ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

AMPLIFICADOR DE ATAQUES DE DENEGACIÒN DE SERVICIOS EN REDES LORAWAN BASANDO EN CONTENEDORES.

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

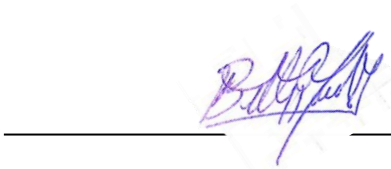
BILLY ORLANDO PROAÑO MORALES
billy.proano@epn.edu.ec

DIRECTOR: MSc. Jhonattan Javier Barriga Andrade
jhonattan.barriga@epn.edu.ec

DMQ, agosto - 2022

CERTIFICACIONES

Yo, BILLY ORLANDO PROAÑO MORALES declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



BILLY ORLANDO PROAÑO MORALES

Certifico que el presente trabajo de integración curricular fue desarrollado por BILLY ORLANDO PROAÑO MORALES, bajo mi supervisión.

JHONATTAN BARRIGA

DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

BILLY ORLANDO PROAÑO MORALES

DEDICATORIA

Dedico este trabajo de integración curricular a mis padres en primer lugar por haberme brindado la oportunidad de formarme profesionalmente y a pesar de las adversidades e inconvenientes siempre me han apoyado en todo momento, a mi hija que es el motivo principal de salir adelante y brindarle un ejemplo a seguir. Este proyecto también está dedicado a mis hermanos ya que también han sido un gran apoyo emocional para seguir adelante.

Billy Proaño.

AGRADECIMIENTO

Agradezco al MSc. Jhonattan Barriga por su apoyo y motivación para salir adelante con el proyecto, su guía ha sido fundamental para que el proyecto sea un éxito y a pesar de los inconvenientes que se presentaron, supo cómo aconsejarme y brindarme una gran cantidad de sus conocimientos.

A mi familia (esposa e hija) por su apoyo incondicional, a mis padres por ser un gran ejemplo para seguir.

A mis compañeros y amigos que a lo largo de la carrera hemos tenido altas y bajas experiencias,

Billy Proaño

ÍNDICE DE CONTENIDO

RESUMEN

Desde el inicio de IoT, se han propuesto varios protocolos de comunicación, diseñados para conectar diversos dispositivos con una potencia de procesamiento limitada, que a veces requiere el funcionamiento con baterías, potencia de procesamiento, a veces con funcionamiento a pilas. Entre estos protocolos destaca LoRaWAN, que es un protocolo LPWAN basado en el estándar IEEE802.15.4.Se centra en las comunicaciones de largo alcance de bajo consumo y en las capacidades de los dispositivos consumo de energía y ofrece la posibilidad de crear aplicaciones complejas y escalables basadas en su arquitectura de red. Aplicaciones escalables sobre su arquitectura de red.

Desde la primera versión de LoRaWAN se han presentado problemas de seguridad que han venido siendo tratados con cada versión. Por su parte, este proecto, pretende aprovechar la vulnerabilidad de Denegación de Servicio (DoS), intentando amplificar su potencia de ataque contra la infraestructura que maneja los datos de LoRaWAN. Para ello, se utilizará un framework, que permita realizar la manipulación de los paquetes para intentar afectar una infraestructura LoRaWAN. Se implementará una infraestructura virtual para realizar las pruebas. Durante las pruebaspretende identificar el comportamiento que tendría un dispositivo dentro de una red LoRaWAN.

.

Palabras clave: LoRaWAN, IoT, LPWAN, Framework**,** DOS, DDOS

ABSTRACT

Since the inception of IoT, various communication protocols have been proposed, designed to connect various devices with limited processing power, sometimes requiring battery operation, processing power, sometimes battery operation. Among these protocols, LoRaWAN stands out, which is an LPWAN protocol based on the IEEE802.15.4 standard. It focuses on low-power long-range communications and the capabilities of energy-consuming devices and offers the possibility of creating complex and scalable applications. based on your network architecture. Scalable applications on your network architecture.

Since the first version of LoRaWAN, there have been security problems that have been addressed with each version. For its part, this project aims to take advantage of the Denial of Service (DoS) vulnerability, trying to amplify its attack power against the infrastructure that handles LoRaWAN data. For this, a framework will be used, which allows packet manipulation to try to affect a LoRaWAN infrastructure. A virtual infrastructure will be implemented to carry out the tests. During the tests, it aims to identify the behavior that a device would have within a LoRaWAN network.

Keywords: LoRaWAN, IoT, LPWAN, Framework, DOS, DDOS

1. INTRODUCIÓN

Durante la última década, la cantidad de objetos que agregaron algo de poder de cómputo y están conectados a Internet (por ejemplo, electrodomésticos, equipos industriales, equipos médicos) ha crecido sustancialmente. Esta realidad se conoce como Internet de las Cosas (IoT). Estas tecnologías están en auge a nivel mundial y han comenzado a utilizarse en varios países. Un ejemplo de esto es el avance de las ciudades inteligentes o smart cities.

En la era del avance tecnológico, el paradigma IoT ha surgido de manera disruptiva, permitiendo incorporarlo a diversos campos y aplicaciones. La tecnología IoT ha entrado Industria, ciudad y hogar. En el llamado hogar inteligente o smart home (aplicado a la seguridad, ciencia de la energía y confort, salud y bienestar, asistencia a personas mayores o discapacitadas) y Al más alto nivel en las llamadas ciudades inteligentes o smart cities (para optimización Servicios públicos tradicionales, seguimiento del consumo energético, calidad del aire y del sonido, gestión del tráfico, reducción de costes de explotación, iluminación, aparcamiento, etc.). [1]

Por supuesto, el nacimiento del Internet de las Cosas no sucedió de la noche a la mañana. Un trabajo de investigación en La Universidad Metropolitana de Manchester presenta una cronología interesante y detallada, desde el desarrollo de las TIC (Tecnologías de la Información y la Comunicación) hasta el Internet de las Cosas, desde la invención del telégrafo electromagnético (1830) hasta el anuncio de Microsoft en 2018 de una inversión de 5.000 millones de dólares en Internet. de las Cosas los próximos cuatro años. [2]

Además, se han desarrollado protocolos a nivel de datos que tienen en cuenta las necesidades de la realidad IoT, buscando principalmente ser livianos. Entre los más destacados se encuentran HTTP/REST (sin cambios para IoT), transporte de telemetría de colas de mensajes (MQTT), MQTT para redes de sensores (MQTT-SN), protocolo de aplicación restringida (CoAP, desarrollado por Internet Engineering Task Force, IETF), protocolo avanzado de colas de mensajes (AMQP)) y Protocolo extensible de mensajería y presencia (XMPP) [1]

Una vez que se ha mencionado protocolos de intercambio de datos, también se considera importante de hablar un poco más a detalle de lo que es el protocolo LPWAN, este protocolo se va a encargar de transmitir pequeñas cantidades de datos a largas distancias. Su aplicabilidad al Internet de las Cosas se basa en que, en este campo, los dispositivos y sensores suelen transmitir pequeñas cantidades de información, a veces de forma no constante en el tiempo [3].

Otro protocolo de comunicación es sigfox, es conocido como un proveedor mundial de comunicaciones LPWAN, y ofrece servicios que mantiene las comunicaciones seguras, de manera bidireccional que facilitan el uso del internet de las cosas. [3] Adicionalmente a esto se conoce que existen protocolos de LPWAN como LoRaWAN, que es uno de los más utilizados por su facilidad y versatilidad para poder hacer cambios sobre el mismo.

Lo que permite aumentar los alcances del proyecto, es poder realizar cambios sobre la arquitectura de LoRaWAN, para este proyecto en específico se realizara cambios sobre los mensajes que se envían hacia los dispositivos que se encuentran creados dentro de la infraestructura,

Para partir con el concepto de LoraWAN es necesario comprender la tecnología inalámbrica Lora, es una tecnología como Wifi, Bluetooth, que utiliza una modulación de radiofrecuencia actualmente, la tecnología Lora está gestionada por la "Lora Alliance", que certifica a los fabricantes de hardware que deseen utilizar la tecnología es una tecnología ideal para conexiones de larga distancia y redes IoT, que requieren sensores sin corriente de la red y

tienen aplicaciones a gran escala como ciudades inteligentes, Áreas con baja cobertura como agricultura.

LoraWAN es un protocolo de red que utiliza tecnología Lora para redes de área amplia y de bajo consumo de energía. El protocolo LoraWAN consta de puertas de enlace (Gateway) y nodos.
Gateway (antena): Son los encargados de recibir y enviar información a los nodos.
Nodos (dispositivos finales): Son dispositivos terminales que envían y reciben información hacia y desde la puerta de enlace.

Permite la interconexión entre objetos inteligentes sin la necesidad de una instalación local compleja, y también proporciona una amplia gama de libertad de uso para usuarios finales, desarrolladores y empresas que deseen instalar su propia red de Internet de las cosas (IoT).

También es necesario conocer sobre los ataques de denegación de servicio en una infraestructura LoRaWAN y estos tienen como objetivo deshabilitar el uso de un sistema, aplicación o máquina para evitar el servicio previsto. Dichos ataques afectan a las fuentes que proporcionan la información, como aplicaciones o canales de transmisión, así como a las redes informáticas [4].

Existen dos técnicas para tales ataques: Denegación de servicio (DoS) y Denegación de servicio distribuida (DDoS). La diferencia entre los dos es la cantidad de computadoras o IP que ejecutan el ataque [4].

En un ataque DoS, una gran cantidad de solicitudes de un servicio se originan desde la misma máquina o dirección IP, consumiendo los recursos proporcionados por el servicio hasta que llega el momento en que deja de responder y comienza a rechazar solicitudes, lo que es una implementación de denegación de servicio [4].

En el caso de un ataque DDoS, la solicitud o conexión se realiza utilizando una gran cantidad de computadoras o direcciones IP. Todas estas solicitudes se realizan al mismo servicio comprometido al mismo tiempo. Los ataques DDoS son más difíciles de detectar porque la cantidad de solicitudes proviene de diferentes IP y los administradores no pueden bloquear la IP que realiza la solicitud, como sucede en los ataques DoS [4].

Es importante conocer previamente a los ataques de denegación de servicios el tipo de seguridad que utiliza el protocolo LoraWAN, que utiliza el estándar de cifrado avanzado (AES) de clave simétrica para cifrar, descifrar y firmar mensajes.

Una forma de amplificar los ataques sobre una red LoRaWAN es el uso de Kubernetes también conocido como k8s o "kube" es una plataforma de código abierto para contenedores que automatiza muchos procesos manuales que implican la implementación, administración y ajuste de aplicaciones alojadas en él.
El principal beneficio de usar Kubernetes en su entorno es que puede obtener una plataforma para programar y ejecutar sus contenedores en máquinas virtuales (VM) o clústeres físicos, especialmente cuando está optimizando el desarrollo de aplicaciones en la nube. Es así como que podemos hacer uso de los kubernetes para amplificar los ataques, partiendo de la idea que los malwares son piezas de software, el objetivo principal para este proyecto es hacer uso de los kubernetes para amplificar los ataques sobre las redes LoRaWAN.

En términos generales, le permite implementar infraestructura basada en contenedores en un entorno de producción y confiar completamente en ella. Además, puede realizar muchas de las mismas tareas que otras plataformas de aplicaciones o sistemas de gestión, pero en sus

contenedores, porque Kubernetes realiza automáticamente las tareas operativas. Los desarrolladores pueden usar Kubernetes como plataforma de tiempo de ejecución para crear aplicaciones en la nube utilizando patrones, que son herramientas necesarias para diseñar aplicaciones y servicios basados en contenedores.

Los contenedores son una forma de virtualización del sistema operativo. Se puede usar un solo contenedor para ejecutar cualquier cosa, desde microservicios o procesos de software hasta aplicaciones más grandes. Incluya todos los archivos ejecutables, código binario, bibliotecas y archivos de configuración necesarios en un contenedor. Sin embargo, a diferencia de los métodos de virtualización de máquinas o servidores, los contenedores no contienen imágenes del sistema operativo. Esto los hace más livianos, más portátiles y reducen significativamente los gastos generales. En implementaciones de aplicaciones más grandes, se pueden implementar varios contenedores como uno o más clústeres de contenedores. Estos clústeres pueden ser administrados por un orquestador de contenedores (como Kubernetes) explicados previamente en esta misma sección.

Los contenedores son una forma simplificada de crear, probar, lanzar y reiniciar aplicaciones en una variedad de entornos, desde la computadora portátil local de un desarrollador hasta un centro de datos local, e incluso en la nube. Algunos de los beneficios de los contenedores son

Beneficios de usar kubernetes:

- Organice sus contenedores en múltiples hosts
- Controle y automatice la implementación y las actualizaciones de aplicaciones
- Agregue almacenamiento para ejecutar aplicaciones con estado
- Amplíe las aplicaciones en contenedores y sus recursos según sea necesario
- Administre los servicios de forma declarativa para garantizar que las aplicaciones implementadas siempre se ejecuten correctamente

1.1 Objetivo general

Desarrollar un generador de paquetes de ataques de DoS sobre las redes LoraWAN, y desplegarlo en contenedores dinámicos para mejorar su efectividad

1.2 Objetivos específicos

1. Diseñar una arquitectura basada en contenedores para desplegar el componente desarrollado con capacidades de auto escalamiento
2. Identificar los requerimientos de hardware y software necesarios para configurar los contenedores.
3. Medir el rendimiento de cada nodo atacante y la cantidad de paquetes maliciosos que se generan.

