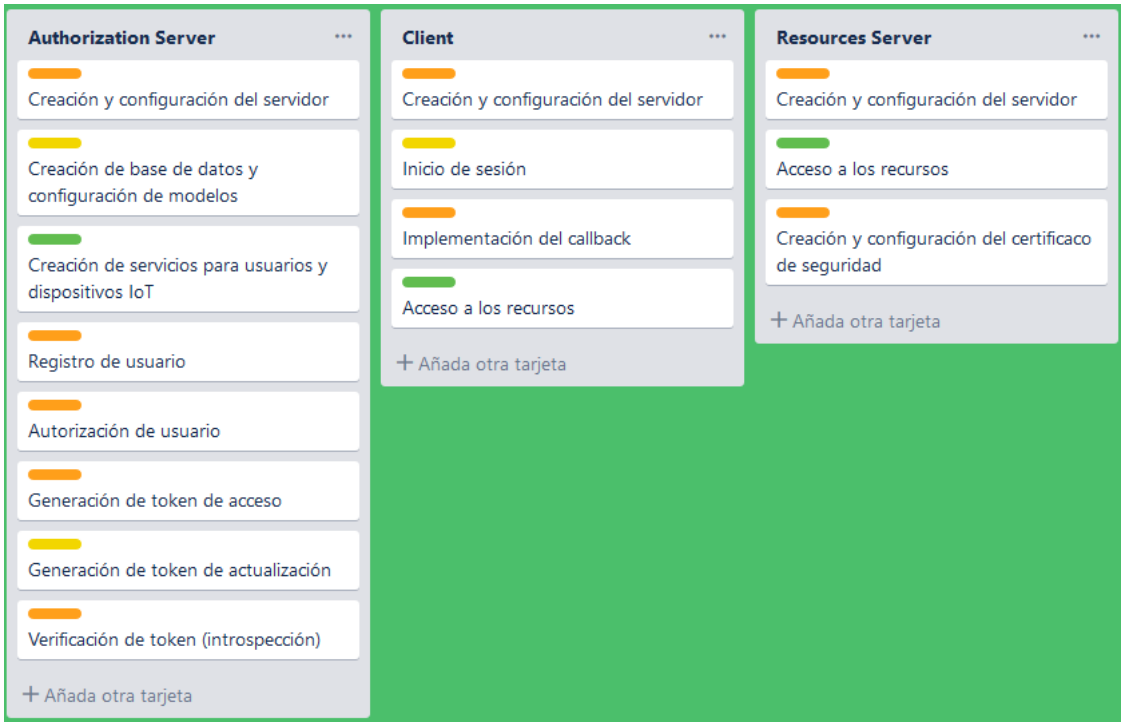


Figura 20. Product Backlog

Para la implementación del proyecto se plantea un tiempo de cuatro semanas, las cuales se van a distribuir en tres sprints. A continuación se presenta el detalle para cada uno de los sprints.

4.2.3. Planning Sprint 1: Authorization Server

- **Sprint Time:** 2 semanas
- **Sprint Backlog:**

Sprint 1: Authorization Server	**mié 01/05/19**	**mié 15/05/19**	
Creación y configuración del servidor	mié 01/05/19	mié 01/05/19	David Valdivieso
Creación de base de datos y configuracion de modelos	jue 02/05/19	jue 02/05/19	David Valdivieso
Creación de servicio para usuarios y dispositivos IoT	vie 03/05/19	dom 05/05/19	David Valdivieso
Registro de usuario	lun 06/05/19	mar 07/05/19	David Valdivieso
Autorización de usuario	mié 08/05/19	jue 09/05/19	David Valdivieso
Generación de tokens	**vie 10/05/19**	**mié 15/05/19**	
Token de acceso	vie 10/05/19	dom 12/05/19	David Valdivieso
Token de actualización	lun 13/05/19	lun 13/05/19	David Valdivieso
Verificación de token (intrsospección)	mar 14/05/19	mié 15/05/19	David Valdivieso

Figura 21. Sprint Backlog Authorization server

- **Entregable:** Como producto entregable de este sprint se tendrá la configuración de un servidor express denominado authorizathion_server, el cual corre en un protocolo HTTP, además de la implementación de APIs tipo REST, que ayudan al envío y recepción de paquetes entre los servidores.

4.2.4. Planning Sprint 2: Client

- **Sprint Time:** 1 semana
- **Sprint Backlog:**

Sprint 2: Client	jue 16/05/19	mié 22/05/19	
Creación y configuración del servidor	jue 16/05/19	jue 16/05/19	David Valdivieso
Inicio de sesión	vie 17/05/19	sáb 18/05/19	David Valdivieso
Implementación del callback	dom 19/05/19	lun 20/05/19	David Valdivieso
Acceso a los recursos	mar 21/05/19	mié 22/05/19	David Valdivieso

Figura 22. Sprint Backlog Client

- **Entregable:** Para este sprint el producto entregable será un servidor express denominado client corriendo en un protocolo HTTP, que a través de sus APIs entregará y recibirá información de las solicitudes realizadas a los otros servidores.

4.2.5. Planning Sprint 3: Resources Server

- **Sprint Time:** 1 semana
- **Sprint Backlog:**

Sprint 3: Resources Server	jue 23/05/19	lun 03/06/19	
Creación y configuración del servidor	jue 23/05/19	jue 23/05/19	David Valdivieso
Acceso a los recursos	vie 24/05/19	sáb 25/05/19	David Valdivieso
Creación y configuración de certificado de seguridad	lun 27/05/19	lun 03/06/19	David Valdivieso

Figura 23. Sprint Resources server

- **Entregable:** Como en los anteriores sprint, este sprint crea el resources_server que es un servidor express, este también se comunica por medio de APIs con los otros servidores. Es importante mencionar que con la realización de última tarea,

el framework en este punto se vuelve más seguro debido a que ahora el protocolo donde corren los tres servidores se convierte en un protocolo HTTPS, esto permite que los paquetes de información viajen de manera encriptada entre servidores, todo gracias a la implementación de un certificado SSL.

4.3. Diseño de la aplicación

4.3.1. Diagrama de despliegue

Para la realización del diagrama de despliegue se han definido tres nodos: el Client que como componentes tiene una interfaz de usuario la cual ayudará a evidenciar la implementación del certificado SSL, además de las APIs tipo REST que permitirán el acceso a los recursos mediante el intercambio del código de autorización por un token de acceso. El Authorization Server a través de sus servicios gestionará la información de los usuarios y dispositivos IoT, su principal función es generar métodos de autorización que controlen el acceso a los dispositivos IoT mediante la introspección y las APIs REST desarrolladas. El principal componente que tiene el Resources Server es el Middleware, el mismo que al ser utilizado como filtro de verificación del token de acceso, se convierte en la principal fuente de seguridad para que este servidor pueda ser utilizado como un bróker de dispositivos IoT.

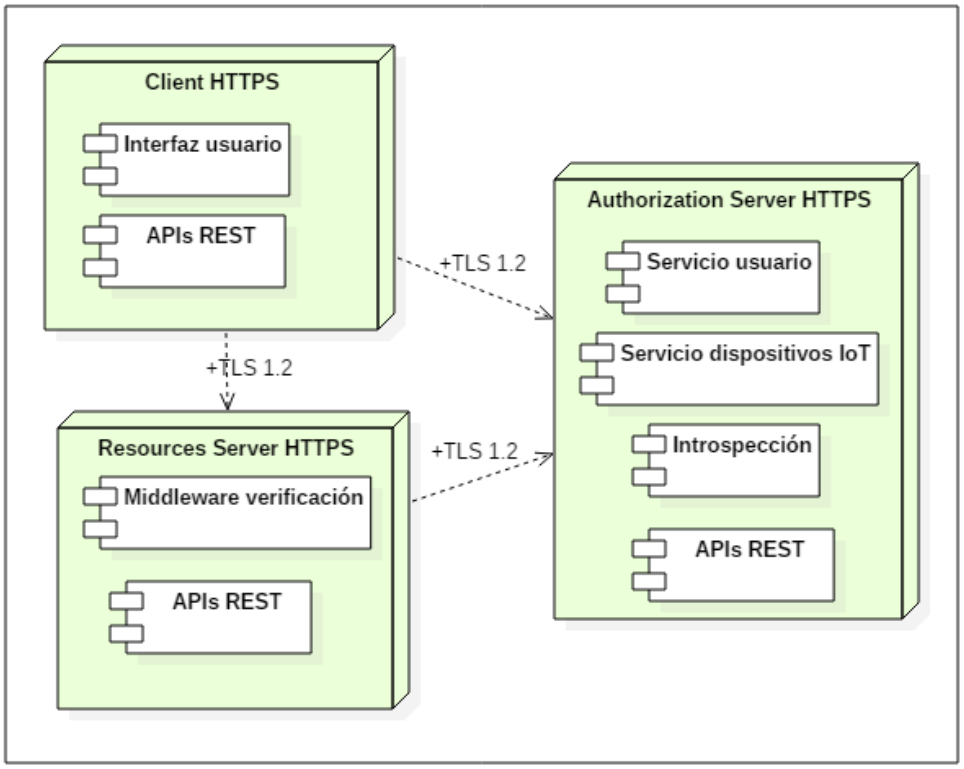


Figura 24. Diagrama de despliegue framework de seguridad

4.4. Arquitectura propuesta

4.4.1. Diagrama de la arquitectura

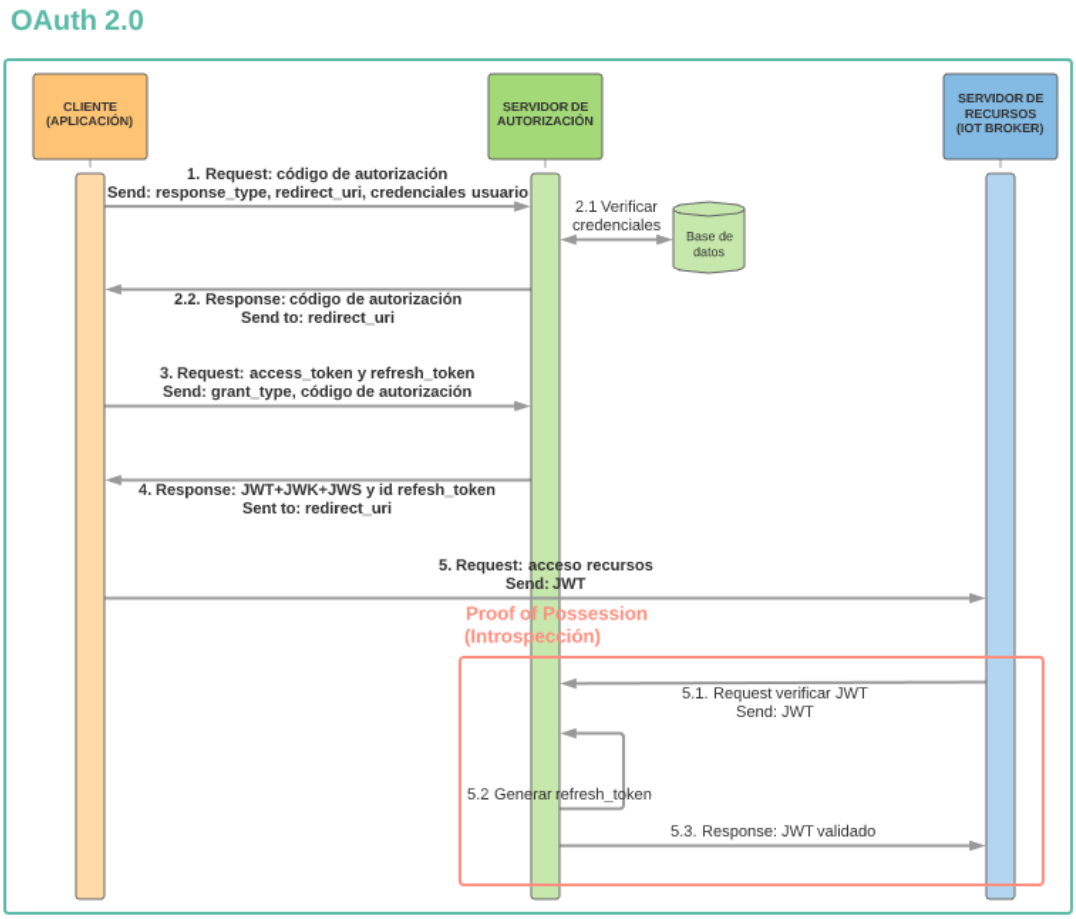


Figura 25. Diagrama de arquitectura

4.4.2. Descripción de la arquitectura

Con la finalidad de comprender la arquitectura del framework de seguridad, a continuación, se detalla los pasos de la figura 19.

