Universidad de Manizales
Facultad de Ciencias e Ingeniería
Ingeniería de Sistemas y Telecomunicaciones

ANÁLISIS DE SEGURIDAD EN REDES LORA

DANIEL ANDRÉS GONZÁLEZ BETANCOURT

UNIVERSIDAD DE MANIZALES
FACULTAD DE CIENCIAS E INGENIERÍA
INGENIERIA DE SISTEMAS Y TELECOMUNICACIONES
MANIZALES
2024

ANÁLISIS DE SEGURIDAD EN REDES LORA

DANIEL ANDRÉS GONZÁLEZ BETANCOURT

Trabajo de Grado presentado como opción parcial para optar
al título de Ingeniero de Sistemas y Telecomunicaciones

Presidente
MICHAEL ALEJANDRO ROJAS
Docente de la Universidad de Manizales

UNIVERSIDAD DE MANIZALES
FACULTAD DE CIENCIAS E INGENIERÍA
INGENIERIA DE SISTEMAS Y TELECOMUNICACIONES
MANIZALES
2024

CONTENIDO
Pág.

LISTA DE FIGURAS

LISTA DE ANEXOS

GLOSARIO

- **Antenas LoRa:** son antenas diseñadas para trabajar con la frecuencia y modulación de la señal de LoRaWAN. (Sánchez ,2023,5)
- **Banda de frecuencia ISM:** son las bandas de frecuencia de espectro no licenciado que se utilizan en redes LoRaWAN. (Sánchez, 2023,5)
- **Cobertura amplia:** capacidad de las redes LoRaWAN de cubrir grandes áreas con una sola red. (Acebedo,2020,6)
- **Dispositivos IoT:** son dispositivos conectados a una red LoRaWAN para recopilar y transmitir datos. (González, 2019, 9)
- **Gateway:** es un dispositivo que actúa como punto de acceso a una red LoRaWAN y permite la comunicación entre dispositivos IoT y la nube. (Torres, 2018, 10)
- **Infraestructura de red:** son los elementos físicos necesarios para implementar una red LoRaWAN, como los Gateway y las antenas. (Garzón, 2020,18)
- **LoRaWAN:** Long Range Wide Área Network, es un protocolo de comunicación inalámbrica de baja potencia y largo alcance utilizado en redes de IoT. (Uribe, 2021, 7)
- **LPWAN:** son redes de baja potencia y largo alcance utilizadas en aplicaciones de IoT. (Pérez, 2022, 11)
- **Modulación LoRa:** técnica de modulación de la señal de radio utilizada en LoRaWAN para transmitir datos de forma inalámbrica. (Giraldo, 2023, 15)
- **Sensores:** son dispositivos que recogen información del entorno y la envían a través de una red LoRaWAN. (Castaño, 2027,20)
- **Seguridad en redes IoT**: es un conjunto de prácticas y técnicas para proteger la información y la integridad de los dispositivos y datos en redes LoRaWAN. (Perico, 2019, 20)
- **Topología estrella:** en la topología estrella, todos los dispositivos IoT están conectados a un único Gateway. (Vaca, 2020,7)
- **Topología malla:** en la topología malla, los dispositivos IoT pueden comunicarse directamente entre sí y no necesitan estar conectados a un Gateway. (Franco, 2018,7)

RESUMEN

El internet de las cosas (IoT), presenta un gran avance tecnológico y forma parte de la cuarta revolución industrial. Estas redes interoperables representan una interconexión de millones de dispositivos IP, que se comunican a través de diferentes protocolos de comunicación. Todo esto con el fin de compartir datos en tiempo real. Esta tecnología se integra con otras como, Big data, machine learning, inteligencia artificial para el análisis, y la toma de decisiones en base a todos los datos recolectados. En esta medida se acrecienta un problema de seguridad ineludible suscitado por todo el tráfico expuesto en las redes y con altos niveles de vulnerabilidades, lo que representa grandes riesgos de ataques informáticos, robos de información o corrupción de la misma. De esta manera se realizará un análisis de riesgos y vulnerabilidades en comunicaciones IoT, específicamente bajo la tecnología LoRa capturando los paquetes en su transmisión y validando la seguridad y el nivel de cifrado de los mismos. Se identificarán las falencias de la tecnología para de esta manera desarrollar mecanismos de seguridad cada vez más robustos que minimicen el nivel de riesgos informáticos presentes en la actualidad.

PALABRAS CLAVES: Internet de las Cosas, Protocolos de comunicación, Riesgos y Vulnerabilidades, Seguridad de la Información.

ABSTRACT

The Internet of Things (IoT) presents a great technological advance and is part of the fourth industrial revolution. These interoperable networks represent an interconnection of millions of IP devices, which communicate through different communication protocols. All this in order to share data in real time. This technology is integrated with others such as Big data, machine learning, artificial intelligence for analysis, and decision making based on all the data collected. To this extent, an unavoidable security problem arises caused by all the traffic exposed on the networks and with high levels of vulnerabilities, which represents great risks of computer attacks, theft of information or corruption thereof. In this way, an analysis of risks and vulnerabilities in IoT communications will be carried out, specifically under LoRa technology, capturing the packets in their transmission and validating their security and level of encryption. The shortcomings of the technology will be identified in order to develop increasingly robust security mechanisms that minimize the level of computer risks currently present.

KEYWORDS: Internet of Things, Communication Protocols, Risks and Vulnerabilities, Information Security.

INTRODUCCIÓN

Las vulnerabilidades del software representan un riesgo de seguridad que crece cada día, en la misma medida que evoluciona la tecnología y se despliegan nuevas formas de adquisición y transmisión de datos a través de la red de internet. Como tal, resulta imperioso poder identificar las vulnerabilidades propias de las tecnologías subyacentes en la nueva era del internet, conocida como el internet de las cosas.

El presente trabajo pretende analizar los riesgos y vulnerabilidades de las redes LoRa. Dichas redes son muy utilizadas para el despliegue del internet de las cosas y como tal su uso se ha incrementado exponencialmente. Poder identificar las vulnerabilidades resulta indispensable para comprender que es lo que debe mejorar la tecnología y de esta manera minimizar el riesgo de posibles ataques.

Este trabajo investigativo se limita a hacer un informe de resultados de las vulnerabilidades detectadas bajo pruebas y simulaciones de ataques a la comunicación empleada entre radios LoRa. Consta de doce capítulos, el primero trata sobre el área problemática en el que se realiza la descripción, delimitación y formulación del proyecto. En el segundo se hace referencia al objetivo general y los objetivos específicos, para describir los alcances generales del proyecto, en el cuarto capítulo se relacionan los antecedentes descritos por trabajos similares en todo el mundo. En el quinto capitulo se precisa la metodología usada para el desarrollo de la investigación, el tipo de trabajo y el procedimiento desarrollado; en el capítulo sexto se indican los resultados esperados, posteriormente, en el capítulo siete se establece el cronograma de actividades a desarrollar y en el capítulo ocho se precisa el presupuesto requerido y, a continuación se realiza todo el desarrollo del proyecto para finalmente exponer los resultados y las conclusiones así como las recomendaciones.

1. ÁREA PROBLEMÁTICA

1.1 DESCRIPCIÓN

Las vulnerabilidades representan falencias en los sistemas informáticos que permiten ser atacados sin el conocimiento del propietario del sistema. En la cuarta revolución industrial se han desarrollado numerosas tecnologías que convergen con las redes IoT para proporcionar una gran cantidad de tráfico. Dicho tráfico nace de los sensores a través de todo el mundo, los cuales se encuentran adquiriendo la información del ambiente cada segundo y enviando dichos datos a la red para ser procesados por las tecnologías emergentes como *Big data*, *machine learning*, inteligencia artificial, entre otras. Esta información representa un activo muy valioso para las organizaciones. Poder proteger sus datos es definitivamente una necesidad latente que se ve en peligro cada día por todas las vulnerabilidades presentes en los dispositivos IP y en los sistemas operativos que corren dichos dispositivos.

1.2 DELIMITACIÓN

Con el auge del internet de las cosas han emergido una serie de tecnologías para hacer posible el manejo de la información. Entre ellas están las redes LoRa, que tienen muy poco consumo permitiendo comunicaciones a grandes distancias (operan en la banda de frecuencias no licenciada de 900 MHz), sin embargo, no son inmunes a los ataques cibernéticos. El proyecto se delimitará a las vulnerabilidades presentes en comunicaciones por radiofrecuencia de los dispositivos del internet de las cosas. Específicamente las redes LoRa mencionadas anteriormente. Haciendo uso de dispositivos ESP32, un emisor y receptor LoRa con el módulo transceptor LYLYGO para un rango de operación de 15 km.

1.3 FORMULACIÓN

Las vulnerabilidades inherentes a los dispositivos que hacen parte del internet de las cosas y las puertas traseras que se encuentran en los firmwares de tales dispositivos representan un problema de seguridad que limita el despliegue de las tecnologías IoT. A pesar de que la tecnología tiene cierto grado de seguridad no es suficiente para todos los ataques que se despliegan.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar un análisis de seguridad para identificar los riesgos presentes debido a las vulnerabilidades inherentes en una comunicación con tecnología LoRa en un caso de uso de laboratorio.

2.2 OBJETIVOS ESPECÍFICOS

- Identificar los dispositivos idóneos y realizar el diseño necesario para un enlace punto a punto LoRa.
- Realizar pruebas de cobertura de la señal y el envío de información entre dos radios LoRa.
- Implementar un sniffer para la captura del tráfico emitida por el dispositivo.
- Realizar un ataque a la red LoRa que permita la captura de la información enviada.
- Analizar resultados de los paquetes y diagnosticar falencias.

3. JUSTIFICACIÓN

3.1 NOVEDAD

En la actualidad las vulnerabilidades encontradas en los dispositivos IoT son cada vez mayores, de la misma manera los adelantos tecnológicos han permitido el desarrollo de nuevas herramientas para contrarrestar los riesgos generados por las vulnerabilidades planteadas. Es indispensable el desarrollo de este tipo de proyectos investigativos para la detección temprana de las falencias en el software que desemboca en problemas de seguridad y en pérdida de información que en última instancia se considera como uno de los activos más importantes para las organizaciones. Con el presente proyecto se pretende abordar un análisis de riesgos y vulnerabilidades presentes en una red IoT, utilizando tecnología LoRa y dispositivos actualizados para ejecutar ataques y documentar los resultados obtenidos. Con este tipo de proyectos de investigación se desarrollan nuevas herramientas que permitan disminuir el riesgo de ataques cibernéticos en las tecnologías usadas y, por lo tanto, permitir el aumento en el despliegue tecnológico de las redes. Debido a que, las personas pueden confiar más en la pertinencia y seguridad de las mismas.

Garantizar la disponibilidad, confidencialidad e integridad de los datos es una tarea vital para el desarrollo tecnológico de la industria 4.0 y, todo inicia desde la puesta a prueba de las tecnologías liberadas y la realización de ataques que validen, reafirmen o tumben la seguridad de la red, solo así, habrá opciones de mejora y al final progreso tecnológico.

3.2 INTERÉS

Las redes LoRa hacen parte del despliegue tecnológico impulsado por el internet de las cosas y, en este orden de ideas todas las tecnologías que se despliegan de la misma, como la domótica, el *machine learning*, el *cloud computing*, representan cambios importantes en el estilo de vida de las personas. En esta medida, la seguridad de las comunicaciones que se ven intrínsecamente ligadas a estas tecnologías resulta indispensable para que todas las personas puedan disfrutar al máximo de los beneficios que traen.

3.3 UTILIDAD

Identificar las falencias emergentes de una tecnología es la base para nuevas actualizaciones del *firmware* de los dispositivos y, como tal, la solución de las vulnerabilidades detectadas. Por tal razón, el análisis de vulnerabilidades en redes IoT resultan imperioso para el aumento de la seguridad de los dispositivos y, la mejora continua de las redes.

4. MARCO TEÓRICO

4.1 MARCO CONCEPTUAL

Una red LoRa es una red de sensores de baja potencia y por ende de un bajo consumo energético que se comunican a través de radiofrecuencia de largo alcance. Esta tecnología se utiliza principalmente para Internet de las cosas (IoT) y aplicaciones que requieren grandes distancias de comunicación y deben consumir poca energía para operar correctamente. Sin embargo, como cualquier red, las redes LoRa también son susceptibles a ataques cibernéticos.

4.1.1 Arquitectura de la red. La arquitectura de una red LoRaWAN consta de tres componentes principales: los nodos finales, las puertas de enlace (gateways) y la red de servidores y aplicaciones. Los nodos finales son dispositivos IoT encargados de la adquisición de los datos y de su envió a través de la red LoRaWAN. Estos nodos son de bajo consumo de energía y se pueden alimentar con baterías durante varios años, lo cual lo hace muy atractivos para aplicaciones del internet de las cosas.

Las puertas de enlace son dispositivos que reciben los datos enviados por los nodos finales y los retransmiten a la red de servidores. Las puertas de enlace se conectan a Internet a través de una conexión de banda ancha, como Wi-Fi o Ethernet y a los nodos finales a través de enlaces de radiofrecuencia utilizando la tecnología LoRa. La red de servidores es el componente central de la red LoRaWAN. Los servidores manejan el enrutamiento de los datos y la gestión de la red, además de agregar una capa de seguridad a toda la red.

4.1.2 Riesgos y Vulnerabilidades de la red LoRaWAN. Para evaluar los riesgos y vulnerabilidades en una red LoRaWAN, se deben seguir los siguientes pasos:

- Identificación del diseño de la red: Identificar todos los dispositivos conectados a la red y las aplicaciones que se ejecutan en ellos. Como, por ejemplo, el número de nodos finales y la clase de información que están captando del ambiente, la capacidad y ubicación de Gateway, así como el medio de comunicación empleado con el nodo final, el servidor de la red y el nivel de seguridad del mismo
- Identificación de las amenazas: Identificar las posibles amenazas que pueden afectar la red y los dispositivos conectados, incluyendo ataques de denegación de servicio, interceptación de datos, explotación de vulnerabilidades y ataques de malware.
- Evaluación de vulnerabilidades: Entre las vulnerabilidades más importantes que se pueden destacar en los sistemas IoT utilizando la tecnología LoRaWAN, están:
- Ataques de denegación de servicio (DoS): Los ataques DoS son una amenaza común para cualquier red, pueden causar interrupciones en el servicio y hacer que

la red sea inutilizable. Los atacantes pueden enviar una gran cantidad de tráfico malicioso a la red para sobrecargarla, lo que ocasiona que los dispositivos de la red no puedan comunicarse adecuadamente.
- Intercepción de datos: La intercepción de datos es una amenaza grave para la seguridad de la red LoRaWAN. Los atacantes pueden interceptar y escuchar los datos transmitidos por los dispositivos de la red, lo que puede revelar información confidencial y comprometer la seguridad de la red. Un ejemplo de este ataque es el hombre en el medio.
- Ataques de repetición: Los atacantes pueden interceptar y grabar los datos transmitidos por los dispositivos de la red y luego reproducirlos en un momento posterior para imitar la actividad legítima. Esto puede permitir que los atacantes accedan a la red y comprometan la seguridad de la red.
- Ataques de spoofing: Los atacantes pueden suplantar la identidad de un dispositivo legítimo en la red para acceder a la red y comprometer su seguridad.
- Ataques de fuerza bruta: Los ataques de fuerza bruta son una amenaza para la seguridad de la red LoRaWAN. Los atacantes pueden intentar adivinar las contraseñas de los dispositivos de la red mediante la prueba de una gran cantidad de combinaciones posibles. Si un atacante tiene éxito en adivinar una contraseña, puede acceder a la red y comprometer su seguridad.

4.1.2.1. Evaluación de riesgos. Se debe evaluar el riesgo de cada amenaza identificada y de cada vulnerabilidad encontrada, teniendo en cuenta su probabilidad de ocurrencia y su impacto potencial.

4.1.2.2 Mitigación de riesgos. Una vez identificados los riesgos, es importante implementar medidas para mitigarlos, como parches de seguridad, medidas de autenticación, encriptación de datos, etc. El proceso de identificación de riesgos es el primer paso en la gestión de la seguridad de una red LoRaWAN. Una vez que se han identificado los riesgos, es importante implementar medidas de seguridad para mitigarlos. La mitigación de riesgos es el proceso de reducir o eliminar el riesgo de un evento no deseado o peligroso que pueda afectar la red. Algunas de las medidas de mitigación de riesgos que se pueden implementar para proteger la red LoRaWAN incluyen:
- Parches de seguridad: los parches de seguridad son actualizaciones de software diseñadas para corregir vulnerabilidades conocidas en el sistema. Es importante asegurarse de que la red LoRaWAN esté actualizada con los últimos parches de seguridad para reducir el riesgo de ataques que exploten vulnerabilidades conocidas.
-Medidas de autenticación: se deben implementar medidas de autenticación para asegurarse de que solo los dispositivos autorizados puedan acceder a la red. Esto puede incluir el uso de contraseñas fuertes, certificados digitales o autenticación de dos factores.
-Encriptación de datos: la encriptación de datos es un proceso de codificación de los datos para que solo puedan ser leídos por aquellos que tienen la clave para

decodificarlos. La encriptación de datos puede ayudar a proteger la privacidad y confidencialidad de los datos transmitidos por la red LoRaWAN.

Otras medidas de mitigación de riesgos que se pueden implementar incluyen la segregación de redes, la monitorización continua de la red y la implementación de políticas de seguridad robustas y actualizadas. Es importante tener en cuenta que la mitigación de riesgos es un proceso continuo y debe ser revisada y actualizada regularmente para asegurarse de que la red LoRaWAN siga siendo segura frente a nuevas amenazas.

4.1.3 Tecnologías Emergentes. El avance tecnológico en la era actual, marcada por la cuarta revolución industrial, ha dado lugar a la proliferación de tecnologías emergentes, entre las cuales destaca el Internet de las Cosas (IoT). Este concepto implica la interconexión de millones de dispositivos IP a través de diversos protocolos de comunicación, con el fin de compartir datos en tiempo real. Sin embargo, este nivel de interconexión también conlleva riesgos significativos en términos de seguridad, ya que expone las redes a vulnerabilidades que pueden ser explotadas por ciber atacantes, lo que podría resultar en la violación de la privacidad, el robo de datos o la corrupción de la información.

En este contexto, surge la necesidad de analizar y comprender los riesgos y vulnerabilidades asociados con las tecnologías IoT, con un enfoque particular en la tecnología LoRa, la cual, es una tecnología de comunicación de baja potencia muy popular en el despliegue del IoT por su bajo consumo de energía. Sin embargo, también presenta sus propios desafíos en términos de seguridad, que deben ser abordados de manera proactiva.

En el ámbito de la seguridad informática, las tecnologías emergentes juegan un papel crucial en la protección de datos y sistemas frente a las crecientes amenazas cibernéticas. Con la rápida evolución tecnológica y la expansión del Internet de las Cosas (IoT), la necesidad de fortalecer las medidas de seguridad se vuelve cada vez más apremiante. En este contexto, la integración de tecnologías como blockchain, inteligencia artificial y aprendizaje automático en el ámbito de la seguridad informática ofrece nuevas oportunidades y desafíos. A continuación, se mencionan los mas relevantes para este estudio:

-Blockchain, como una tecnología de registro distribuido y descentralizado, ofrece un alto nivel de seguridad al proporcionar un sistema inmutable y resistente a la manipulación de datos. Su aplicación en la seguridad informática abarca desde la autenticación de identidad hasta la protección de transacciones financieras, proporcionando un medio confiable para garantizar la integridad de los datos y la

trazabilidad de las acciones. Tal como lo expresa Mitzi[1], el blockchain es una forma de Tecnología Ledger Distribuida (DTL) en la que los datos se replican, comparten y sincronizan entre diferentes sitios sin un administrador central. Está compuesto por bloques conectados por un algoritmo criptográfico conocido como "hash", formando una cadena de bloques. El blockchain se puede desglosar en varias partes, como la base de datos distribuida, el bloque de transacciones, el nonce, los nodos y el algoritmo de encriptación. Cada una de estas partes desempeña un papel importante en el almacenamiento, compartición y conocimiento de la información contenida en la cadena.

En cuanto a la seguridad, el blockchain utiliza criptografía para garantizar la integridad de la información almacenada y distribuida en los nodos de la red. Además, la trazabilidad permite auditar todas las operaciones realizadas en la cadena de bloques, mientras que la privacidad protege la identidad de los usuarios. La transparencia se logra mediante la publicación de reglas que definen el funcionamiento del blockchain. En este orden de ideas se puede apreciar las características de seguridad y transparencia presente en el blockchain, lo que lo hace confiable para transacciones financieras entre muchas otras aplicaciones.