1.3 Marco teórico

1.3.1 Internet de las cosas (IoT)

Los orígenes de IoT se remontan a la Segunda Guerra Mundial, cuando se usaba el radar para identificar aviones enemigos que se acercaban, pero era imposible determinar si esos aviones eran aliados o enemigos. Así que los alemanes descubrieron que, si devolvían el avión a la base aérea, cambiaba la señal que el avión reflejaba de vuelta al sistema de radar. Así, en los años siguientes se siguió investigando en la tecnología RFID. y en agricultura, comercio, transporte e Incluso como una forma de abrir puertas. Produce especificaciones como UHF RFID utiliza altas frecuencias que dan como resultado un mayor ancho de banda y rango para equipo El futuro será la base de todo el Internet de las Cosas con otras tecnologías. [5]

Según [6], [7] CyberPhysical System (CPS) es un sistema informático y proceso físico. Consiste en una red y computadoras integradas para monitorear y Ellos controlan el proceso. CPS va más allá de la identificación y el control de dispositivos individuales a un nivel en el que la información se comparte entre dispositivos para lograr objetivos. con mayor eficiencia. Es una red de dispositivos que interactúan entre sí a través de la entrada. salida física en lugar de un dispositivo separado. [5] Individual, el nivel en el que la información se comparte entre dispositivos para lograr objetivos con mayor eficiencia. Es una red de dispositivos que interactúan entre sí a través de la entrada. salida física en lugar de un dispositivo separado [5]. Por otro lado, desde el punto de vista de la aplicación, IoT comienza con la lectura de datos Comunicación entre dispositivos conectados estáticamente en tarjetas RFID

Entonces, en este caso, IoT funcionará como CPS, pero con dispositivo Disponible por separado en Internet. En conclusión, IoT apunta a más Una amplia gama de objetos conectados, CPS está diseñado para coordinar objetos en la red para lograr un objetivo común. Una red inalámbrica de sensores (WSN) según [8] consiste en un conjunto de dispositivos Bajo consumo de energía para conectar uno o más sensores o actuadores entre sí o con un dispositivo central formando una red. WSN es principalmente Las características se utilizan para medir las características ambientales y transmitir datos. Obtenido de la estación base. En general, una WSN no necesita ser muy grande infraestructura.

Según [5], WSN es una red parcialmente distribuida de sensores autónomos, Supervisar las condiciones físicas o ambientales y comunicar datos de forma cooperativa De la red a la ubicación central. El objetivo principal de WSN es coordinar Recopilación de datos. Comparando la definición anterior, es obvio que su enfoque Lo principal es recopilar y centralizar datos de una red de nodos de bajo consumo. Entonces, la principal diferencia entre IoT y WSN es que En IoT, tienden a ser más inteligentes y pueden incluir sensores y actuadores Para lograr un objetivo determinado sin intervención humana [5]. Por otro lado, las redes inalámbricas de sensores carecen Dos características básicas de IoT, como son la conexión a Internet e Identificación única del dispositivo.

Similar a CPS, WSN puede usar IoT como su herramienta operaciones, pero no todas las WSN son necesariamente un sistema IoT. aceptar Dada la comparación con CPS y WSN, y una breve historia de IoT, ya existe una

Obtenga una comprensión más clara de lo que es IoT y sus principales características. Por lo tanto, a continuación, se analizarán las diferentes definiciones de existencia de IoT para establecer Sentó las bases para el desarrollo de este trabajo.

1.3.2 Protocolos LP-WAN y LR-WPAN

Los protocolos Low-Power Wide Area Network (LPWAN) y Low-Rate Wireless Personal Area Network (LR-WPAN) nacieron en base a las necesidades de comunicación de IoT para una gran cantidad de dispositivos de bajo costo con consumo de energía limitado y bajo consumo de energía Según [9]. La especificación IEEE 802.15.4-2020 surge de la necesidad de dispositivos de bajo consumo que no requieran altas velocidades de transferencia de datos. Está diseñado para dispositivos móviles o no móviles sin batería con un uso limitado de la batería. También, este protocolo IEEE describe las capas físicas (PHY) y de control de acceso al medio (MAC) que implementan el estándar. Se han realizado varias iteraciones del estándar para mejorar su seguridad, eficiencia y funcionalidad compatibles con los dispositivos. De esta forma, IEEE 802.15.4 sirve como base para la creación del protocolo de comunicación Low Rate Wireless Personal Area Network o LRWPAN. y LPWAN como lo indica [9] [10]

Los protocolos LR-WPAN y LPWAN se enfocan en conectar dispositivos a Bajo poder de procesamiento y poder limitado. La diferencia clave entre los dos Existe en el enfoque que tiene; donde LR-WPAN se enfoca en baja velocidad transmisión y corto alcance; LPWAN, por otro lado, se enfoca en tener el mayor rango posible con el menor consumo de energía, por lo que es común para varios protocolos Las LPWAN también tienen velocidades de transferencia más bajas porque se derivan de IEEE 802.15.4 [10] Porque estos protocolos son más parecidos que diferentes, y porque se ocupan de Para abordar problemas de comunicación similares, pueden incluirse bajo el término Protocolo IoT. Incluso el protocolo LR-WPAN permite una gran área Superposición a través de topologías de árbol o malla como ZigBee, dos de las tecnologías más utilizadas en el mundo del ioT son LoRa y LoRaWAN.

1.3.3 LoRa y LoRaWAN

LoRa (Long Range) es la capa física o modulación de radio para Establecer un enlace de comunicación remota. LoRa se basa en la tecnología de modulación Chirp Spread Spectrum (CSS), que se caracteriza por baja potencia y largo alcance. CSS tiene Utilizado durante décadas para comunicaciones militares y espaciales debido a las distancias alcanzadas y robustez contra interferencias, pero LoRa es la primera implementación de bajo costo en usar comercial. Es una tecnología LPWAN y, por lo tanto, proporciona años de duración de la batería (hasta 10 años), diseñado para sensores y aplicaciones que requieren poco transporte Transmita datos a largas distancias varias veces por hora en un entorno muy variable [11]

Por otro lado, LoRaWAN (LoRa Wide Area Network) define el protocolo de comunicación y arquitectura de red, mientras que LoRa admite enlaces remotos. Los protocolos y arquitecturas de red tienen el mayor impacto en la determinación de la capacidad y la calidad de la red. Servicios, seguridad, diversas aplicaciones para servicios web y duración de la batería nodo. Ayuda con el último punto, evitando la sincronización con la red durante una

sincronización de red. La "activación" del dispositivo es la clave para reducir el consumo de energía. equipo Las LoRaWAN son asincrónicas y transmiten cuando están listas para enviar datos, o según Eventos o planificados, es decir, siguen el método ALOHA.

Los dispositivos de LoRaWAN pueden ser de tres diferentes tipos de clases [11]

Clase A: Permiten la comunicación bidireccional, enviando cuando se necesita, aunque Solo se recibió dentro de dos breves períodos de tiempo después del envío. el momento El envío depende de las necesidades del equipo y de pequeñas variaciones aleatorias de tiempo Porque es un protocolo tipo ALOHA. La clase A debe ser Todo el equipamiento. Es el más eficiente energéticamente, pero produce el mayor Retardo aguas abajo.

Clase B: Se ajusta a las características de la Clase A, pero admite la configuración de ventanas de recepción adicionales. Esto su cede después de recibir una baliza de sincronización para determinar cuándo un dispositivo está escuchando.

Clase C: la ventana de recepción es continua y solo se corta cuando es necesario enviar.

1.3.3.1 Arquitectura de una red LoRaWAN

Una arquitectura de una red LoRaWAN esta compuesta por los siguientes elementos como se muestra en la Figura 2. Desde esta parte del documento por facilidad de redacción los dispositivos serán mencionados por su abreviatura.

- **End – Nodes o End – Device (ED)** Son los dispositivos IoT que se encuentran unidos a la red LoRaWAN, se comunican de forma bidireccional con los gateways.

- **Gateway (GW)** Un dispositivo con una antena para recibir mensajes LoRa desde el ED. Su única tarea es reenviar el mensaje LoRa completo a un servidor web a través de un enlace TCP/IP directo basado en tecnologías tradicionales como Ethernet. o 3G. A su vez, se encarga de transmitir los mensajes desde el servidor web al ED. Según las necesidades, pueden coexistir varios GW, lo que en realidad es un ED No se comunica con un GW específico, pero todos los GW capturan mensajes y Enviado al servidor web responsable de descartar los duplicados.

- **Network Server (NS)** es responsable de administrar las sesiones de LoRaWAN con todos los ED, validar los mensajes de los ED y reenviarlos al servidor de aplicaciones. A su vez, este último envía Los ajustes para la aplicación de referencia NS, ED y GW dependen de cómo se defina la red. Se pueden integrar varias instancias de NS en LoRaWAN. En resumen, es un enrutador Autenticación y Autorización de Mensajes LoRaWAN y ED de Control

- **Aplication Server (AS)** Es el destino de los datos provenientes de ED como payload de LoRaWAN. Tiende a implementarse por usuarios finales de acuerdo con el propósito de los usuarios consumidores.

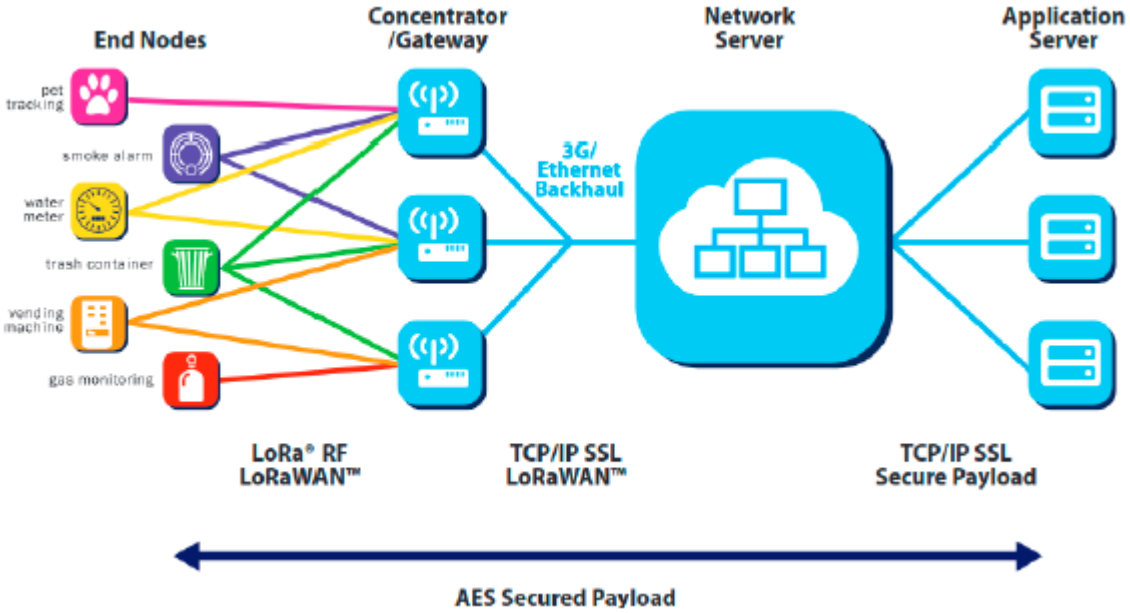


Figura 1. Arquitectura de LoRaWAN [11]

1.3.4 Seguridad en LoRaWAN

La falta de estándares viene con la falta de medidas de seguridad. La gran mayoría de los dispositivos se caracterizan por su simplicidad, razón por la cual sus mecanismos de protección suelen ser subóptimos y los fabricantes no recomiendan proteger los dispositivos antes de su implementación. Un problema común en IoT es que los dispositivos están expuestos a Internet y su configuración de fábrica (es decir, usan credenciales predeterminadas o no están presentes), lo que les permite ubicarlos en la gran cantidad de escaneos de motores de búsqueda como Shodan. [12] Si consigue controlar un número suficiente de dispositivos, el atacante Puede deshabilitarlos o usarlos para otros fines, como denegación de servicio distribuida (DDoS), minería de Bitcoin, robo y manipulación de datos, extorsión/extorsión, acoso o robo a terceros. [1]

Diseñado teniendo en cuenta la seguridad de los datos en LoRaWAN, la comunicación es obligatoria porque hay dos claves de sesión [13], llamadas Clave de sesión de red (NwkSKey) y Clave de sesión de aplicación (AppSKey), las cuales se almacenan en el ED. NwkSKey es la clave para firmar mensajes LoRa, conforme al control de integridad. Cuando el NS recibe el mensaje, verifica la integridad por última vez y, de ser así, extrae la carga útil de datos para enviarla al AS. los datos son Cifrado punto a punto usando AppSKey entre ED y AS Tenga en cuenta que el control de integridad termina en NS, por lo que cualquier entidad capaz de manipular la comunicación después de esto puede alterar el contenido de los datos en tránsito y su integridad se verifica en el nivel LoRaWAN (independientemente de corrección de la modificación).

También tenga en cuenta que NS técnicamente tiene los datos necesarios para exportar NwkSKey y AppSKey.

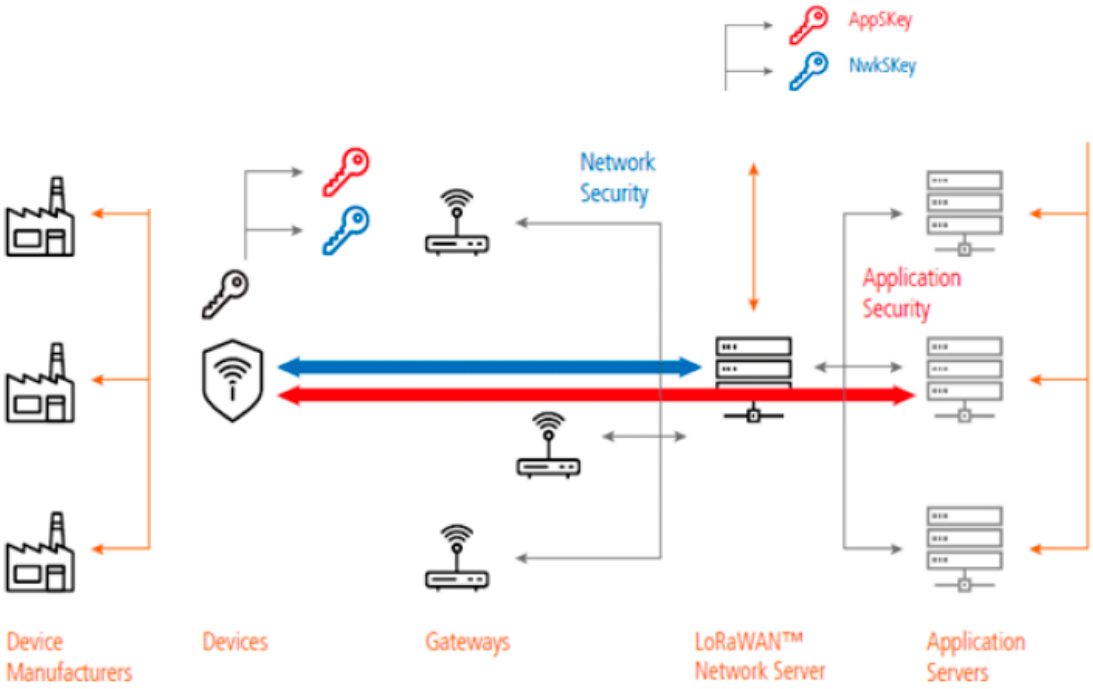


Figura 2. Esquema de seguridad en LoRaWAN

2. METODOLOGÍA

Para este proyecto se seleccionó la metodología LEAN, ya que los principios de esta han comprobado ser una útil herramienta a la hora del desarrollo de software [14], adaptación de librerías entre otras varias utilidades, esta metodología generalmente consiste en aplicar los principios: optimizar el todo, eliminar los desperdicios, construir calidad, aprender constantemente, entregas rápidas, involucrar a todos y mejorar contantemente.