1. El CLIENTE envía el response_type, el redirect_uri y las credenciales de usuario (email, password) al SERVIDOR DE AUTORIZACIÓN.

2. EL SERVIDOR DE AUTORIZACIÓN verifica en la base de datos que las credenciales sean correctas y devuelve un código de autorización al redirect_uri enviado por el CLIENTE.
3. El CLIENTE envía el grant_type y el código de autorización al SERVIDOR DE AUTORIZACIÓN.
4. El SERVIDOR DE AUTORIZACIÓN genera y envía al CLIENTE un identificador del refresh_token y un access_token de tipo JSON Web Token (JWT), el cual, mediante una clave simétrica JSON Web Key (JWK) es firmado con JSON Web Signature (JWS).
5. El acceso a los recursos relacionados con el usuario se produce, cuando el SERVIDOR DE AUTORIZACIÓN verifica que el JSON Web Token (JWT) enviado por el CLIENTE hacia el SERVIDOR DE RECUROS sea correcto o no haya expirado. A continuación, el proceso de este paso.
    a. El CLIENTE envía el JWT al SERVIDOR DE RECURSOS.
    b. El SERVIDOR DE RECURSOS reenvía el JWT al SERVIDOR DE AUTORIZACIÓN para la verificación (introspección).
    c. EL SERVIDOR DE AUTORIZACIÓN valida el JWT mediante la verificación de la firma (JWS) con la que fue creado y comprueba el tiempo de vida del JWT. Si el JWT es correcto y no ha expirado se devuelve la información del usuario junto con los recursos a los cual puede acceder, si el JWT es correcto y ha expirado, se hace uso del refresh_token para generar un nuevo JWT y comunicar al CLIENTE (ciclo repetitivo de los pasos 4 y 5) y si el JWT es incorrecto se devuelve un mensaje de error.

d. El SERVIDOR DE RECURSOS permite el acceso si se evidencia los datos y los recursos del usuario devueltos desde el SERVIDOR DE AUTORIZACIÓN.

4.4.3. Beneficios de la arquitectura

El desarrollo del framework está pensado principalmente para la seguridad de la autenticación de los dispositivos IoT, a continuación, se describen los beneficios, los cuales están centrados en los mecanismos OAuth 2.0, JSON Web Token (JWT) y Proof of Possession (PoP) considerados los más importantes para desarrollar el framework.

- Con el uso del grant type “código de autorización” propuesto por OAuth 2.0, el SERVIDOR DE AUTORIZACIÓN permitirá la generación de un JWT si se verifica la autenticidad del código de autorización, restringiendo de esta manera el acceso a los usuarios que no concuerden con dicho código.
- Al manejar OAuth 2.0 juntamente con los JWT (access_token y refresh_token), se pueden definir tiempos de expiración cortos, de esta manera se protege de una posible suplantación de identidad.
- Los JWT al ser firmados con JWK y JWS, verifica que la información no haya sido modificada, asegura que la identidad del usuario sea legítima y que solo se pueda acceder a los dispositivos IoT vinculados al usuario.
- La utilización del método de introspección de PoP, faculta que únicamente el SERVIDOR DE AUTORIZACIÓN conozca la firma (JWS) del JWT, así se evita el

envío de la clave simétrica (JWK) como parte del JWT, lo que nos permite impedir la reclamación de la información si esta clave es robada.

CAPÍTULO V

DESARROLLO Y PRUEBAS

5.1. Construcción del framework de seguridad

5.1.1. Base de datos

La base de datos creada en MongoDB contiene dos tablas con un modelamiento sencillo, que tiene como objetivo representar la relación entre los dispositivos IoT y los usuarios.

5.1.2. Authorization server

- **Servicios**

  El desarrollo de servicios tanto para el usuario como para los dispositivos IoT, permiten la interacción entre los modelos y los esquemas de la base de datos, sus principales funcionalidades son la creación de usuarios, la obtención de un usuario, la actualización del estado del usuario y la obtención de los dispositivos IoT vinculados al usuario.

- **Introspección**

  La introspección es la funcionalidad más relevante para demostrar que el usuario es quien dice ser y pueda tener acceso a sus dispositivos IoT. Se puede decir que la introspección es la base para que se produzca el Proof of Possession, ya que el servidor de autorización es el encargado de verificar que el token de acceso del usuario sea válido.

- **APIs REST**

  ‘/register’: Configurado como POST, es un api que permite el registro y almacenamiento de un usuario en la base de datos, tiene una validación para la no duplicidad de usuarios. Los atributos que deben enviarse en esta solicitud son: el email y password.

  ‘/authorize’: Configurado como POST, es un api que genera el código de autorización si los datos del usuario son correctos. Los atributos que deben ser enviados para esta solicitud son: response_type de tipo code y el redicet_uri que básicamente es una dirección de callback a donde se va a enviar el código de autorización.

  ‘/token’: Configurado como POST, es un api para el intercambio de un código de autorización por un token de acceso en este caso un JWT. Cabe mencionar que únicamente se generará el token de acceso si el código enviado es el correcto. Los atributos que deben enviarse para esta solicitud son: el grant_type de tipo authorization_code y el code que su valor es el código de autorización generado por el servidor de autorización. Es importante mencionar que en este api también se puede generar un token de actualización en caso de que un token de acceso haya expirado por lo que ahora el grant_type enviado será de tipo refresh_token.

  ‘/introspection’: Configurado como POST, es un api que sirve para la verificación del token. El atributo que debe enviarse a esta solicitud es el token de acceso (JTW) generado para el usuario.

5.1.3. Client

- **Interfaz de Usuario**

Con el propósito de verificar el uso del certificado SSL, se creó dos interfaces simples para el usuario, una interfaz tendrá el inicio de sesión y la otra un botón que permita evidenciar los dispositivos IoT a los que el usuario tiene acceso si la comprobación del token es correcta.

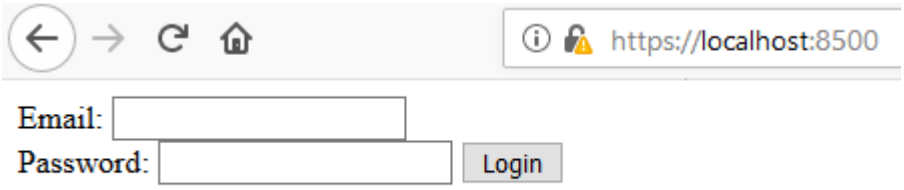


Figura 26. Interfaz inicio de sesión

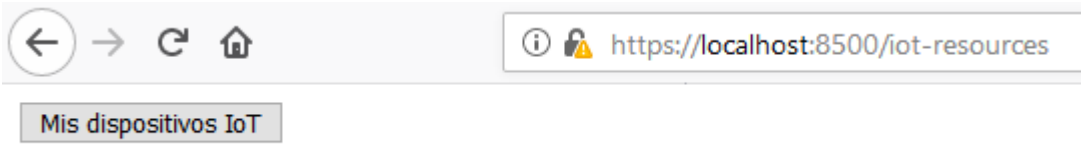


Figura 27. Interfaz obtener dispositivos IoT

- **APIs REST**

‘/login’: Configurado como POST, este api sirve para el inicio de sesión del usuario. Los parámetros que debe ser enviado en la solicitud debe ser el email y el password.

'/callback': Configurado como GET, este api sirve para la devolución del código de autorización y el token de acceso generados por el servidor de autorización.

Dependiendo del caso, este api requiere el envío de parámetros como el code o el token.

'/iot-resources': Configurado como POST, desde este api se solicita el acceso a los dispositivos IoT del usuario. El parámetro a enviar en esta solicitud es el token de acceso devuelto en el callback.

5.1.4. Resources server

- **APIs REST**

  '/iot-resources': Configurado como POST, este api es el encargado de la comunicación entre el cliente y el servidor de recursos al momento de acceder a los dispositivos IoT del usuario.

- **Middleware verificación de token**

  El middleware de verificación de token es una función que se ejecuta antes de la solicitud a los dispositivos IoT, esta función da paso únicamente cuando el token de acceso enviado por el cliente es válido. Desde aquí se produce la introspección (PoP) entre el servidor de recursos y el servidor de autorización.

5.1.5. Certificados SSL

La utilización de los certificados SSL dentro del framework de seguridad, permite que el envío de información entre los servidores viaje de manera encriptada. Al implementar dichos certificados garantizamos la integridad de la información debido a que el protocolo de comunicación pasa de ser un Transmission Control Protocol (TCP) a ser un protocolo Transport Layer Security TLS.

5.2. Pruebas del framework de seguridad

5.2.1. Definición de pruebas de seguridad

Las pruebas de seguridad dentro de un ecosistema IoT están orientadas a diferentes ámbitos, como es el caso de la capa de percepción, la capa de red, y la capa de aplicación, cada una de las pruebas se las enfoca dependiendo de la capa a la que se le quiera dar seguridad (Verma & Chahal, 2017). Es relevante mencionar que el objetivo de las pruebas es determinar vulnerabilidades que se puedan presentar dentro de un ecosistema IoT, una vez analizadas dichas vulnerabilidades se deben realizar las implementaciones necesarias para mitigar de esta manera los ataques que puedan afectar la seguridad del ecosistema.

5.2.2. Fundamento para las pruebas del framework de seguridad

Dado que el desarrollo del framework está orientado a brindar seguridad en la autenticación de un ecosistema IoT y, además los ataques de Man-in-the-Middle, Brute Force Attack e Impersonation constan en las preguntas de investigación planteadas, han permitido dar la apertura necesaria para que se puedan realizar pruebas de concepto a dichos ataques.

Para realización de las pruebas se propone dos escenarios diferentes, un escenario inseguro donde se demostrará las vulnerabilidades por cada uno de los ataques, y un escenario seguro que permitirá demostrar la mitigación de dichos ataques.

5.2.3. Pruebas Man-in-the-middle attack

Escenario inseguro

En este escenario la información enviada entre servidores se la realiza en texto plano, lo cual puede traer consecuencias graves como: robo de credenciales, robo tokens es decir robo de información susceptible.