-**IDaaS (Identity as a Service).** Es un modelo de entrega de servicios de gestión de identidad que se basa en la nube. En este modelo, las organizaciones pueden externalizar la gestión de identidades y accesos a un proveedor de servicios en la nube en lugar de gestionarlos internamente. Caiza[2] afirma que “este modelo como enfoque principal se encarga de analizar las plataformas de nube de IoT populares a la luz de la solución de varios dominios de servicio, como el desarrollo de aplicaciones, la gestión de dispositivos, la gestión de sistemas, la gestión e heterogeneidad, la gestión de datos, las herramientas para el análisis, la implementación, la supervisión, la visualización y la investigación”. Al utilizar IDaaS, las organizaciones pueden beneficiarse de una mayor flexibilidad, escalabilidad y agilidad en la gestión de identidades, así como de la reducción de costos. Además, IDaaS puede ayudar a mejorar la seguridad al proporcionar un acceso más seguro a las aplicaciones y recursos.

-IA. El uso de inteligencia artificial (IA) y aprendizaje automático (ML) permite detectar y prevenir ataques cibernéticos de manera más efectiva. Estas tecnologías pueden analizar grandes volúmenes de datos en tiempo real para identificar patrones y anomalías que podrían indicar actividades maliciosas. Además, la IA y el ML pueden mejorar la capacidad de respuesta ante amenazas al automatizar procesos de detección y respuesta, reduciendo así el tiempo de reacción ante incidentes de seguridad.

---

[1]Mitzi. Vulnerabilidades existentes en técnicas tradicionales de almacenamientos de datos vs tecnologías emergentes como blockchain en entidades gubernamentales”, 2022. p. 44-46.

[2] Caiza, et al. Arquitectura basada en tecnologías emergentes y tecnología de monitoreo de tráfico de red, 2021. p. 4

Por otro lado, resulta preocupante que la IA también está siendo utilizada por ciberdelincuentes para realizar a taques, como lo menciona Ayerbe[3], "Además de la utilización del uso ofensivo de la IA por parte de los ciber atacantes para conocer patrones de comportamiento de las futuras víctimas, también puede utilizarse para romper más rápidamente contraseñas y captchas, construir malware que evite la detección, esconderse donde no puedan ser encontrados, adaptarse lo antes posible a las contramedidas que puedan tomarse, así como para la obtención automática de información utilizando métodos de procesamiento del lenguaje natural y la suplantación y generación de audios, vídeos y textos falsos. Los atacantes también están utilizando redes generativas para imitar patrones de tráfico de comunicaciones normales con el objetivo de distraer la atención de un ataque y encontrar y extraer datos sensibles rápidamente".

De la misma manera Ayerbe[4], afirma que los atacantes pueden emplear sistemas de Inteligencia Artificial (IA) no solo para tomar decisiones, sino también para influir en las decisiones de otros. Simplificando, un sistema de IA es esencialmente un software que utiliza datos, modelos y algoritmos de procesamiento. Sin embargo, si no se ha integrado la ciberseguridad desde la fase de diseño, estos sistemas pueden ser vulnerables a ataques cibernéticos dirigidos a los datos, modelos o algoritmos de la IA, lo que puede resultar en resultados no deseados o decisiones erróneas. Se han identificado diversas tácticas de ataque contra sistemas de IA, tanto durante el entrenamiento como durante la operación. Estas tácticas incluyen el acceso a los datos, la manipulación de los mismos, la evasión de los algoritmos y el uso de interfaces de modelo para obtener información. Por ejemplo, los atacantes pueden acceder a los datos de entrenamiento para crear un modelo sustituto, alterar los datos o el modelo directa o indirectamente, o incluso influir en la lógica del algoritmo de aprendizaje automático para obtener resultados maliciosos. Según un informe de Gartner para el 2022, se espera que el 30% de todos los ciberataques de IA utilicen tácticas como el envenenamiento de datos, el robo de modelos de IA o la manipulación de muestras para atacar sistemas de IA.

En resumen, la integración de tecnologías emergentes en seguridad informática, ofrecen nuevas oportunidades y desafíos en la protección de datos y sistemas contra las crecientes amenazas cibernéticas. Estas tecnologías proporcionan herramientas poderosas para garantizar la integridad, confidencialidad y disponibilidad de la información, al tiempo que permiten una gestión más eficiente y segura de identidades y accesos. Sin embargo, es de vital importancia reconocer que estas mismas tecnologías pueden ser aprovechadas por los ciberdelincuentes para perpetrar ataques sofisticados. Por lo tanto, es fundamental abordar estos

[3] Ibíd., p. 7
[4] Ayerbe. La ciberseguridad y su relación con la inteligencia artificial,2020. p. 4

riesgos de manera proactiva, integrando la ciberseguridad desde la fase de diseño de los sistemas y adoptando medidas de protección adecuadas.

4.1.3 Análisis Comparativo de Tecnologias IoT. El análisis comparativo de las vulnerabilidades en redes LoRa con otras tecnologías de IoT es crucial para contextualizar los hallazgos dentro del marco de desarrollo de esta tecnología. Si bien, las redes LoRa ofrecen ventajas significativas para el despliegue de IoT en cuanto a su bajo consumo y comunicaciones de larga distancia, también presentan riesgos de seguridad que deben abordarse de manera proactiva.

En comparación con otras tecnologías de IoT, como Zigbee o Wi-Fi, las vulnerabilidades en las redes LoRa pueden ser diferentes debido a sus características. Por ejemplo, mientras que Wi-Fi puede ser más vulnerable a ataques de fuerza bruta debido a su amplia disponibilidad y facilidad de acceso en el espectro y uso generalizado, las redes LoRa pueden ser susceptibles a ataques de interferencia debido a su uso de bandas de frecuencia no licenciadas y comunicaciones de larga distancia.

Además, es importante considerar el contexto de implementación de estas tecnologías. Por ejemplo, en entornos industriales donde se despliegan redes de sensores para monitorear equipos o procesos, las vulnerabilidades en las redes LoRa pueden tener repercusiones significativas en la seguridad operativa y la integridad de los datos en procesos críticos. En contraste, en aplicaciones domésticas donde se utilizan dispositivos IoT para automatizar el hogar, las preocupaciones de seguridad pueden centrarse más en la privacidad y la protección de datos personales. Como lo expresa Rueda & Talavera[5], las investigaciones sobre redes IoT centran sus esfuerzos en estudiar y generar soluciones para que los nodos sensores hagan uso óptimo de sus recursos, como, por ejemplo, uso de la energía, comunicación de corto alcance y conectividad entre nodos sensores. De este tipo de investigaciones nacieron protocolos y tecnologías de comunicación que optimizan el uso de energía como LoRaWAN y ZigBee, este último basado en el estándar IEEE 802.15.4.

A continuación, se describen las principales tecnologías inalámbricas implementadas en el IoT y su funcionalidad desde un enfoque comparativo en cuanto a la seguridad que ofrecen ante los ataques cibernéticos:

- Bluetooth. Tecnología de comunicación inalámbrica de corto alcance que permite la transferencia de datos entre dispositivos electrónicos. Fue desarrollada para reemplazar los cables de conexión entre dispositivos, como auriculares.

[5] Rueda & Talavera. Similitudes y diferencias entre Redes de Sensores Inalámbricas e Internet de las Cosas: Hacia una postura clarificadora. 2017. p-2

La tecnología Bluetooth funciona mediante ondas de radio de corto alcance, lo que permite que los dispositivos se comuniquen entre sí sin necesidad de cables físicos. Utiliza un protocolo de comunicación estándar para establecer y mantener conexiones inalámbricas seguras y confiables. Bluetooth se utiliza ampliamente en la industria de la electrónica de consumo debido a su conveniencia y facilidad de uso**. U**tiliza cifrado AES de 128 bits para proteger la privacidad y confidencialidad de la información transmitida. Admite varios métodos de autenticación para verificar la identidad de los dispositivos emparejados.

- Zigbee. El estándar ZigBee es un conjunto de protocolos diseñados para redes inalámbricas de corto alcance y baja velocidad de datos. Desarrollado por la Alianza ZigBee en 2002, ZigBee utiliza el estándar IEEE 802.15.4 para sus capas física (PHY) y de acceso al medio (MAC), y agrega capas de red y aplicación para crear una arquitectura completa.

Una de las características principales de ZigBee es su enfoque en el bajo consumo de energía, lo que lo hace ideal para dispositivos alimentados por batería. Además, busca ofrecer dispositivos asequibles y una instalación y mantenimiento económicos. Operando en bandas de frecuencia como 868 MHz, 915 MHz y 2.4 GHz, ZigBee puede transferir datos a una velocidad de hasta 250 Kbps. Según Matta[6], la seguridad es una preocupación fundamental en ZigBee. Utiliza el estándar AES (Advanced Encryption Standard) para encriptación y autenticación de datos, lo que garantiza la integridad y confidencialidad de la información transmitida. ZigBee también incorpora métodos para la distribución segura de claves, asegurando que solo dispositivos autorizados puedan acceder a la red. En cuanto a la autenticación, ZigBee admite la autenticación de dispositivos y datos. Para asegurar la integridad de los mensajes, se utiliza un código especial llamado Código de Integridad de Mensaje (MIC), que se acompaña con cada mensaje transmitido. Esto permite verificar la autenticidad de los datos recibidos y proteger contra modificaciones no autorizadas.

-**Wifi.** Es una tecnología que dispone de una alta tasa de transferencia de datos, es por ello que mantiene un mayor consumo de energía que otras tecnologías. Villamar[7] afirma que, aunque esta tecnología tiene muy buenas ventajas, también presenta una limitación a resolver que es la seguridad, debido a que, por la simplicidad de su instalación de forma abierta, pueden ser desplegadas incluso por hackers con el fin de robar información de los usuarios de la red.

[6] Matta. Sistema de monitoreo Vehicular como herramienta Para el sistema de seguridad Ciudadana utilizando Tecnología zigbee. 2018. p- 27-56
[7] Villamar. Análisis comparativo de tecnologías wsn en Base a la seguridad y su forma de operación, Aplicado al entorno iot. 2022. p-27

En cuanto a su seguridad cabe resaltar que su fácil despliegue lo convierte en un objetivo común para ataques cibernéticos. El WiFi ofrece varios protocolos de encriptación, como WEP (Wired Equivalent Privacy), WPA (WiFi Protected Access) y WPA2, cada uno con diferentes niveles de seguridad. La seguridad en las redes WiFi es una preocupación importante debido a su naturaleza inalámbrica y su omnipresencia en entornos domésticos y empresariales.

-**IoT.** El despliegue rápido de la tecnología IoT se apoya en diversas tecnologías de comunicación inalámbrica que han facilitado su adopción. Sin embargo, para garantizar un funcionamiento óptimo, es fundamental un adecuado procesamiento de datos, que incluye la recolección, almacenamiento y análisis inteligente de la información proveniente de los sensores. Además, se requiere minimizar el consumo de energía para prolongar la vida útil de las baterías y reducir costos operativos. Por último, la seguridad de la información juega un papel crítico en la protección de los datos frente a posibles ataques cibernéticos, asegurando la integridad y confidencialidad de la información transmitida y almacenada.

Villamar[8] realizo un estudio comparativo de las tecnologías involucradas en el despliegue de IoT y desarrolló un análisis que pondera la relevancia de estos requisitos en el entorno del IoT. La heterogeneidad es fundamental (24%) para la interoperabilidad de dispositivos. La escalabilidad (18%) garantiza una respuesta eficiente al integrar nuevos dispositivos. La minimización de costos (16%) y el consumo de energía (14%) son vitales para una implementación rentable y eficiente. El procesamiento de datos (12%) y la seguridad de la información (9%) son esenciales para garantizar un manejo adecuado de la información y protegerla contra posibles amenazas. La transmisión de datos (9%) y la calidad del servicio (7%) también son importantes para una comunicación eficiente y una experiencia de usuario satisfactoria.

En la siguiente figura se puede apreciar el análisis comparativo de las tecnologías descritas anteriormente:

[8] Ibíd., p. 44-50

Figura 1. Análisis comparativos de tecnologias.

Parámetro	Bluetooth	WiFi	Zigbee	LoRa
Velocidad	1 Mbps	54 Mbps	250 Kbps	0.3 - 50 kbps
Nodos/dispositivo	7	32	64 mil	Hasta 1 millón
Latencia	Up to 10s	Up to 3s	30 ms	10 - 100 ms
Variedad de datos	Audio	Gráficos	Películas	Datos, voz, comandos de voz
				Pequeños paquetes de datos
Duración de batería	1 semana	48 horas	1000 días	Hasta 10 años
Costos	Muy bajos	Altos	Bajos	Bajos
Cobertura	10 metros	100 metros	70-100 m	Hasta 15 km en entornos urbanos, hasta 50 km en áreas rurales o abiertas
Nivel de seguridad	E1-SAFER+	WEP/WPA/WPA2 AES	AES	AES

Fuente: Villamar. Análisis comparativo de tecnologías wsn en Base a la seguridad y su forma de operación, Aplicado al entorno iot. 2022. P. 50

La comparación entre las tecnologías de comunicación inalámbrica como WiFi, Bluetooth, LoRaWAN y Zigbee revela diferencias significativas en términos de seguridad ante ataques cibernéticos. En primer lugar, es importante destacar que LoRa no proporciona un cifrado incorporado como AES, mientras que LoRaWAN sí incluye medidas de seguridad como cifrado AES, lo que brinda encriptación en las comunicaciones y autenticación de dispositivos y datos.

Por otro lado, Zigbee es la tecnología que ofrece un mayor nivel de seguridad, ya que maneja el algoritmo de seguridad AES. Esto permite garantizar la integridad y confidencialidad de la información transmitida y almacenada. En contraste, aunque WiFi también integra el cifrado AES en su protocolo de seguridad WPA2, aún es vulnerable a ciertos tipos de ataques cibernéticos, como denegación de servicios, inundación de datos o secuestro de sesión, que afectan la disponibilidad de la red.

Además de la seguridad, las tecnologías también difieren en otros aspectos, como el consumo energético y la tasa de transferencia de datos. Por ejemplo, en la figura se puede apreciar la proporcionalidad entre el consumo energético y la tasa de transferencia de datos, mostrando que cada tecnología tiene sus propias características en este aspecto.

En conclusión, al elegir una tecnología de comunicación inalámbrica para aplicaciones de IoT, es fundamental considerar no solo la seguridad ante posibles ataques cibernéticos, sino también otros factores como el consumo energético, la velocidad de transferencia de datos y la compatibilidad con los requisitos específicos de la aplicación. La tecnología LoRa permite abarcar grandes distancias y un bajo consumo energético, pero no ofrece la capa de seguridad adecuada para aplicaciones que manejen datos críticos, a diferencia de zigbee que, si ofrece ese nivel se seguridad y también representa un bajo consumo energético, aunque no abarca las distancias que puede cubrir LoRa.

4.2 MARCO LEGAL

En Colombia, la implementación y uso de tecnologías de Internet de las Cosas (IoT, por sus siglas en inglés) y de redes LoRaWAN están regulados por varias leyes y regulaciones. En primer lugar, la Constitución Política de Colombia reconoce la importancia de las tecnologías de la información y la comunicación (TIC) para el desarrollo del país, en el artículo 77. Además, la Ley 1341 de 2009 establece el marco legal para el desarrollo de la sociedad de la información en Colombia y fomenta el uso de las TIC para el acceso universal a la información y la promoción de la competitividad.

En cuanto a las redes LoRaWAN, la Comisión de Regulación de Comunicaciones (CRC) es la entidad encargada de regular el espectro radioeléctrico en Colombia. La CRC estableció la Resolución 5050 de 2016, que regula el uso de las bandas de frecuencia para redes de baja potencia y largo alcance, incluyendo las redes LoRaWAN. Esta resolución establece las condiciones y requisitos para el uso de estas redes, como la necesidad de obtener una autorización previa de la CRC y el cumplimiento de las normas técnicas aplicables.

Además, el Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC) es la entidad encargada de promover el desarrollo de las TIC en Colombia. El MinTIC ha lanzado varios programas y proyectos para fomentar el uso de tecnologías IoT y redes LoRaWAN en el país, como el programa "IoT y Big Data para la productividad" y el proyecto "Ciudades Inteligentes".

En resumen, en Colombia existen varias leyes y regulaciones que establecen el marco legal para el desarrollo de las TIC y fomentar el uso de tecnologías IoT y redes LoRaWAN. La CRC y el MinTIC son las entidades encargadas de regular y promover el uso de estas tecnologías en el país.

4.3 ANTECEDENTES

Con base en los riesgos y vulnerabilidades presentes en los dispositivos IoT se realiza una investigación de los antecedentes referentes a la problemática planteada, en el presente proyecto se abordan las redes que implementan la tecnología LoRa. Esta tecnología opera en la banda de los 900Mhz**.**

- En un análisis de seguridad desarrollado por González et al.,[9] se presenta el diseño de una red LpWan con un dispositivo IoT y se ejecutan las pruebas de validación con herramientas de Kali Linux (hydra y medusa). De esta manera se logra determinar que la tecnología sigfox, en su backend si es segura gracias al uso del protocolo https, aunque en el aplicativo no tiene restricción para el ingreso por intentos fallidos por lo que resulta vulnerable para ataques de fuerza bruta. También se aprecia que los dispositivos sigfox no cuentan con actualizaciones de firmware, lo que genera una vulnerabilidad al tener las mismas contraseñas de acceso durante toda su vida útil. Pese a todo esto, dicha tecnología cuenta con una infraestructura de comunicación muy robusta y se prevé mejoras en cuanto a la limitación del número de intentos para ingresar al backend.

- Rondón & Bravo[10] desarrollaron una comunicación con un dispositivo RTL-2832U implementando la tecnología LoRa y haciendo uso de un sniffer para la captura de paquetes en la transmisión. Con esto, lograron capturar información y verificar que la única sección del mensaje que se encuentra cifrada son los datos. El contador y la dirección del dispositivo están en texto plano. Lo que genera un riesgo ya que dicha información es útil para los atacantes.

- Fernández & Robayo[11] implementaron una metodología scrum para atrapar los paquetes en una red LoRa, diseñaron e implementaron una red inalámbrica LoRa con ESP32 LoRa y un código con el ide de Arduino para enviar datos a través de la red. De la misma manera, utilizaron un sniffer para la captura de paquetes y el software wireshark para poder descifrar todo el tráfico que entra y sale de la red LoRa, Puertos de origen y destino, si se usa TCP o UDP, direcciones IP de Destino y Emisor, etc. En conclusión, obtuvieron acceso a las tramas enviadas a través de la red lo que evidencia los problemas de seguridad que tiene la tecnología.

9 González, et al. Análisis de seguridad en redes LPWAN para dispositivos IoT, 2019. p. 252-261.
10 Rondón & Bravo. Esquema de seguridad de datos entre los nodos y el gateway en una red LoRaWAN, 2020. p. 38-61
11 Fernández & Robayo. Ensamble de un adaptador inalámbrico para el desarrollo del software sniffer en una red LoRawan y análisis con wireshark, 2022. p. 51-79

- Vega[12] Desarrolló un montaje de red LoRa con un kit de desarrollo LoRa Technology Evaluation Kit 800. Este sistema se configuró en software y hardware utilizando node.js como lenguaje de programación y la base de datos MongoDb para el almacenamiento de los datos obtenidos. Con esto, se realizó y comprendió el funcionamiento de la red utilizando tecnología LoRa. En síntesis, este trabajo permite identificar el montaje y funcionamiento de una red LoRa utilizando microcontroladores y lenguajes de programación para la adquisición de datos, también se enfatiza en sistema operativo Linux para su correcto funcionamiento, demostrando la asequibilidad del despliegue de la tecnología.

- Rodríguez[13] define el protocolo LoRa y recrea escenarios que muestran algunas de las vulnerabilidades y ataques presentes en la versión 1.0.2 de este protocolo. Adicionalmente, se desarrolla la extracción y los procesos que se deben activar, instalando una red LAF y configurando el Gateway. Mediante la recreación del ataque Password Cracking se pretende demostrar que es posible recuperar la AppKey y las claves de sesión (NwkSKey, AppSKey) de un nodo a partir de la captura de los mensajes JoinRequest y Join-Accept utilizando el lenguaje de programación phyton. En esta medida se indica que los paquetes LORA que capture el Gateway y que dirija al puerto 1702 del Network Server (localhost), van a ser duplicados para almacenarlos en la base de datos a través del puerto 1800 y reenviarlos al Network Server a través del puerto 1700 utilizando la nube (Chipstack). Con esto, se redirigen los paquetes para enlazarlos a la base de datos y mostrar el password cracking. La herramienta LAF resulta bastante útil para auditar y encontrar fallos en las redes y se demuestra la importancia de encontrar fallos y vulnerabilidades para poder corregirlos.