Fase de Identificación:
En esta fase se identificará y se explora lo que es el mundo de IoT, en específico los protocolos LoRaWAN, conocer cuál es su funcionamiento, y las medidas de seguridad que tiene este protocolo con el fin de entender sus vulnerabilidades, y realizar una experimentación lanzado ataques hacia dispositivos virtuales, amplificando estos haciendo uso de contenedores y kubernetes.

Fase de Planificación y Diseño:
En esta fase se revisará el estado del arte concerniente a vulnerabilidades en redes LoRaWAN, contenedores y ataques tipo DoS.
Se propondrá una arquitectura con capacidades auto escalables que permita desplegar nodos atacantes bajo demanda.
Se revisará y utilizará una metodología de desarrollo ágil para la identificación de requisitos funcionales y no funcionales del prototipo.

Fase de implementación:
Se utilizará la metodología identificada (SCRUM) para el desarrollo de las historias de usuario identificadas.
Se identificarán los componentes de hardware y software requeridos para el nodo atacante optimizando el uso de recursos tecnológicos para poner a disponibilidad la mayor cantidad de recursos computaciones a la generación de paquetes maliciosos.
Se pondrá en marcha el componente y se afinarán los recursos tecnológicos a ser utilizados.

Fase de evaluación y análisis de resultados:
Esta fase comprende la documentación de los resultados obtenidos durante las pruebas que se realizarán para medir consumo de memoria RAM, CPU, tráfico E/S, operaciones de disco de E/S en cada uno de los nodos atacantes. De la misma manera se afinará el prototipo para determinar las características mínimas que garanticen su funcionamiento. Adicionalmente, se propondrán escenarios ideales y se medirá el comportamiento del componente generador de paquetes maliciosos, la intención de estos escenarios será medir la eficacia del componente al hacer que los recursos crezcan de manera automática sin la intervención manual de un tercero. Por otro lado, también en esta fase se medirá el tiempo que toma desplegar nuevos nodos.

Para cumplir con el propósito de este proyecto de integración curricular se trabajará bajo la siguiente metodología:

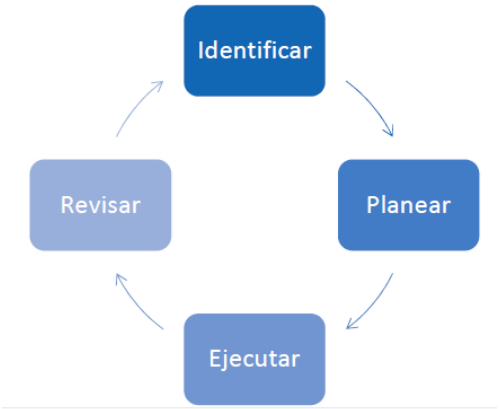


figura 3. Metodología de trabajo

Como se muestra arriba, la metodología utilizada en este proyecto se basa en Lean Software Development y el marco SCRUM para el desarrollo de software. Puedes ver cada paso de la metodología Lean a través de las tareas específicas del proyecto. En el paso de planificación y ejecución, puede ver la aplicación del marco Scrum mediante la creación de un trabajo pendiente del proyecto durante la planificación y el trabajo pendiente del sprint, la ejecución del sprint y la revisión del sprint en el paso de ejecución como se aprecia en la Figura 4.

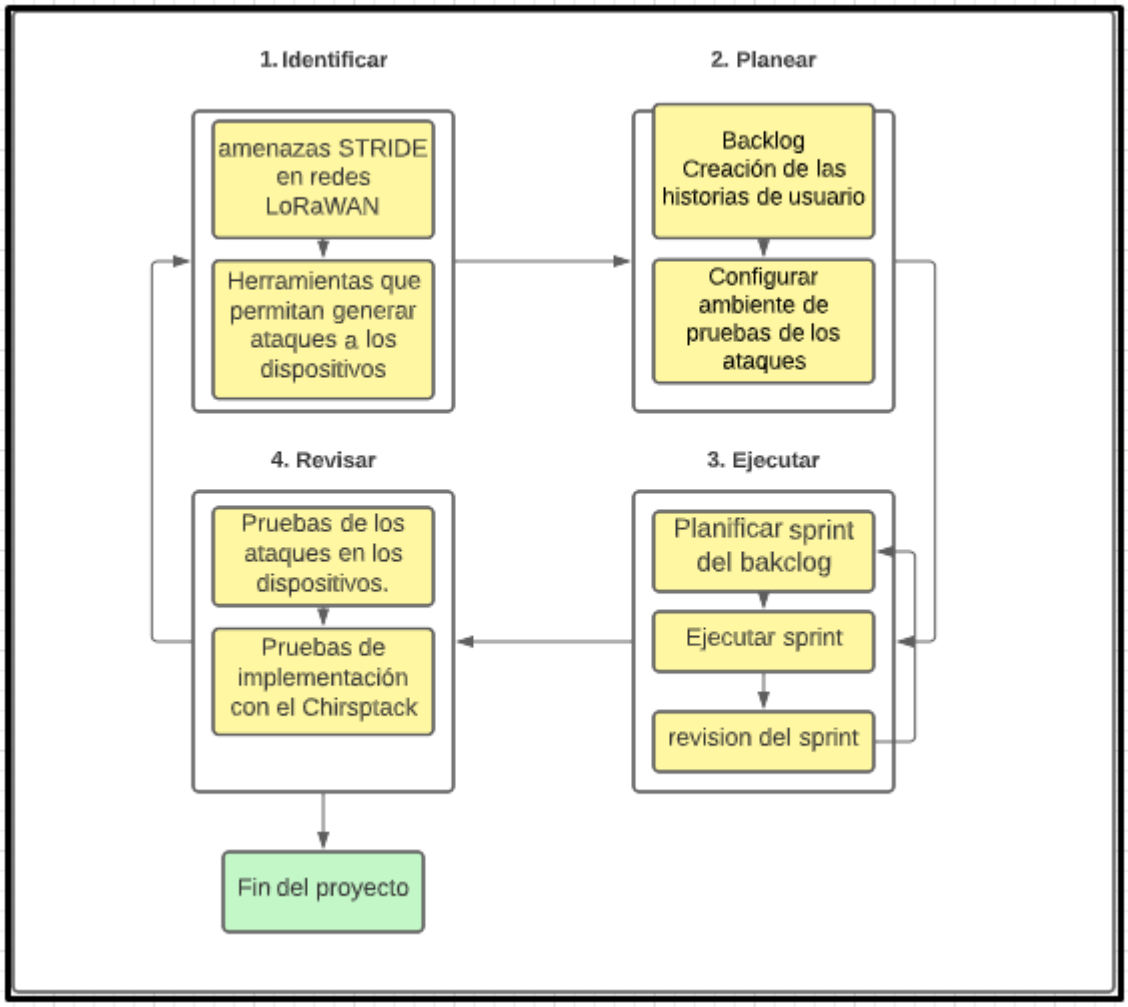


Figura 4. Metodología aplicada al proyecto

2.1 Identificar

2.1.1 Amenazas STRIDE en redes LoRaWAN

OWASP recomienda identificar amenazas de acuerdo con un modelo estándar, teniendo en cuenta posibles actores de amenazas tiene descripciones de los actores, sus objetivos habituales, probabilidades se espera que se encuentren en los escenarios de LoRaWAN del mundo real que se están analizando, y se los priorice en el modelado de amenazas.

Para este modelo se optó por seguir STRIDE [15], que sugería preguntarse cómo acceder a determinados recursos o entorpecer el proceso y qué pasaría si se implementara. Su sigla se puede utilizar para identificar amenazas desde el principio, o para clasificarlas una vez adquiridas. Las letras de STRIDE representan diferentes propiedades y significados de seguridad y tiene los siguientes significados Spoofing (autenticidad), Tampering (integridad), Repudiation (no repudio), Information disclosure (confidencialidad), Denial of Service (disponibilidad) y Elevation of privileges (autorización).

ID	Nombre
Autenticidad	
T01	Atacante autenticado en ED como personal de mantenimiento.
T02	ED falso
T03	GW falso
T04	Atacante autenticado en GW como personal de mantenimiento.
Integridad	
T05	Alterar comunicación entre EDs y GWs
T06	Mantenimiento local altera mediciones, configuración o firmware del ED
T07	Mantenimiento local altera configuración o firmware de GW
No repudio	
T08	Modificación de firmware anónima en ED
T09	ED falso suplantando la identidad de un ED legítimo
T10	ED niega haber recibido/enviado un mensaje
T11	Modificación de software/firmware anónimo en GW
Confidencialidad	
T12	ED filtra claves de sesión
T13	ED filtra sus parámetros de configuración
T14	ED filtra mediciones
T15	Escucha de comunicación entre ED y GW
T16	Escucha de comunicación entre GW y NS
Disponibilidad	
T17	Abuso de medida para bloquear/ignorar medidas legítimas
T18	ED no puede comunicarse
T19	ED deja de cumplir funciones que afectan al ambiente
T20	Agotamiento de batería del ED
T21	GW no puede comunicarse
T22	Sumidero de mensajes desde el GW (sinkhole)
Autorización	
T23	Acceso remoto a ED
T24	Acceso remoto a GW

Tabla 1. Listado de amenazas siguiendo el modelo STRIDE

Algo importante que destacar sobre la exhaustividad de la lista es que el alcance se reduce a amenazas que involucran ED y GW porque son poco conocidos se sabe de sí mismo y de sus intercambios, y se cree que ha aportado más al campo de la investigación. Sin embargo, vale la pena señalar que NS y AS también son

importantes para asegurar LoRaWAN (especialmente NS). Si se considera al GW como un reenviador de mensajes, el NS puede verse como el siguiente punto de comunicación con el ED, por lo que los problemas de seguridad que lo involucran se incluyen inevitablemente en el trabajo.

2.1.1.1 Autenticidad

Abusar de la autenticidad es la posibilidad de que un atacante pueda autenticarse en el ED como si fueran mantenedores (T01 y T04 similares a GW), posiblemente requiriendo acceso físico al dispositivo, aunque podría ser remoto. Los ED falsos (T02) también se pueden usar para reemplazar parcial o completamente los ED legítimos en la red en lugar de agregar ED maliciosos, ya que esto requiere obtener parámetros (AppEUI y AppKey en OTAA y clave de sesión en ABP) para la autenticación mutua. Dicho esto, para para lograr esto, hay que controlar NS y AS (más complicado, más allá de lo ya comentado). Para la amenaza T03, sucede algo similar, porque para reemplazar la identidad del GW, necesita: en el lado ED, conocer su clave de sesión o abusar del proceso de unión, y poder enviar un JA malicioso que conoce otros parámetros (todos solo porque solo un GW, ya que cancelar múltiples señales complica el ataque).

2.1.1.2 Integridad

Para afectar la integridad, la amenaza más sofisticada pero poderosa es comprometer el canal ED-GW para alterar el mensaje (T05). Los requisitos varían (acceso físico, acceso a claves de sesión, destrucción de hardware, etc.). Esta amenaza está asociada con T02 y T03. Luego están T06 y T07, diseñados para cambiar datos, configuración o firmware con acceso de mantenimiento local.

2.1.1.3 No repudio

De las amenazas de no repudio, todas se basan en la falta de registros que permitan identificar a un usuario o dispositivo. El T08 y el T11 son similares en el sentido de que ambos han completado los cambios de firmware y no pueden asociar ninguna identidad con este proceso. El T09 es similar al T02, pero desde una perspectiva innegable. T10 ocurre cuando el ED se niega a recibir o enviar un mensaje. En la recepción, hay un mecanismo de acuse de recibo (ACK), pero el ED no está obligado a hacerlo cumplir, por lo que no hay forma de verificar que el mensaje ha llegado si el ED no mantiene un registro del mensaje (que por lo general no lo hace, ya que esto requeriría demasiados recursos) o usa un ACK ed. Cuando se envía, no hay más evidencia que la firma en el mensaje, que debe tener la clave NwkSKey única del ED. Debería ser suficiente evidencia, pero hay formas de obtener esa clave, según la configuración en NS.

2.1.1.4 Confidencialidad

Las amenazas T12-T14 son filtraciones en el ED, posiblemente a través de una interfaz (p. ej., serie, USB u otra conexión) que expone claves de sesión, AppKeys, ajustes de configuración relacionados o acciones que se están realizando. Hay otras formas de almacenar estos valores en la memoria desprotegida para que un atacante pueda leerlos. Luego, escuchar los canales en T15 es algo que cualquiera puede hacer en el espectro LoRa, pero los canales codificados no se pueden leer sin una clave de sesión (excepto JR)

que se puede obtener como se describe anteriormente. Lo mismo sucede en T16, aunque para poder escuchar el canal, primero se debe romper el enlace TCP/IP.

2.1.1.5 Disponibilidad

La gravedad de la denegación de servicio varía con la práctica, pero en IoT y más allá de los entornos en frecuencias de radio, puede ser relativamente simple de lograr. En las vías respiratorias, afectan a las técnicas de interferencia, por ejemplo, emitiendo ruidos que anulan los mensajes. Combinado con cualquier otra tecnología que interrumpa la transmisión/recepción, da lugar a amenazas T18 y T21. Luego está la amenaza relacionada con la seguridad física y el suministro de energía (T20, posiblemente junto con T18 y T21). Recuerde que el ED se analiza en la implementación de medidas y entornos que pueden verse físicamente afectados para que no se realicen medidas legales (por ejemplo, con un ruido muy alto en relación con la cinta del medidor negativo). Si ED también es un efecto ambiental (por ejemplo, luz), puede verse afectado para que su función precisa no funcione (extinguir las luces o la emisión continuamente) en T19. Finalmente, si GW está comprometida, con ataques como la modificación de la tabla de enrutamiento a NS, los mensajes no se enviarán a ninguna parte (T22).