1. Ingresar a la página de inicio de sesión

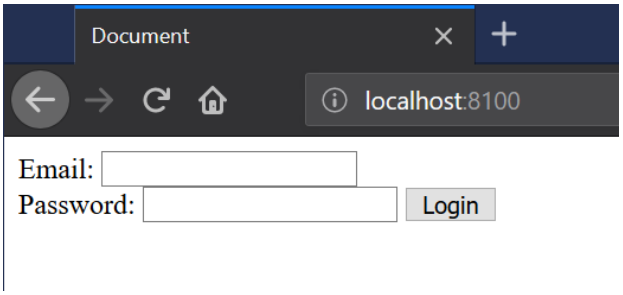


Figura 28. Inicio de sesión escenario inseguro de Man-in-the-middle

2. Colocar las credenciales de usuario

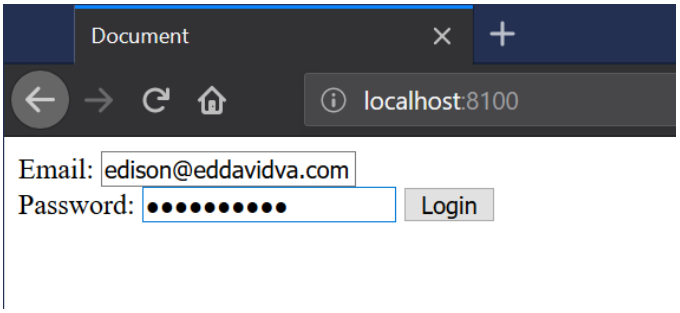


Figura 29. Credenciales escenario inseguro de Man-in-the-middle

3. Verificar mediante Wireshark que la información puede ser robada por cualquier atacante dentro de la red.

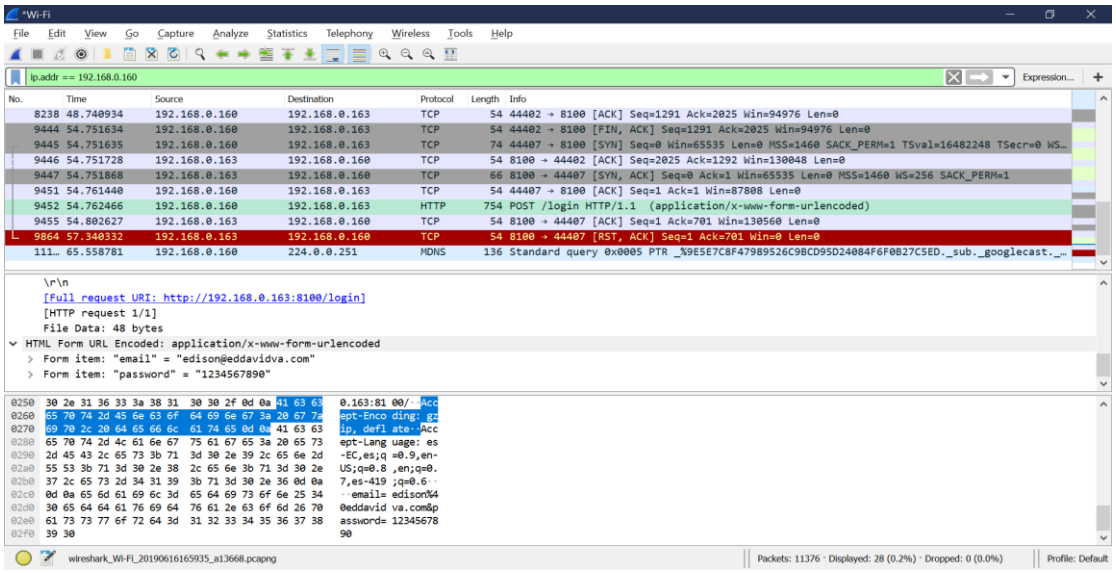


Figura 30. Verificación de información, escenario Inseguro de Man-in-the-Middle

a. Para poder saber cuál es el paquete de información entre el servidor de autorización y el cliente, revisamos que en source esté la IP del servidor de autorización, en destination la IP del cliente y en protocol debe estar TCP que es el protocolo de comunicación usado para comunicaciones no encriptadas.

Source	Destination	Protocol
192.168.0.160	192.168.0.163	TCP
192.168.0.160	192.168.0.163	TCP
192.168.0.160	192.168.0.163	TCP
192.168.0.163	192.168.0.160	TCP
192.168.0.163	192.168.0.160	TCP
192.168.0.160	192.168.0.163	TCP
192.168.0.160	192.168.0.163	HTTP

Figura 31. Paquete de información, escenario inseguro de Man-in-the-Middle

b. La etiqueta de info muestra la información sobre el método de comunicación, en este caso es un POST, mediante el cual estamos enviando el email y password.

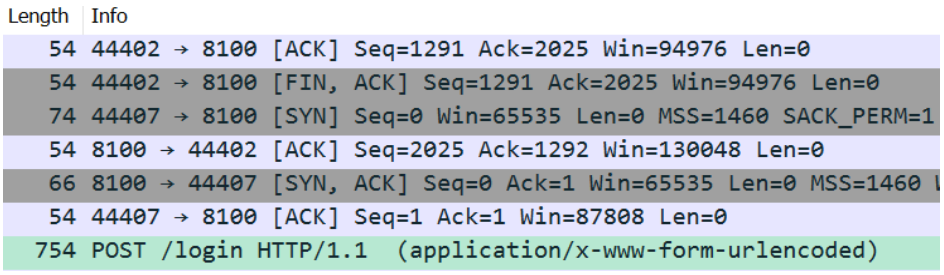


Figura 32. Información del método de comunicación escenario inseguro Man-in-the-Middle

c. Al extender los datos del paquete, en la parte inferior podemos ver que la información del email y password se encuentran en texto plano, lo que facilita el robo de datos al ser interceptada por cualquier atacante que se encuentre dentro de la red.

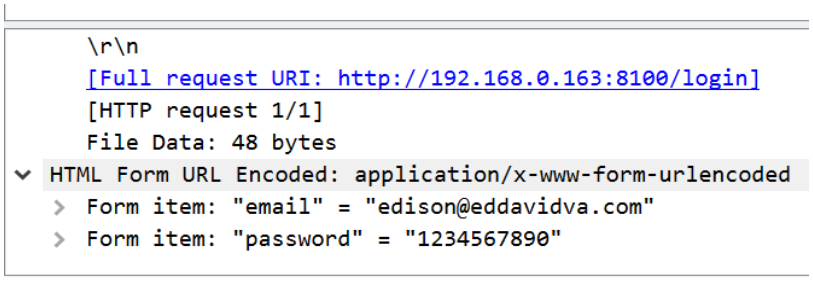


Figura 33. Información URL escenario inseguro de Man-in-the-Middle

4. Revisar el paquete de comunicación

Figura 34. Paquete de comunicación escenario inseguro de Man-in-the-Middle

En el paquete de comunicación, se puede visualizar la siguiente información.

Header: Son los datos de la cabecera de la solicitud.

```
POST /login HTTP/1.1
Host: 192.168.0.163:8100
Connection: keep-alive
Content-Length: 48
Cache-Control: max-age=0
Origin: http://192.168.0.163:8100
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
```

Figura 35. Header del paquete de comunicación

User-agent: Es el medio desde donde se hizo la solicitud, en este caso fue desde un celular con sistema operativo Android 7.

```
User-Agent: Mozilla/5.0 (Linux; Android 7.0; Moto G (4)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
74.0.3729.157 Mobile Safari/537.36
```

Figura 36. User-agent del paquete de comunicación

Body: Son los datos que se envían en el cuerpo de la solicitud.

```
email=edison%40eddavidva.com&password=1234567890
```

Figura 37. Body del paquete de comunicación

Escenario seguro

Para la verificación de este escenario se usará un protocolo seguro, como lo es HTTPS. Para esto se generó un certificado de seguridad mediante OpenSSL con el siguiente comando:

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout ./selfsigned.key -out selfsigned.crt

Tabla 16

Descripción parámetros de comando openssl

Parámetro	***Descripción***
Req	Especifica el cifrado de una llave privada.
x509	Es un estándar para infraestructura de claves públicas.
Nodes	Especifica el cifrado de una llave privada.
Days	Especifica el número de días por los que el certificado es válido, en este caso 365 días.
newkey	Especifica el algoritmo de encriptación y el número de bits de este, en este caso se utilizará RSA de 4098 bits.
keyout	Especifica el lugar en el que se guardara la llave generada.
Out	Especifica el lugar en el que se guardara el certificado generado.

Para poder utilizar el certificado SSL generado, se realizaron modificaciones en los servidores, a continuación el detalle de los cambios.

- Se especificó la librería *https.*

```
var https = require('https');
```

Figura 38. Uso de la librería HTTPS

- Se asignó en el objeto *credentials* las variables privateKey y certificate, las cuales tienen los datos de la llave privada y del certificado respectivamente, estas se encuentran almacenadas en diferentes archivos de tipo pem.

```
var privateKey  = fs.readFileSync('../certs/client-key.pem', 'utf8');
var certificate = fs.readFileSync('../certs/client-cert.pem', 'utf8');

var credentials = {key: privateKey, cert: certificate};
```

Figura 39. Uso de los archivos del certificado SSL

- Luego se creó la variable *httpsServer,* en la cual se especifica las credenciales del certificado y el servidor que se está utilizando, en este caso el servidor lo generamos con express.

```
var httpsServer = https.createServer(credentials, app);
```

Figura 40. Declaración del servidor con HTTPS

- Para finalizar inicializamos los servidores en el puerto 8500 (client), 8600 (authorization_server) y 8700 (resources_server).

```
httpsServer.listen(8500, function () {
    console.log('Starting server on port 8500');
});
```

Figura 41. Inicialización de los servidores

Una vez realizada la configuración en los servidores antes mencionados se procede a realizar la prueba.

1. Ingresar a la página de inicio de sesión y aceptar el riesgo de usar un certificado no seguro, debido a que es un certificado realizado como prueba para el proyecto.

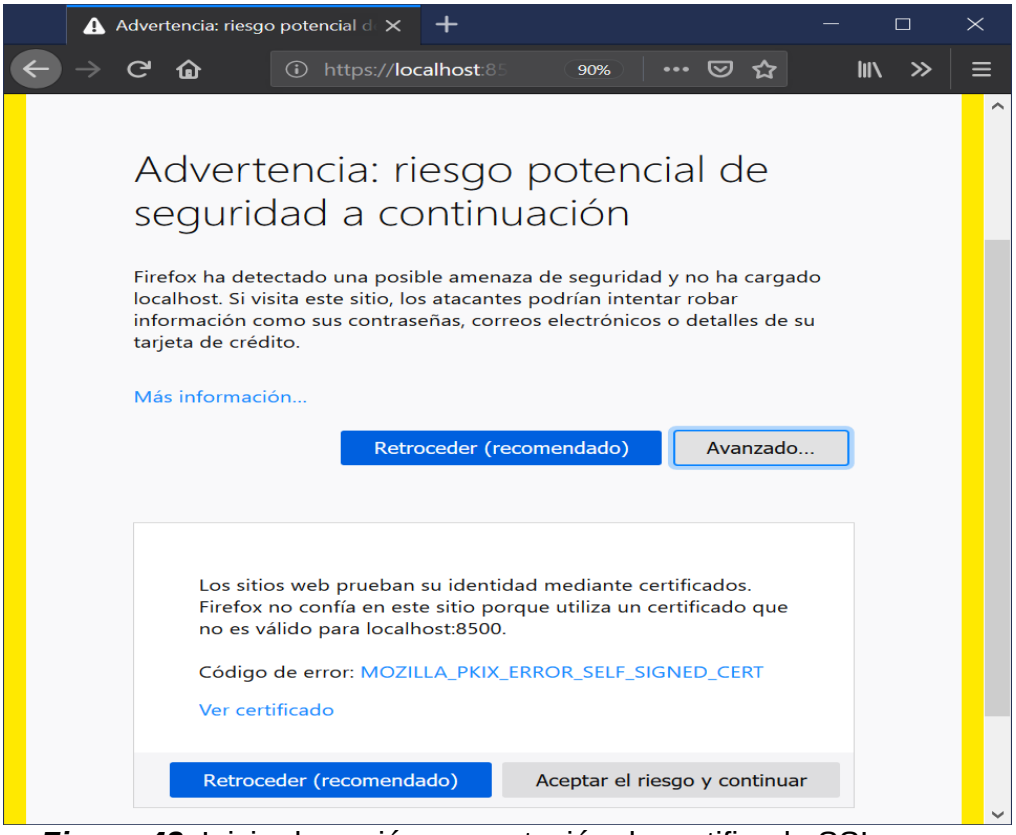


Figura 42. Inicio de sesión y aceptación de certificado SSL escenario seguro de Man-in-the-Middle