- Peña[14] realiza una comparación entre la tecnología LoRa Edge y su versión previa, implementando una red LoRa. A continuación, se realizaron experimentos en diferentes circunstancias, variando las frecuencias de transmisión y las localizaciones geográficas para determinar el comportamiento de las transmisiones realizadas y en esta medida discernir entre los desempeños de cada tecnología empleada. Peña determino que en el empleo de tecnología LoRa el chip LR 1110 arroja buenos resultados cuando se necesitan aplicaciones de localización. Se concluyo también que la gama LoRa Edge ofrece soluciones versátiles al poder utilizar distintos rangos de escaneo RF.

---

12 Vega. Desarrollo de un manual de prácticas para el uso de equipos LoRawan en redes de sensores inalámbricos, 2020. p. 27-57

13 Rodríguez. Demostración de los ataques Password Cracking y Spoofing en redes LoRawan, 2022. p. 59 - 100

14 Peña. Comparación y validación experimental de la gama LoRa Edge sobre redes LoRawan, 2022. p. 35 - 69

- Del Rosario & Meza[15] desarrollan un proyecto con enfoque institucional en el cual plantearon la creación de una red IoT bajo la tecnología LoRa para la comunicación local en la universidad de Guayaquil. Dicha red consta de un Gateway que actúa como concentrador de datos para nodos finales, una antena que proporciona un rango de alcance muy alto para la comunicación. También tiene una interfaz web para el registro y un servidor web para la información de los diferentes proyectos realizados en la universidad en tiempo real. Se realizan montajes de hardware como son un Arduino ESP32. La conclusión final del proyecto ratifica la red LoRa como la tecnología con más despliegue en el Internet de las cosas gracias a su poco consumo y gran cobertura. Ideal para dar conectividad a gran cantidad de sensores.

- Rodríguez[16] realiza un montaje completo y configuración de una red privada LoRA utilizando sensores de temperatura, agua, humedad y nivel, así como un equipo Gateway ODU Macro V1.5 – 8. Describe el proceso de montaje y configuración de cada dispositivo, así como el lenguaje de programación empleado. En este caso, JavaScript. Adicionalmente se hace uso de un sistema de control y monitoreo scada para el proyecto creado. Este sistema permite visualizar en tiempo real el estado de las variables empleadas y registradas a través de los sensores. Se demuestra el uso de la tecnología LoRa para la adquisición y comunicación de datos en entornos industriales, gracias al sistema scada. Se concluye con la eficacia de la tecnología y se realiza una comparación con las redes 5G en donde a futuro, dichas tecnologías se pueden complementar, ya que, si se deben captar datos de dispositivos muy lejanos, se podrá utilizar LoRa para la comunicación desde una puerta de enlace con los dispositivos finales. Mientras que la red 5G podrá comunicar dicha puerta de enlace con el servidor de red.

- Pérez[17] aborda un proyecto de Smart city con tecnología LoRa. Se enfatiza en la necesidad de proteger los datos en una aplicación tan sensible como la comunicación de información a lo largo de una ciudad. Las redes LoRa incorporan varias capas de cifrado con algoritmos AES 128. Se emplean módulos RN 2483. Se estudian las cabeceras de las tramas en los paquetes transmitidos a lo largo de la red. También se especifican las diferencias con la tecnología LoRa ya que mientras que en las redes LoRa el único intercambio de mensajes se realiza entre los nodos y el gateway, con LoRa

15 Del Rosario & Meza. Diseño e implementación de una red iot basado en LoRawan para el uso de proyectos desarrollados por estudiantes de la universidad de guayaquil, con un sistema de control de acceso a la red IoT, 2021. p. 34-77
16 Rodríguez. Red privada LoRaWAN para el ámbito de aplicaciones en la Industria 4.0, 2021. p.57-92
17 Pérez. Evaluación de LoRa/LoRaWAN para escenarios de Smart City. 2017. p. 22-56

el servidor de red se comunica con los sensores a través del gateway. El proyecto concluye con la falta de mecanismos de seguridad en la tecnología LoRa, mientras que en LoRaWAN se mejoran dichas falencias.

- Bravo et al.,[18] desarrollan un proyecto investigativo en el que se realizan ataques a una red de sensores inalámbrica de alcance amplio o LoRa. Para dichos ataques se implementa un sniffing y replay. Estas herramientas permiten analizar los paquetes enviados de manera inalámbrica desde el dispositivo final hasta los nodos. El Sniffer se implementa utilizando el hardware RTL2832U y se visualiza en Wireshark, a través de GNU-Radio. El proyecto concluye con un informe dando como resultado el éxito en los ataques realizados. Con esto se reafirma que se puede amenazar la disponibilidad y confidencialidad de los datos a través de ataques de replay con verificación en el LoRa server utilizando hardware HackRF One y GNU-Radio. A pesar que la tecnología LoRa tiene contadores para evitar ataques de replay, con los equipos adecuados se puede llegar a vulnerar la red llegando a realizar la denegación del servicio del nodo en el servidor.

Con los antecedentes mencionados anteriormente resulta evidente el interés en todo el mundo por la tecnología LoRa y como tal por la seguridad que puede proporcionar en las comunicaciones inalámbricas de largo alcance. Esta comunicación de largo alcance y su bajo consumo son los aspectos más atractivos para utilizar la red en los sistemas IoT, ya que, todos los dispositivos finales del internet de las cosas se encuentran en la mayoría de los casos a grandes distancias de su nodo principal.

Los resultados obtenidos de los antecedentes mencionados demuestran las fallas de seguridad inherentes en la tecnología y los diferentes medios y herramientas utilizadas para detectarlas y documentarlas. Se tiene mucho trabajo aun por realizar, pues cada vez se liberan más herramientas para realizar nuevos ataques a la red y de esta manera, violentar sus mecanismos de seguridad para robar información sensible.

18 Bravo et al., Desarrollo y prueba de un Sniffer en tiempo real de una red LoRaWAN usando GNU-Radio. 2019. p 185-194

5. METODOLOGÍA

5.1 TIPO DE TRABAJO

Este proyecto corresponde a una investigación **experimental** ya que, según Rojas[19], se caracteriza por la aplicación de estímulos a la unidad experimental, se observa la reacción y se registran los resultados. Consiste en una relación causa – efecto. En este caso constituye la realización de ataques (como estímulos) a la unidad experimental (redes LoRa) y la documentación de los resultados obtenidos

En él se incluirán aspectos de las disciplinas de las telecomunicaciones, desarrollo de software, seguridad informática.

El proyecto está avalado por el Grupo de Investigación y desarrollo en Informática y Telecomunicaciones, GIDIT en su línea seguridad en IoT.

5.2 PROCEDIMIENTO

El proyecto se realizará en 5 fases, así:

5.2.1 Fase 1. Identificación del despliegue de una red LoRa, así como, el diseño de la red y cobertura. El diseño y montaje de la red constituyen el ambiente adecuado para la realización de las pruebas experimentales. Se divide en las siguientes actividades

- **Actividad 1. Diseño**. Elaboración del diseño, distribución y determinación de los equipos específicos necesarios para el despliegue de la arquitectura de red. Se deben determinar todos los dispositivos necesarios que conforman la red LoRa, así como sus respectivas configuraciones. Esto, para implementar el ambiente de pruebas necesario para cumplir con el objetivo planteado.

- **Actividad 2. Implementación**. Después de haber realizado el diseño correspondiente de la red, se deben configurar los equipos utilizados. Para esto, se hace uso del ide de desarrollo de Arduino. Se realiza la instrumentación electrónica de los sensores y el acondicionamiento de las señales a la tarjeta de adquisición de datos y el cableado correspondiente.

5.2.2 Fase 2. Realización de pruebas de cobertura de la señal. Para poder llevar a cabo el desarrollo del objetivo del proyecto se deben realizar pruebas de ataques en un entorno de laboratorio controlado. En este caso, una red IoT implementando

19 Rojas. Tipos de Investigación científica: Una simplificación de la complicada incoherente nomenclatura y clasificación. 2015. p.7

la tecnología de comunicación inalámbrica LoRa. Esta fase conlleva las siguientes actividades.

- **Actividad 1. Pruebas de Comunicación.** Se ejecutan pruebas de conectividad entre los dispositivos empleados para verificar la comunicación y la transmisión y recepción de información entre ellos.

5.2.2 Fase 3. Implementación de un sniffer para la captura del tráfico emitida por el dispositivo. Se implementarán herramientas de captura de tráfico para poder lograr la interceptación de paquetes en la comunicación.

- **Actividad 1. Sniffer.** Se implementará un sniffer para capturar los paquetes que se generen en la red

5.2.2 Fase 4. Realización de un ataque a la red LoRa que permita la captura de la información enviada. Después de haber puesto en marcha la red y validado la comunicación y la captura de paquetes se desplegarán los ataques a la red

- **Actividad 1. Ataques**. Se deben realizar ataques a la red anteriormente implementada. Para tal fin, se debe hacer uso de herramientas cibernéticas. En este caso el IDE de desarrollo de Arduino. Adicionalmente, se empleará un sniffer para la captura de los paquetes enviados en la transmisión y de esta manera poder interceptar las comunicaciones.

- **Actividad 2. Análisis.** Después de capturar los ataques con las herramientas especificadas se debe proceder con la actividad de análisis en donde se desglosa la información capturada y se realizan las validaciones correspondientes. Para esta actividad se dispondrá de la técnica de observación, esto conlleva a registrar datos e interpretarlos.

5.2.3 Fase 5. Análisis de resultados de los paquetes y diagnostico de falencias. Por último, después de haber ejecutado las pruebas de ataques y el análisis correspondientes se desarrollará el informe de resultados registrando los hallazgos encontrados.

- **Actividad 1. Informe**. Se identifican las vulnerabilidades presentes descubiertas en la fase de ejecución de ataques, se documentan las fallas y se realiza el informe de análisis de resultados para dar pie al cumplimiento del objetivo trazado del proyecto.

6. RESULTADOS ESPERADOS

El objetivo general de este proyecto consiste en desarrollar un análisis de seguridad para identificar y predecir los riesgos presentes debido a las vulnerabilidades inherentes en una comunicación con tecnología LoRa utilizando radios Lylygo. Como tal, se espera poder identificar falencias en la tecnología. Se pretende realizar un informe de vulnerabilidades encontradas en los sistemas IoT bajo la red LoRa, documentarlos y proponer posibles soluciones.

Objetivo No.	**Resultado esperado**	**Medio de verificación**	**Semana de obtención**
1	Montaje de la red.	Captura de información a través los radios LoRa Informé de avance N 1	4
2	Comunicación exitosa entre los dispositivos finales de la red implementada	Conectividad y latencia de la red. Informé de avance N 2	8
3	Montaje del sniffer y captura de paquetes en la transmisión.	Arduino IDE para la captura y análisis de los paquetes interceptados. Informé de avance N 3	12
4	Códigos implementados para efectuar ataques a la red	Realización de ataques exitosos. Informé de avance N 4	14
5	Informe con los hallazgos encontrados	Nivel y cantidad de vulnerabilidades presentes. Informé de avance N 5	16
6	Análisis de riesgos y vulnerabilidades detectados	Análisis de los riesgos encontrados, conclusiones del proyecto N 6	20

7. CRONOGRAMA DE ACTIVIDADES

ACTIVIDAD	**SEMANAS**									
	2	4	6	8	10	12	14	16	18	20
Fase 1.										
Actividad 1	x	x								
Actividad 2			x	x						
Fase 2.										
Actividad 1					x	x				
Fase 3.										
Actividad 1						x	x			
Fase 4.										
Actividad 1							x	x		
Actividad 2								x	x	
Fase 5.										
Actividad 1									x	x

8. PRESUPUESTO

Ítem	Unidad	Cantidad	Valor unitario	Valor Total ($000)	Aporte ($000)		
					Personal	U. M.	Otro
Autor	**hora**	**x**	**y**	**x*y**	**x*y**		
Autor	hora	220	20	4.400	4.400		
Asesoría	hora	40	12,5	500		500	
Internet	hora	220	1,2	264	264		
Sniffer 2,4Ghz		1	180	180	180		
Sensores LoRa		1	280	280	280		
Gateway		1	200	200	200		
Subtotal				**5824**	**5324**	**500**	
Imprevistos	**(15 %)**			**873**	**798**	**75**	
Total				**6697**	**6122**	**575**	

9. DESARROLLO

Realización de la primera fase del proyecto:

9.1 FASE 1. DISEÑO DE LA RED. El diseño de la red constituye el ambiente adecuado para la realización de las pruebas experimentales.

9.1.1 Actividad 1. Diseño. Elaboración del diseño, distribución y determinación de los equipos específicos necesarios para el despliegue de la arquitectura de red. Se deben determinar todos los dispositivos necesarios que conforman la red LoRa, así como sus respectivas configuraciones. Esto, para implementar el ambiente de pruebas necesario para cumplir con el objetivo planteado.

9.1.1.1 Red LoRa. Es una red inalámbrica de bajo consumo de energía que se ha diseñado para proporcionar conectividad a dispositivos IoT (Internet de las cosas) en un amplio rango geográfico, permitiendo la transmisión de datos a larga distancia. Los objetivos clave de la tecnología son establecer una infraestructura de comunicaciones confiable y eficiente para admitir una variedad de aplicaciones IoT y recopilar datos en entornos remotos.

Según lo expresa Rodríguez[20] las redes inalámbricas se basan en tres ejes: consumo energético, alcance y capacidad de trasmisión (data rate). Sin embargo, estas redes sacrifican uno de estos ejes. Por ejemplo, las redes 4G, tiene un gran alcance y data rate, pero la transmisión consume mucha energía. En cambio, Bluetooth consume muy poca energía, tiene un buena data rate, pero la distancia de cobertura es muy reducida. La elección de una red LoRa radica en su capacidad para superar obstáculos geográficos y su bajo consumo de energía, lo que la convierte en una solución viable para los entornos del internet de las cosas

A lo largo de esta fase, se abordan aspectos clave relacionados con la definición de requisitos, el diseño de topología de red, la selección de dispositivos LoRa, las medidas de seguridad. Además, se mencionan las consideraciones para la futura etapa de implementación física.

Para determinar el diseño adecuado de la red es necesario conocer los aspectos técnicos de la misma. En el análisis de LoRa, un aspecto clave es el factor de dispersión (SF), que influye en la velocidad de transmisión de datos y la distancia alcanzada. Cuando se disminuye el SF, aumenta la velocidad de transmisión de datos al incrementar el número de pulsos por segundo, pero se reduce la distancia cubierta. Por otro lado, aumentar el SF permite alcanzar distancias mayores a costa de una menor velocidad de transmisión. Como lo indica Rodríguez[21], LoRa ofrece hasta 6 SF diferentes (SF7 - SF12), cada uno con características específicas para

[20] Rodríguez, Op. cit., p. 36
[21] Ibíd., p. 45

adaptarse a las necesidades de la red y los dispositivos. Es importante destacar que los SF son ortogonales entre sí, lo que significa que señales con SF distintos en el mismo canal no se interferirán entre sí, tratándose como ruido. Los factores de dispersión se aplican en mensajes de enlace ascendente (UL) para canales de 125 kHz. Aumentar el SF aumenta la distancia, pero reduce la tasa de bits y aumenta el tiempo en el aire. Disminuir el SF resulta en tasas de bits más altas y menos tiempo en el aire, pero con una cobertura menor. El factor de dispersión (SF) está vinculado al Data Rate (DR) en la comunicación LoRaWAN. En la banda europea EU868, se observa que cuanto mayor es el SF (y, por lo tanto, menor el DR), más lenta es la comunicación, pero mayor es el alcance. Por el contrario, un SF más bajo implica una comunicación más rápida, pero con menor alcance. Para favorecer la escalabilidad de una red LoRaWAN, es óptimo que los dispositivos utilicen un SF bajo para minimizar el tiempo en el aire y permitir que otros dispositivos transmitan en el mismo canal. Por ejemplo, transmitir en SF7 es aproximadamente 30 veces más rápido que en SF12.

Los objetivos clave de la red LoRa en el proyecto incluyen:

1. Conectividad a Larga Distancia: Proporcionar una infraestructura de comunicaciones que permita la transmisión de datos a larga distancia, lo que resulta fundamental para recopilar información desde áreas remotas o de difícil acceso.
2. Eficiencia Energética: Minimizar el consumo de energía de los dispositivos IoT para garantizar una vida útil prolongada de las baterías y reducir la necesidad de mantenimiento.
4. Escalabilidad: Diseñar la red de tal manera que sea escalable, lo que permite agregar más dispositivos IoT según sea necesario sin comprometer el rendimiento.
5. Seguridad: Implementar medidas de seguridad sólidas para proteger la integridad y confidencialidad de los datos transmitidos a través de la red LoRa.

9.1.1.2 Requisitos de la Red LoRa para IoT:

Tipos de Dispositivos IoT:

- Nodos Finales (End Nodes): Son los dispositivos finales que recopilan datos o controlan activos. Pueden incluir sensores, actuadores, dispositivos de monitoreo, etc.
- **Gateway LoRa:** Dispositivos que actúan como puentes entre los nodos finales y la red LoRa. Reciben y transmiten datos entre los nodos y la red.

Alcance:
- **Largo Alcance:** La red LoRa se destaca por su capacidad de comunicación a larga distancia, permitiendo la cobertura de áreas extensas, incluso en entornos rurales o remotos.

- **Penetración de Objetos:** Debe ser capaz de comunicarse a través de obstáculos físicos, como edificios o vegetación, lo que es fundamental en aplicaciones de IoT al aire libre.

Eficiencia Energética:

- Los dispositivos IoT que operan en una red LoRa deben ser de baja potencia y capaces de funcionar con baterías durante largos períodos de tiempo. La eficiencia energética es un requisito crítico.

Ancho de Banda:

- La red LoRa utiliza un ancho de banda estrecho para transmitir datos, lo que permite una mayor capacidad de canalización y una mejor coexistencia con otras redes inalámbricas.

Seguridad:

- Los requisitos de seguridad incluyen la autenticación de dispositivos, la confidencialidad de los datos y la integridad de las comunicaciones para proteger la red LoRa contra amenazas potenciales.

Bajo Costo:

- Los componentes de la red LoRa y los dispositivos IoT deben ser asequibles para permitir una implementación económica y sostenible.

Compatibilidad con Diversas Aplicaciones:

- La red LoRa debe ser capaz de satisfacer los requisitos de una amplia gama de aplicaciones IoT, desde la agricultura y la logística hasta la gestión de activos y las ciudades inteligentes.

Capacidad de Bidireccionalidad:

- La red LoRa debe admitir tanto la transmisión de datos desde los nodos finales como la recepción de comandos o configuraciones desde una plataforma central.

Administración de Colisiones:

- La red LoRa debe ser capaz de gestionar colisiones de tráfico de dispositivos y garantizar una comunicación confiable.

En resumen, los requisitos de la red LoRa se centran en lograr una comunicación eficiente a larga distancia, con dispositivos de baja potencia y un alto nivel de seguridad, lo que permite la conexión de una amplia variedad de aplicaciones de manera rentable y confiable.

9.1.1.3 Diseño de la Topología:

Exploración de Topologías de Red LoRa y Justificación de Elección:

La elección de la topología de red LoRa es un paso crítico en el diseño de una red IoT eficiente. A continuación, exploráremos las tres topologías comunes de red

LoRa y justificaremos la elección de una topología específica para el escenario del proyecto.

Topología Estrella:

- Descripción: En una topología estrella, todos los dispositivos IoT (nodos finales) se comunican directamente con un único gateway central. Los nodos no se comunican entre sí.
- Ventajas:
  - Sencillez: Es fácil de implementar y gestionar, ya que todos los nodos se conectan a un punto central.
  - Bajo consumo de energía: Los nodos solo necesitan comunicarse con el gateway, lo que ahorra energía.
- Desventajas:
  - Punto único de falla: Si el gateway central falla, toda la red se ve afectada.
  - Limitación de alcance: El alcance de la red está limitado por la distancia entre los nodos y el gateway.

Topología Malla:

- Descripción: En una topología malla, los nodos IoT pueden comunicarse directamente entre sí y retransmitir datos para extender el alcance de la red.
- Ventajas:
  - Redundancia: La comunicación entre nodos permite rutas alternativas, reduciendo el riesgo de puntos únicos de falla.
  - Mayor alcance: La red puede extenderse a través de múltiples saltos entre nodos.
- Desventaja:
  - Mayor complejidad: La configuración y gestión de la red son más complejas.
  - Más consumo de energía: Los nodos deben estar activos durante más tiempo para retransmitir datos.