2.1.1.6 Autorización

Para dispositivos como ED o GW, no es necesario tener derecho al usuario cada vez, se vuelve común distinguir entre la amenaza de la escala de franquicia. Pero dado que era una licencia, la gente decidió alcanzar una dimensión a ED y GW para incluirlos en esta lista (T23 y T24).

2.1.2 Herramientas que permiten generar los ataques hacia los dispositivos.

Para este proyecto se utilizará una herramienta de auditoría y monitoreo para las redes LoRaWAN, esta herramienta va a permitir generar ataques dentro de la infraestructura, que se va a implementar mas adelante en el proyecto.

Para ello es necesario conocer los orígenes de esta herramienta y cuales son los usos que se le pueden dar, esta herramienta ha sido implementada por los investigadores de IOActive [16].

Debido a que el protocolo LoRaWAN usa encriptación y se anuncia como un protocolo seguro, los usuarios y desarrolladores lo adoptaron rápidamente y se espera que aumente su popularidad porque también ofrece otros beneficios, como un menor costo y una fácil instalación y mantenimiento. Sin embargo, a través de su nuevo artículo, los investigadores de IOActive quieren resaltar que muchas de estas redes están expuestas a riesgos de seguridad y deben ser auditadas y monitoreadas en busca de debilidades y ataques.

Afortunadamente, algunos ataques dejan rastros y el marco de auditoría LoRaWAN (LAF) de código abierto de IOActive se puede utilizar para descubrir los compromisos existentes. No ayudará a bloquear nuevos ataques, pero puede servir como una herramienta de detección pasiva. Por ejemplo, se puede usar para configurar verificaciones de mensajes duplicados o contadores de mensajes que son

más bajos de lo esperado, lo que podría ser un signo de suplantación de identidad del dispositivo.

Se debe evitar el uso de dispositivos con claves de sesión codificadas porque corren un mayor riesgo de verse comprometidos. Estos se conocen como dispositivos de activación por personalización (ABP) y LAF se puede usar para descubrirlos y marcarlos para su reemplazo. El marco también se puede usar para descubrir claves débiles para que puedan regenerarse y reemplazarse. El documento de IOActive incluye recomendaciones sobre cómo proteger las claves, incluido el uso de dispositivos con elementos seguros de hardware (SE) y servidores con módulos de seguridad de hardware (HSM).

"El mejor enfoque para prevenir ataques es holístico, donde se asegura el ecosistema LoRaWAN completo", dijeron los investigadores. “Esto solo se puede lograr si toda la tecnología que forma parte del ecosistema (dispositivos, puertas de enlace, servidores de red, servidores de unión, servidores de aplicaciones y aplicaciones) se audita adecuadamente en materia de seguridad. De esta manera, se identifican y solucionan los posibles problemas de seguridad. Esto debe hacerse al menos dos veces al año, ya que el ecosistema no es estático. Las redes LoRaWAN son muy dinámicas y se agregan nuevos componentes regularmente". [16]

Las implementaciones de IoT siguen creciendo y una parte de ese crecimiento significativo se compone de millones de sensores LPWAN (red de área amplia de baja potencia) implementados en cientos de ciudades (ciudades inteligentes) en todo el mundo, también en industrias y hogares. Una de las tecnologías LPWAN más utilizadas es LoRa, para la cual LoRaWAN es el estándar de red (capa MAC). LoRaWAN es un protocolo seguro con cifrado incorporado, pero los problemas de implementación y las debilidades afectan la seguridad de la mayoría de las implementaciones actuales. [17]

2.2 Planear

2.2.1 Backlog Creación de las historias de usuario

Para proceder con la creación del backlog del proyecto, se describen los pasos a realizar, para la implementación de la infraestructura en la que se van a realizar las pruebas de los ataques, esto se realiza con el fin de que los requerimientos se entiendan de manera clara y posteriormente se puedan transformar en ítems realizables para el backlog del proyecto.

Se toma al atacante como actor del sistema sobre el cual se elaborarán las historias de usuarios que se presentan en la tabla 2.

Como	Deseo	Con el fin
Usuario	Implementar la infraestructura de LoRaWAN (NS-AS-GW) en un ambiente virtual.	Realizar la emulación del funcionamiento y realizar los ataques de manera segura.
Usuario	Implementar un clúster de kubernetes con varios nodos.	Amplificar los ataques hacia la infraestructura LoRaWAN.
Usuario	Implementar un script de generación de paquetes maliciosos.	Ejecutar ataques del tipo DoS contra la infraestructura LoRaWAN.

Atacante	Ejecutar ataques hacia un dispositivo virtual en la arquitectura ya implementada.	De verificar el comportamiento de los recursos de hardware.
Atacante	Identificar los dispositivos que fueron atacados	De verificar que fue lo que cambio dentro de los mensajes que estos reciben.

Tabla 2. Historias de usuario

Para entender las historias de usuario que se mencionan en la tabla 2, que antecede a esta sección, es importante considerar la arquitectura virtual del proyecto que se está utilizando para realizar este ambiente de pruebas.

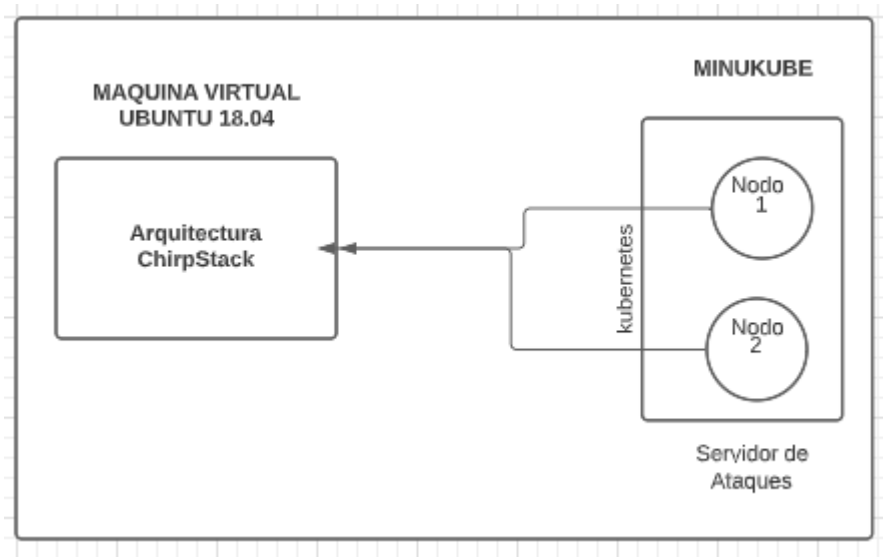


Figura 5. Arquitectura Macro del Proyecto

Como se observa, en la figura 5 es una arquitectura macro de como esta implementado el proyecto, al lado derecho se puede observar la arquitectura ChirpStack, esta se detalla a continuación y se la muestra a detalle en la figura 6.

En el lado izquierdo se muestra un servidor virtual minikube, este está implementado de manera virtual, y contiene un kubernete en el que se encuentra un contenedor, que contiene la infraestructura de la herramienta que permitirá generar los ataques hacia los dispositivos o ED.

La arquitectura del minikube, se muestra a detalle en la figura 7. Donde se entenderá como se implementó la infraestructura, que nos va a permitir realizar los ataques.

2.2.2 Configurar ambiente de pruebas

Para configurar el ambiente de pruebas para el desarrollo de este proyecto se utilizará el ambiente virtual proporcionado por ChirpStack, la implementación se realizará en una maquina con las siguientes características, procesador Intel core i3-7020u, Memoria 8GB RAM, sistema operativo Ubuntu 22.04, dentro de esta maquina se

instala una maquia virtual Ubuntu 18.04 con las siguientes características memoria RAM 4GB virtualizada.

En la máquina virtual, se implementará únicamente la arquitectura de red LoRaWAN, que constará de la siguiente arquitectura como se muestra en figura 5.

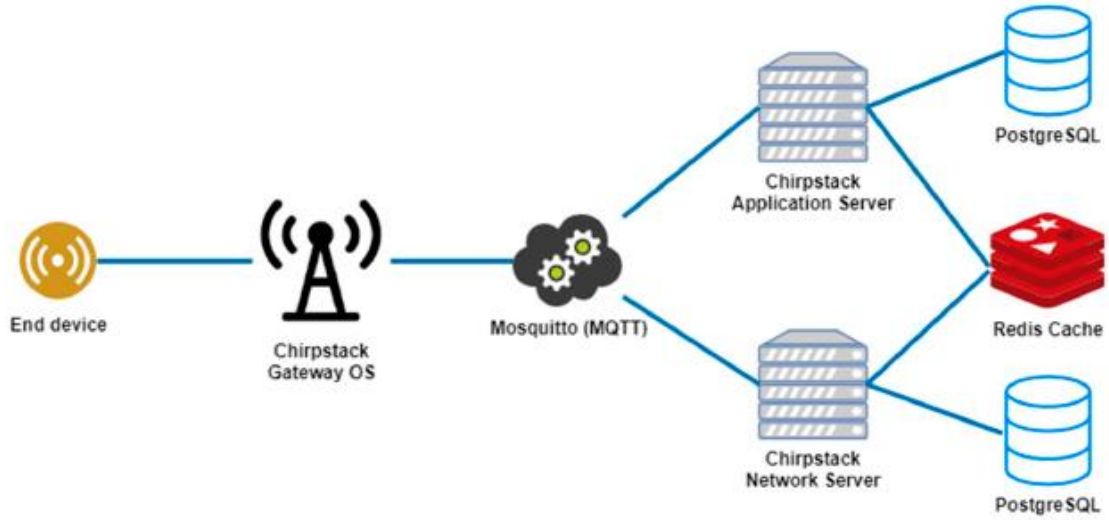


Figura 6. Arquitectura ChirpStack

Como se muestra en la figura, la arquitectura de ChirpStack cuenta con tres componentes fundamentales para la emulación de una red LoRaWAN.

2.2.2.1 ChirpStack Apllication Server

Es una implementación de código abierto del servidor de aplicaciones LoRaWAN [18]. Sin embargo, incluye la aplicación ChirpStack dentro del servidor de aplicaciones, el repositorio del terminal y la gestión del proceso de unión; Características de la unión al servidor LoRaWAN. Se puede decir que, sin configurar el servidor de enlace, el servidor de aplicaciones ChirpStack actúa como un servidor de aplicaciones y un servidor de serialización al mismo tiempo.

2.2.2.2 ChirpStack Network Server

Es una implementación de código abierto del Network Server de LoRaWAN [19], se comunica con los GW y el AS para brindar servicio a los dispositivos

La responsabilidad del componente del servidor de red es la de duplicación de las tramas LoRaWAN recibidas por las puertas de enlace LoRa® y las tramas recopiladas se encargan de manejar:

- Autenticación.
- LoRaWAN capa mac y comandos mac.
- La comunicación con el AS
- Programación de tramas de enlace descendente.
- Responder a los comandos MAC del dispositivo

Este puede ser utilizado para diferentes aspectos y esto va a depender del ED.

Puede ser utilizado como un servidor, esto va a depender si se encuentra en roaming o no, si se encuentra en roaming controla la capa MAC del ED.

Puede ser utilizado como un alojador o HOME, y aquí almacenara el perfil del dispositivo, el perfil del servicio, el perfil de rutas, este va a ser implícitamente el que se utilizara en este proyecto.

2.2.2.3 ChirpStack Gateway OS

ChirpStack Gateway OS es un sistema operativo integrado basado en Linux que se puede instalar en múltiples puertas de enlace LoRa en un solo módulo LoRaWAN. Gateway OS se puede instalar de dos maneras: Versión básica: Gateway Bridge y Concentratord Versión completa: Gateway Bridge, Concentratord, Mosquitto Server, Application Server, Network Server, PostgreSQL, Configured Redis. ChirpStack Concentratord es un demonio LoRaWAN de código abierto cuya función es interactuar con el hardware de la puerta de enlace. ChirpStack Gateway Bridge [20] es un servicio responsable de convertir los protocolos LoRa Packet Forwarder en un "formato de datos común" para un servidor de red ChirpStack. En otras palabras, la puerta de enlace se comunica con el servidor de la red convirtiendo los mensajes a un formato que el servidor de la red pueda entender.

2.2.2.4 Otros componentes

Se requieren otros tres componentes para que ChirpStack funcione. Una base de datos PostgreSQL para el servidor de red ChirpStack y otra para el servidor de aplicaciones ChirpStack. Estas bases de datos son responsables de almacenar datos impopulares, como la composición periférica, el bloqueo y la EUI. ChirpStack también requiere que el mosquito neto (MQTT) se conecte al puente, el servidor de red y el servidor de aplicaciones. Finalmente, uso Redis; Infraestructura en el almacenamiento de la memoria para almacenar datos volátiles a los que necesita acceder muy rápidamente.

2.2.2.5 Fase de implementación

Para la fase de implementación se establece la configuración del ambiente, como se mencionó anteriormente las características del ordenador en el que se va a trabajar, se procede a realizar la configuración de este.

2.2.2.5.1 Implementación de requisitos previos.

Antes de comenzar con la implementación, es importante seguir la guía de instalación [21] de los requerimientos para que la infraestructura de ChirpStack funcione correctamente**,** en este proceso se va a instalar:

Mosquitto que es un popular servidor MQTT de código abierto, pero cualquier corredor de MQTT que implemente MQTT 3.1.1 sirve para publicar y recibir cargas útiles de aplicaciones [21].

PostgreSQL para el almacenamiento de datos persistente. Tenga en cuenta que se requiere PostgreSQL 9.5+ y que cada componente requiere su propia base de datos para evitar conflictos de esquema [21]

Redis ya que los componentes de ChirpStack almacenan todos los datos no persistentes hay que tener en cuenta que se requiere al menos

Redis 5.0.0. Redis es utilizado por ChirpStack Network Server y ChirpStack Aplication Server [21].

2.2.2.5.2 Implementación del GW

Una vez se han instalado por completo los requerimientos previos para el correcto funcionamiento del ambiente previamente mencionado se proceden con la instalación del GW (Gateway Bridge), para esto seguimos al pie de la letra la guía que nos proporciona ChirpStack [22].

Una vez que se ha implementado el GW, se debe verificar que este servicio se encuentre habilitado y corriendo como se muestra en el anexo1 en la sección de anexos.