2. Colocar las credenciales del usuario.

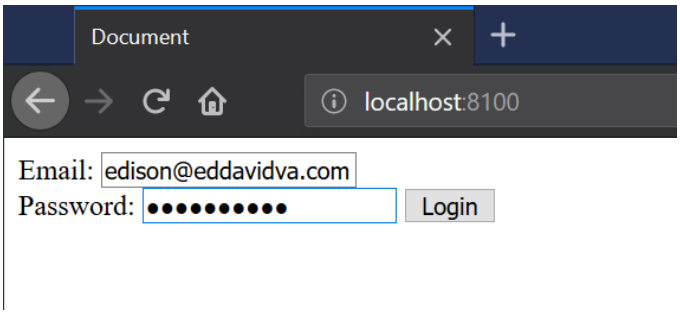


Figura 43. Credenciales escenario seguro de Man-in-the-Middle

3. Revisar en Wireshark que la información no pueda ser robada.

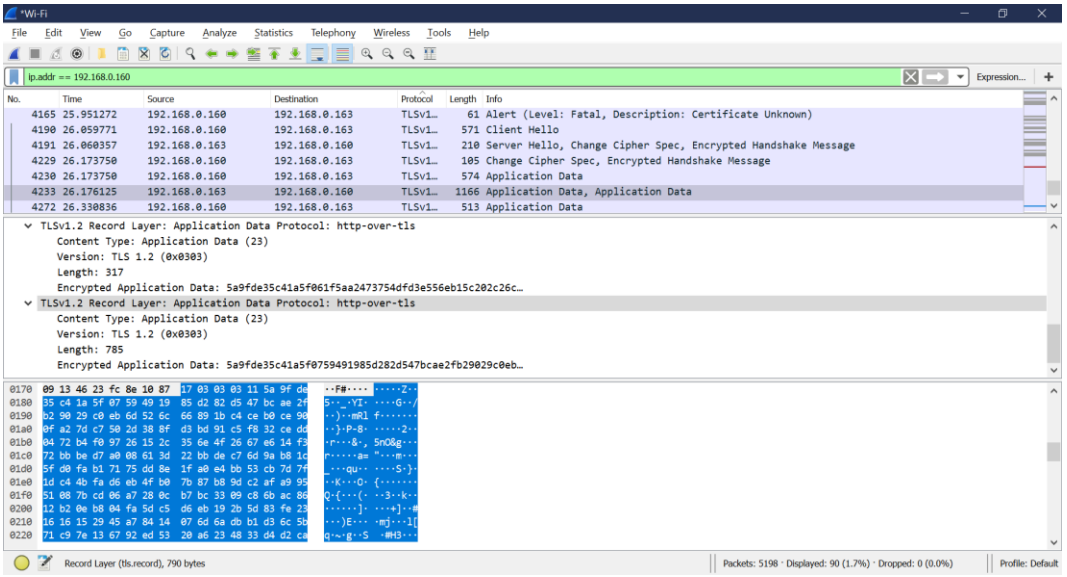


Figura 44. Verificación de información, escenario seguro de Man-in-the-Middle

a. Verificar que en source esté la IP del servidor de autorización, en destination esté la IP del cliente y en protocol esté TLS, este protocolo indica que los datos que viajan a través del mismo están encriptados.

Source	Destination	Protocol
192.168.0.160	192.168.0.163	TLSv1…
192.168.0.160	192.168.0.163	TLSv1…
192.168.0.163	192.168.0.160	TLSv1…
192.168.0.160	192.168.0.163	TLSv1…
192.168.0.160	192.168.0.163	TLSv1…
192.168.0.163	192.168.0.160	TLSv1…
192.168.0.160	192.168.0.163	TLSv1…

Figura 45. Paquete de información escenario seguro de Man-in-the-Middle

b. Además se puede ver que en la etiqueta info ya no se ve el método por el cual se realizó la solicitud (POST o GET), ahora aparece un mensaje que dice application data, el cual indica que se está realizando un intercambio de información entre los dos servidores.

ength	Info
61	Alert (Level: Fatal, Description: Certificate Unknown)
571	Client Hello
210	Server Hello, Change Cipher Spec, Encrypted Handshake Message
105	Change Cipher Spec, Encrypted Handshake Message
574	Application Data
1166	Application Data, Application Data
513	Application Data

Figura 46. Información de comunicación escenario seguro Man-in-the-Middle

4. Revisar el paquete de comunicación y verificar que todo se encuentra encriptado, a diferencia del escenario inseguro en el cual se mostraba toda la información en texto plano.

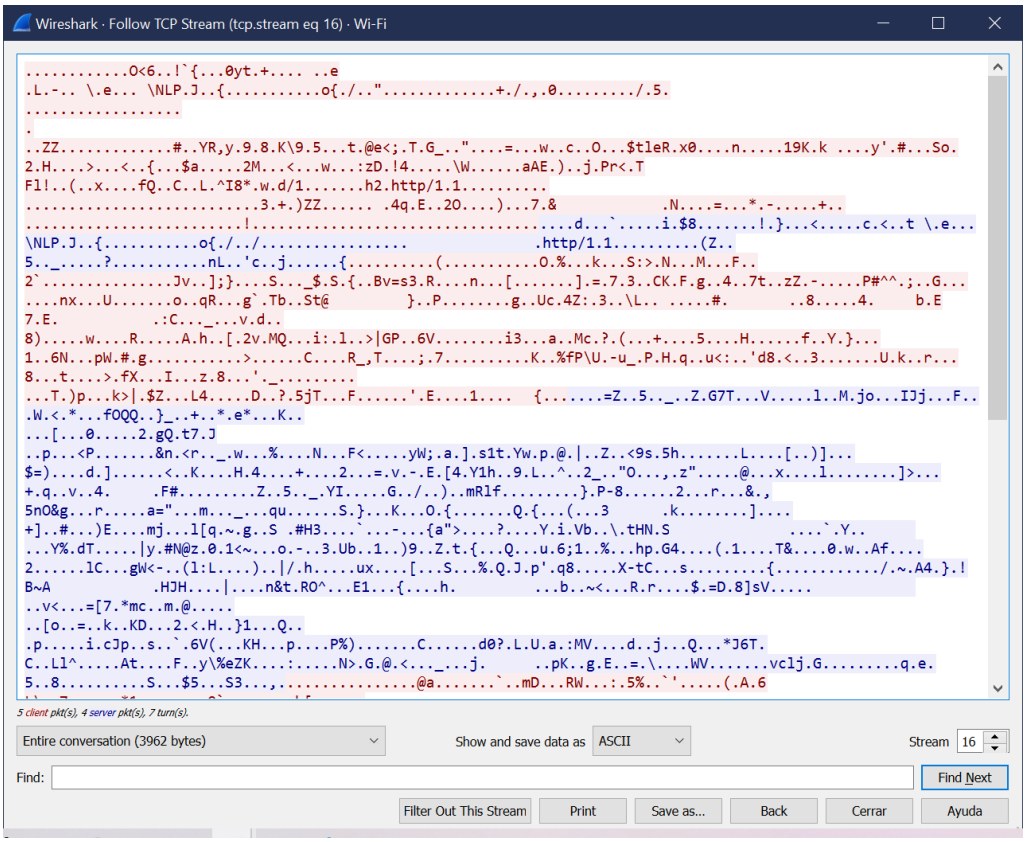


Figura 47. *Paquete de comunicación escenario seguro de* Man-in-the-Middle

Podemos concluir que nuestro framework de seguridad no es vulnerable al ataque Man-in-the-Middle ya que todos los paquetes de información enviados entre servidores se encuentra encriptados, gracias a TLS 1.2.

5.2.4. Pruebas Brute Force Attack

Para demostrar este ataque se utilizará el programa Hydra, que permite realizar ataques de fuerza bruta. Para la ejecución de este ataque se hará uso del siguiente comando.

hydra 192.168.0.163 https-form-post "/login:email=^USER^&password=^PASS^:Bad login" -s 8500 -l edison@eddavidva.com -P /home/invest/Documents/Hack/pass.txt -t 10 -w 30

Tabla 17

Descripción parámetros de comando para ataque con Hydra

Parámetro	Descripción
192.168.0.163	Es la dirección del servidor o la victima
https-form-post	Es el método de comunicación que queremos atacar, en este caso es un método POST.
"/login:email=^USER^&password=^PASS^:Bad login"	Aquí se detalla la url del ataque y los campos a ser atacados.
-s 8500	Especificación del puerto de comunicación en este caso es el 8500.
-l edison@eddavidva.com	Especificación de una lista de usuarios (-L) o un usuario único (-l).
-P /home/invest/Documents/Hack/pass.txt	Especificación de una lista de contraseñas (-P) o una sola contraseña (-p).

CONTINÚA

-t 10	Especificación del número de tareas paralelas que se va a utilizar para hacer el ataque.
-w 30	Definición del tiempo máximo de espera para las respuestas.