Topología Híbrida:

- Descripción: Una topología híbrida combina elementos de las topologías estrella y malla. Algunos nodos pueden comunicarse directamente con el gateway central, mientras que otros actúan como nodos repetidores en una red malla
- Ventajas:
  - Equilibrio entre simplicidad y alcance: Permite una red relativamente simple con la capacidad de extenderse a través de repetidores.
  - Redundancia selectiva: Solo los nodos críticos actúan como repetidores, lo que reduce la complejidad.
- Desventajas:

- Mayor gestión: Requiere la identificación y configuración de nodos repetidores.

Elección de Topología: Para este escenario de pruebas, la topología en estrella es la elección más apropiada. A continuación, se justifica esta elección:

- Alcance y Cobertura: La topología estrella suple a cabalidad con los requerimientos del enlace punto a punto que se va a realizar para implementar el entorno de pruebas. La topología malla proporciona alcance, pero la complejidad puede ser abrumadora.
- Redundancia y Confiabilidad: La topología híbrida ofrece una redundancia selectiva. Algunos nodos pueden actuar como repetidores para garantizar que la comunicación sea confiable sin aumentar significativamente la complejidad de la red.
- Gestión Sostenible: La topología estrella permite un equilibrio entre simplicidad y alcance, lo que facilita la gestión de una red en constante crecimiento.

En resumen, para el ámbito de este proyecto, una topología de red LoRa en estrella ofrece un equilibrio óptimo entre alcance, redundancia y eficiencia energética, lo que la convierte en la elección más adecuada.

9.1.1.4 Arquitectura de la Red. Según lo previsto en la actividad de diseño de la primera fase de ejecución del proyecto se realiza la Identificación del despliegue de una red LoRa, así como, el diseño de la red y cobertura. Como se mencionó anteriormente la red LoRa se compone de varios dispositivos para su implementación. A continuación, se detallan uno a uno los dispositivos requeridos para llevar a cabo el proyecto:

•Lylygo LoRa32 868/915mhz Esp32 LoRa Oled Pantal. Es una tarjeta de desarrollo basada en ESP32 que incorpora un módulo LoRa 868 / 915 Mhz que permite transmitir de forma bidireccional datos a grandes distancias Este módulo será conectado directamente al sensor de temperatura para procesar los datos y gracias a su antena de radiofrecuencia puede enviar los datos capturados por el sensor a través de LoRa en la banda de frecuencia no licenciada de 915 MHz. Posee las siguientes características:
-Frecuencia de funcionamiento: 868 M / 915 MHz
-Potencia de transmisión: + 20dBm
- Sensibilidad de recepción:
-139dBm @ LoRa y 62.5Khz y SF = 12 y 146bps
-136dBm @ LoRa y 125Khz y SF = 12 y 293bps
-118dBm @ LoRa y 125Khz y SF = 6 y 9380bps
-123dBm@FSK&5Khz&1.2Kbps
-Error de frecuencia: +/- 15 KHz
-Espacio FIFO: 64 bytes
- Velocidad de datos:

1.2K~300Kbps@FSK
0.018K~37.5Kbps@LoRa
Modo de modulación: FSK, GFSK, MSK, GMSK, LoRa TM, OOK

Figura 2. Radio LYLIGO LoRa

Fuente: ElectronicLab. LYLIGO LoRa32 V2.1 ESP32 OLED 0.96 SX1278

9.1.1.5 Configuración de los Radios LoRa en el Proyecto: La ubicación estratégica y la configuración adecuada de los radios son esenciales para garantizar un rendimiento óptimo de la red LoRa en el proyecto. Aquí se proporcionan detalles específicos sobre la ubicación, frecuencias, potencia de transmisión y antenas de los radios:

Frecuencias:

- En el proyecto, se ha configurado para operar en las frecuencias autorizadas y disponibles para la región geográfica en la que se encuentra la ciudad. Esto asegura que la red cumpla con las regulaciones de espectro radioeléctrico y no cause interferencias con otras redes de comunicación.

Potencia de Transmisión:

- La potencia de transmisión de los radios se debe ajustar para garantizar una cobertura efectiva sin un consumo excesivo de energía.
- La potencia se configura de acuerdo con la distancia y las condiciones del entorno. Se utiliza una potencia más alta cuando es necesario cubrir áreas extensas o superar obstáculos.

Es importante destacar que la configuración de los radios se realiza con un enfoque en la eficiencia y la optimización del rendimiento de la red LoRa. Esto garantiza que los nodos finales puedan comunicarse de manera confiable con la red y que los datos se recopilen de manera efectiva en todas las áreas de pruebas.

Configuraciones de Red: La tecnología LoRa permite las siguientes configuraciones:

- Clase A: Se ha optado por la Clase A. Esto es crucial para optimizar el consumo de energía, ya que muchos de los dispositivos son alimentados por batería.
- Tiempo de Espera (Duty Cycle): se establecen tiempos de espera que cumplen con las regulaciones locales. Estos tiempos de espera son esenciales para evitar la congestión del espectro radioeléctrico en nuestra ubicación.
- Ciclos de Trabajo: Se definen ciclos de trabajo específicos para nuestros dispositivos que garantizan un uso eficiente del espectro y cumplen con las restricciones locales. Esto garantiza una comunicación sin problemas y cumple con los requisitos legales.
- Frecuencias y Canales: Se configuran los dispositivos para utilizar las frecuencias y canales autorizados en nuestra ubicación geográfica. Esto asegura que estemos operando dentro de los límites legales y evita interferencias.
- Potencia de Transmisión: se configuran los niveles de potencia de transmisión según nuestras necesidades específicas. Esto asegura que tengamos la cobertura adecuada sin consumir demasiada energía.

Estas configuraciones de red garantizan un funcionamiento eficiente de nuestra red LoRa. Con esto se consigue que los dispositivos operen adecuadamente, respetando las regulaciones locales y cumpliendo con los requisitos de seguridad.

Software de Simulación: Se utiliza el IDE de Arduino para simular y diseñar nuestra red LoRa en un entorno virtual. Este enfoque nos permitió probar y validar nuestra arquitectura antes de implementarla en el mundo real. Con los requerimientos de diseño suplidos siguiendo el cronograma, la siguiente fase es la implementación de todos los dispositivos mencionados en esta sección, así como su configuración y pruebas de conectividad.

9.1.2 Actividad 2. Implementación. Después de haber realizado el diseño correspondiente de la red, se deben configurar los equipos utilizados. Para esto, se hace uso del ide de desarrollo de Arduino. Se realiza la instrumentación electrónica de los sensores y el acondicionamiento de las señales a la tarjeta de adquisición de datos y el cableado correspondiente.

9.1.2.1 Implementación de la Red LoRa: Para proceder con la implementación de la red propuesta se emplean los siguientes dispositivos:
-2 tarjetas de comunicación LoRa Lylygo 32 para establecer una comunicación bidireccional.
-Herramientas de programación, IDE de Arduino

Como primera medida se implementa una comunicación punto a punto entre las tarjetas LoRa, a continuación, se enviará información a través de las tarjetas por medio de la tecnología de radiofrecuencia LoRa, la cual opera en la banda de frecuencia de 915 MHz. Esta información será recibida por la segunda tarjeta.

9.1.2.2 Procedimiento de configuración de los dispositivos LoRa: A continuación, se detalla el procedimiento de configuración:

1. Se conecta la antena al puerto UFL del dispositivo LYLIGO, se enciende y se conecta al computador mediante un cable USB para proceder con su programación.
2. Se realiza la configuración de la biblioteca LMIC, para reconocer los dispositivos LoRa desde el IDE de Arduino

A continuación, se presenta la configuración del dispositivo LYLIGO emisor, el cual recibe la señal del sensor y la transmite a través de tecnología LoRa. Desarrollo implementado con el IDE de Arduino

```
#include <SPI.h>
#include <LoRa.h>
#include <DHT.h>
```

Se definen los pines para la conexión del sensor DHT11 y los pines SPI para LoRa.

```
#define DHTPIN 15
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DI0 26
```

Se inicia la configuración en el setup().

```
void setup() {
  Serial.begin(115200);
  while (!Serial);
  dht.begin();
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(915E6)) {
    Serial.println("Fallo en conexion
    while (true);
  }
}
```

En el método loop(), se lee la temperatura y humedad utilizando el sensor DHT11.

```
void loop() {
```

```
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
```

Crear una cadena de texto que contiene los valores de temperatura y humedad

```
  String payload = String(temperature) + "|" + String(humidity);
  Serial.println("Sending payload: " + payload);
```

Iniciar la transmisión a través de LoRa utilizando LoRa.beginPacket()

```
  LoRa.beginPacket();
  LoRa.print(payload);
  LoRa.endPacket();
  delay(5000);
```

En la figura 3 se presenta los radios de comunicación LoRa y se puede observar el sensor de temperatura y humedad DTH11

Figura 3. Radios LoRa

Fuente: Elaboración propia.

Código para el radio receptor en la Implementación de la Red LoRa:

```
#include <SPI.h>
#include <LoRa.h>
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DI0 26
void setup() {
  Serial.begin(115200);
  while (!Serial);
  SPI.begin(SCK, MISO, MOSI, SS);
```

```
  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(915E6)) {
    Serial.println("Conexion LoRa fallo.");
    while (true);
  }
}
void loop() {
  if (LoRa.parsePacket()) {
    String receivedData = "";
    while (LoRa.available()) {
      char receivedChar = (char)LoRa.read();
      receivedData += receivedChar;
    }
    Serial.println("Datos recibidos: " + receivedData);

  }
}
```

El código establece la comunicación LoRa en la frecuencia de 915 MHz y espera la recepción de paquetes. Cuando se recibe un paquete, se lee la cadena de datos y se imprime en el puerto serial para su análisis. La implementación del dispositivo receptor es crucial para completar el sistema de comunicación LoRa. Este código recibe datos del dispositivo emisor a través de la tecnología de radiofrecuencia.

9.2 FASE 2. REALIZACIÓN DE PRUEBAS DE COBERTURA DE LA SEÑAL: A continuación, se realizan pruebas exhaustivas de comunicaciones entre los radios para tratar de obtener la mejor relación señal ruido

9.2.1 RESULTADOS DE LA TRANSMISIÓN: En el proceso de implementación y prueba del sistema de comunicación LoRa con radios, se han obtenido los siguientes resultados:

- La comunicación entre el dispositivo emisor y el receptor ha sido exitosa.

Se ha logrado la transmisión de datos del sensor de humedad DHT11 desde el emisor al receptor.

- La cadena de datos transmitida ha sido recibida correctamente por el receptor. Los datos recibidos han sido procesados según las necesidades específicas de la aplicación.

Variables Clave para una Comunicación Exitosa:

- Configuración de Frecuencia: Es esencial que ambas radios LoRa estén configuradas para operar en la misma frecuencia. En el código proporcionado, la frecuencia se establece en 915 MHz
- Configuración de Pines: Los pines de conexión, tanto para la interfaz SPI como para los pines específicos de LoRa (SS, RST, DI0), deben coincidir en ambas radios.
- Alimentación y Energía: Se debe garantizar que ambos dispositivos tengan una fuente de alimentación adecuada. Problemas de energía pueden afectar negativamente la comunicación.

La comunicación exitosa en una red LoRa depende de la coherencia en la configuración de frecuencia, parámetros de red, conexiones físicas y lógica de procesamiento. La atención a estas variables clave garantizará una transmisión de datos confiable y efectiva entre dispositivos LoRa.

9.3 FASE 3. IMPLEMENTACIÓN DE UN SNIFFER PARA LA CAPTURA DEL TRÁFICO EMITIDA POR EL DISPOSITIVO. Se implementan herramientas de captura de tráfico para poder lograr la interceptación de paquetes en la comunicación.

9.3.1 ACTIVIDAD 1: Esta fase comprende las actividades correspondientes a la adquisición y configuración del sniffer como el dispositivo encargado de realizar los ataques a la red previamente implementada.

9.3.1.1 Sniffer CatWAN USB-Stick: El sniffer es el dispositivo encargado interceptar y registrar el tráfico de red. Esto implica capturar y analizar los datos que fluyen a través del aire por medio de señales electromagnéticas como es habitual en la comunicación por radiofrecuencia y en este caso en particular en la frecuencia de 915MHz. El sniffer puede capturar paquetes de datos, contraseñas, correos electrónicos, mensajes de texto u otros tipos de información.

Para el propósito de este proyecto se hizo uso del sniffer "Catwan Stick USB" él es un dispositivo de hardware que funciona como un sniffer de red. Está diseñado para ser conectado a un puerto USB y puede ser configurado como dispositivo transmisor o receptor o LoRa o en particular un sniffer para capturar todo el tráfico LoRa presenta en su espacio de cobertura.

Según la página del fabricante ElectronicCats[22], el Catwan Stick USB generalmente viene con software que permite al usuario ver y analizar el tráfico capturado. Puede proporcionar información detallada sobre los paquetes de datos, direcciones IP, puertos utilizados y otros detalles relevantes. Además, algunos dispositivos de este

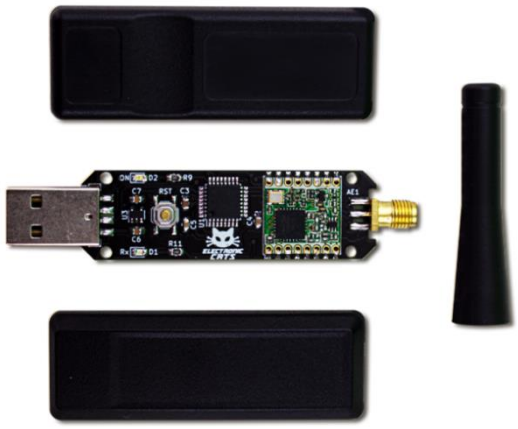
[22] ElectronicCats. CatWAN USB Stick. 2019

tipo pueden ser configurados para filtrar ciertos tipos de datos o realizar otras funciones avanzadas de análisis de red. A continuación, se detallan sus principales características:

El USB CatWAN está programada con una imagen de firmware especial que la convierte en un rastreador LoRa fácil de usar. Puede capturar pasivamente los intercambios de datos entre dos dispositivos LoRa. Este dispositivo puede funcionar en redes LoRaWAN compatibles con las clases A, B y C. Cuenta con un microcontrolador SAMD21 ARM Cortex a 48Mhz con USB 2.1 nativo, con 256Kb para programación, compatible con Arduino y Circuit Python.
Características principales

- Funciona con cualquier PC, Raspberry Pi o BeagleBone, incluso un teléfono inteligente o tableta
- Soporta modo paquete LoRa® (modo paquete) o LoRaWAN™ Clase A, B y C
- Compatible con The Things Network y otras redes LoRaWAN
- Basado en el RFM95
- LED RX como indicador de recepción, programable por el usuario
- Fácil reprogramación compatible con Arduino y Circuit Python
- Compatible con la aplicación LoRa Sniffer
- Fuente abierta
- Especificaciones
- Conectividad: USB 2.1
- Consumo de energía: 140 ma TX típico, 20 ma inactivo (con LED de encendido)
- Sensibilidad del receptor: hasta -146 dBm
- Potencia TX: ajustable hasta +18,5 dBm
- Alcance: hasta 15 km de cobertura en áreas suburbanas y hasta 5 km de cobertura en áreas urbanas. En la figura 4 se puede apreciar el sniffer mencionado:

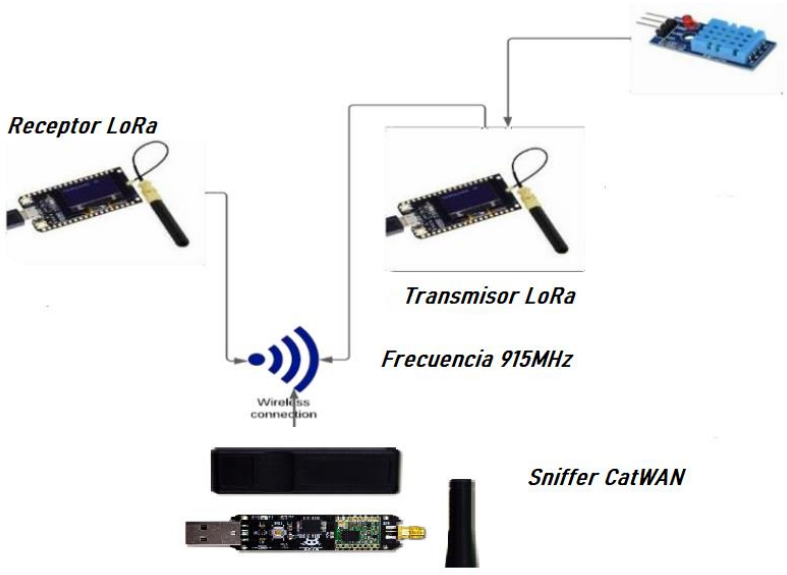
Figura 4. CatWAN USB-STICK



Fuente: ElectronicCats. CatWAN, 2019

Como se ha mencionado la red LoRa de este proyecto consta de dos radios comunicados entre sí y envían señales, uno como transmisor y el otro como receptor, el objetivo consiste en poner un sniffer CatWAN en medio de los dos para escuchar y capturar los paquetes que están transmitiendo estos radios. En la siguiente figura se puede apreciar el diseño lógico de la red:

Figura 5. Diseño de la red



Fuente: Elaboración propia

El fabricante proporciona los códigos de configuración base para el IDE ARDUINO que permiten poner el dispositivo en modo sniffer y realizar el escaneo del ambiente en busca de paquetes LoRa. Por lo cual, se deben configurar todas las características técnicas de los equipos para simular el ambiente de pruebas. A continuación, se presentan la configuración base de los radios:

```
LoRa.begin(915E6); // Frecuencia: 915 MHz
LoRa.setSpreadingFactor(8); // Factor de esparcimiento: 8
LoRa.setSignalBandwidth(125E3); // Ancho de banda: 125 kHz
LoRa.setCodingRate4(5); // Tasa de codificación: 5
LoRa.setSyncWord(0x12); // Palabra de sincronización: 0x12
LoRa.setPreambleLength(8); // Longitud del preámbulo: 8
```

Inicialmente se realizaron pruebas de comunicación entre los dos radios, uno como transmisor y el otro como receptor, enviando un mensaje:

Figura 6. Comunicación inicial radios LoRa.

Fuente: Elaboración propia.

En la siguiente figura se puede observar como el radio transmisor está enviando un mensaje el cual es recibido por el receptor de manera exitosa.

Figura 7. Envió de mensajes entre radios LoRa.

Fuente: Elaboración propia.

9.3.1.2 Configuración del Sniffer: El programa de configuración del CatWAN es proporcionado por el fabricante. Este código establece la configuración inicial de los pines y muestra un mensaje de bienvenida en el puerto serie. A continuación, Se configura el LoRa, se establecen las frecuencias y se registran callbacks para manejar la recepción de paquetes.

```
SerialCommand SCmd;
float fwVersion= 0.2;
float frequency = 915;
int spreadFactor = 8;
int bwReference = 7;
int codingRate = 5;
byte syncWord = 0x12;
int preambleLength = 8;
int txPower = 17;
int channel = 0;
bool rx_status = false;
int inv_iq = 0;
```

Consta de varias funciones que permiten configurar los parámetros del dispositivo LoRa, como frecuencia (set_freq()), factor de propagación (set_sf()), ancho de banda (set_bw()), tasa de codificación (set_cr()), palabra de sincronización (set_sw()), longitud del preámbulo (set_pl()), potencia de transmisión (set_tp()),

Estas funciones leen los argumentos proporcionados por el usuario a través del puerto serial y actualizan las variables correspondientes con los valores configurados.

También tiene funciones para obtener información sobre la configuración actual del dispositivo LoRa, como frecuencia (get_freq()), factor de propagación (get_sf()), ancho de banda (get_bw()), tasa de codificación (get_cr()), palabra de sincronización (get_sw()), longitud del preámbulo (get_pl()), potencia de transmisión (get_tp()), entre otros. Estas funciones simplemente imprimen los valores actuales de las variables al puerto serial para que el usuario pueda consultarlos. El programa principal se ejecuta en un bucle infinito (loop()), donde se llama continuamente a SCmd.readSerial() para leer los comandos entrantes del puerto serial y ejecutar las funciones correspondientes.