2.2.2.5.3 Implementación del NS

Para proceder con la implementación del NS, seguimos la guía [19], es importante mencionar que se sigue la guía de instalación sobre Debian/Ubuntu 18.04 [23], sobre la máquina virtual instalada, se siguen los comandos de instalación.

Una vez que se ha implementado el NS, se debe verificar que este servicio se encuentre habilitado y corriendo como se muestra en el anexo2 en la sección de anexos.

2.2.2.5.4 Implementación del AS

Para la instalación del AS, se debe seguir las instrucciones proporcionadas por ChirpStack, [24] el AS nos proporciona una interfaz web, que se habilita en el puerto 8080 que nos ayuda a gestionar de manera cómoda tantos los ED, como los NS y los GW.

Una vez que se ha implementado el NS, se debe verificar que este servicio se encuentre habilitado y corriendo como se muestra en el anexo3 en la sección de anexos

2.2.2.5.5 Implementación nodos – kubernetes

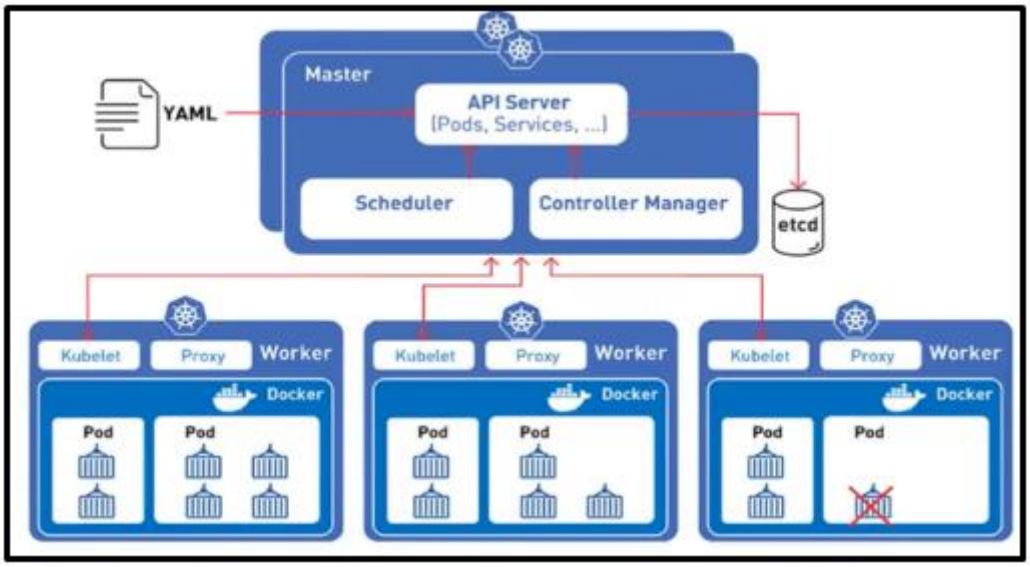


Figura 6. Arquitectura Kubernetes [25].

Una vez implementada la infraestructura de ChirpStack, el siguiente paso a seguir, es implementar los nodos atacantes, desde donde se van a enviar los paquetes maliciosos. En esta sección se implementará lo que es una infraestructura de kubernetes, en donde se van a almacenar los nodos atacantes, y de este nodo se van a crear varias replicas que van a aumentar la capacidad de ataques hacia los dispositivos.

Para este proyecto se seguirán los pasos de la guía de instalación de IOACtive / laf [17], en esta guía encontrará dos métodos de implementación, una para realizar la instalación de la herramienta de manera local en una maquina / servidor físico, y dos que es una implementación en un contenedor en Docker.
Para este proyecto se hace uso del segundo método, ya que la herramienta de kubernetes, los nodos son instancias de contenedores de Docker.

Como primer paso a realizar se realiza la instalación de minikube [26] en la maquina en la cual se van a alojar los nodos, para ello seguimos la guía de instalación de kubernetes en ambiente Linux para este caso Ubuntu 22.04, una vez instalado se inicia el ambiente y se muestra una consola como se muestra en el Anexo4, en la sección de Anexos.

Una vez se ha implementado el ambiente de minikube y el entorno Kubectl, se implementa la herramienta la herramienta laf usando Docker como contenedores siguiendo la guía [17].

Una vez implementado ya el nodo y colocado en kubernetes, se debe ingresar por consola y se ingresa el comando que se muestra en el Anexo5, en la sección de Anexos.

Una vez se ha implementado todo el ambiente tanto de LoRaWAN (ChirpStack) y kubernetes (minikube – kubectl), se verifican los puertos habilitados de cada servicio, como se muestra en el Anexo6, en la

sección de Anexos. Para esta visualización se utiliza el comando **“sudo lsof -i -P -n”**

2.3 Ejecutar

Juntamente con el backlog del proyecto, y el ambiente de pruebas una vez implementado en la fase de planeación se puede continuar con el proyecto, es decir comenzar a realizar las pruebas.

Es importante mencionar que para la ejecución del sprint se van a realizar cambios en los mensajes que enviamos en la herramienta que genera los ataques hacia los ED, para esto es importante conocer la estructura de estos y cómo funciona el script que esta generando los mensajes.

Para realizar la ejecución de los ataques, es necesario entender la estructura de los paquetes que reviven los dispositivos LoRaWAN, para este proyecto en específico se utilizara mensajes ascendentes, que cuentan con la estructura que se muestra en la figura 7.

Campos de mensaje de enlace ascendente de LoRaWan

Parámetro	Descripción	Tipo	Obligatorio
`WirelessDeviceID`	ID del dispositivo inalámbrico que envía los datos.	Cadena	Sí
`PayloadData`	Mensaje binario recibido del dispositivo, codificado en base64.	Cadena	Sí
`WirelessMetadata`	Metadatos sobre el dispositivo LoraWAN y la solicitud de mensaje. Esto incluye información como los identificadores de dispositivo, la velocidad de datos y código, la marca de hora del mensaje, si ADR (velocidad de datos adaptativa) está habilitado y los metadatos de la puerta de enlace.	Enumeración	No

Figura 7. Estructura de mensajes ascendentes en LoRaWAN [27]

A continuación, se muestra un ejemplo de cómo se estructuran los mensajes para una mejor comprensión de lo que significa estas tres partes fundamentales de los mensajes en las redes LoRaWAN, se puede excluir la informacion de metadatos de la puerta de enlace, esto se realiza deshabilitando la AddGwMetadata.

```
{
    "WirelessDeviceId": "0d9a439b-e77a-4573-a791-49d5c0f4db95",
    "PayloadData": "AAAAAAAA//8=",
    "WirelessMetadata": {
        "LoRaWAN": {
            "ClassB": false,
            "CodeRate": "4/5",
            "DataRate": "1",
            "DevAddr": "01920f27",
            "DevEui": "ffffff10000163b0",
            "FCnt": 1,
            "FPort": 5,
            "Timestamp": "2021-04-29T05:19:43.646Z"
    }
  }
}
```


Figura 8. Ejemplo de la estructura de un mensaje ascendente en LoRaWAN [27].

Una vez que se tiene entendida la estructura de los mensajes en las redes LoRaWAN, se procede hacer uso de la herramienta LAF [17], en esta herramienta se debe cambiar algunos parámetros en los mensajes.

Estos parámetros en específico son la dirección del dispositivo y se va a colocar la dirección del dispositivo creado en la infraestructura ChirpStack, en el Anexo 7 se encuentran los cambios realizados sobre el mensaje de ataque.

Una vez que se ha realizado un preámbulo de lo que se va a revisar en esta sección, se procede a planificar el sprint, en la planificación se observa un proceso a seguir para la ejecución del backlog visto en la sección de planear.

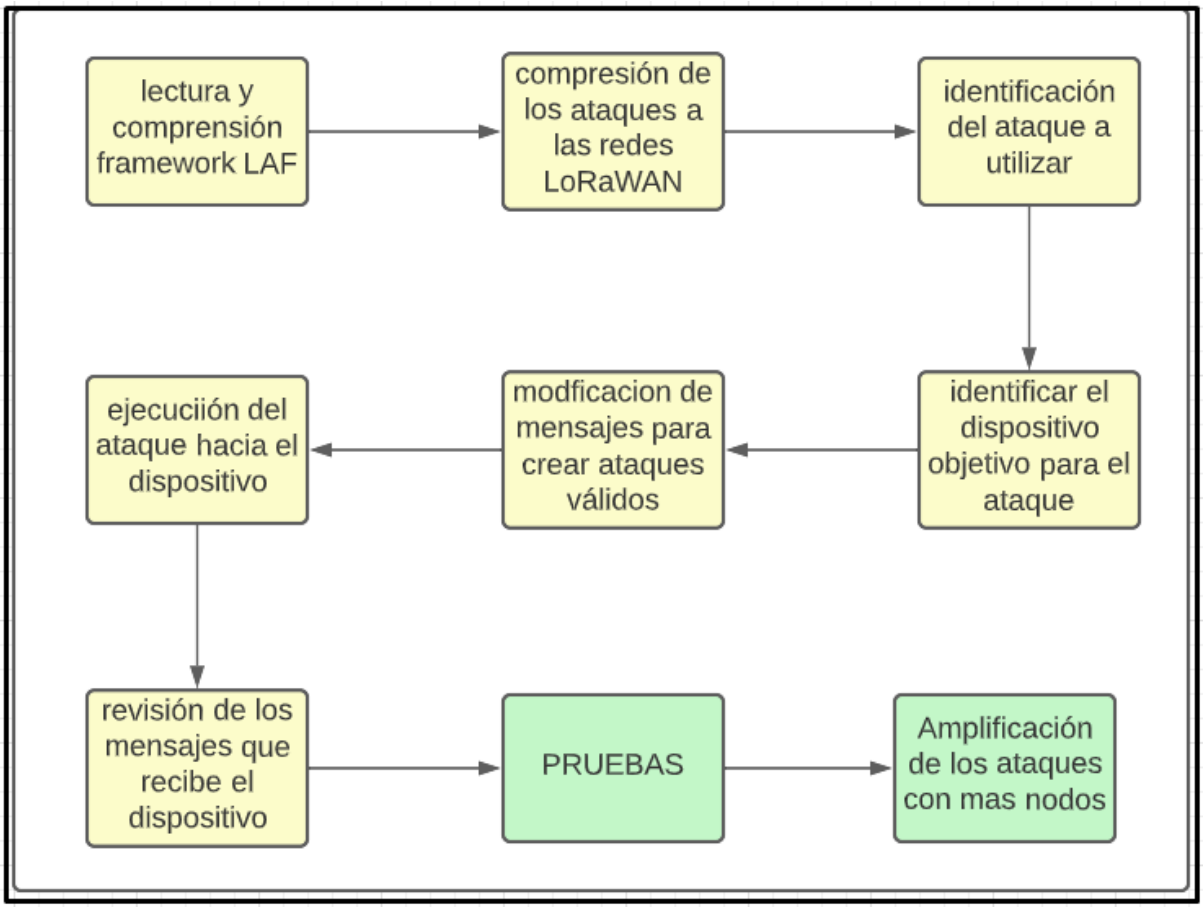


Figura 9: Ejecución de los ataques hacia los dispositivos

Cuando se tiene por completo entendido el framework LAF [17], y las bases de este, se realiza la instalación de la librería, se dirige a la sección de herramientas (tools), en donde se encuentran 4 archivos (UdpSender.py, UdpProxy.py, TcpProxy.py, GateviceSender.py).

El mensaje en específico que se utiliza para este proyecto es el UdpSender.py [17], este script está diseñado para enviar paquetes de enlace ascendente hacia el NS o GW, opcionalmente puede calcular MIC válidos.

Se tiene comprendido y seleccionado el ataque a utilizar y se continua con la siguiente fase del sprint que se trata de identificar cual es el dispositivo que se va atacar para esto tenemos la dirección del dispositivo, esta dirección se encuentra en hexadecimal de 8 bits, en específico la dirección del dispositivo a utilizar es la que se muestra en la tabla3.

DevAdr	AB CD 01 24

Tabla 3: dirección del dispositivo objetivo.

Para realizar la modificación del mensaje, utilizamos una calculadora de base64 a hexadecimal [28], esta herramienta lo que nos permitirá realizar es convertir un texto a base 64 y se podrán visualizar los bytes, que se están generando.

Con esta herramienta se verifica que los mensajes que se estén enviando se dirijan hacia el dispositivo objetivo, algo más que se va a modificar es el mensaje que se está enviando para verificar que los paquetes estén llegando hacia el dispositivo.

Esta verificación se la realiza con la herramienta de wireshark, en donde se puede captar el tráfico de la red, y con ello verificar que los paquetes lleguen hacia el dispositivo.
Cuando se tiene modificado el mensaje por completo como se muestra en el Anexo 8, en la sección de anexos. Se ejecuta el ataque hacia el dispositivo, el fin de ejecutar este script es provocar una caída del dispositivo o forzar que este se cuelgue provocando un ataque de DOS.

Ingresamos al AS, para verificar el estado del dispositivo y comprobamos que los paquetes hayan llegado al dispositivo objetivo, con esto nos encontramos ya en la sección de pruebas, si el mensaje llego correctamente.

Continuamos a la última fase del sprint que se trata de amplificar los ataques hacia el dispositivo objetivo, para realizar esto hacemos uso de la infraestructura de kubernetes, se tiene ya un nodo con el que se esta ejecutando los ataques, para proceder crear réplicas de este nodo utilizamos el comando: **kubectl scale –replicas=7 ataques-pod2.**

La estructura del comando se muestra en la tabla 4

Kubectl	**Escalar al pod**	**Numero de replicas**	**Nombre del pod**
Kubectl	Scale	--replicas = 7	Ataques-pod 2

Tabla 4: estructura del mensaje para amplificar los ataques

2.4 Revisar

En esta sección del documento se pretende hacer un análisis a los ataques que se realizaron en la sección anterior, es decir verificar si el dispositivo objetivo recibió el ataque por parte del nodo atacante.

Adicional a esto se debe verificar si el ataque se amplifico, con las réplicas implementadas del nodo.

Nos podemos basar en un checklist de pruebas de acuerdo con los ítems de el sprint que se planteó en la sección, de ejecutar basándonos en la tabla 5.