Escenario inseguro

1. Iniciar los servidores

   client

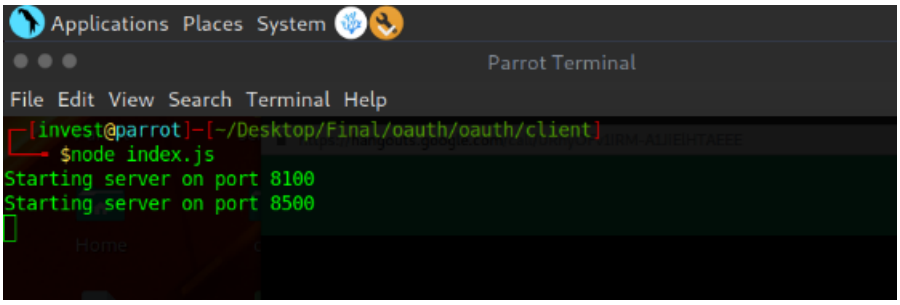


Figura 48. Inicio del client escenario inseguro
Brute Force Attack

   authorization_server

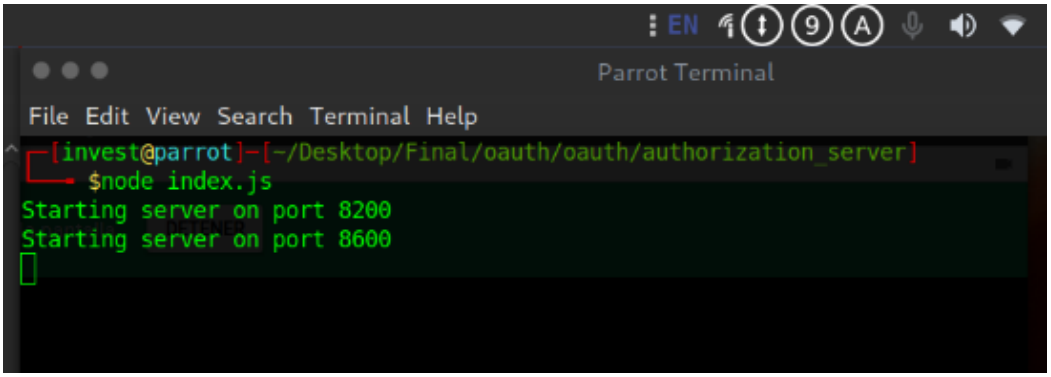


Figura 49. Inicio del authorization_server escenario inseguro
Brute Force Attack

resources_server

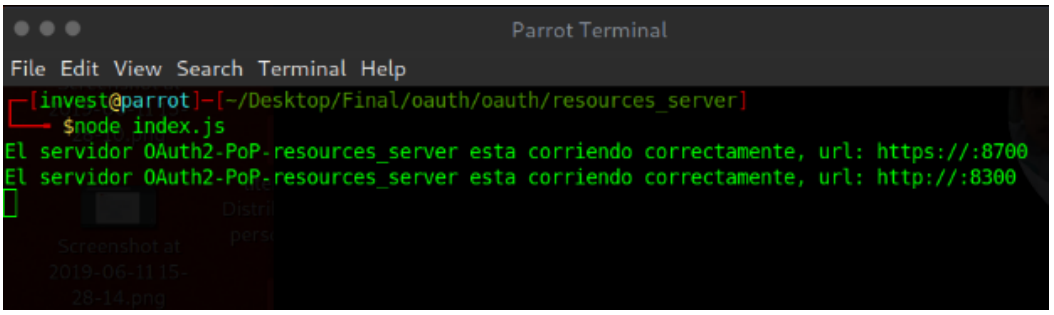


Figura 50. Inicio del resources_server escenario inseguro Brute Force Attack

2. Enviar el ataque mediante Hydra

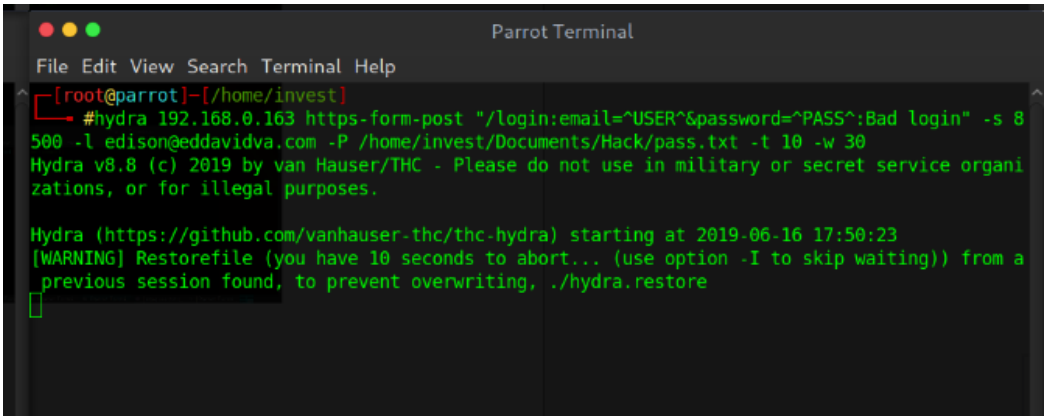


Figura 51. Ataque Hydra escenario inseguro Brute Force Attack

3. Se puede visualizar que el atacante puede seguir intentando ingresar al sistema, sin que este se lo niegue.

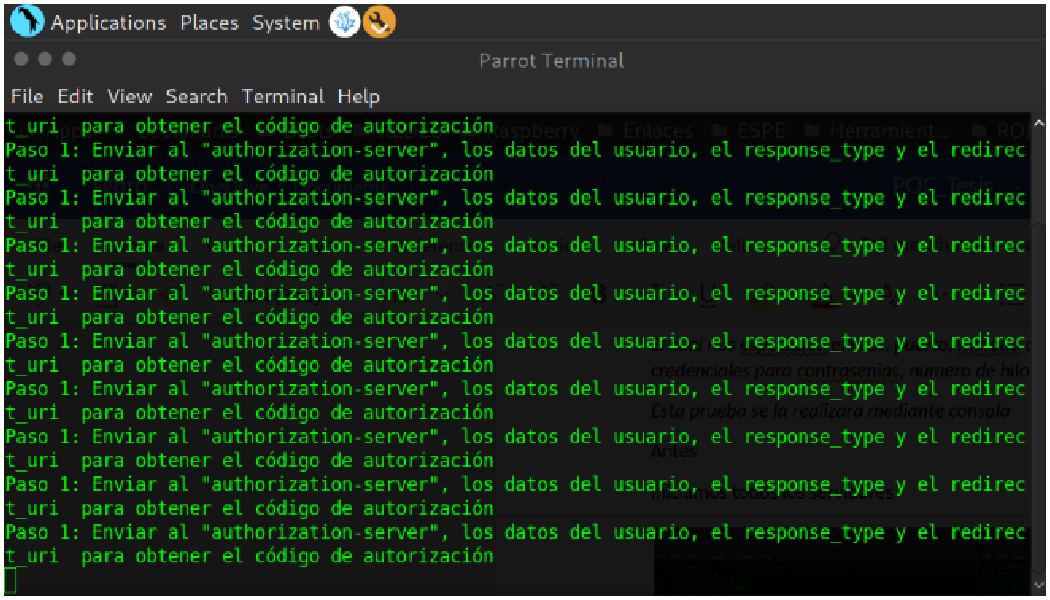


Figura 52. Verificación de intento de inicio de sesión escenario inseguro Brute Force Attack

De esta forma el atacante puede generar diccionarios de palabras y seguir intentando vulnerar la seguridad de inicio de sesión.

Escenario seguro

Para poder mitigar estos ataques es necesario controlar el acceso de los usuarios, es decir si un usuario erra por tres veces el ingreso del password, este se inactivará, con esto se evita que un atacante siga probando distintas contraseñas para vulnerar las seguridades del framework.

1. Iniciar los servidores

client

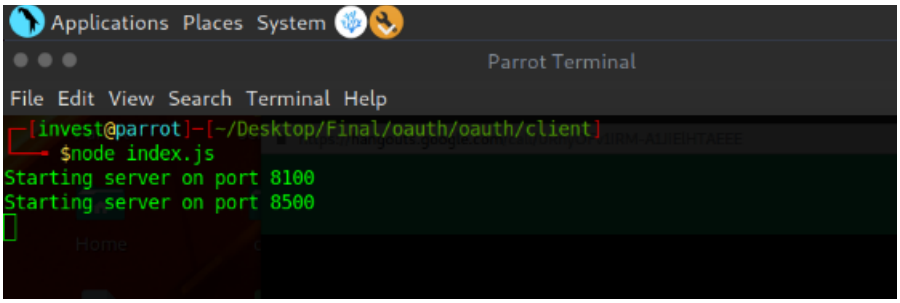


Figura 53. Inicio del client escenario seguro Brute Force Attack

authorization_server

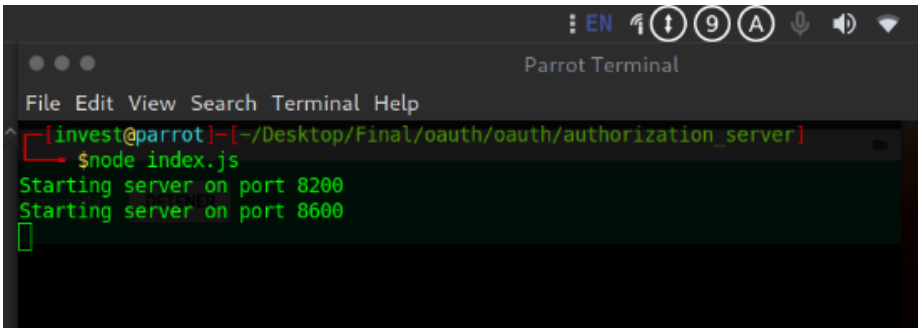


Figura 54. Inicio del authorization_server escenario seguro Brute Force Attack

resources_server

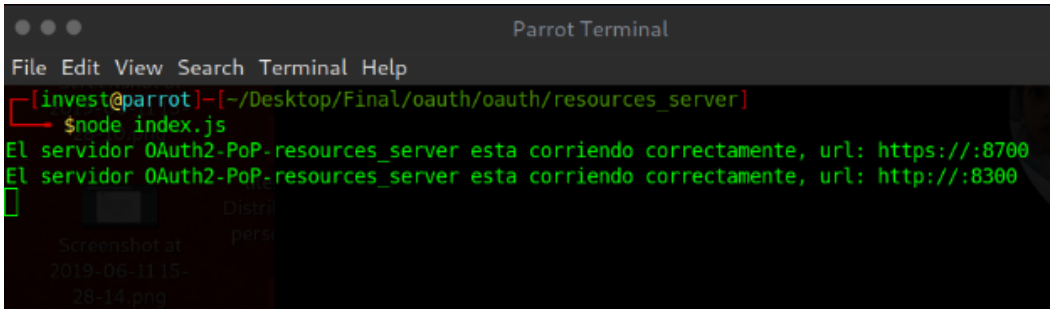


Figura 55. Inicio del authorization_server escenario seguro Brute Force Attack

2. Enviar el ataque mediante Hydra

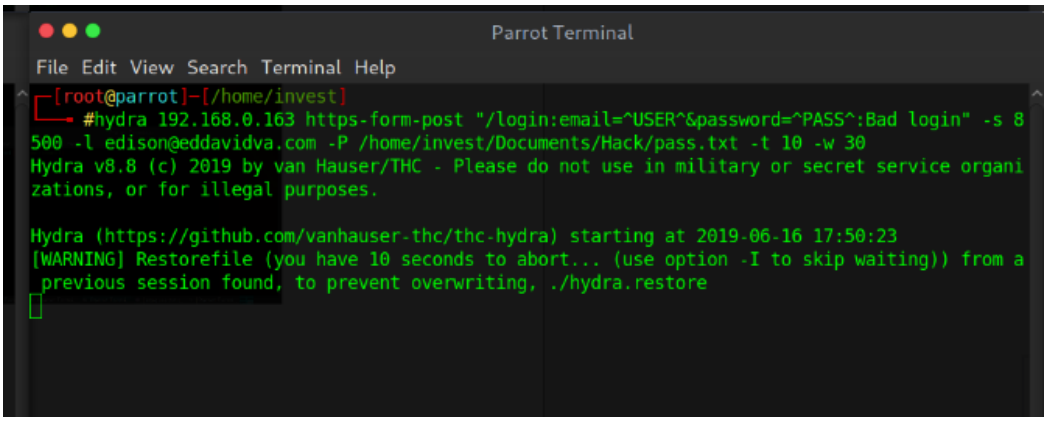


Figura 56. Ataque Hydra escenario seguro Brute Force Attack

3. El atacante tiene un máximo de tres intentos para vulnerar la seguridad del framework.

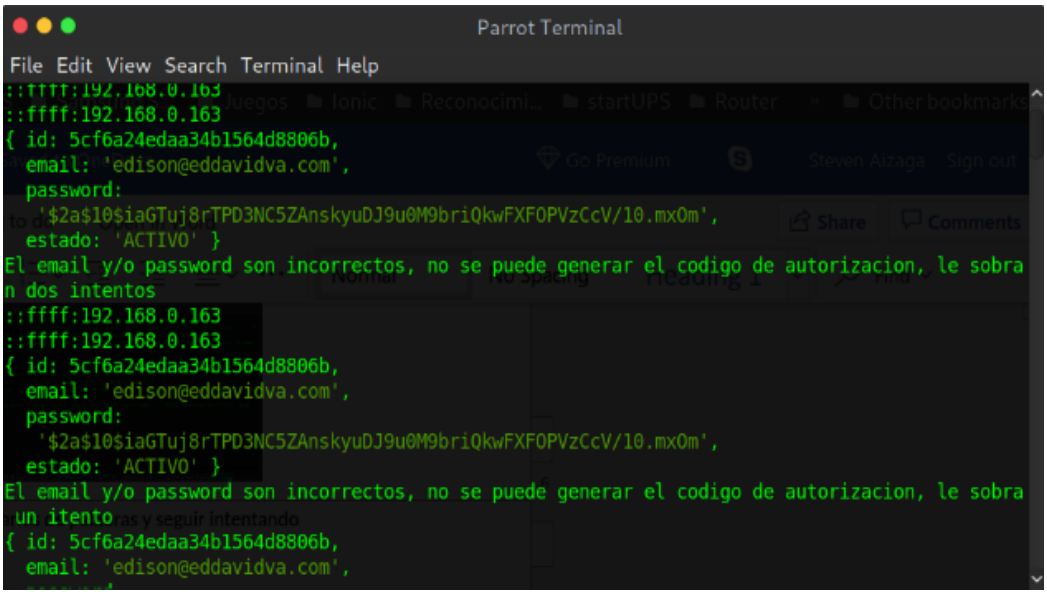


Figura 57. Intentos de inicio de sesión escenario seguro Brute Force Attack

4. A partir del intento número cuatro, el usuario se inactiva, lo cual evita que el atacante siga probando con distintas contraseñas el poder vulnerar la seguridad del framework.

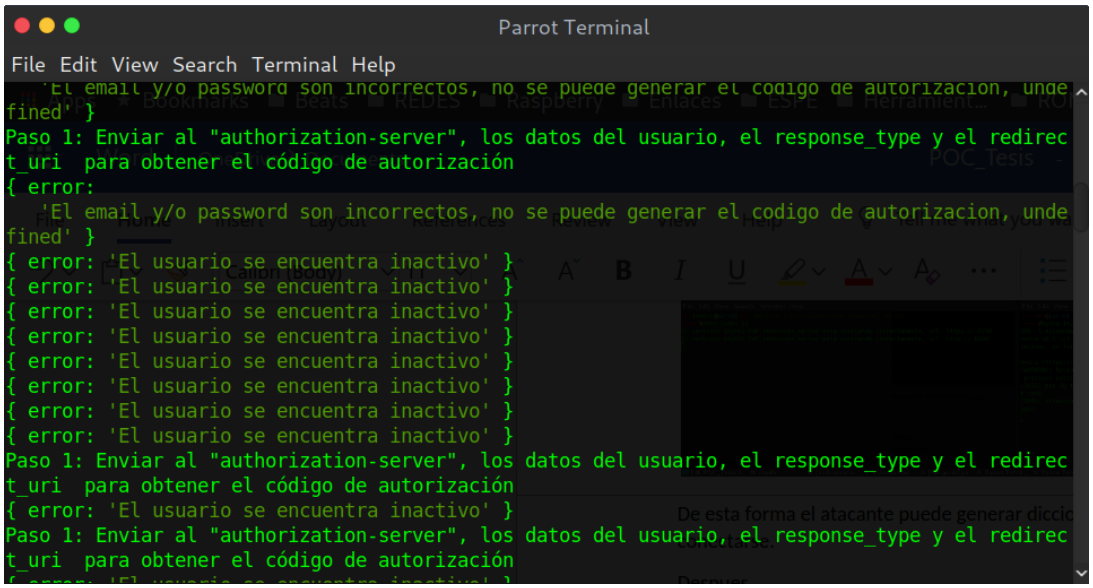


Figura 58. Verificación de inactivación de usuario escenario seguro Brute Force Attack

5.2.5. Pruebas Impersonation

Para este caso de ataque se ha planteado una situación en la que un atacante logra robar parte de la base de datos que es utilizada por el framework.