Para activar el CatWAN y detectar paquetes a través del monitor serie, se implementan los siguientes comandos:

set_rx: Este comando configura el dispositivo en modo receptor, lo que le permite escuchar y detectar paquetes LoRa en el aire.
set_freq: Este comando permite establecer la frecuencia en deseada para que el dispositivo escuche los paquetes LoRa. Se debe proporcionar la frecuencia en MHz como argumento.
set_sf: Este comando permite configurar el factor de esparcimiento, que determina la sensibilidad y la velocidad de transferencia de datos. Se puede proporcionar un valor entre 6 y 12 como argumento.
set_bw: Este comando permite configurar el ancho de banda del canal LoRa. Se puede proporcionar un valor entre 0 y 9 como argumento, donde cada valor corresponde a un ancho de banda específico.
set_cr: Este comando permite configurar la tasa de codificación del dispositivo. Se puede proporcionar un valor entre 5 y 8 como argumento.
set_pl: Este comando sirve para establecer la longitud del preámbulo. Esto puede afectar la sensibilidad y la capacidad de detección del dispositivo.
set_tp: Este comando permite configurar la potencia de transmisión del dispositivo. Se puede proporcionar un valor entre 2 y 20 como argumento.
set_inv_iq: Este comando sirve para habilitar o deshabilitar la inversión de IQ, que puede ser útil en algunas situaciones de RF.

9.3.1.3 Implementación del sniffer: El sniffer de paquetes es una técnica comúnmente utilizada para interceptar y analizar el tráfico de red. En el contexto de redes LoRa, un sniffer de paquetes se utiliza para capturar los mensajes transmitidos entre los dispositivos LoRa. Este proceso implica el uso de hardware especializado o software que puede monitorear y registrar las transmisiones de

radio LoRa en un área determinada. A continuación, se presenta la configuración inicial para el dispositivo encargado de escuchar la comunicación.

```
// Setup callbacks for SerialCommand commands
SCmd.addCommand("help",help);
SCmd.addCommand("set_rx",set_rx);
SCmd.addCommand("set_tx",set_tx);
SCmd.addCommand("set_tx_hex",set_tx_hex);
SCmd.addCommand("set_tx_ascii",set_tx_ascii);
SCmd.addCommand("set_freq",set_freq);
SCmd.addCommand("set_sf",set_sf);
SCmd.addCommand("set_bw",set_bw);
SCmd.addCommand("set_cr",set_cr);
SCmd.addCommand("set_sw",set_sw);
SCmd.addCommand("set_pl",set_pl);
SCmd.addCommand("set_tp",set_tp);
SCmd.addCommand("set_chann",set_chann);
SCmd.addCommand("set_inv_iq",set_inv_iq);
SCmd.addCommand("get_config",get_config);
SCmd.addCommand("get_freq",get_freq);
SCmd.addCommand("get_sf",get_sf);
SCmd.addCommand("get_bw",get_bw);
SCmd.addCommand("get_cr",get_cr);
SCmd.addCommand("get_sw",get_sw);
SCmd.addCommand("get_pl",get_pl);
SCmd.addCommand("get_tp",get_tp);
```

Esta es la función principal que permite iniciar el dispositivo como sniffer y capturar paquetes:

```
void set_rx(){
  char *arg;
  arg = SCmd.next();
  if (arg != NULL){
    frequency = atof(arg);
//    if(frequency > 902 && frequency < 923){
      long freq = frequency*1000000;
      LoRa.setFrequency(freq);
      Serial.println("LoRa radio receiving at " + String(frequency) + " MHz");
      while (digitalRead(RFM_DIO5) == LOW){
         Serial.print(".");
       }
      LoRa.receive();
      rx_status = true;
//    }
//    else{
//      Serial.println("Error setting the frequency");
```

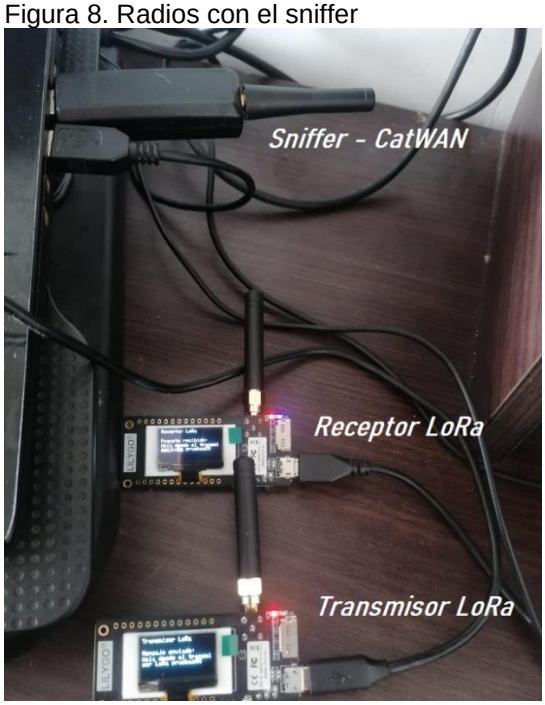
```
//     Serial.println("Value must be between 902 MHz and 923 MHz");
//   }
  }
  else {
    Serial.println("LoRa radio receiving at " + String(frequency) + " MHz");
    LoRa.receive();
    rx_status = true;
  }
}
```

9.4 FASE 4. REALIZACIÓN DE UN ATAQUE A LA RED LORA QUE PERMITA LA CAPTURA DE LA INFORMACIÓN ENVIADA. Después de haber puesto en marcha la red y validado la comunicación y la captura de paquetes se desplegaron los ataques a la red

9.4.1 Actividad 1. Ataques. Se deben realizar ataques a la red anteriormente implementada. Se empleará un sniffer para la captura de los paquetes enviados en la transmisión y de esta manera poder interceptar las comunicaciones.

9.4.1.1 Despliegue del Sniffer: Una vez configurado y compilado el código en el sniffer se procede a conectar para rastrear los paquetes disponibles, en la siguiente imagen se pueden observar todos los dispositivos involucrados:

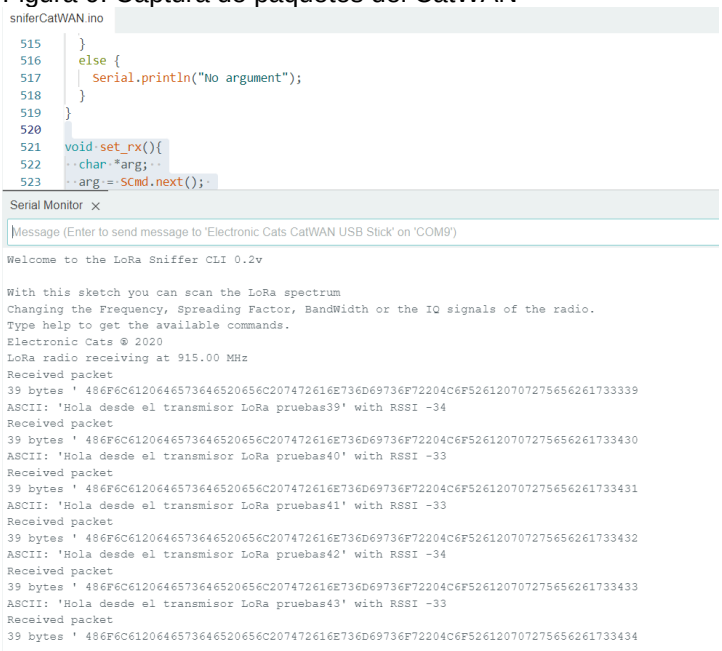
Figura 8. Radios con el sniffer



Fuente: Elaboración propia.

A continuación, observamos la captura de paquetes obtenidos por el CatWAN:

Figura 9. Captura de paquetes del CatWAN



Fuente: Elaboración propia.

En la figura 9 se observan los paquetes que capturo el sniffer y se puede apreciar el paquete y el nivel de RSSI. Como lo indica Lucas[23] el RSSI (received signal strength indicator) es una escala de referencia (en relación a 1 mW) para medir el nivel de potencia de las señales recibidas por un dispositivo en las redes inalámbricas. La escala tiene al valor 0 como centro; representa 0 RSSI, o 0 dBm. La escala se expresa dentro de valores negativos; cuanto más negativo, mayor pérdida de señal. Valor de la medición de RSSI Interpretación 0 Señal ideal -40 a -60 Señal idónea con tasas de transferencia estables -60 Enlace bueno, con el cual se puede lograr una conexión estable al 80%. -70 Enlace normal a bajo, es una señal medianamente buena, se puede sufrir problemas con lluvia y viento -80 a -110 Señal mínima para el enlace. Puede ocurrir caídas, que se traducen en corte de comunicación.

9.4.1.2 Ataque de denegación de servicio (DoS): Este ataque tiene como objetivo abrumar un dispositivo o red LoRa con una gran cantidad de tráfico falso o solicitudes, lo que impide que los dispositivos legítimos accedan a la red o se comuniquen de manera efectiva. Como lo indican Rondón & Bravo [24] el propósito

[23] LUCAS BULLIAN. Implementación de una red LoRa en el ámbito de la Universidad Nacional de San Martín. 2018, p 44

[24] Rondón & Bravo. Op. cit., p. 59

del ataque es impedir que los paquetes lleguen al destino correspondiente o se sature la red haciendo que esta colapse en su funcionamiento, por lo tanto, la vulnerabilidad encontrada se puede disminuir igualmente añadiendo canales y/o cambiando el SF, además se requiere de un buen diseño de red para evitar en la medida de lo posible colisiones que causen un DoS.

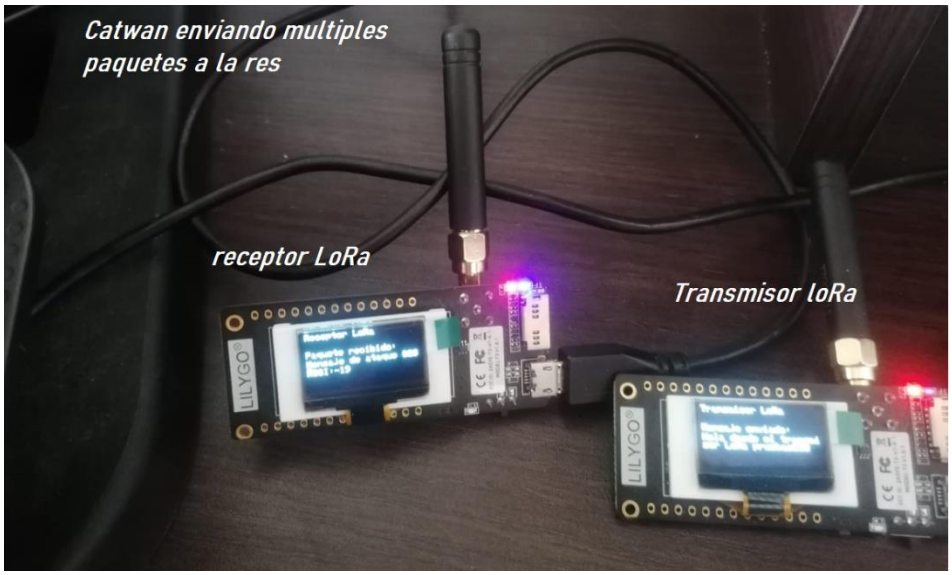
Para la implementación de este ataque se utilizó en CatWAN enviando una gran cantidad de paquetes LoRa a la red objetivo, ocupando todo el ancho de banda disponible y saturando los recursos de los dispositivos receptores.

Se configura los comandos para ejecutar los ataques desde la catwan a través se funciones automatizadas en la cli:

```
    SCmd.addCommand("dos", attackDoSCommand);
    SCmd.addCommand("replay", attackReplayCommand);
    SCmd.addCommand("injection", attackInjectionCommand);

void attackDoSCommand() {
    char *arg1 = SCmd.next();
    char *arg2 = SCmd.next();
    if (arg1 != NULL && arg2 != NULL) {
        int numPackets = atoi(arg1);
        int delayTime = atoi(arg2);
        attackDoS(numPackets, delayTime);
    } else {
        Serial.println("Usage: dos <num_packets> <delay_time>");
    }
}
// Función para realizar un ataque de denegación de servicio (DoS)
void attackDoS(int numPackets, int delayTime) {
  static int contador = 0;
  for (int i = 0; i < numPackets; i++) {
  contador++;
    String mensaje = "Mensaje de ataque #" + String(contador);
    // Enviar el mensaje por LoRa
    LoRa.beginPacket();
    LoRa.print(mensaje);
    LoRa.endPacket();
        Serial.println(mensaje);
    delay(delayTime);
  }
```

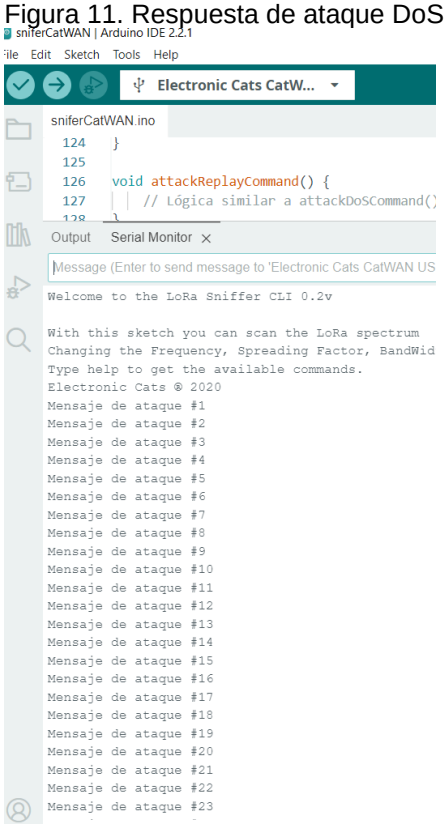
Figura 10. Ataque DoS



Fuente: Elaboración propia.

Se puede apreciar a continuación la respuesta obtenida por el sniffer:

Figura 11. Respuesta de ataque DoS



Fuente: elaboración propia.

9.4.1.3 Ataque de repetición: En este ataque, un atacante intercepta y graba un paquete LoRa legítimo y luego lo retransmite repetidamente en la red. Esto puede conducir a la reutilización de mensajes antiguos, lo que puede comprometer la seguridad de la red. Rondón & Bravo[25] afirman que el ataque de replay en una red LoRa consiste en la copia y reenvío de señales transmitidas por un nodo, con el objetivo de suplantar la identidad del nodo original y acceder ilegalmente a la red.

Para la implementación de este ataque se hizo uso de sniffer CatWAN para interceptar y grabar paquetes LoRa legítimos. Luego, se retransmitieron estos paquetes de manera repetida a la red para llevar a cabo el ataque de repetición.

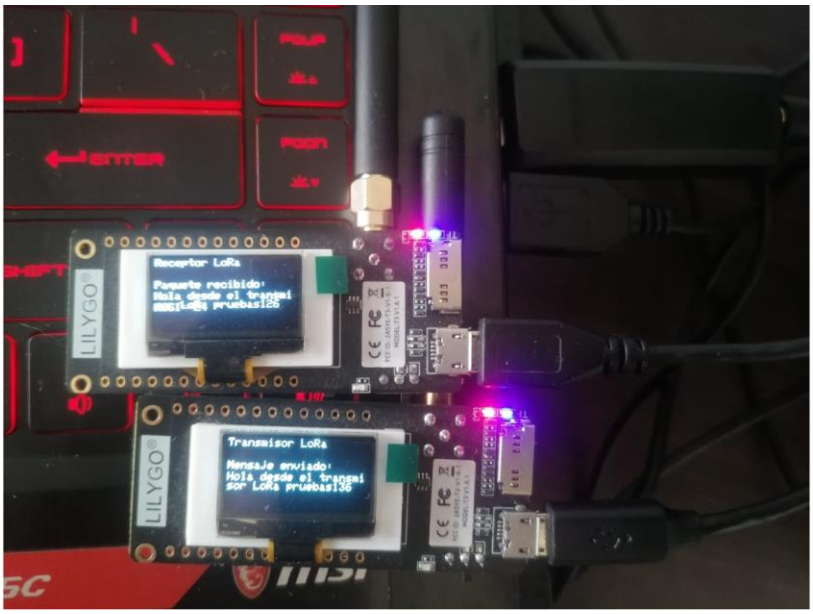
```
void attackReplayCommand() {
   char *arg1 = SCmd.next();
   char *arg2 = SCmd.next();

   if (arg1 != NULL && arg2 != NULL) {
      int numRepeats = atoi(arg1);
      int delayTime = atoi(arg2);

      if (capturedPacketSize > 0) {
                  attackReplay(capturedPacket, capturedPacketSize, numRepeats,
delayTime);
      } else {
         Serial.println("No packet captured to replay.");
      }
   } else {
      Serial.println("Usage: replay <num_repeats> <delay_time>");
   }
}
// Función para realizar un ataque de repetición
void attackReplay(byte* packet, int packetSize, int numRepeats, int delayTime) {
   for (int i = 0; i < numRepeats; i++) {
      // Retransmitir el paquete capturado repetidamente al nodo objetivo
      LoRa.beginPacket();
      LoRa.write(packet, packetSize);
      LoRa.endPacket();
      Serial.println("replicando paquete");
      delay(delayTime);
   }
```

Figura 12. Ataque reply

[25] Ibíd., p. 55-58



Fuente: Elaboración propia

Figura 13. Respuesta del ataque reply

sniferCatWAN.ino

```
161       LoRa.print(mensaje);
162       LoRa.endPacket();
163           Serial.println(mensaje);
164       delay(delayTime);
165     }
166   }
167
168   // Función para realizar un ataque de repetición
169   void attackReplay(byte* packet, int packetSize, int numRepeats, int delayTi
170       for (int i = 0; i < numRepeats; i++) {
171           // Retransmitir el paquete capturado repetidamente al nodo objetivo
172           LoRa.beginPacket();
173           LoRa.write(packet, packetSize);
174           LoRa.endPacket();
175           Serial.println("replicando paquete");
176           delay(delayTime);
177       }
178   }
179
180
```

Output Serial Monitor ×

Not connected. Select a board and a port to connect automatically.

```
replicando paquete
replicando paquete
replicando paquete
replicando paquete
replicando paquete
replicando paquete
replicando paquete
replicando paquete
replicando paquete
replicando paquete
replicando paquete
replicando paquete
replicando paquete
```

Fuente: Elaboración propia.

9.5 FASE 5. ANÁLISIS DE RESULTADOS. Por último, después de haber ejecutado las pruebas de ataques y el análisis correspondiente se desarrolla el informe de resultados registrando los hallazgos encontrados.

9.5.1 Actividad 1. Informe De Resultados. Se identifican las vulnerabilidades presentes descubiertas en la fase de ejecución de ataques, se documentan las fallas y se realiza el informe de análisis de resultados para dar pie al cumplimiento del objetivo trazado del proyecto.

9.5.1.2 Resultados y Conclusiones: El análisis de seguridad en redes LoRa mediante sniffer de paquetes revela varias vulnerabilidades y riesgos potenciales. Algunos de los hallazgos encontrados incluyen:

- Interceptación de Paquetes: En la figura 9 se puede observar la captura de paquetes obtenida por el CatWAN al configurarlo para escuchar en la red. La respuesta proporcionada por el CatWAN muestra que está recibiendo paquetes de datos LoRa transmitidos por los radios que están enviando mensajes seguido de un número de prueba. Se puede apreciar:

- **Frecuencia y RSSI:** El CatWAN está configurado para recibir en una frecuencia de 915 MHz. Cada paquete recibido muestra su intensidad de señal recibida (RSSI), indicando la fuerza de la señal recibida, que es -34 o -33 en este caso.
- **Paquetes recibidos:** Se muestran varios paquetes recibidos, cada uno con una longitud de 39 bytes. Los paquetes contienen datos codificados en hexadecimal, seguidos de la representación ASCII de esos datos. La capacidad de interceptar y leer los mensajes transmitidos puede plantear preocupaciones de seguridad y privacidad si los mensajes contienen información sensible.

La respuesta obtenida es la capacidad de interceptar y leer los mensajes transmitidos por las radios LoRa. Esto puede proporcionar información sobre las comunicaciones en la red, incluidos los datos transmitidos y la ubicación de los dispositivos.

Vulnerabilidades encontradas: En este caso, la principal vulnerabilidad es la falta de cifrado o seguridad en la capa física de la comunicación LoRa. Esto permite que un atacante intercepte y lea los datos transmitidos sin autenticación ni cifrado.
Además, si los mensajes transmitidos contienen información sensible, como datos personales o contraseñas, su exposición a terceros podría representar un riesgo de privacidad y seguridad. En conclusión, este análisis resalta la importancia de implementar medidas de seguridad adecuadas en las redes LoRa, como el cifrado de datos y la autenticación, para proteger la confidencialidad e integridad de las comunicaciones.