Ítem del sprint	**Cumple / No cumple**
Comprensión del framework LAF	
Comprensión de los ataques a redes LoRaWAN	
Identificación del ataque a utilizar	
Identificación del dispositivo objetivo	
Modificación del mensaje (Mensaje Válido)	
Ejecución del ataque hacia el dispositivo	
Revisión de los mensajes des dispositivo objetivo	
Pruebas	
Amplificación del ataque con 7 replicas del nodo	

Tabla 5: checklist de verificación del sprint.

Cada uno de los elementos que se observan en el checklist, permite que se verifique si el proyecto cumplió con el alcance del proyecto propuesto.

3. Resultados y discusión

En esta sección del presente documento se presentan los resultados de las pruebas planteadas en la sección (2.4), el objetivo de esta es poder revisar, validar y verificar que los ataques realizados hacia l infraestructura LoRaWAN, una parte que será de gran importancia son los requisitos que se han planteado en la sección de identificación (2.1).

Luego de realizar las pruebas planteadas en la sección 2.4, se obtienen resultados exitosos sobre la infraestructura virtual implementada en el presente proyecto.

Adicional se pudo observar el comportamiento del dispositivo al recibir mensajes, que ocasionan un comportamiento anómalo en el mismo, en caso de no tener ningún comportamiento anómalo es decir que no se está mandando la suficiente cantidad de paquetes, hacia el dispositivo.

En esta sección se discutirá sobre los diferentes resultados que se obtuvieron una vez ejecutado el script UdpSender.py.

Se toma como guía la tabla 5 vista en la sección (2.4) Revisar, se parte desde el primer ítem y se va a detallar si el proyecto cumple o no cumple con cada ítem propuesto.

3.1 Comprensión del Framework LAF

Una vez se ha revisado la documentación del framework LAF [17], se conoce con qué fin fue desarrollado, y con qué tipo de herramientas cuenta este con ello se llega a determinar qué tipo de ataque es el que se puede utilizar para cumplir con el propósito del proyecto.

Como se ha mencionado en secciones anteriores el ataque del cual se hace uso es el UdpSender.py. Este tipo de ataque lo que realiza específicamente es mandar un conjunto de mensajes de forma ascendente, hacia el dispositivo objetivo este mensaje cuenta con parámetros que son obligatorios y parámetros que pueden ser enviados opcionalmente.

Los parámetros que se envían en este proyecto se detallan a continuación:

IP de destino = La dirección IP del NS (infraestructura ChirpStack) donde este alojado el dispositivo objetivo.

Puerto de destino = El puerto habilitado el NS.

Datos para enviar = Se trata de un paquete UDP en este también se pueden agregar más paquetes en matriz de "datos" al final de este script. El paquete debe ser una cadena de bytes

Dirección del dispositivo: Dirección del dispositivo objetivo, esta debe tener un formato hexadecimal.

3.2 Comprensión de los ataques a las redes LoRaWAN

Como se le leyó en la documentación del framework laf se comprende con que objetivo se desarrollo [17] , un preámbulo de esto se da debido a que por el aumento de dispositivos IoT, y millones de estos sensores están compuestos por el protocolo LPWAN, y una de las tecnologías más utilizadas es LoRaWAN, el objetivo de este framework es brindar una serie de herramientas que van a permitir realizar ataques hacia una infraestructura LoRaWAN (ChirpStack), con esto se podrá observar las vulnerabilidades de esta tecnología.

3.3 identificar del ataque a utilizar

Una vez se ha realizado la lectura y compresión de lo que son los ataques en las redes LoRaWAN, el siguiente paso a seguir es identificar el ataque a utilizar para realizar las pruebas en el ambiente creado de la infraestructura de LoRaWAN.

El ataque que se va a utilizar como ya se ha mencionado en secciones anteriores de este documento es el ataque UdpSender.py, como se muestra en la figura 10.

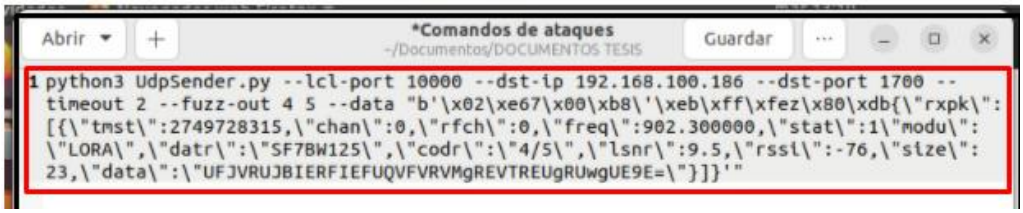


Figura 10. Mensaje UdpSender.py

Al mensaje que se aprecia en la figura se le tuvieron que realizar cambios, para que este tenga sentido y se pueda realizar el ataque hacia el dispositivo objetivo y se pueda verificar si este ataque ha sido recibido exitosamente por el dispositivo objetivo.

3.4 Identificación del dispositivo objetivo

Para realizar la identificación del dispositivo objetivo, ingresamos a la infraestructura ChirpStack que se creó previamente y se revisa los dispositivos que se tienen creados dentro del NS, el dispositivo que se ha creado previamente tiene una dirección en hexadecimal con la siguiente nomenclatura **AB CD 01 24,** como se muestra en la figura 11.

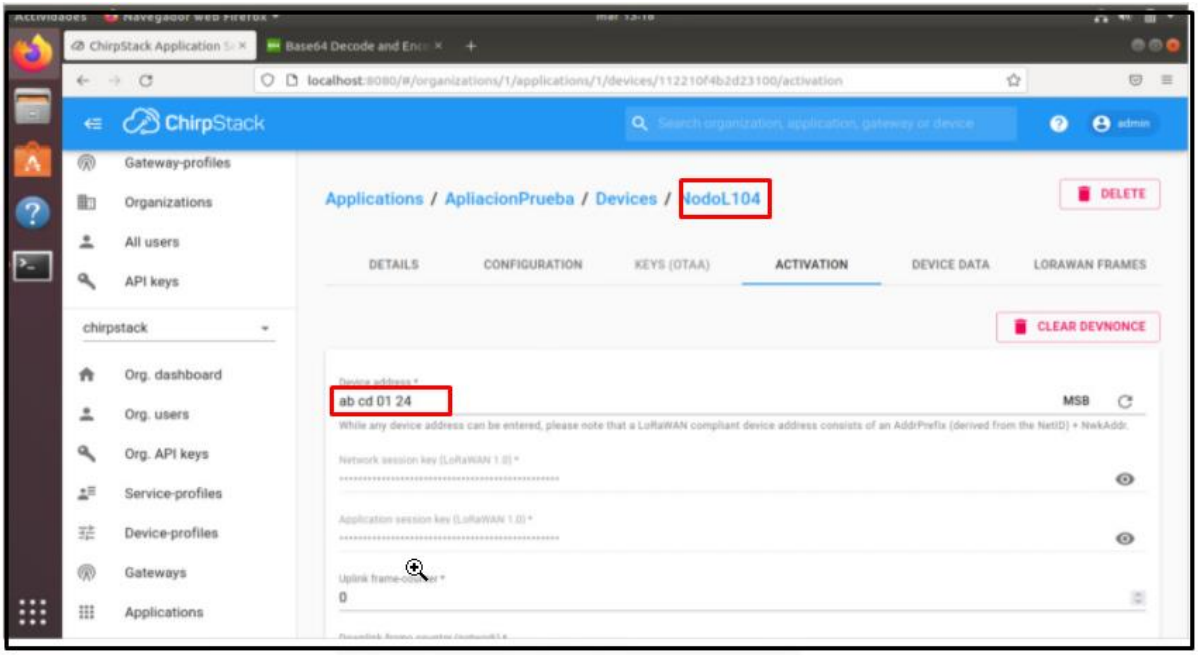


Figura 11. Dispositivo objetivo

Como se aprecia en la figura anterior ya se tienen identificado el dispositivo que va a ser el que reciba los ataques que se van a enviar desde el framework implementado en la maquina física, este dispositivo se encuentra levantando dentro de un ambiente virtual como ya se especificó en secciones anteriores en este documento

3.5 Modificación del mensaje (Parámetros Válidos)

En este punto del documento se va a verificar los resultados en la modificación del mensaje que se ja realizado, como se mencionó en la sección 3.3, los cambios que se van a realizar son los siguientes: cambio en la dirección del dispositivo objetivo, y se pueden realizar algunos cambios en el mensaje que contiene la data, esto con el fin de verificar que el mensaje se esté enviando correctamente hacia el dispositivo objetivo.

Los cambios realizados se aprecian en la Figura 12 que se muestra a continuación.

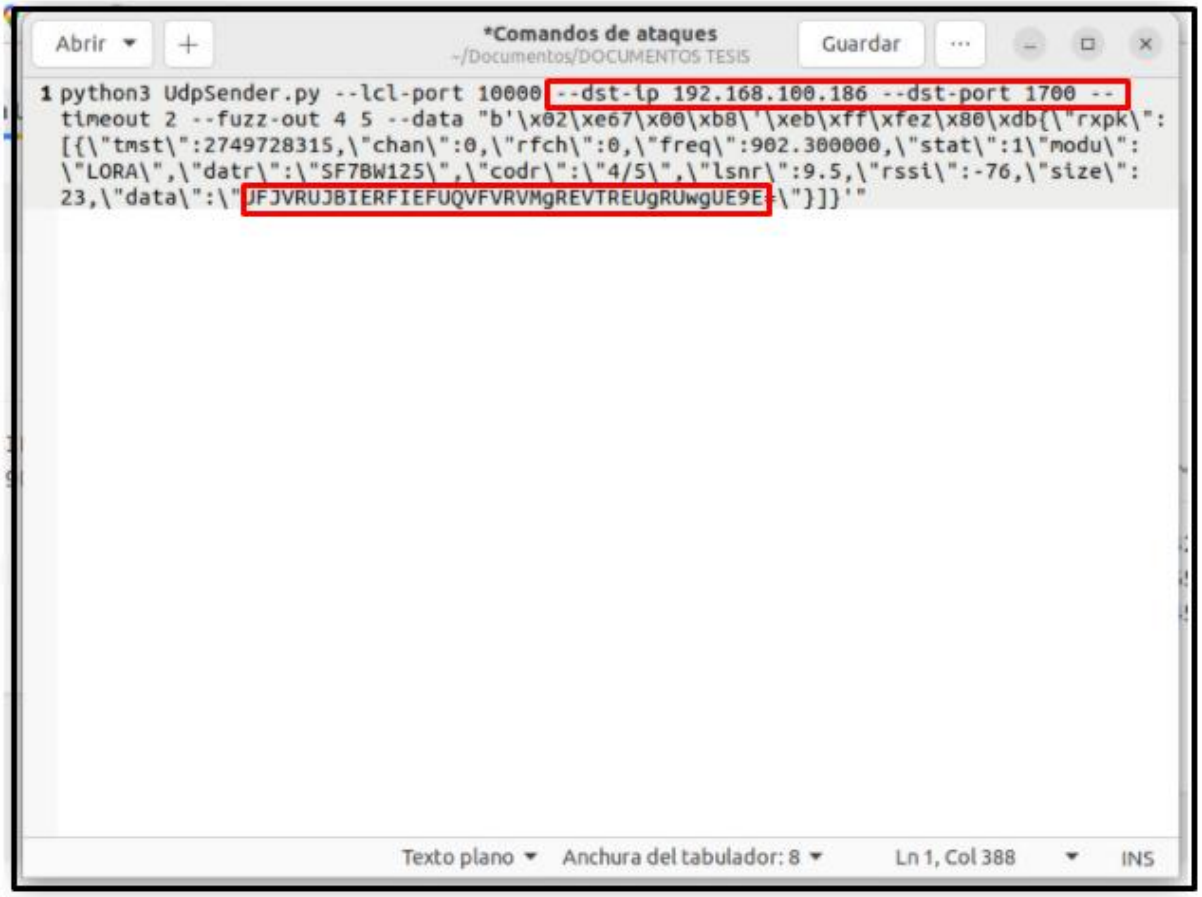


Figura 12. Modificación del mensaje (Dirección del dispositivo objetivo)

Como se puede apreciar los cambios realizados, el mensaje ya contienen la dirección del dispositivo objetivo y se encuentra listo para poder realizar las pruebas.

3.6 Ejecución del ataque hacia el dispositivo

una vez que se ha realizado los cambios al mensaje que ha sido elegido para realizar las pruebas en la infraestructura el siguiente paso a seguir es realizar el ataque hacia el NS

En este paso se van a realizar dos actividades principales, la primera va a ser abrir la herramienta laf implementada dentro de los kubernetes, y nos vamos a ubicar en la ruta que muestra en la Figura 13.