El framework utiliza MongoDB como base de datos y para la realización de las pruebas se van a ejecutar los comandos mediante consola.

Escenario inseguro

1. Para ingresar a la base de datos MongoDB, se debe abrir una ventana de comandos y ejecutar el comando use oauth2_pop, el cual permite acceder a las colecciones que contiene la base.

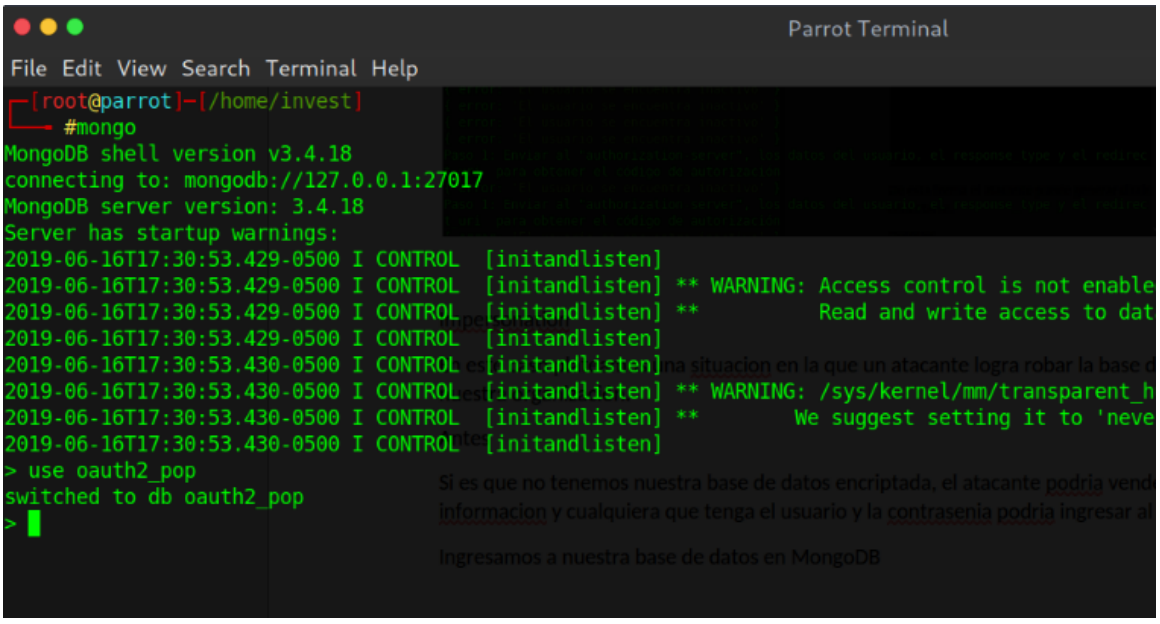


Figura 59. Acceso a la base de datos escenario inseguro Impersonation

2. Previo a la ejecución del comando db.usuarios.find().pretty(), que servirá para listar los usuarios que interactúan con el framework, se creó una nueva colección denominada usuarios, la cual servirá como ejemplo para mostrar la vulnerabilidad de almacenar contraseñas en texto plano.

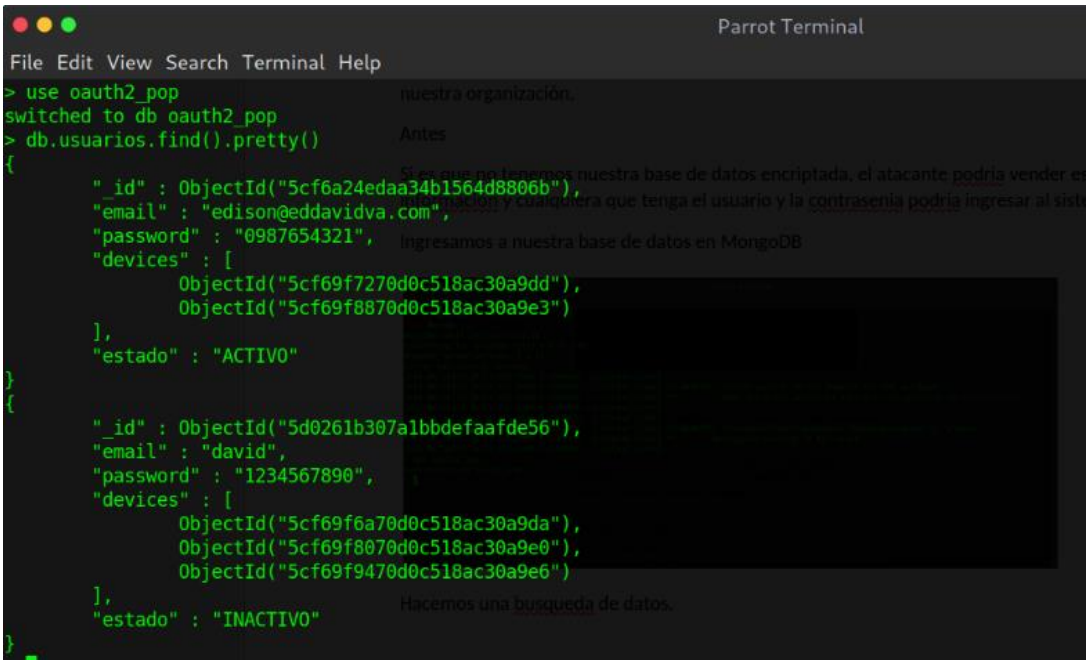


Figura 60. Listado de usuarios de la colección usuarios escenario inseguro Impersonation

3. En la imagen que se muestra a continuación, se puede observar que la información de los usuarios es leída a simple vista.

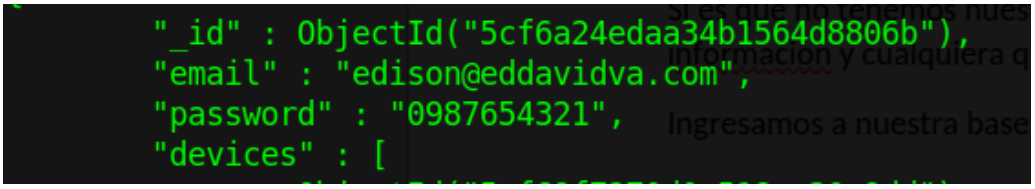


Figura 61. Información de usuario escenario inseguro Impersonation

Al no tener la debida seguridad sobre los campos más relevantes de la base de datos, el atacante podría obtener fácilmente las credenciales del usuario, con las cuales sería inevitable el robo de más información que se transmite a través de los servidores.

Escenario seguro

Para mitigar el escenario anterior se implementó una funcionalidad en el framework, el cual a través del uso de la librería de encriptación bcrypt permitirá generar un hash (algoritmo matemático) del password antes de que sea almacenada en la base de datos, cuando el usuario es creado.

```
const salt = 10;
user.password = bcrypt.hashSync(password, salt);
```

Figura 62. Configuración del hash con BCrypt

El uso del salt en bcrypt hace referencia al tiempo de procesamiento para generar el hash del password, entre más alto es el salt, el tiempo que requiere un atacante para poder descifrar es mucho más alto, por lo que también en un método que sirve para mitigar el Brute Force Attack.

1. Para ingresar a la base de datos MongoDB, se debe abrir una ventana de comandos y ejecutar el comando *use oauth2_pop*, el cual permite acceder a las colecciones que contiene la base.

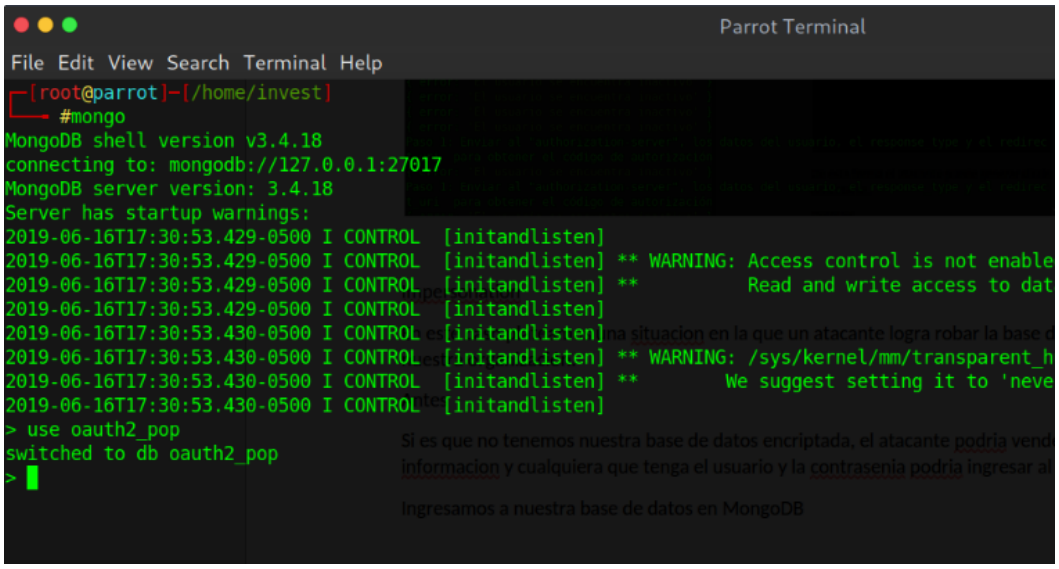


Figura 63. Acceso a la base de datos escenario seguro Impersonation

2. El comando *db.user.find().pretty()*, muestra la lista de los usuarios que pueden interactuar con el framework. En este caso la colección *users,* es la colección real con la que el framework trabaja.

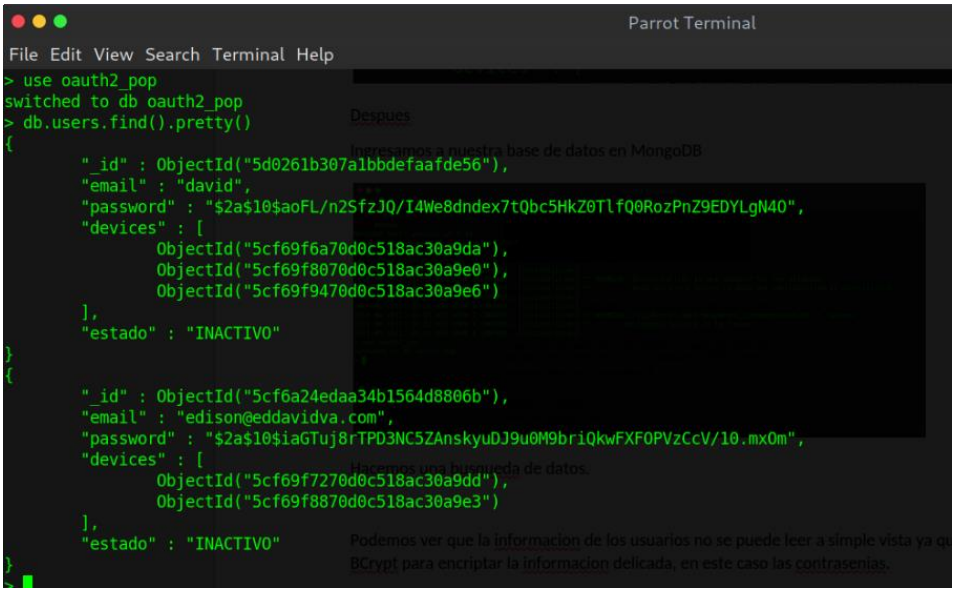


Figura 64. Listado de usuarios de la colección users escenario seguro Impersonation

3. En la imagen que se muestra a continuación, se puede verificar que parte de la información de los usuarios no puede ser leída a simple vista.