- Informe de Análisis de Vulnerabilidades y Conclusiones de un Ataque DoS en una Red LoRa: Como se presentó en el apartado anterior, se realizó el montaje de una red con dos equipos LoRa que estaban generando tráfico punto a punto, se implementó un sniffer CatWAN en la red y se configuró para hacer un envió masivo de paquetes y así congestionar la red buscando de esta manera impedir la comunicación exitosa entre los radios legítimos. Tal como se presenta en la figura 10 se puede observar que a pesar que el radio transmisor está enviando mensajes específicos el radio receptor está recibiendo los mensajes que está enviando el CatWAN por lo que no le es posible recibir los mensajes del transmisor debido a la saturación del canal. Esto demuestra la efectividad del ataque realizado y la inoperatividad de la tecnología para contrarrestar este ataque.

Vulnerabilidades Identificados:

- Falta de autenticación en las comunicaciones entre dispositivos.
- Utilización de encriptación débil o nula en los mensajes transmitidos.
- Deficiencias en la gestión de claves, incluyendo el almacenamiento no seguro de claves de cifrado.
- Riesgo de interrupción del servicio debido a ataques DoS.
- Posibilidad de degradación del rendimiento debido a los ataques efectuados.
- Riesgo de interrupción del servicio debido a ataques DoS por la falta de mecanismos de defensa adecuados que exponen la red a ataques de Denegación de Servicio, como el observado en el presente análisis, que pueden afectar gravemente la disponibilidad de los servicios IoT asociados.

Impacto del Ataque DoS:

- Los dispositivos de la red LoRa fueron incapaces de procesar solicitudes legítimas debido a la saturación de tráfico generada por el ataque.
- La conectividad entre los dispositivos se vio afectada, resultando en la pérdida de datos y la interrupción de los servicios IoT asociados.

Medidas de Mitigación:

- Es imperativo establecer procedimientos de respuesta a incidentes para abordar rápidamente cualquier brecha de seguridad o interrupción del servicio.
- Se recomienda realizar una planificación de la capacidad y optimización del rendimiento para garantizar que el sistema pueda escalar eficientemente según sea necesario.

Vulnerabilidades encontradas: La falta de autenticación adecuada y el uso de encriptación débil representan riesgos significativos para la seguridad de la red

LoRa, lo que facilita la interceptación de datos y la realización de ataques de Denegación de Servicio.

- Informe de Análisis de Vulnerabilidades y Conclusiones de un Reply en una Red LoRa: Otro de los ataques realizados a la red LoRa fue el ataque de reply, el cual consistió en la captura de un paquete y su posterior reenvió a la red en intervalos de tiempo específicos, esta circunstancia genera errores en el receptor al no recibir los paquetes legítimos. El objetivo fue enviar información falsa al dispositivo receptor y comprometer la integridad de la comunicación.

Vulnerabilidades Identificados:

- Reutilización de Mensajes Antiguos: El hecho de retransmitir paquetes grabados permite la reutilización de mensajes antiguos, lo que puede comprometer la seguridad de la red y facilitar la suplantación de identidad.
- Compromiso de la Integridad de la Red: La retransmisión de paquetes falsos puede comprometer la integridad de la comunicación en la red LoRa, permitiendo la inserción de información falsa y la realización de acciones maliciosas.

Impacto del Ataque de Repetición:

- La retransmisión repetida de paquetes falsos puede generar confusión en los dispositivos receptores y comprometer la confiabilidad de la información transmitida.

Medidas de Mitigación:

- Implementación de Contramedidas Efectivas: Es crucial implementar contramedidas efectivas para evitar los ataques de repetición, como la utilización de mecanismos de autenticación robustos y la validación de la integridad de los paquetes recibidos.

Vulnerabilidades encontradas: El ataque de repetición representa una amenaza significativa para la seguridad de la red LoRa, ya que compromete la integridad de la comunicación y facilita la suplantación de identidad. Es fundamental implementar medidas de mitigación efectivas para proteger la red contra este tipo de ataques y garantizar su seguridad y confiabilidad a largo plazo.

10. RESULTADOS

10.1 DESCRIPCIÓN DE RESULTADOS. En esta sección, se presentan los resultados obtenidos durante la evaluación de la red LoRa, incluyendo el diseño topológico, la configuración de los radios LoRa, las pruebas de cobertura y los ataques realizados para determinar las vulnerabilidades de la red.

Se realizó el diseño topológico de la red LoRa y se implementaron las configuraciones correspondientes de los radios para generar tráfico en la red. Además, se configuró y se implementó un sniffer para escuchar los paquetes transmitidos, lo que permitió realizar ataques simulados para validar la seguridad de la red.

Durante las pruebas de cobertura en un ambiente controlado, se obtuvieron los siguientes resultados:

- Se logró establecer una comunicación bidireccional entre las tarjetas LYLIGO con una tasa de transferencia de datos de 50 Kbps.
- El rango máximo de comunicación alcanzado fue de 2 km utilizando una antena omnidireccional de 5 dBi y una altura de 3 metros sobre el nivel del suelo.
- Se observó una degradación significativa en la calidad de la señal en presencia de obstáculos físicos como edificios o árboles.
- Se evaluó la sensibilidad de las tarjetas a interferencias electromagnéticas y se determinó que son susceptibles a interferencias de otras señales en la misma banda de frecuencia.

Estos resultados demuestran que las tarjetas LYLIGO LoRa son una opción viable para implementaciones de redes LoRa de larga distancia, siempre y cuando se tenga en cuenta el entorno y la posible interferencia de otras señales.

En cuanto a los resultados obtenidos en las pruebas de laboratorio realizadas se obtuvieron:

- Interceptación de Paquetes. Durante la captura de paquetes, se evidenció la capacidad de interceptar y leer los mensajes transmitidos por radios LoRa. Esta vulnerabilidad plantea preocupaciones sobre la privacidad y seguridad de los datos transmitidos a través de la red.
- Ataque de Denegación de Servicio (DoS). Se identificó la posibilidad de realizar ataques de denegación de servicio (DoS) mediante el envío masivo de paquetes falsos, lo que saturó la red y dificultó la comunicación efectiva entre los dispositivos legítimos.

- Ataque de Repetición. El análisis reveló la vulnerabilidad de repetición, donde un atacante puede interceptar y retransmitir paquetes legítimos para comprometer la integridad de la comunicación en la red LoRa.

Estos resultados subrayan la importancia de implementar medidas de seguridad adecuadas en las redes LoRa para proteger la confidencialidad, integridad y disponibilidad de las comunicaciones. Además, resaltan la necesidad de una mayor investigación y desarrollo en el campo de la seguridad de IoT para abordar estos desafíos emergentes.

10.2 DISCUSIÓN DE RESULTADOS. Al comparar los hallazgos obtenidos en este trabajo con la literatura existente y las investigaciones realizadas en el campo de los ataques a redes LoRa, se pueden identificar varias similitudes y diferencias a tener en cuenta. Se identificaron varias vulnerabilidades en la red LoRa, como la falta de autenticación adecuada, encriptación débil, y deficiencias en la gestión de claves, lo que podría exponer la red a ataques de interceptación, denegación de servicio (DoS) y suplantación de identidad. Esta situación se asemeja a los hallazgos de Bravo et al.,[26] quienes también encontraron que las redes LoRa son susceptibles a ataques de replay y denegación de servicio, demostrando la necesidad de abordar estas vulnerabilidades para garantizar la seguridad de la red.

La investigación previa ha abordado temas como la interceptación de paquetes, los ataques de denegación de servicio (DoS) y los ataques de repetición en redes LoRa, así como los medios de simulación para validar la seguridad de la tecnología y de esta manera concluir con el análisis de seguridad en redes LoRa, el cual es el objetivo general de este proyecto de grado. Esto permite contrastar y validar los resultados obtenidos.

- Interceptación de Paquetes: Los hallazgos obtenidos en este trabajo muestran que la interceptación de paquetes en redes LoRa es factible mediante el uso de sniffers, como se demostró en la Figura 9 con el sniffer CatWAN. Esta capacidad para capturar y leer los mensajes transmitidos plantea preocupaciones de seguridad y privacidad, ya que los mensajes pueden contener información sensible y, de esta manera, la tecnología sería inútil para suplir los requerimientos de la aplicación.

Investigaciones previas, como las de Yang et al[27], también han destacado la vulnerabilidad de las comunicaciones LoRa a la interceptación de paquetes. En su trabajo investigativo presentaron cinco vulnerabilidades en el protocolo y se describieron ataques reales. Para verificarlos, se han implementado todos los vectores de ataque en un prototipo de software. Los ataques apuntan a los tres aspectos principales de la seguridad de la comunicación: primero, demostraron que es posible interceptar y descifrar el contenido de un marco en ciertas circunstancias. Segundo, mostraron que el contenido de un paquete puede ser modificado fuera de la integridad verificada por el protocolo. Tercero, destacaron que los mensajes podrían ser retransmitidos a un nodo engañado para creer que un mensaje ha sido recibido por la pasarela cuando en realidad no.

En este orden de ideas, se destacan los siguientes puntos de contraste:
- Identificación de vulnerabilidades: Ambas investigaciones identifican vulnerabilidades en las redes LoRa. Tanto el análisis de seguridad en redes LoRa mediante sniffer de paquetes como el análisis de vulnerabilidades en el protocolo

[26] Bravo et al., Op. cit., p 185-194
[27] Yang, et al. Security Vulnerabilities in LoRaWAN, 2018. p. 3-7.

LoRaWAN revelan deficiencias en la autenticación, encriptación y gestión de claves, así como la exposición a ataques de denegación de servicio (DoS).
- Uso de sniffer para capturar paquetes: se emplea un sniffer para capturar los paquetes enviados en la transmisión de la red LoRa. Esta técnica permite examinar el tráfico de red en tiempo real y analizar los mensajes transmitidos, lo que resulta fundamental para identificar vulnerabilidades y evaluar la seguridad de la red.
- Implementación de ataques: Ambos estudios realizan ataques específicos a la red LoRa para demostrar las vulnerabilidades identificadas. El presente análisis de seguridad implementó ataques como DoS y ataques de repetición, mientras que el análisis de vulnerabilidades en el protocolo LoRaWAN desarrollado por Yang describe pruebas de concepto para explotar las debilidades del protocolo, como ataques de repetición y eavesdropping.
- Enfoque en la seguridad: Ambos trabajos investigativos destacan la importancia de abordar las vulnerabilidades de seguridad en las redes LoRa mediante sniffer de paquetes lo que pone de manifiesto la necesidad de implementar medidas de seguridad adecuadas para proteger la confidencialidad, integridad y disponibilidad de los datos transmitidos. Se proponen medidas de mitigación para abordar las vulnerabilidades identificadas. Estas incluyen establecer procedimientos de respuesta a incidentes, realizar planificación de capacidad y optimización del rendimiento, implementar tecnologías de cifrado robustas y mecanismos de autenticación sólidos. Lo cual se alinea con las conclusiones obtenidas en el presente documento
- Metodología: El estudio sobre LoRaWAN utiliza un enfoque teórico para analizar la efectividad de los mecanismos de seguridad del protocolo LoRa, mientras que este análisis de seguridad utiliza un enfoque práctico para identificar vulnerabilidades en la tecnología implementada y realizar pruebas de seguridad en condiciones reales.

- Ataques de Denegación de Servicio (DoS): Los resultados muestran que los ataques de DoS pueden saturar la red LoRa y afectar gravemente su funcionamiento, como se ilustra en la Figura 10. La congestión del canal impide la comunicación entre los dispositivos legítimos, demostrando la efectividad de este tipo de ataque. Este proyecto reveló que los ataques de denegación de servicio (DoS) afectaron la capacidad de los dispositivos LoRa para procesar solicitudes legítimas, lo que resultó en la pérdida de datos y la interrupción de los servicios asociados. Estos resultados son consistentes con los hallazgos de Rodríguez[28], quienes demostraron que los ataques de replay podrían comprometer la disponibilidad y confidencialidad de los datos, lo que confirma la gravedad de estos tipos de ataques en redes LoRa. Adicionalmente, este hallazgo está respaldado por la investigación de Kuntke et al[29], quienes identificaron la vulnerabilidad de las redes

[28] Rodríguez. Op. cit., p. 59 - 100
[29] Kuntke, et al. LoRaWAN security issues and mitigation options by the example of agricultural IoT scenarios, 2022. p. 13-18.

LoRa a los ataques de DoS. La denegación de servicio (DoS) mediante la interferencia de radiofrecuencia es un problema general para las tecnologías IoT, particularmente en aplicaciones agrícolas. Un atacante transmite una señal de radio potente cerca de los dispositivos de la aplicación, interrumpiendo las transmisiones y afectando la calidad del servicio. Para contrarrestar estos ataques, se pueden implementar medidas como la creación de una red densa de LoRa con cobertura de GW superpuesta, el uso máximo de salto de canal, la reducción de la frecuencia de señal y el tamaño del paquete para vencer el tiempo de reacción de los atacantes, y la detección de ataques mediante análisis de tráfico.

Impacto del Ataque DoS: Tanto en el estudio desarrollado por Kuntke como en el presente trabajo, se destaca el impacto significativo que un ataque DoS puede tener en las redes LoRa. La saturación del tráfico genera interrupciones en la comunicación entre dispositivos, resultando en la pérdida de datos y la interrupción de los servicios IoT asociados.

Medidas de Mitigación: Ambos informes resaltan la necesidad de implementar medidas de mitigación para abordar las vulnerabilidades identificadas y proteger las redes LoRa contra posibles ataques. Se propusieron acciones, como establecer procedimientos de respuesta a incidentes y realizar planificación de capacidad y optimización del rendimiento para garantizar la escalabilidad del sistema. Estas recomendaciones están alineadas con las sugerencias de Pérez[30], quien resalta la importancia de proteger los datos en las redes LoRa mediante el empleo de capas de cifrado y la implementación de mecanismos de seguridad robustos.

- Ataques de Repetición: La realización de ataques de repetición, como se muestra en la Figura 12, revela la vulnerabilidad de las redes LoRa a la reutilización de mensajes antiguos, comprometiendo su integridad y seguridad. Investigaciones anteriores, como la de Aras et al[31], han identificado este tipo de ataque como una amenaza potencial para las redes LoRa. Su investigación destaca que la modulación LoRa es susceptible a ataques de interferencia y jamming. Los dispositivos LoRa pueden sufrir problemas de coexistencia, haciéndolos vulnerables a ataques de interferencia. La amplia ventana de transmisión de LoRa permite que los atacantes interrumpan las comunicaciones o corrompan los paquetes antes de llegar a su destino. Esto es especialmente preocupante debido a los largos tiempos de transmisión de LoRa, que van desde cientos de milisegundos hasta 1.5 segundos, proporcionando oportunidades para ataques selectivos de jamming. Su estudio demuestra que los ataques de jamming contra redes LoRa pueden llevarse a cabo con hardware comercial. Esto implica que los atacantes podrían perturbar las comunicaciones LoRa con un costo relativamente bajo utilizando dispositivos Arduino y módulos LoRa.

[30] Pérez. Op. cit., p. 22-56
[31] Aras, et al. Exploring the Security Vulnerabilities of LoRa, 2017.

La efectividad de estos ataques es alarmante, con resultados que muestran que aproximadamente el 99% de las transmisiones LoRa pueden verse afectadas utilizando esta técnica. Esto subraya la urgencia de abordar las vulnerabilidades en la capa física de LoRa.

El estudio señala la posibilidad de realizar ataques de repetición y ataques de gusano (wormhole) en redes LoRa. Estos ataques implican la captura y retransmisión de paquetes, lo que podría engañar a los dispositivos o redes al utilizar mensajes antiguos o falsos. Los ataques de repetición podrían ser particularmente problemáticos en entornos donde los contadores de tramas no se gestionan adecuadamente, lo que permite la reutilización de mensajes previamente capturados. Destacan la necesidad del uso adecuado de cifrado y gestión de claves, así como la vigilancia activa de las comunicaciones para detectar y prevenir ataques.
En conclusión, el estudio revela que, si bien la tecnología LoRa ofrece beneficios significativos en términos de alcance y eficiencia energética para las redes IoT, también presenta desafíos importantes en cuanto a seguridad. Abordar estas vulnerabilidades es crucial para garantizar la integridad y confidencialidad de las comunicaciones en las redes LoRa y proteger contra posibles ataques maliciosos.

Al contrastar los resultados obtenidos en este trabajo con las investigaciones anteriores, se confirma la relevancia y validez del estudio desarrollado de la seguridad en las redes LoRa.

Los resultados obtenidos en el proyecto se suman a una serie de investigaciones que resaltan las fallas de seguridad inherentes en la tecnología LoRa, lo que subraya la necesidad de abordar estos desafíos para garantizar la seguridad de las comunicaciones inalámbricas de largo alcance en el contexto del Internet de las Cosas (IoT). Además, se reconoce que, si bien la tecnología LoRa ofrece ventajas significativas en términos de alcance y consumo de energía, aún existen desafíos en cuanto a la protección de la seguridad y la integridad de los datos transmitidos.

En resumen, se destaca la importancia de abordar las vulnerabilidades identificadas en las redes LoRa para garantizar su seguridad y confiabilidad en el contexto del IoT. Los hallazgos del proyecto actual se alinean con investigaciones previas, lo que subraya la necesidad de continuar investigando y desarrollando medidas de seguridad efectivas para proteger las comunicaciones.

11. CONCLUSIONES

El análisis detallado de las redes LoRa revela la existencia de vulnerabilidades críticas que pueden ser explotadas debido a la falta de estandarización en los protocolos del IoT. Estas vulnerabilidades pueden afectar la confidencialidad, integridad y disponibilidad de los datos transmitidos a través de la red.

La ausencia de estándares sólidos en los protocolos del IoT puede dejar a las redes LoRa vulnerables a una variedad de ataques cibernéticos, como la interceptación de datos, la inyección de código malicioso y la denegación de servicio. Estos ataques pueden tener consecuencias graves, incluida la pérdida de datos sensibles y la interrupción de servicios críticos. La falta de estandarización en los protocolos del IoT también puede afectar la interoperabilidad entre dispositivos y plataformas en las redes LoRa. Esto puede dificultar la integración de nuevos dispositivos y la implementación de soluciones de gestión centralizadas, lo que limita la escalabilidad y la eficiencia de la red.

Las redes LoRaWAN son una innovadora tecnología de red de área amplia de baja potencia que ha revolucionado las comunicaciones inalámbricas de larga distancia en el Internet de las Cosas (IoT). Utilizando el protocolo LoRaWAN, estas redes ofrecen una solución rentable, escalable y segura para una amplia gama de aplicaciones, desde la agricultura de precisión hasta la monitorización ambiental y la gestión de activos.

La tecnología LoRaWAN se basa en el protocolo LoRa, que implementa la tecnología de espectro ensanchado en su capa física. Esto permite una transmisión de datos eficiente y confiable incluso en entornos con interferencias electromagnéticas y obstáculos físicos. Sin embargo, este proyecto ha revelado una preocupación significativa en cuanto a la seguridad de las comunicaciones punto a punto utilizando radios LoRa. Se identificó que la falta de cifrado y seguridad en los mensajes transmitidos representa una vulnerabilidad potencial para las aplicaciones que requieren un nivel más alto de seguridad. Esto puede exponer los datos sensibles a posibles interceptaciones y manipulaciones por parte de terceros no autorizados.

Para abordar esta vulnerabilidad, es esencial considerar opciones de cifrado y autenticaciones adecuadas al diseñar sistemas basados en redes LoRa. Esto puede incluir la implementación de tecnologías de cifrado robustas y mecanismos de autenticación sólidos para proteger la confidencialidad e integridad de los datos transmitidos. Además, es importante tener en cuenta que la seguridad en las comunicaciones LoRa es un aspecto clave para su adopción generalizada en aplicaciones críticas para la seguridad. Es fundamental realizar evaluaciones de riesgos y pruebas de seguridad exhaustivas para garantizar que las redes LoRa cumplan con los estándares de seguridad requeridos para cada aplicación específica.

Las tarjetas LoRa tienen un gran potencial para implementaciones de redes de larga distancia con baja potencia y consumo de energía. Sin embargo, su desempeño puede variar significativamente dependiendo del entorno y de las condiciones de transmisión, especialmente en presencia de obstáculos físicos y/o interferencias electromagnéticas.

12. RECOMENDACIONES

Identificar las necesidades de su proyecto: Antes de implementar una red LoRa, es importante identificar las necesidades específicas de su proyecto, incluyendo el alcance geográfico, los dispositivos finales requeridos y los requisitos de seguridad, ya que si la seguridad es crítica lo recomendable es utilizar todo el esquema LoRaWAN, en cambio si los requerimientos de seguridad no son una prioridad el uso de radios LoRa resulta más factible por su velocidad de despliegue, costos y funcionalidad.