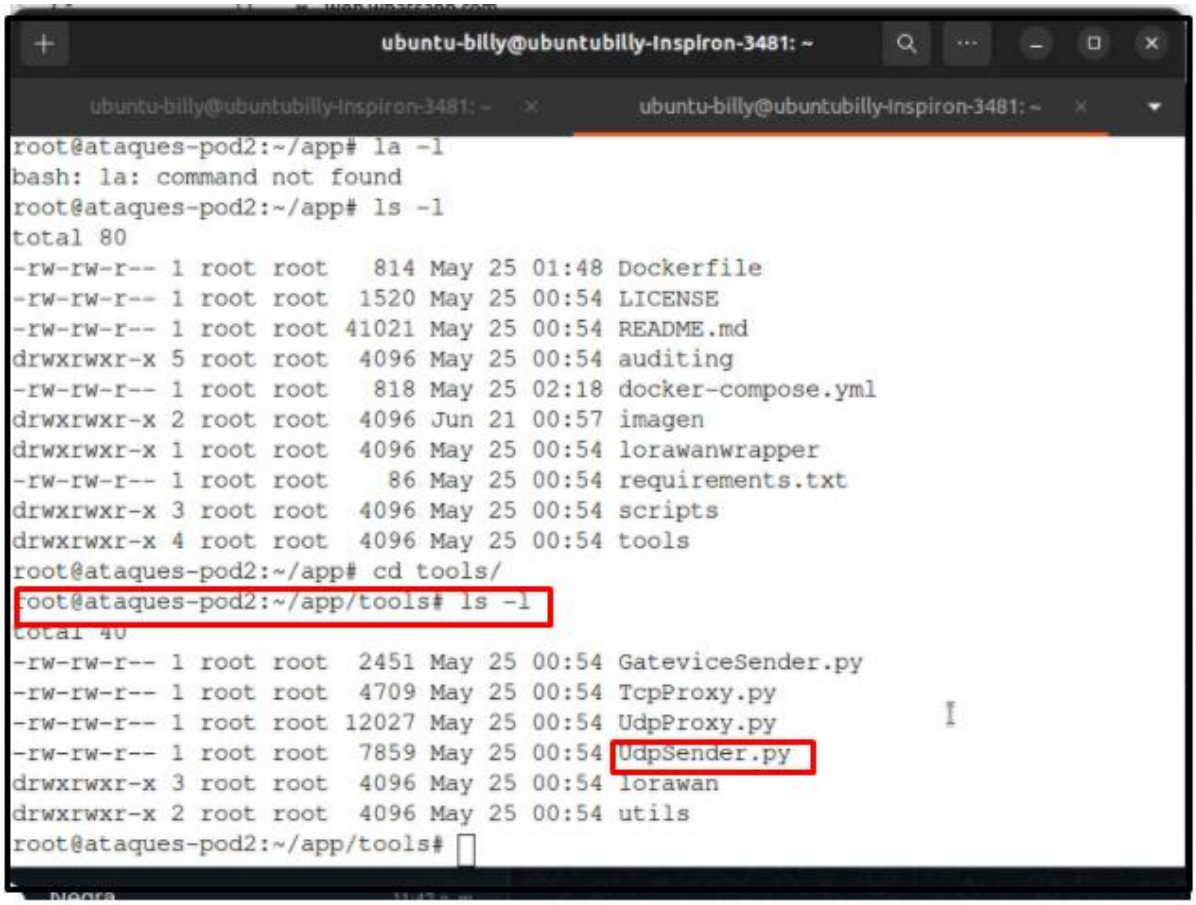


Figura 13. Ruta de donde se encuentra el archivo UdpSender.py

Cuando nos encontramos ya en la ruta, el siguiente paso a seguir es escribir el comando que se va a enviar, como ya se revisó en las secciones anteriores el mensaje ya se lo tiene listo y con los cambios preparados listo para realizar el ataque. Como se muestra en la Figura 14.

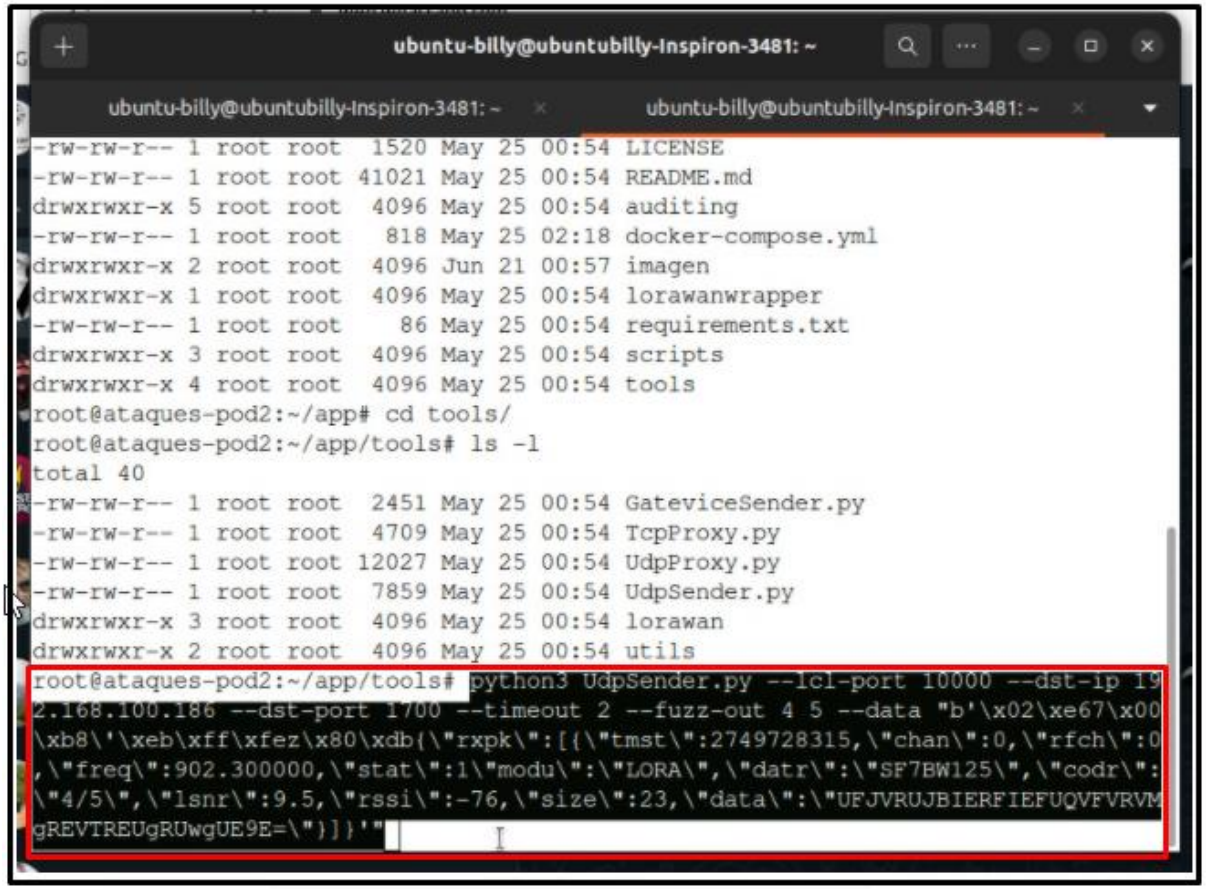


Figura 14. Mensaje preparado para ser enviado

Se puede apreciar la arquitectura del ataque, en la Figura 15, que se muestra a continuación, con esto se tiene una mejor comprensión de cómo se está realzando el ataque hacia el NS.

Y de qué manera llegan los mensajes incluso de las réplicas, es importante mencionar que las réplicas ejecutan el mismo ataque hacia el NS, ya que son réplicas del nodo atacante.

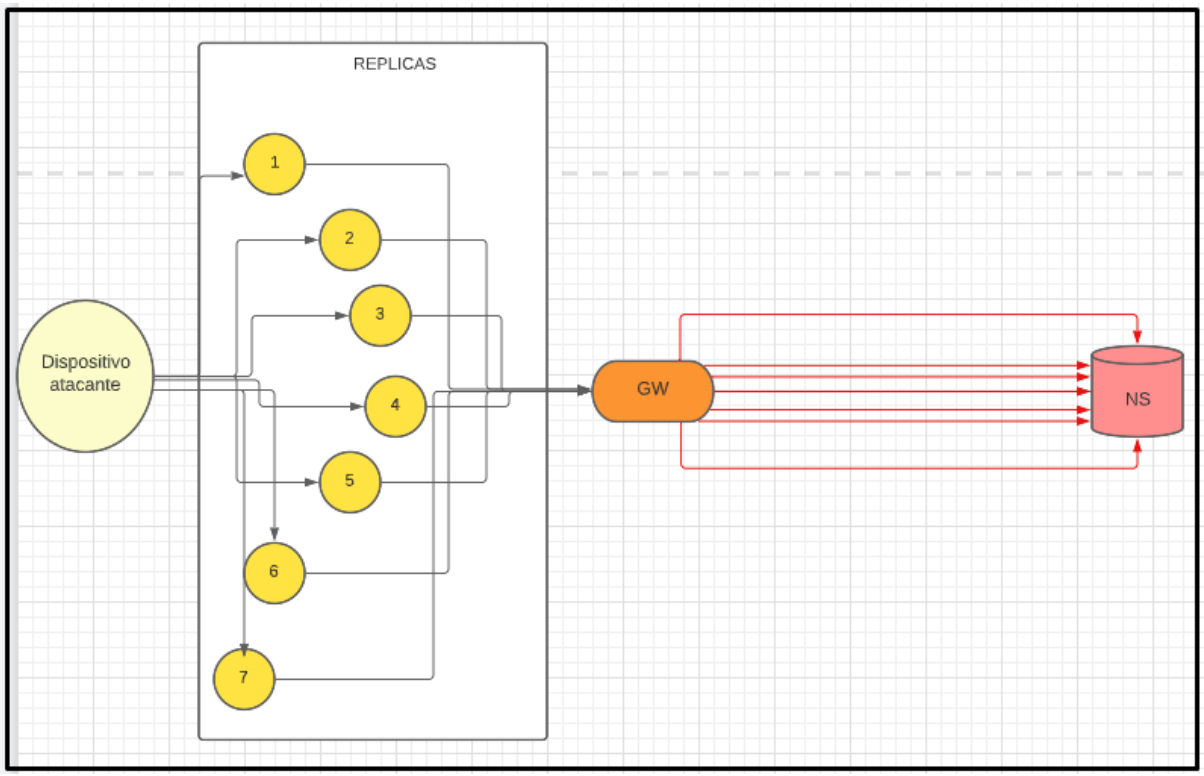


Figura 15. Arquitectura de los ataques hacia el NS

3.7 Revisión de los mensajes en el dispositivo objetivo

Para realzar la verificación de si los mensajes han llegado al dispositivo objetivo, se va a hacer uso de la herramienta Wireshark, esta herramienta nos permite analizar el flujo de red, en este caso nos vamos a concentrar en el flujo de red que esta llegado hacia la máquina virtual que tiene implementado la infraestructura ChirpStack.

En la Figura 16. Se muestra el tráfico de red que está recibiendo la máquina virtual y como se aprecia los mensajes están llegando correctamente hacia el NS, para esto analizamos la data que se está enviado en el paquete, para esto la herramienta Wireshark es de gran utilizad.

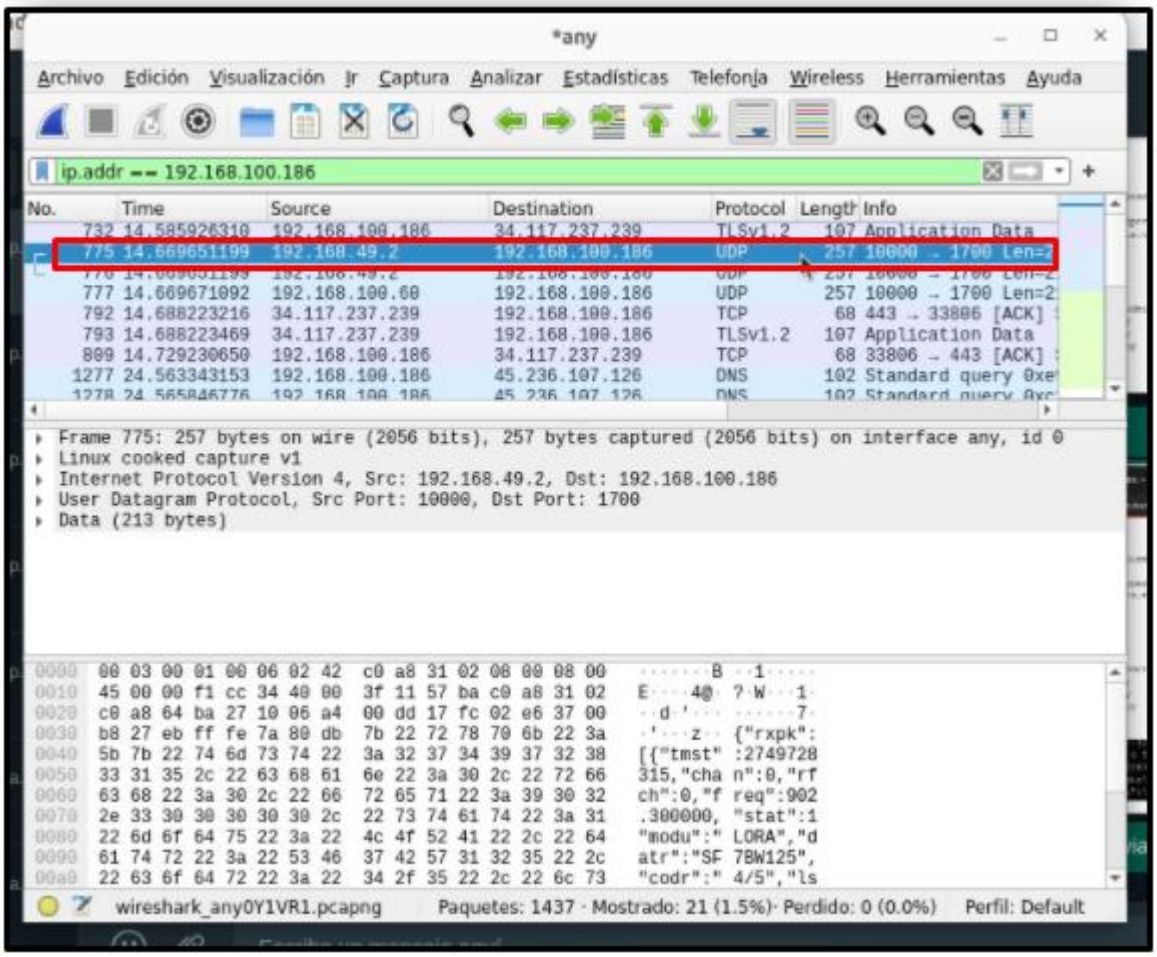


Figura 16. Trafico de red en la herramienta wireshark.

Como se puede observar en la figura los paquetes están llegando de forma correcta, con lo que se puede decir que el resultado de esta sección es exitoso y cumple con el objetivo de este proyecto.

3.8 Pruebas

Una vez que el resultado de la sección anterior ha sido exitoso, el siguiente paso a seguir es realizar pruebas, es decir continuar mandando mensajes hacia el dispositivo objetivo, en este caso se realiza el envió de 5 paquetes con los mismos parámetros de mensajes con el fin de seguir realizando las pruebas y comprobar que el dispositivo sigue recibiendo los mensajes.

Para esto, hacemos uso de la misma herramienta Wireshark en la cual analizamos todo el tráfico de la máquina virtual, como se puede observar en la Figura 17. Todos los paquetes han sido recibidos de manera exitosa lo que nos permite llegar al último paso de la verificación de este sprint que es la amplificación de los ataques utilizando comandos que pertenecen a kubernetes.