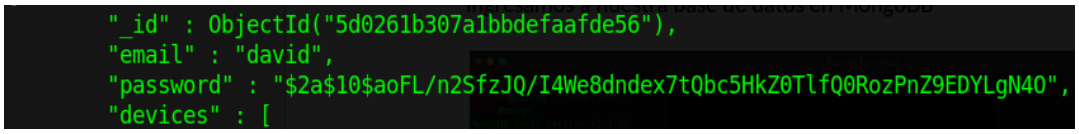


Figura 65. Información de usuario escenario seguro Impersonation

Por lo tanto al utilizar una función de hash encriptar datos, evitamos que los atacantes puedan robar la información, en este caso el password. De esta forma se disminuye los ataques de impersonation.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- Después de haber realizado el proceso de revisión inicial de literatura se puede concluir que el protocolo OAuth en su versión 2.0. es el más utilizado para el proceso de autorización en aplicaciones y ecosistemas IoT.
- En respuesta a una de las preguntas de investigación planteadas se logró identificar varios tipos de ataque, siendo los más comunes, Man-in-the-Middle, Brute Force Attack e Impersonation, los cuales buscan vulnerar la seguridad de la autenticación.
- Al hacer uso del protocolo OAuth 2.0 considerado en la actualidad uno de los métodos más seguros para la autorización de acceso a cualquier tipo de recursos, se está garantizando la autenticación dentro de un ecosistema IoT.
- Al utilizar el mecanismo de Proof of Possession juntamente con OAuth 2.0., se está brindando un middleware de seguridad para la verificación del token de acceso (JWT), lo que permite controlar el acceso no autorizado a los recursos dentro de una red IoT.
- Con la implementación de certificados SSL y BCrypt dentro del framework de seguridad desarrollado, y después de realizar pruebas de concepto al mismo, se logró evidenciar la mitigación de vulnerabilidades causadas por ataques como: Man-in-the-Middle, Brute Force Attack e Impersontation.

6.2. Recomendaciones

- Se recomienda el uso de introspección propuesto por Proof of Possesión (PoP) ya que la firma de seguridad del token de acceso (JWT) al no ser enviada como parte de este, garantiza la integridad de la información dentro de una red, en el caso de que esta información sea interceptada por un atacante.
- Con el propósito de brindar seguridad en la confidencialidad e integridad de la información, se recomienda utilizar herramientas actualizadas que vayan acorde con las técnicas de ataque existentes en la actualidad. Como ejemplo tenemos TLS 1.2+ para cifrar los paquetes de datos en la capa de comunicación y BCrypt para el hashing de password.
- Si bien es cierto el framework desarrollado garantiza la integridad de los datos en el proceso de autenticación, se recomienda concientizar a los usuarios sobre la utilización de sus credenciales y datos personales, ya que con el crecimiento tecnológico se han abierto grandes ventanas que permiten el robo de información.

6.3. Trabajos futuros

El alcance del framework de seguridad se concentra en el desarrollo de una parte del backend, específicamente en la autenticación dentro de un ecosistema IoT, por lo que se sugiere:

- Analizar, diseñar y desarrollar un frontend, pudiendo ser este una aplicación web o una aplicación móvil la cual permita enlazarse a través de sus APIs REST con el framework desarrollado.

- Implementar un sistema de comunicación entre el bróker y los dispositivos IoT mediante el uso de un protocolo de comunicación liviano y seguro, evitando así posibles conflictos con los dispositivos de baja potencia y memoria.

BIBLIOGRAFÍA

Abomhara, M., & Koien, G. (2015). *Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks.* Obtenido de https://www.riverpublishers.com/journal_read_html_article.php?j=JCSM/4/1/4

Abufouda, M., & Abukwaik, H. (2017). *On Using Network Science in Mining Developers Collaboration in Software Engineering: A Systematic Literature Review.* Obtenido de https://www.researchgate.net/publication/321511153_On_Using_Network_Science_in_Mining_Developers_Collaboration_in_Software_Engineering_A_Systematic_Literature_Review

Afifi, M. H., Zhou, L., Chakrabartty, S., & Ren, J. (2018). *Dynamic Authentication Protocol Using Self-Powered Timers for Passive Internet of Things.* Obtenido de https://ieeexplore.ieee.org/document/8053739

Albors, J. (2018). *Seguridad en dispositivos IoT: ¿Aún a tiempo de ganar la batalla?* Obtenido de https://www.welivesecurity.com/la-es/2018/07/25/seguridad-iot-a-tiempo-ganar-batalla/

Al-Halabi, Y., Raeq, N., & Abu-Dabaseh, F. (2017). *Study on access control approaches in the context of Internet of Things: A survey.* Obtenido de https://ieeexplore.ieee.org/document/8308153

Ali, S., Hassan, H., Qayyum, S., Sohail, F., Tahir, S., Maqsood, S., & Adil, M. (2019). *Multi-agent System Using Scrum Methodology for Software Process Management.* Obtenido de https://link.springer.com/chapter/10.1007/978-981-13-6052-7_68

Aman, M. N., Taneja, S., Sikdar, B., Chua, K. C., & Alioto, M. (2018). *Token-Based Security for the Internet of Things With Dynamic Energy-Quality Tradeoff.* Obtenido de https://ieeexplore.ieee.org/document/8489899

Aman, M., Chua, K., & Sikdar, B. (2017). *Mutual Authentication in IoT Systems Using Physical Unclonable Functions.* Obtenido de https://ieeexplore.ieee.org/document/7924368

Ammar, M., Washha, M., Ramabhadran, G. S., & Crispo, B. (2018). *SlimIoT: Scalable Lightweight Attestation Protocol for the Internet of Things.* Obtenido de https://ieeexplore.ieee.org/document/8625142

Andy, S., Rahardjo, B., & Hanindhito, B. (2017). *Attack scenarios and security analysis of MQTT communication protocol in IoT system.* Obtenido de https://ieeexplore.ieee.org/document/8239179

Anicas, M. (2018). *Una introducción a OAuth 2.* Obtenido de https://www.digitalocean.com/community/tutorials/una-introduccion-a-oauth-2-es

Arasteh, S., Ashouri-Talouki, M., & Aghili, S. F. (2017). *Lightweight and Secure Authentication Protocol for the Internet of Things in Vehicular Systems.* Obtenido de https://ieeexplore.ieee.org/document/8488371

auth0. (2018). *OAuth 2.0 Authorization Framework.* Obtenido de https://auth0.com/docs/protocols/oauth2

Babar, S., Mahalle, P., Stango, A., Prasad, N., & Prasad, R. (2010). *Proposed Security Model and Threat Taxonomy for the Internet of Things (IoT).* Obtenido de https://link.springer.com/chapter/10.1007/978-3-642-14478-3_42

Bate, K. O., Kumar, N., & Khatri, S. K. (2017). *Framework for authentication and access control in IoT.* Obtenido de https://ieeexplore.ieee.org/abstract/document/8343530

Bertino, E., & Islam, N. (2017). *Botnets and Internet of Things Security.* Obtenido de https://www.computer.org/csdl/magazine/co/2017/02/mco2017020076/13rRUxZRbvu

Beydemir, A., & Soğukpinar, İ. (2017). *Lightweight zero knowledge authentication for Internet of things.* Obtenido de https://ieeexplore.ieee.org/document/8093410

certsuperior. (2016). *¿Qué es un certificado SSL?* Obtenido de https://www.certsuperior.com/QueesunCertificadoSSL.aspx

Chamoun, M., El-hajj, M., Fadlallah, A., & Serhrouchni, A. (2017). *Analysis of authentication techniques in Internet of Things (IoT).* Obtenido de https://ieeexplore.ieee.org/document/8242006

Chifor, B. C., Bica, I., & Patriciu, V. V. (2017). *Mitigating DoS attacks in publish-subscribe IoT networks.* Obtenido de https://ieeexplore.ieee.org/document/8166463

Choudhary, S., & Kesswani, N. (2018). *Detection and Prevention of Routing Attacks in Internet of Things.* Obtenido de https://ieeexplore.ieee.org/document/8456088

Cirani, S., Picone, M., Gonizzi, P., Veltri, L., & Ferrari, G. (2014). *IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios.* Obtenido de https://ieeexplore.ieee.org/abstract/document/6915840

Crossman, M. A., & Liu, H. (2015). *Study of authentication with IoT testbed.* Obtenido de https://ieeexplore.ieee.org/abstract/document/7225303

Dammak, M., Merad Boudia, O. R., Messous, M. A., Senouci, S. M., & Gransart, C. (2019). *Token-Based Lightweight Authentication to Secure IoT Networks.* Obtenido de https://ieeexplore.ieee.org/document/8651825

Das, A. K., Wazid, M., Yannam, A. R., Rodrigues, J. J., & Park, Y. (2019). *Provably Secure ECC-Based Device Access Control and Key Agreement Protocol for IoT Environment.* Obtenido de https://ieeexplore.ieee.org/document/8698231

Davis, M. J. (2016). *BCrypt: Hash Passwords Correctly.* Obtenido de http://davismj.me/blog/bcrypt/

De Ryck, P. (2019). *THE HARD PARTS OF JWT SECURITY NOBODY TALKS ABOUT.* Obtenido de https://www.pingidentity.com/en/company/blog/posts/2019/jwt-security-nobody-talks-about.html

Doaa, A., Esraa, A.-S., & Areej, A. (2017). *A Survey: Authentication Protocols for Wireless Sensor Network in the Internet of Things; Keys and Attacks.* Obtenido de https://ieeexplore.ieee.org/document/8250300

Eldefrawy, M. H., Pereira, N., & Gidlund, M. (2018). *Key Distribution Protocol for Industrial Internet of Things Without Implicit Certificates.* Obtenido de https://ieeexplore.ieee.org/document/8435930

El-hajj, M., Chamoun, M., Fadlallah, A., & Serhrouchni, A. (2017). *Taxonomy of authentication techniques in Internet of Things (IoT).* Obtenido de https://ieeexplore.ieee.org/document/8305419

Florea, I., Ruse, L. C., & Rughinis, R. (2017). *Challenges in security in Internet of Things.* Obtenido de https://ieeexplore.ieee.org/document/8123739

Gope, P., & Sikdar, B. (2018). *Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices.* Obtenido de https://ieeexplore.ieee.org/document/8382158

Griffin, P. H. (2017). *Secure authentication on the Internet of Things.* Obtenido de https://ieeexplore.ieee.org/document/7925274

Gurabi, M. A., Alfandi, O., Bochem, A., & Hogrefe, D. (2018). *Hardware based Two-Factor User Authentication for the Internet of Things.* Obtenido de https://ieeexplore.ieee.org/document/8450397

Harsha, M. S., Bhavani, B. M., & Kundhavai, K. (2018). *Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs.* Obtenido de https://ieeexplore.ieee.org/document/8554472

Hossain, M., & Hasan, R. (2017). *Boot-IoT: A Privacy-Aware Authentication Scheme for Secure Bootstrapping of IoT Nodes.* Obtenido de https://ieeexplore.ieee.org/document/8039048

Hossain, M., Hasan, R., & Skjellum, A. (2017). *Securing the Internet of Things: A Meta Study of Challenges, Approaches, and Open Problems.* Obtenido de https://ieeexplore.ieee.org/document/7979820

Hovsmith, S. (2017). *ADAPTING OAUTH2 FOR INTERNET OF THINGS (IOT) API SECURITY.* Obtenido de https://blog.approov.io/adapting-oauth2-for-internet-of-things-iot-api-security