Diseño de la red: Diseñe su red LoRa teniendo en cuenta la topografía y las características del terreno para garantizar una cobertura y conectividad óptimas, además de mejorar la tasa de transferencia de bit, sacándole el máximo provecho a los equipos.

Asegúrese de implementar medidas de seguridad adecuadas, como el cifrado de datos y la autenticación de dispositivos, para garantizar la integridad de la red y la privacidad de los datos en la capa LoRa.

BIBLIOGRAFÍA

Aras, Emekcan. Ramachandran, Gowri. Lawrence, Piers & Hughes, Danny. Exploring the Security Vulnerabilities of LoRa. Delft: Delft University of Technology, 2017. pp. 1-6. DOI: 10.1109/CYBConf.2017.7985777. [Fecha de Consulta 28/03/2024].

AYERBE, Ana. La ciberseguridad y su relación con la inteligencia artificial. [en línea]. En: Real Instituto Elcano Príncipe de Vergara, 51. 28006 Madrid. [Fecha de Consulta 04/03/2024]. Disponible en: <https://media.realinstitutoelcano.org/wp-

content/uploads/2021/10/ari128-2020-ayerbe-ciberseguridad-y-su-relacion-con-inteligencia-artificial.pdf>

BRAVO MONTOYA, Andrés; RONDÓN SANABRIA, Jefferson S & GAONA GARCÍA, Elvis E. Desarrollo y prueba de un Sniffer en tiempo real de una red LORA usando GNU-Radio [en línea]. En: Tecnológicas. Medellín: Instituto Tecnológico Metropolitano. Vol. 22, No. 46(Sep.-dic de 2019); p. 185-194. ISSN-p 0123-7799 / ISSN-e 2256-5337. [Fecha de consulta: 02/10/2022]. Disponible en:< https://repositorio.itm.edu.co/bitstream/handle/20.500.12622/1422/Rep_Itm_CEA.pdf?sequence=1&isAllowed=y>

DEL ROSARIO LITARDO, Raúl Fernando & MEZA MORA, Eder Javier. Diseño e implementación de una red iot basado en LORA para el uso de proyectos desarrollados por estudiantes de la universidad de guayaquil, con un sistema de control de acceso a la red iot [en línea]. Guayaquil, 2021, 174 p. Trabajo de grado (Ingeniero en networking y telecomunicaciones). Universidad de guayaquil. Facultad de ciencias matemáticas y físicas. Carrera de ingeniería en networking y telecomunicaciones. [Fecha de consulta: 23/09/2022]. Disponible en: < http://repositorio.ug.edu.ec/handle/redug/56463>

ElectronicCats. CatWAN USB Stick. [En línea]. Disponible en: https://github.com/ElectronicCats/CatWAN_USB_Stick. [Fecha de consulta: 14/12/2023].

FERNANDEZ PAUCAR, Luis Javier & ROBAYO TIPÁN, Alexander Sebastián. Ensamble de un adaptador inalámbrico para el desarrollo del software sniffer en una red LORA y análisis con wireshark [en línea]. Quito,2022, 80 p. Trabajo de grado (Título de Ingeniero en Ciencias de la Computación). Universidad politécnica salesiana sede quito. [Fecha de consulta: 16/09/2022]. Disponible en: < https://dspace.ups.edu.ec/bitstream/123456789/23442/1/UPS%20-%20TTS1028.pdf>

GONZÁLEZ GONZÁLEZ, Cristian Arley; ARÉVALO TAPIAS, Fernando & HERNÁNDEZ GUTIÉRREZ, Jairo (2019). Análisis de seguridad en redes LPWAN para dispositivos IoT [en línea]. En: Vínculos, Ciencia, Tecnología y Sociedad, Vol. 16, No. 2 (julio-diciembre). Bogotá (Colombia): Universidad Distrital Francisco José de Caldas. p. 252-261 e-ISSN 2322-939X DOI: 10.14483/2322939X.15712 < https://www.researchgate.net/publication/343626335_Analisis_de_seguridad_en_redes_LPWAN_para_dispositivos_IoT> [Fecha de consulta: 10/09/2022].

J. Caiza-Narváez., K. Márceles-Villalba., y s. Amador-Donado. Arquitectura basada en tecnologías emergentes y tecnología de monitoreo de tráfico de red.

Investigación e Innovación en Ingenierías, vol. 9, n°3, 18-31, 2021. DOI: https://doi.org/10.17081/invinno.9.3.5340

Kuntke, Franz, Romanenko, Vladimir, Linsner, Sebastian, Steinbrink, Enno, & Reuter, Christian. LoRaWAN security issues and mitigation options by the example of agricultural IoT scenarios. En: Transactions on Emerging Telecommunications Technologies. Delft: Delft University of Technology. Vol. 33 (mayo, 2022). DOI: 10.1002/ett.4452.

LUCAS BULLIAN, Cristian Urbina. Implementación de una red LoRa en el ámbito de la Universidad Nacional de San Martín. Buenos Aires. 2018. 64 p. Proyecto de grado (Ingeniería en Telecomunicaciones). Institución Nacional San Martin. [Fecha de consulta: 10/01/2024]. Disponible en <https://ri.unsam.edu.ar/bitstream/123456789/1806/1/TING%20ESCYT%202018%20BLE-UC.pdf

MATTA HERNANDEZ, Jorge Antonio. Sistema de monitoreo vehicular como herramienta para el sistema de seguridad ciudadana utilizando tecnología zigbee [en línea]. Arequipa, 2018, 154 p. Trabajo de grado (pregrado en ingeniería electrónica). Universidad Nacional de San Agustín de Arequipa [Fecha de consulta: 05/03/2024]. Disponible en: < https://repositorio.unsa.edu.pe/server/api/core/bitstreams/74a79dfc-eb5c-437c-ab11-392e12392a04/content>

MITZI NOEMI, Saquicela tigua. Vulnerabilidades existentes en técnicas tradicionales de almacenamientos de datos vs tecnologías emergentes como blockchain en entidades gubernamentales. Ecuador. 2022. 105 p. Proyecto de grado (ingeniera en tecnologías de la información). Universidad estatal Península de santa elena. [Fecha de consulta: 04/03/2024]. Disponible en < https://repositorio.upse.edu.ec/bitstream/46000/8781/1/UPSE-TTI-2022-0045.pdf>

PEÑA ESPINOZA, Ignacio. Comparación y validación experimental de la gama LoRa Edge sobre redes LORA [en línea]. Santiago de Chile, 2022, 118 p. Trabajo de grado (Título de ingeniero civil eléctrico). Universidad de chile. Facultad de ciencias físicas y matemáticas. Departamento de ingeniería eléctrica. [Fecha de consulta: 23/09/2022]. Disponible en: <https://repositorio.uchile.cl/handle/2250/185548>

PÉREZ GARCÍA, Rubén. Evaluación de LoRa/LORA para escenarios de Smart City [en línea]. Catalunya,2017, 89 p. Trabajo final de grado (Ingeniería Telemática). Universidad Politécnica de Catalunya. [Fecha de consulta: 23/09/2022]. Disponible en:<https://upcommons.upc.edu/bitstream/handle/2117/100922/memoria.pdf?sequence=1&isAllowed=y>

RODRÍGUEZ DÍAZ, Daniel. Red privada LORA para el ámbito de aplicaciones en la Industria 4.0 [en línea]. Catalunya, 2021, 101 p. Trabajo final de grado (Ingeniería electrónica industrial y automática). Universidad Politècnica de Catalunya. [Fecha de consulta: 25/09/2022]. Disponible en: < http://hdl.handle.net/2117/359214>

RODRÍGUEZ BAUTISTA, Juan Antonio. Demostración de los ataques Password Cracking y Spoofing en redes LORA [en línea]. Sevilla, 2022, 110 p. Trabajo Fin de grado (Grado en Ingeniería de las Tecnologías de Telecomunicación). Universidad de Sevilla. Departamento de Ingeniería Telemática. Escuela Técnica Superior de Ingeniería. [Fecha de consulta: 19/09/2022]. Disponible en: < https://idus.us.es/bitstream/handle/11441/137449/TFG4245_Rodr%c3%adguez_2022.pdf?sequence=1&isAllowed=y>

ROJAS CAIRAMPOMA, Marcelo. Tipos de Investigación científica: Una simplificación de la complicada incoherente nomenclatura y clasificación Radio [en línea]. En: Redvet: Revista electrónica de veterinaria. Málaga Veterinaria Organización. Vol. 16, No. 1(2015); p. 1-14. ISSN: 1695-7504. [Fecha de consulta: 15/10/2022]. Disponible en:< https://www.redalyc.org/pdf/636/63638739004.pdf>

RONDON SANABRIA, Jefferson Sebastián & BRAVO MONTOYA, Andrés Felipe. Esquema de seguridad de datos entre los nodos y el gateway en una red LORA [en línea]. Bogotá, 2020, 77 p. Trabajo de Investigación (pregrado en ingeniería de telecomunicaciones). Bogotá (Colombia). Universidad Distrital Francisco José de Caldas. Facultad De Ingeniería. [Fecha de consulta: 16/09/2022]. Disponible en: < https://repository.udistrital.edu.co/bitstream/handle/11349/25252/RondonSanabriaJefersonSebastian2020.pdf?sequence=6&isAllowed=y>

RUEDA R, Johan & TALAVERA P, Jesús M. Similitudes y diferencias entre Redes de Sensores Inalámbricas e Internet de las Cosas: Hacia una postura clarificadora los medios en la sociedad del conocimiento. [en línea]. En: Revista Colombiana de Computación, 2017, Vol. 18, No. 2, pp. 58–74. [Fecha de consulta: 05/03/2024]. Disponible en <https://doi.org/10.29375/25392115.3218 >

VEGA LAZZO, Fernando Xavier. Desarrollo de un manual de prácticas para el uso de equipos LORA en redes de sensores inalámbricos [en línea]. Cuenca, 2020, 219 p. Trabajo de grado (pregrado en ingeniería de electrónica). Universidad Politécnica Salesiana sede Cuenca. [Fecha de consulta: 16/09/2022]. Disponible en: < https://dspace.ups.edu.ec/handle/123456789/19349

VILLAMAR ALVARADO, Antony Alexander. Análisis comparativo de tecnologías wsn en base a la seguridad y su forma de operación, aplicado al entorno iot [en línea]. Ecuador, 2022, 83 p. Trabajo de grado (pregrado en ingeniería de

teleinformática). Guayaquil (Ecuador). Universidad de guayaquil facultad de ingenieria industrial [Fecha de consulta: 05/03/2024]. Disponible en: < https://repositorio.ug.edu.ec/server/api/core/bitstreams/64f49d37-7b20-4114-99d8-7c55dd89a8f9/content>

X. Yang, E. Karampatzakis, C. Doerr and F. Kuipers, "Security Vulnerabilities in LoRaWAN," 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), Orlando, FL, USA, 2018, pp. 129-140, doi: 10.1109/IoTDI.2018.00022.

ANEXO A
CODIGO CONFIGURACION RADIOS LORA

Se presenta el código completo empleado para la configuración de los radios LoRa en el IDE de Arduino

Código del radio transmisor

```
#include <LoRa.h>
```

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DIO0 26

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET    -1
int cont = 0;
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  Serial.begin(115200);
  // Iniciamos la comunicación con la pantalla OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("Fallo al iniciar SSD1306"));
    for (;;)
      ;
  }
  // Configuramos la pantalla OLED
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.print("TRANSMISOR LORA");
  display.display();
  // Configuración LoRa
  pinMode(OLED_RESET, OUTPUT);
  digitalWrite(OLED_RESET, LOW);
  delay(20);
  digitalWrite(OLED_RESET, HIGH);
  Wire.begin(OLED_SDA, OLED_SCL);
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C, false, false)) {
    Serial.println(F("Fallo al iniciar SSD1306"));
    for (;;)
      {}
```

```
  }
  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(1);
  display.setCursor(0, 10);
  display.print("TRANSMISOR LORA");
  display.display();
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DIO0);
  if (!LoRa.begin(915E6)) {
    Serial.println("Error iniciando LoRa");
    while (1)
      ;
  }
  Serial.println("Inicio de LoRa!");
  display.setCursor(0, 30);
  display.print("Inicio de LoRa!");
  display.display();
  delay(2000);
//configuraciones
  LoRa.begin(915E6);  // Frecuencia: 915 MHz
  LoRa.setSpreadingFactor(8);  // Factor de esparcimiento: 8
  LoRa.setSignalBandwidth(125E3);  // Ancho de banda: 125 kHz
  LoRa.setCodingRate4(5);  // Tasa de codificación: 5
  LoRa.setSyncWord(0x12);  // Palabra de sincronización: 0x12
  LoRa.setPreambleLength(8);  // Longitud del preámbulo: 8
}
void loop() {
 static int contador = 0;
  contador++;
  String mensaje = "Hola desde el transmisor LoRa pruebas"  +
String(contador);
  // Enviar el mensaje por LoRa
  LoRa.beginPacket();
  LoRa.print(mensaje);
  LoRa.endPacket();
  Serial.println("Mensaje enviado: " + mensaje);
  // Mostrar información en el display OLED
  display.clearDisplay();
  display.setCursor(0, 0);
  display.print("Transmisor LoRa");
  display.setCursor(0, 20);
  display.print("Mensaje enviado:");
```

```
  display.setCursor(0, 30);
  display.print(mensaje);
  display.display();
  // Esperar antes de enviar el próximo mensaje
  delay(500);
}
```

Código del radio receptor:

```
#include <LoRa.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "WiFi.h"
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DIO0 26
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET    -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
String DatoLoRa;
void setup() {
  Serial.begin(115200);
  // Iniciamos la comunicación con la pantalla OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("Fallo al iniciar SSD1306"));
    for (;;)
      ;
  }
  // Configuramos la pantalla OLED
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  // Configuración LoRa
  pinMode(OLED_RESET, OUTPUT);
  digitalWrite(OLED_RESET, LOW);
  delay(20);
  digitalWrite(OLED_RESET, HIGH);
```

```
  Wire.begin(OLED_SDA, OLED_SCL);
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C, false, false)) {
    Serial.println(F("Fallo al iniciar SSD1306"));
    for (;;)
      {}
  }
  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(1);
  display.setCursor(0, 10);
  display.print("RECEPTOR LORA ");
  display.display();
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DIO0);
  if (!LoRa.begin(915E6)) {
    Serial.println("Error iniciando LoRa");
    while (1)
      ;
  }
  Serial.println("Inicio de LoRa!");
  display.setCursor(0, 30);
  display.print("Inicio de LoRa!");
  display.display();
  delay(2000);
  // Configuración WiFi
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
  //configuracion
    LoRa.begin(915E6);  // Frecuencia: 915 MHz
  LoRa.setSpreadingFactor(8);  // Factor de esparcimiento: 8
  LoRa.setSignalBandwidth(125E3);  // Ancho de banda: 125 kHz
  LoRa.setCodingRate4(5);  // Tasa de codificación: 5
  LoRa.setSyncWord(0x12);  // Palabra de sincronización: 0x12
  LoRa.setPreambleLength(8);  // Longitud del preámbulo: 8
}
void loop() {
  // Escaneo de redes WiFi
  display.clearDisplay();
  display.setCursor(0, 0);
  int n = WiFi.scanNetworks();
  if (n == 0) {
    display.println("No se encontraron redes WiFi.");
```

```
  } else {
    display.print(n);
    display.println(" redes WiFi encontradas:");
    for (int i = 0; i < n; ++i) {
      display.print(i + 1);
      display.print(": ");
      display.print(WiFi.SSID(i));
      display.print(" (");
      display.print(WiFi.RSSI(i));
      display.print(" dBm)");
      display.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN) ? " " :
"*");
    }
  }
  // Escucha de paquetes LoRa
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    Serial.print("Paquete LoRa recibido: ");
    while (LoRa.available()) {
      DatoLoRa = LoRa.readString();
      Serial.print(DatoLoRa);
    }
    int rssi = LoRa.packetRssi();
    Serial.print(" con RSSI ");
    Serial.println(rssi);
    // Mostrar información en el display OLED
    display.clearDisplay();
    display.setCursor(0, 0);
    display.print("Receptor LoRa");
    display.setCursor(0, 20);
    display.print("Paquete recibido:");
    display.setCursor(0, 30);
    display.print(DatoLoRa);
    display.setCursor(0, 40);
    display.print("RSSI:");
    display.setCursor(30, 40);
    display.print(rssi);
    display.display();
  }
  // Esperar antes de volver a escanear o escuchar
  delay(1000);
}
```

ANEXO B
CÓDIGO DE CONFIGURACIÓN DEL CATWAN

Se presenta el código completo utilizado para la configuración del sniffer, este código es libre y lo proporciona la empresa electronicCats. De igual manera, se adicionan las funciones que implementé para la realización de los ataques expuestos en el proyecto.

```
/*
```

```
  Created by Eduardo Contreras @ Electronic Cats 2020
  PLEASE REFER TO THESE LIBRARIES:
    https://github.com/kroimon/Arduino-SerialCommand
    https://github.com/sandeepmistry/arduino-LoRa
  This example code works as a CLI to control your CatWAN USB-Stick
  As a LoRa Sniffer to catch any LoRa Packet
  TODO: Set frequency and channels depending on the region
  This code is beerware; if you see me (or any other Electronic Cats
  member) at the local, and you've found our code helpful,
  please buy us a round!
  Distributed as-is; no warranty is given.
*/
#define SERIALCOMMAND_HARDWAREONLY
#include <SerialCommand.h>
#include <SPI.h>
#include <LoRa.h>
SerialCommand SCmd;
float fwVersion= 0.2;
float frequency = 915;
int spreadFactor = 8;
int bwReference = 7;
int codingRate = 5;
byte syncWord = 0x12;
int preambleLength = 8;
int txPower = 17;
int channel = 0;
bool rx_status = false;
int inv_iq = 0;
#define MAX_PACKET_SIZE 256
byte capturedPacket[MAX_PACKET_SIZE];
int capturedPacketSize = 0;
void setup(){
  pinMode(LED_BUILTIN,OUTPUT);      // Configure the onboard LED for output
  digitalWrite(LED_BUILTIN,LOW);    // default to LED off
  pinMode(RFM_DIO5,INPUT);
  Serial.begin(115200);
  while (!Serial);
  Serial.println("Welcome to the LoRa Sniffer CLI " + String(fwVersion,1) +
"v\n");
  Serial.println("With this sketch you can scan the LoRa spectrum");
  Serial.println("Changing the Frequency, Spreading Factor, BandWidth or the
IQ signals of the radio.");
```

```
  Serial.println("Type help to get the available commands.");
  Serial.println("Electronic Cats ® 2020");
  // Setup callbacks for SerialCommand commands
  SCmd.addCommand("help",help);
  SCmd.addCommand("set_rx",set_rx);
  SCmd.addCommand("set_tx",set_tx);
  SCmd.addCommand("set_tx_hex",set_tx_hex);
  SCmd.addCommand("set_tx_ascii",set_tx_ascii);
  SCmd.addCommand("set_freq",set_freq);
  SCmd.addCommand("set_sf",set_sf);
  SCmd.addCommand("set_bw",set_bw);
  SCmd.addCommand("set_cr",set_cr);
  SCmd.addCommand("set_sw",set_sw);
  SCmd.addCommand("set_pl",set_pl);
  SCmd.addCommand("set_tp",set_tp);
  SCmd.addCommand("set_chann",set_chann);
  SCmd.addCommand("set_inv_iq",set_inv_iq);
  SCmd.addCommand("get_config",get_config);
  SCmd.addCommand("get_freq",get_freq);
  SCmd.addCommand("get_sf",get_sf);
  SCmd.addCommand("get_bw",get_bw);
  SCmd.addCommand("get_cr",get_cr);
  SCmd.addCommand("get_sw",get_sw);
  SCmd.addCommand("get_pl",get_pl);
  SCmd.addCommand("get_tp",get_tp);
  //*********************************************************vectores de
ataque***************************
    SCmd.addCommand("dos", attackDoSCommand);
    SCmd.addCommand("replay", attackReplayCommand);
  //*********************************************************vectores de
ataque***************************
  SCmd.setDefaultHandler(unrecognized);  // Handler for command that isn't
matched  (says "What?")
  LoRa.setPins(SS, RFM_RST, RFM_DIO0);
  if (!LoRa.begin(915E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  //LoRa.setFrequency(915E6);
  LoRa.setSpreadingFactor(spreadFactor);
  //LoRa.setSignalBandwidth(125E3);
  LoRa.setCodingRate4(codingRate);
  //LoRa.setSyncWord(0x12);
```

```
  LoRa.setSyncWord(syncWord);
  //LoRa.setPreambleLength(8);
  LoRa.setPreambleLength(preambleLength);
  rx_status = false;
  LoRa.onReceive(onReceive);
}
void loop()
{
  SCmd.readSerial();     // We don't do much, just process serial commands
}
//*************************************************************Vectores
de Ataque LoRa************************************************
void attackDoSCommand() {
    char *arg1 = SCmd.next();
    char *arg2 = SCmd.next();
    if (arg1 != NULL && arg2 != NULL) {
        int numPackets = atoi(arg1);
        int delayTime = atoi(arg2);
        attackDoS(numPackets, delayTime);
    } else {
        Serial.println("Usage: dos <num_packets> <delay_time>");
    }
}
void attackReplayCommand() {
    char *arg1 = SCmd.next();
    char *arg2 = SCmd.next();
    if (arg1 != NULL && arg2 != NULL) {
        int numRepeats = atoi(arg1);
        int delayTime = atoi(arg2);
        if (capturedPacketSize > 0) {
            attackReplay(capturedPacket, capturedPacketSize, numRepeats,
delayTime);
        } else {
            Serial.println("No packet captured to replay.");
        }
    } else {
        Serial.println("Usage: replay <num_repeats> <delay_time>");
    }
}
// Función para realizar un ataque de denegación de servicio (DoS)
void attackDoS(int numPackets, int delayTime) {
  static int contador = 0;
```

```
  for (int i = 0; i < numPackets; i++) {
  contador++;
    String mensaje = "Mensaje de ataque #" + String(contador);
    // Enviar el mensaje por LoRa
    LoRa.beginPacket();
    LoRa.print(mensaje);
    LoRa.endPacket();
    Serial.println(mensaje);
    delay(delayTime);
  }
}
// Función para realizar un ataque de repetición
void attackReplay(byte* packet, int packetSize, int numRepeats, int
delayTime) {
    for (int i = 0; i < numRepeats; i++) {
        LoRa.beginPacket();
        LoRa.write(packet, packetSize);
        LoRa.endPacket();
        Serial.println("replicando paquete");
        delay(delayTime);
    }
}
//**************************************************************Vectores
de Ataque LoRa**************************************************
void help(){
  Serial.println("Fw version: " + String(fwVersion,1)+"v");
  Serial.println("\tConfiguration commands:");
  Serial.println("\tset_rx");
  Serial.println("\tset_tx");
  Serial.println("\tset_sf");
  Serial.println("\tset_bw");
  Serial.println("\tset_cr");
  Serial.println("\tset_sw");
  Serial.println("\tset_pl");
  Serial.println("\tset_tp");
  Serial.println("\tset_inv_iq");
  Serial.println("Monitor commands:");
  Serial.println("\tget_freq");
  Serial.println("\tget_sf");
  Serial.println("\tget_bw");
  Serial.println("\tget_cr");
  Serial.println("\tget_sw");
  Serial.println("\tget_pl");
```

```
  Serial.println("\tget_tp");
  Serial.println("\tget_config");
  Serial.println("..help");
}
/**********Set configuration*************/
void set_freq(){
  char *arg;
  arg = SCmd.next();
  frequency = atof(arg);
  if (arg != NULL){
      long freq = frequency*1000000;
      LoRa.setFrequency(freq);
      Serial.println("Frequency set to " + String(frequency) + " MHz");
      rx_status = false;
  }
  else {
    Serial.println("No argument");
  }
}
void set_sf(){
  char *arg;
  arg = SCmd.next();
  if (arg != NULL){
    spreadFactor = atoi(arg);
    if(spreadFactor < 6 || spreadFactor > 12){
      Serial.println("Error setting the Spreading factor");
      Serial.println("Value must be between 6 and 12");
      return;
    }
    else{
      LoRa.setSpreadingFactor(spreadFactor);
      Serial.println("Spreading factor set to " + String(spreadFactor));
      rx_status = false;
    }

  }
  else {
    Serial.println("No argument");
  }
}
void set_cr(){
  char *arg;
```

```
  arg = SCmd.next();
  if (arg != NULL){
    codingRate = atoi(arg);
    if(codingRate > 8 || codingRate < 5){
      Serial.println("Error setting the Coding Rate");
      Serial.println("Value must be between 5 and 8");
      return;
    }
    else{
      LoRa.setCodingRate4(codingRate);
      Serial.println("CodingRate set to 4/" + String(codingRate));
      rx_status = false;
    }
  }
  else {
    Serial.println("No argument");
  }
}
void set_sw(){
  char *arg;
  byte data;
  int i;
  arg = SCmd.next();    // Get the next argument from the SerialCommand
object buffer
  if(arg != NULL){
      if((arg[0] > 64 && arg[0]< 71 || arg[0] > 47 && arg[0]< 58) && (arg[1]
> 64 && arg[1]< 71 || arg[1] > 47 && arg[1]< 58) && arg[2] == 0){
          data = 0;
          data = nibble(*(arg))<<4;
          data = data|nibble(*(arg + 1));
          LoRa.setSyncWord(data);
          syncWord = data;
          Serial.print("Sync word set to 0x");
          Serial.println(data, HEX);
      }
      else{
        Serial.println("Use yy value. The value yy represents any pair of
hexadecimal digits. ");
        return;
      }
  }
  else {
    Serial.println("No argument");
```

```
  }
}
void set_pl(){
  char *arg;
  arg = SCmd.next();
  if (arg != NULL){
    preambleLength = atoi(arg);
    if(preambleLength < 6 || preambleLength > 65536){
      Serial.println("Error setting the Preamble Length");
      Serial.println("Value must be between 6 and 65536");
      return;
    }
    else{
      LoRa.setPreambleLength(preambleLength);
      Serial.println("Preamble lenght set to " + String(preambleLength));
      rx_status = false;
    }

  }
  else {
    Serial.println("No argument");
  }
}
void set_tp(){
  char *arg;
  arg = SCmd.next();
  if (arg != NULL){
    txPower = atoi(arg);
    if(txPower > 1 || txPower < 21){
      Serial.println("Error setting the TX Power");
      Serial.println("Value must be between 2 and 20");
      return;
    }
    else{
      LoRa.setTxPower(txPower);
      Serial.println("TX Power set to " + String(txPower));
      rx_status = false;
    }
  }
  else {
    Serial.println("No argument");
  }
```

```
}
void set_bw(){
  char *arg;
  arg = SCmd.next();    // Get the next argument from the SerialCommand
object buffer
  int bwRefResp = bwReference; //save the previous data
  bwReference = atoi(arg);
  if (arg != NULL){
      switch (bwReference){
        case 0:
          LoRa.setSignalBandwidth(7.8E3);
          rx_status = false;
          Serial.println("Bandwidth set to 7.8 kHz");
          break;
        case 1:
          LoRa.setSignalBandwidth(10.4E3);
          rx_status = false;
          Serial.println("Bandwidth set to 10.4 kHz");
          break;
        case 2:
          LoRa.setSignalBandwidth(15.6E3);
          rx_status = false;
          Serial.println("Bandwidth set to 15.6 kHz");
          break;
        case 3:
          LoRa.setSignalBandwidth(20.8E3);
          rx_status = false;
          Serial.println("Bandwidth set to 20.8 kHz");
          break;
        case 4:
          LoRa.setSignalBandwidth(31.25E3);
          rx_status = false;
          Serial.println("Bandwidth set to 31.25 kHz");
          break;
        case 5:
          LoRa.setSignalBandwidth(41.7E3);
          rx_status = false;
          Serial.println("Bandwidth set to 41.7 kHz");
          break;
        case 6:
          LoRa.setSignalBandwidth(62.5E3);
          rx_status = false;
          Serial.println("Bandwidth set to 62.5 kHz");
```

```
          break;
        case 7:
          LoRa.setSignalBandwidth(125E3);
          rx_status = false;
          Serial.println("Bandwidth set to 125 kHz");
          break;
        case 8:
          LoRa.setSignalBandwidth(250E3);
          rx_status = false;
          Serial.println("Bandwidth set to 250 kHz");
          break;
        case 9:
          LoRa.setSignalBandwidth(500E3);
          rx_status = false;
          Serial.println("Bandwidth set to 500 kHz");
          break;

        default:
          Serial.println("Error setting the bandwidth value must be between
0-8");
          bwReference = bwRefResp; //if there's no valid data restore
previous value
          break;
      }
  }
  else {
    Serial.println("No argument");
  }
}
byte nibble(char c)
{
  if (c >= '0' && c <= '9')
    return c - '0';

  if (c >= 'a' && c <= 'f')
    return c - 'a' + 10;
  if (c >= 'A' && c <= 'F')
    return c - 'A' + 10;
  return 0;  // Not a valid hexadecimal character
}
void set_tx(){
  char *arg;
```

```
  byte data[64];
  int i;
  arg = SCmd.next();    // Get the next argument from the SerialCommand
object buffer
  if(arg != NULL){
    for(i = 0; arg != NULL; i++){
        if((arg[0] > 47 && arg[0]< 58) && (arg[1] > 47 && arg[1]< 58) &&
(arg[2] > 47 && arg[2]< 58) && arg[3] == 0){
          data[i] = (byte)strtoul(arg, NULL, 10);
          //Serial.println(data[i],BIN);
        }
        else {
          Serial.println("Use a series of xxx values separated by spaces.
The value xxx represents a 3-digit number. ");
          return;
        }
        arg = SCmd.next();    // Get the next argument from the
SerialCommand object buffer
    }
    for(int j = 0; j < i; j++){
      Serial.print(data[j]);
      Serial.print(" ");
    }
    LoRa.beginPacket();                   // start packet
    LoRa.write(data, i);                  // add payload
    LoRa.endPacket(true);                 // finish packet and send it
    Serial.println();
    Serial.print(i);
    Serial.println(" byte(s) sent");
    rx_status = false;
  }
  else {
    Serial.println("No argument");
  }
}
void set_tx_hex(){
  char *arg;
  byte data[64];
  int i;
  arg = SCmd.next();    // Get the next argument from the SerialCommand
object buffer
  if(arg != NULL){
    for(i = 0; arg != NULL; i++){
```

```
      if((arg[0] > 64 && arg[0]< 71 || arg[0] > 47 && arg[0]< 58) && (arg[1]
> 64 && arg[1]< 71 || arg[1] > 47 && arg[1]< 58) && arg[2] == 0){
          data[i] = 0;
          data[i] = nibble(*(arg))<<4;
          data[i] = data[i]|nibble(*(arg + 1));
      }
      else{
        Serial.println("Use a series of yy values separated by spaces. The
value yy represents any pair of hexadecimal digits. ");
        return;
      }
      arg = SCmd.next();    // Get the next argument from the SerialCommand
object buffer
    }
    for(int j = 0; j < i; j++){
      Serial.print(data[j]);
      Serial.print(" ");
    }
    LoRa.beginPacket();                   // start packet
    LoRa.write(data, i);                  // add payload
    LoRa.endPacket(true);                 // finish packet and send it
    Serial.println();
    Serial.print(i);
    Serial.println(" byte(s) sent");
    rx_status = false;
  }
  else {
    Serial.println("No argument");
  }
}
void set_tx_ascii(){
  char *arg;
  arg = SCmd.next();    // Get the next argument from the SerialCommand
object buffer
  if (arg != NULL){
      LoRa.beginPacket();               // start packet
      for(int i = 0;;i++){
        if(arg[i] == 0)
          break;
        Serial.print(arg[i]);
        LoRa.write(arg[i]);             // add payload
      }
```

```
      LoRa.endPacket(true);             // finish packet and send it
      Serial.println(" ASCII message sent");
      rx_status = false;
  }
  else {
    Serial.println("No argument");
  }
}
void set_chann(){
  char *arg;
  arg = SCmd.next();    // Get the next argument from the SerialCommand
object buffer
  channel = atoi(arg);
  if (arg != NULL){
    if(channel > -1 && channel < 64){
    long freq = 902300000 + channel*125000;
    frequency = (float)freq/1000000;
    LoRa.setFrequency(freq);
    Serial.println("Uplink channel set to " + String(channel));
    rx_status = false;
    }
    else if(channel > 63 && channel < 72){
    long freq = 903000000 + (channel - 64)*500000;
    frequency = (float)freq/1000000;
    LoRa.setFrequency(freq);
    Serial.println("Uplink channel set to " + String(channel));
    rx_status = false;
    }
    else{
      Serial.println("Error setting the channel");
      Serial.println("Value must be between 0 and 63");
    }
  }
  else {
    Serial.println("No argument");
  }
}
void set_inv_iq(){
  char *arg;
  arg = SCmd.next();
  if (arg != NULL){
    inv_iq = atoi(arg);
    if(inv_iq != 0 && inv_iq != 1){
```

```
      Serial.println("Error setting the InvertIQ parameter");
      Serial.println("Value must be 0 or 1");
      return;
    }
    else{
      if(inv_iq){
        LoRa.enableInvertIQ();
        Serial.println("InvertIQ enabled ");
        rx_status = false;
      }
      else {
        LoRa.disableInvertIQ();
        Serial.println("InvertIQ disabled");
        rx_status = false;
      }
    }

  }
  else {
    Serial.println("No argument");
  }
}
void set_rx(){
  char *arg;
  arg = SCmd.next();
  if (arg != NULL){
    frequency = atof(arg);
//    if(frequency > 902 && frequency < 923){
      long freq = frequency*1000000;
      LoRa.setFrequency(freq);
      Serial.println("LoRa radio receiving at " + String(frequency) + "
MHz");
      while (digitalRead(RFM_DIO5) == LOW){
          Serial.print(".");
        }
      LoRa.receive();
      rx_status = true;
  }
  else {
    Serial.println("LoRa radio receiving at " + String(frequency) + " MHz");
    LoRa.receive();
    rx_status = true;
```

```
  }
}
/**********Get information*************/
void get_freq(){
  Serial.print("Frequency = ");
  Serial.println(frequency);
}
void get_sf(){
  Serial.print("Spreading factor = ");
  Serial.println(spreadFactor);
}
void get_cr(){
  Serial.print("Coding Rate = ");
  Serial.println(codingRate);
}
void get_sw(){
  Serial.print("Sync Word = 0x");
  Serial.println(syncWord, HEX);
}
void get_pl(){
  Serial.print("Preamble Length = ");
  Serial.println(preambleLength);
}
void get_tp(){
  Serial.print("TX Power = ");
  Serial.println(txPower);
}
void get_bw(){
  Serial.println("Bandwidth = ");
  switch (bwReference){
    case 0:
      Serial.println("7.8 kHz");
      break;
    case 1:
      Serial.println("10.4 kHz");
      break;
    case 2:
      Serial.println("15.6 kHz");
      break;
    case 3:
      Serial.println("20.8 kHz");
      break;
    case 4:
```

```
      Serial.println("31.25 kHz");
      break;
    case 5:
      Serial.println("41.7 kHz");
      break;
    case 6:
      Serial.println("62.5 kHz");
      break;
    case 7:
      Serial.println("125 kHz");
      break;
    case 8:
      Serial.println("250 kHz");
      break;
    case 9:
      Serial.println("500 kHz");
      break;
    default:
      Serial.println("Error setting the bandwidth value must be between 0-
8");
      break;
  }
}
void get_config(){
  Serial.println("\nRadio configurations: ");
  Serial.println("Frequency = " + String(frequency) + " MHz");
  Serial.print("Bandwidth = ");
  switch (bwReference){
    case 0:
      Serial.println("7.8 kHz");
      break;
    case 1:
      Serial.println("10.4 kHz");
      break;
    case 2:
      Serial.println("15.6 kHz");
      break;
    case 3:
      Serial.println("20.8 kHz");
      break;
    case 4:
      Serial.println("31.25 kHz");
```

```
      break;
    case 5:
      Serial.println("41.7 kHz");
      break;
    case 6:
      Serial.println("62.5 kHz");
      break;
    case 7:
      Serial.println("125 kHz");
      break;
    case 8:
      Serial.println("250 kHz");
      break;
    case 9:
      Serial.println("500 kHz");
      break;
  }
  Serial.println("Spreading Factor = " + String(spreadFactor));
  Serial.println("Coding Rate = 4/" + String(codingRate));
  Serial.print("Sync Word = 0x");
  Serial.println(syncWord, HEX);
  Serial.println("Preamble Length = " + String(preambleLength));
  Serial.print("InvertIQ = ");
  Serial.println(inv_iq?"enabled":"disabled");
  Serial.println("TX Power = " + String(txPower));
  Serial.println("Rx active = " + String(rx_status));
}
// This gets set as the default handler, and gets called when no other
command matches.
void unrecognized(const char *command) {
  Serial.println("Command not found, type help to get the valid commands");
}
void onReceive(int packetSize) {
  char buf[256];
  int i;
  // received a packet
  Serial.println("Received packet ");
  Serial.print(packetSize);
  //Serial.print(" bytes ' ");
    // Almacenar el paquete capturado
    capturedPacketSize = packetSize;
    for (int i = 0; i < packetSize; i++) {
        capturedPacket[i] = LoRa.read();
```

```
    }
  // read packet
  for (i = 0; i < packetSize; i++) {
    buf[i] = LoRa.read();
    //Serial.print("<0x");
    Serial.print(buf[i] < 16 ? "0" : "");
    Serial.print(buf[i], HEX);
  }
  buf[i] = 0;
  Serial.println();
  Serial.print("ASCII: '");
  for (i = 0; i < packetSize; i++) {
    Serial.print(buf[i]);
  }
  // print RSSI of packet
  Serial.print("' with RSSI ");
  Serial.print(LoRa.packetRssi());
  Serial.println();
  LoRa.receive();
}
```

ANEXO
RESUMEN ANALÍTICO

Título del proyecto	Análisis de seguridad en redes LoRa
Autor(es)	GONZALEZ BETANCOURT, Daniel Andrés Dagonzalez85533@umanizales.edu.co
Presidente / Asesor temático	ALEJANDRO ROJAS, Michael. mrojas@umanizales.edu.co
Tipo de documento	Trabajo de Grado

Referencia documento	Manizales, 2024, 91pg. Trabajo final. (Estudiante). U. de Manizales. Ciencias e Ingeniería.
Institución	Ingeniería de sistemas y telecomunicaciones, Facultad de ciencias e ingeniería, Universidad de Manizales
Palabras claves	Redes, vulnerabilidad, LoRaWAN, Implementación, Seguridad
Descripción	El internet de las cosas (IoT), presenta un gran avance tecnológico y forma parte de la cuarta revolución industrial. Estas redes interoperables representan una interconexión de millones de dispositivos IP, que se comunican a través de diferentes protocolos de comunicación. Todo esto con el fin de compartir datos en tiempo real. Esta tecnología se integra con otras como, Big data, machine learning, inteligencia artificial para el análisis, y la toma de decisiones en base a todos los datos recolectados.
Fuentes	-logitek (juan Vargas), [Conceptos técnicos básicos que te ayudarán a entender LoRa LoRaWAN]. España, Barcelona (https://www.m2mlogitek.com/conceptos-tecnicos-basicos-que-te-ayudaran-a-entender-LoRa-y-LoRawan-low-power-wide-area-network-en-pocos-minutos/),Junio/14/2022 -Unit Electronics, LILYGO LYLIGO LoRa32 915Mhz V1.6.1]. México, Barcelona (https://uelectronics.com/producto/lilygo-Lyligo-LoRa32-915mhz-v1-6-1/)Enero/01/2016.
Metodología	Este proyecto corresponde a una investigación experimental ya que, según Rojas, se caracteriza por la aplicación de estímulos a la unidad experimental, se observa la reacción y se registran los resultados. Consiste en una relación causa – efecto. En este caso constituye la realización de ataques (como estímulos) a la unidad experimental (redes LoRaWAN) y la documentación de los resultados obtenidos. Se elaboró el diseño de la red, la distribución y determinación de los equipos específicos necesarios para el despliegue de la arquitectura LoRa. Se validaron las características técnicas de todos los dispositivos necesarios para la correcta implementación de la comunicación y el entorno de pruebas idóneo
Conclusiones	En conclusión, las redes LoRa son una tecnología de red de área amplia de baja potencia que permite la comunicación inalámbrica de larga distancia entre dispositivos finales y servidores de red. Al ser una solución rentable, escalable, las redes LoRaWAN han demostrado ser útiles en una variedad de aplicaciones de IoT, como la agricultura de precisión, la gestión de activos y la monitorización ambiental.
Anexos	ANEXO A: Código de configuración de los radios ANEXO B: Código de configuración del CatWAN