Iglesias, J. (2014). *Diferencias entre autenticación y autorización.* Obtenido de https://programadorburgos.es/blog/diferencias-entre-autenticacion-y-autorizacion/

IoT World Online. (2018). *Las grandes estadísticas del Internet de las Cosas (IoT).* Obtenido de https://www.iotworldonline.es/las-grandes-estadisticas-del-internet-de-las-cosas-iot/

ISO 27001:2013. (2017). *¿Qué es el CIA (Confidencialidad, Integridad, Disponibilidad) en la seguridad de la información?* Obtenido de https://www.pmg-ssi.com/2017/07/cia-confidencialidad-integridad-disponibilidad-seguridad-de-la-informacion/

jalaluddin Khan, J. P. (2018). *An Authentication Technique Based on Oauth 2.0 Protocol for Internet of Things (IoT) Network.* Obtenido de https://www.semanticscholar.org/paper/An-Authentication-Technique-Based-on-Oauth-2.0-for-Khan-Li/e6b4a7b5a79d5ec3fa7ee9a3c5fa5de0e066811c

Jalaluddin, K., Ghufran, a. K., Mudassir, K., Asif, K., Amin, U. H., & Mohammad, S. (2018). *An Authentication Technique Based on Oauth 2.0 Protocol for Internet of Things (IoT) Network.* Obtenido de https://ieeexplore.ieee.org/document/8632587

Jenkov, J. (2014). *OAuth 2.0 Roles.* Obtenido de http://tutorials.jenkov.com/oauth2/roles.html

Jianming, C., Zuowen, Z., Hengzhong, L., & Rongquan, S. (2018). *An Improved User Authentication Protocol for IoT.* Obtenido de https://ieeexplore.ieee.org/document/8644677

Jones. (2012). *JSON Web Signature (JWS).* Obtenido de https://openid.net/specs/draft-jones-json-web-signature-04.html

Jones, M. (2015). *JSON Web Key (JWK).* Obtenido de https://tools.ietf.org/html/rfc7517#appendix-A.1

Jones, M. (2015). *JSON Web Signature.* Obtenido de https://www.rfc-editor.org/rfc/rfc7515.txt

Jones, M. (2015). *JSON Web Token (JWT).* Obtenido de https://www.rfc-editor.org/rfc/rfc7519.txt

Jones, M. (2016). *Proof of Possession Key Semantics for JSON Web Tokens.* Obtenido de https://tools.ietf.org/html/rfc7800

Kamble, A., & Bhutad, S. (2018). *Survey on Internet of Things (IoT) security issues & solutions.* Obtenido de https://ieeexplore.ieee.org/document/8399084

Khan, R. (2019). *Dynamically Configurable Architecture for User Identification and Authentication for Internet of Things Platform.* Obtenido de https://ieeexplore.ieee.org/document/8679282

Kim, H., & Lee, E. A. (2017). *Authentication and Authorization for the Internet of Things.* Obtenido de https://ieeexplore.ieee.org/document/8057722

Kitchenham, B. (2009). *Systematic literature reviews in software engineering – A systematic literature review.* Obtenido de https://www.sciencedirect.com/science/article/abs/pii/S0950584908001390

Krall, C. (2019). *¿Qué es y para qué sirve UML? Versiones de UML (Lenguaje Unificado de Modelado). Tipos de diagramas UML.* Obtenido de https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=688:i que-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46&Itemid=163

Kurera, C., & Navoda, D. (2018). *Node-to-Node Secure Data Transmission Protocol for Low-power IoT Devices.* Obtenido de https://ieeexplore.ieee.org/document/8615511

Lara, W. (2017). *SCRUM.* Obtenido de https://platzi.com/blog/metodologia-scrum-fases/

Lee, J.-Y., Lin, W.-C., & Huang, Y.-H. (2014). *A lightweight authentication protocol for Internet of Things.* Obtenido de https://ieeexplore.ieee.org/abstract/document/6839375

Li, C., & Yang, C. (2018). *(WIP) Authenticated Key Management Protocols for Internet of Things.* Obtenido de https://ieeexplore.ieee.org/document/8473449

Li, X., Niu, J., Bhuiyan, M. Z., Wu, F., Karuppiah, M., & Kumari, S. (2018). *A Robust ECC-Based Provable Secure Authentication Protocol With Privacy Preserving for Industrial Internet of Things.* Obtenido de https://ieeexplore.ieee.org/document/8110708

Ling, Z., Liu, K., Xu, Y., Jin, Y., & Fu, X. (2017). *An End-to-End View of IoT Security and Privacy.* Obtenido de https://ieeexplore.ieee.org/document/8254011

Liu, J., Xiao, Y., & Chen, P. (2012). *Authentication and Access Control in the Internet of Things.* Obtenido de https://ieeexplore.ieee.org/abstract/document/6258209

Losada Perez, I. (2015). *¿Cuál es la diferencia entre HTTP y HTTPS?* Obtenido de https://omicrono.elespanol.com/2015/02/diferencia-entre-http-y-https-2/

Malik, M., Dutta, M., & Granjal, J. (2019). *A Survey of Key Bootstrapping Protocols Based on Public Key Cryptography in the Internet of Things.* Obtenido de https://ieeexplore.ieee.org/document/8648325

Martínez, G., Camacho, G., & Gutierrez, B. (2010). *DISEÑO DE FRAMEWORK WEB PARA EL DESARROLLO DINÁMICO DE APLICACIONES.* Obtenido de https://www.redalyc.org/html/849/84917316032/

Node js Foundation. (2018). *Acerca de Node.js.* Obtenido de https://foundation.nodejs.org/

Núñez, R. (2017). *Las grandes estadísticas del Internet de las Cosas (IoT).* Obtenido de https://www.bebee.com/producer/@rafael-nunez-vazquez/las-grandes-estadisticas-del-internet-de-las-cosas-iot

Oliveira, C. T., Moreira, R., Silva, F. d., Sanches Miani, R., & Frosi Rosa, P. (2018). *Improving Security on IoT Applications Based on the FIWARE Platform.* Obtenido de https://ieeexplore.ieee.org/document/8432306

Ory. (2019). *https://www.ory.sh/docs/hydra/jwks.* Obtenido de https://www.ory.sh/docs/hydra/jwks

Ouaissa, M., Rhattoy, A., & Chana, I. (2018). *New Security Level of Authentication and Key Agreement Protocol for the IoT on LTE Mobile Networks.* Obtenido de https://ieeexplore.ieee.org/document/8629767

Owoude Bate, K., Kumar, N., & Kumar Khatri, S. (2017). *Framework for authentication and access control in IoT.* Obtenido de https://ieeexplore.ieee.org/document/8343530

Peng, H., Xianbin, W., & Weiming, S. (2018). *A Collaborative PHY-Aided Technique for End-to-End IoT Device Authentication.* Obtenido de https://ieeexplore.ieee.org/document/8419693

Peña, M. (2016). *Qué es el Internet de las Cosas y cómo afecta tu vida diaria.* Obtenido de https://es.digitaltrends.com/tendencias/que-es-el-internet-de-las-cosas/

Premalatha, T., & Duraisamy, S. (2017). *A certificate based authorization and protected application layer protocol for IoT.* Obtenido de https://ieeexplore.ieee.org/document/8117739

Purohit, K. C., Bisht, S., Joshi, A., & Bhatt, J. (2017). *Hybrid approach for securing IoT communication using authentication and data confidentiality.* Obtenido de https://ieeexplore.ieee.org/document/8344678

Rajashree, S., Soman, K. S., & Shah, P. G. (2018). *Security with IP Address Assignment and Spoofing for Smart IOT Devices.* Obtenido de https://ieeexplore.ieee.org/document/8554660

Rescorla, E. (2018). *The Transport Layer Security (TLS) Protocol Version 1.3.* Obtenido de https://www.rfc-editor.org/rfc/pdfrfc/rfc8446.txt.pdf

Rivas, G. (2018). *El Internet de las Cosas: 5 medidas para garantizar la seguridad del IoT.* Obtenido de https://www.gb-advisors.com/es/iot-seguridad-el-internet-de-las-cosas/

Saadeh, M., Sleit, A., Qatawneh, M., & Almobaideen, W. (2016). *Authentication Techniques for the Internet of Things: A Survey.* Obtenido de https://ieeexplore.ieee.org/abstract/document/7600206

Sahu, A. K., Sharma, S., Puthal, D., Pandey, A., & Shit, R. (2017). *Secure Authentication Protocol for IoT Architecture.* Obtenido de https://ieeexplore.ieee.org/document/8423911

Sandoval, K. (2017). *OAuth 2.0 – Why It's Vital to IoT Security.* Obtenido de https://nordicapis.com/why-oauth-2-0-is-vital-to-iot-security/

Sethia, D., Gupta, D., & Saran, H. (2018). *NFC Secure Element-Based Mutual Authentication and Attestation for IoT Access.* Obtenido de https://ieeexplore.ieee.org/document/8477059

Shah, T., & Venkatesan, S. (2018). *Authentication of IoT Device and IoT Server Using Secure Vaults.* Obtenido de https://ieeexplore.ieee.org/document/8455985

Sobers, R. (2018). *¿Qué es OAuth? Definición y cómo funciona.* Obtenido de https://www.varonis.com/blog/what-is-oauth/

Sridhar, S., & Smys, S. (2017). *Intelligent security framework for iot devices cryptography based end-to-end security architecture.* Obtenido de https://ieeexplore.ieee.org/document/8068718

Tschofenig, H. (2016). *Fixing User Authentication for the Internet of Things (IoT).* Obtenido de https://link.springer.com/article/10.1007/s11623-016-0582-1

Unwala, I., Taqvi, Z., & Lu, J. (2018). *IoT Security: ZWave and Thread.* Obtenido de https://ieeexplore.ieee.org/document/8373623

Valois, M. A. (2018). *Qué es internet de las cosas y cómo funciona.* Obtenido de https://www.hostgator.mx/blog/internet-de-las-cosas/

Velasco, R. (2019). *Hydra 9.0: conoce esta completa herramienta para romper contraseñas.* Obtenido de https://www.redeszone.net/2019/06/08/hydra-9-0-herramienta-romper-contrasenas/

Verma, H., & Chahal, K. (2017). *A review on security problems and measures of Internet of Things.* Obtenido de https://ieeexplore.ieee.org/document/8250560

Vicente, D. (2017). *Encriptación de password en NodeJS y MongoDB: bcrypt.* Obtenido de https://solidgeargroup.com/password-nodejs-mongodb-bcrypt?lang=es

Villagómez, C. (2017). *Protocolo TCP.* Obtenido de https://es.ccm.net/contents/281-protocolo-tcp

Wireshark. (2019). *About Wireshark.* Obtenido de https://www.wireshark.org/

Xiao, L., Wan, X., Lu, X., Zhang, Y., & Wu, D. (2018). *IoT Security Techniques Based on Machine Learning.* Obtenido de https://arxiv.org/abs/1801.06275

Xiaohong, F., & Baoli, N. (2017). *Security of a new lightweight authentication and key agreement protocol for internet of things.* Obtenido de https://ieeexplore.ieee.org/document/8230088

Yerlikaya, Ö., & Dalkılıç, G. (2017). *Security of message queue telemetry transport protocol.* Obtenido de https://ieeexplore.ieee.org/document/7960353

Yerlikaya, O., & Dalkilic, G. (2018). *Authentication and Authorization Mechanism on Message Queue Telemetry Transport Protocol.* Obtenido de https://ieeexplore.ieee.org/document/8566599

York, P. (2017). *Uber fue hackeado, datos de 57 millones de usuarios comprometidos.* Obtenido de https://www.fayerwayer.com/2017/11/uber-fue-hackeado-datos-de-57-millones-de-usuarios-comprometidos/

Zhen, L., Junzhou, L., Yiling, X., Chao, G., Kui, W., & Xinwen, F. (2017). *Security Vulnerabilities of Internet of Things: A Case Study of the Smart Plug System.* Obtenido de https://ieeexplore.ieee.org/document/7932855

Zhou, Y., Liu, T., Tang, F., & Tinashe, M. (2019). *An Unlinkable Authentication Scheme for Distributed IoT Application.* Obtenido de https://ieeexplore.ieee.org/document/8620304