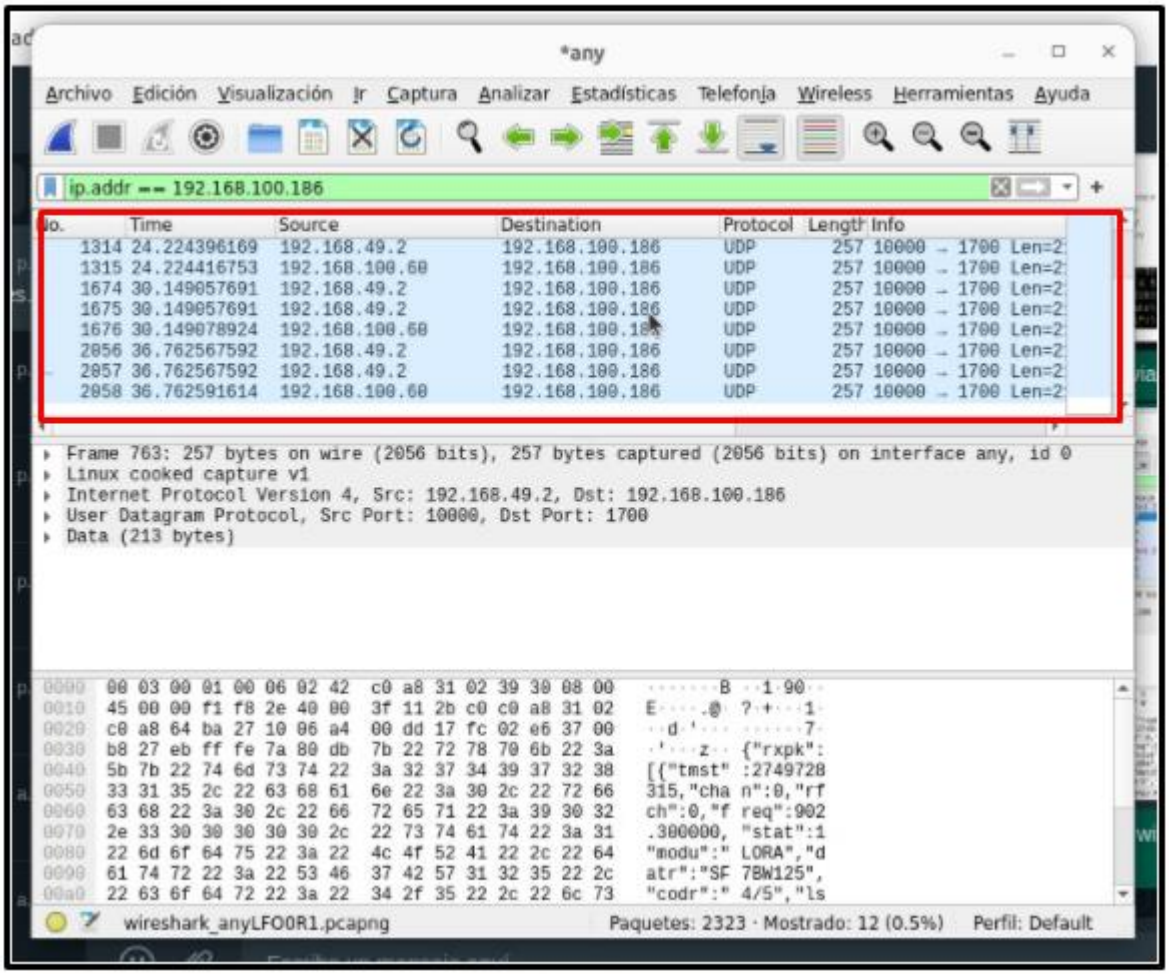


Figura 17. Trafico de wireshark, 5 paquetes recibidos

3.9 Amplificación del ataque con 7 réplicas del nodo atacante

Cuando se tienen resultados exitosos con las pruebas anteriores, se comprueba que el funcionamiento de la arquitectura es correcto, con lo que se procede a entender que para cumplir con el objetivo principal de este proyecto el cuál es amplificar los paquetes maliciosos enviados hacia el dispositivo objetivo, para realizar esta amplificación se utiliza un comando que ya cuenta los kubernetes, el cual permite realizar réplicas del nodo que está realizando los ataques como se muestra en la Figura 18.

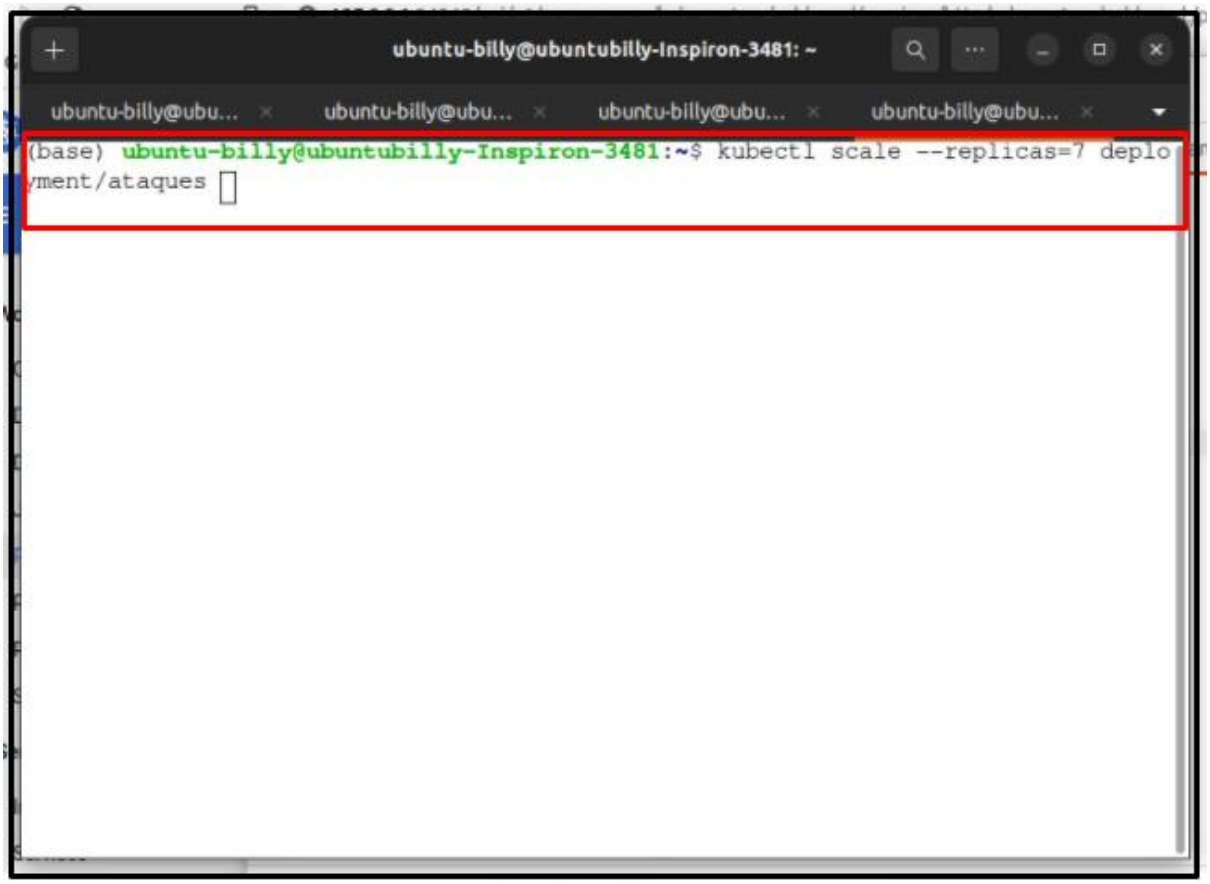


Figura 18. Comando para realizar las réplicas del nodo atacante.

Para verificar que las réplicas se crearon correctamente se accede al Dashboard de kubernetes y se observa que las réplicas estén creadas y se encuentren funcionando como se muestra en la Figura 19.

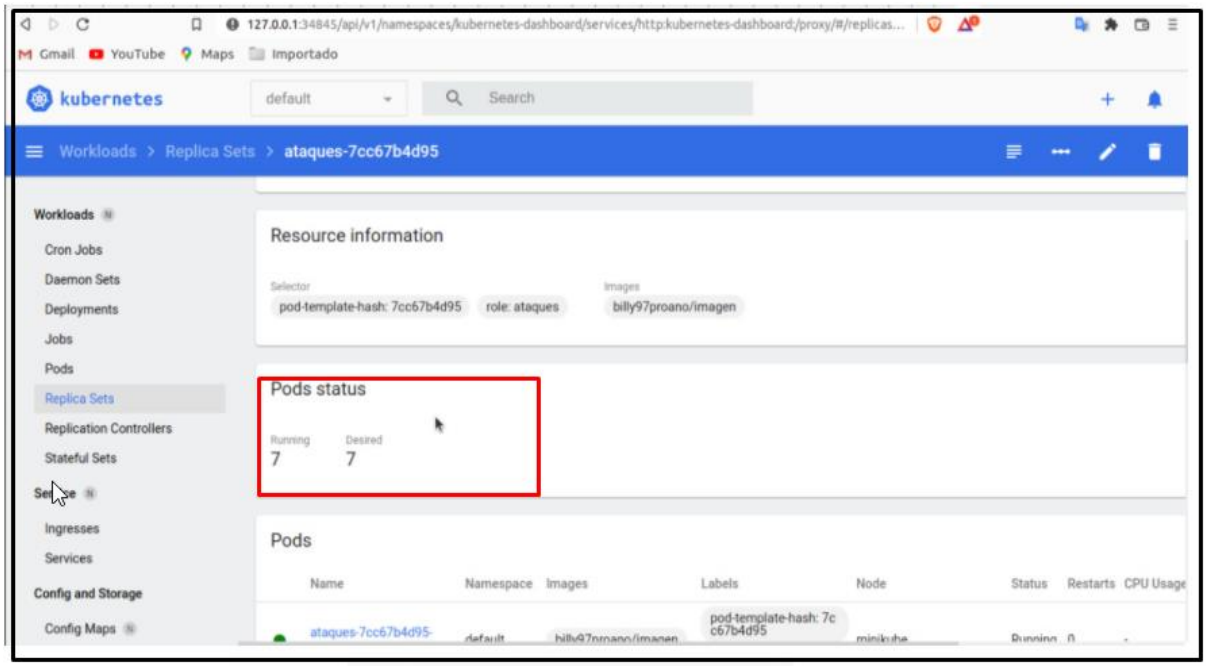


Figura 19. Dashboard de kubernetes con réplicas de nodos atacantes

Como se aprecia en la Figura 20 las réplicas se han creado correctamente y están listas para amplificar los ataques hacia el dispositivo objetivo.

Si notamos la Figura 19 se puede observar cómo las 7 réplicas del dispositivo atacante se encuentran corriendo y todas realizan el ataque hacia el NS, que es el objetivo.

Figura 20. Réplicas del nodo atacante activas y corriendo

Como se puede observar en la figura anterior cada replica tiene un identificador diferente con esto se puede detectar de que cada una de las réplicas está cumpliendo con su objetico que es enviar mensajes al NS.

Cabe tomar en cuenta que debido al hardware en el que se encuentra virtualizado el servidor de kubernetes, nos permite una cantidad máxima de replicas como se muestra en la Figura 21, por lo tanto, se utilizan 7 réplicas, una cantidad suficiente pero que no exceda la cantidad de replicas máxima, para evitar conflictos antes de realizar los ataques hacia el NS, y asi llevar a cabo el correcto funcionamiento del prototipo.

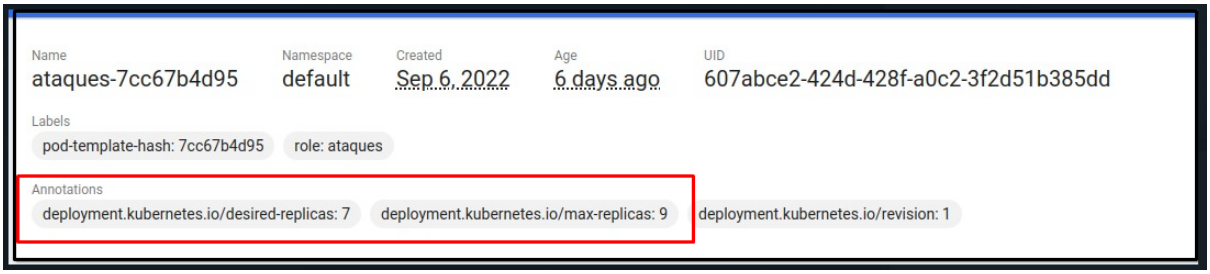


Figura 21. Cantidad máxima de replicas, cantidad utilizada de replicas

Una vez se evidencia las configuraciones necesarias para las réplicas se procede a realizar los ataques hacia el NS, para determinar si los objetivos planteados se cumplieron.

4 Conclusiones y Recomendaciones

Conclusiones

- Se logro desarrollar una infraestructura, que permite generar paquetes de DOS, sobre una red LoRaWAN, desplegando en contenedores dinámicos (Kubernetes), como se muestra en la sección de resultados (3.9), observando que se utilizaron 7 réplicas, para maximizar el ataque hacia el NS.
- Se diseño una arquitectura basada en contenedores dinámicos, para el caso específico de este componente se hizo uso de kubernetes, por su simplicidad y eficacia al momento de realizar réplicas de un nodo, con un contenedor atacante.
- Dados los recursos de Hardware y Software del equipo de trabajo, se levantó un ambiente virtual como se mencionó en secciones anteriores en el presente documento, es asi como luego de ejecutar varios ataques con las réplicas, la máquina virtual empezó a ralentizarse, sin embargo, el ambiente no se estropeo.
- Se logró realizar la adaptación del mensaje de la herramienta LAF [17], con el fin de que este mensaje tenga sentido cuando se dirija hacia el dispositivo objetivo, dando como resultado exitoso que el dispositivo objetivo recibe el ataque como se muestra en la sección de resultados (3.5).
- Cada nodo atacante, está realzando él envió de un mensaje hacia el NS, al ser 7 réplicas las que se están utilizando, se concluye que se envían 7 paquetes en cada ejecución del nodo con sus réplicas, como se pudo evidenciar en la sección de resultados (3.8), se evidencia la cantidad de paquetes que recibe el NS, por ejecución del ataque.

Recomendaciones:

- La principal recomendación que sale de este proyecto es estudiar y analizar el framework LAF, además de realizar pruebas con los demás ataques y herramientas que este framework proporciona, ya que a un futuro cercano las redes LoRaWAN, serán unas de las más utilizadas en el mundo de IoT.
- Se recomienda dar una lectura pausada y acceder a las referencias que se mencionan para tener una mejor compresión de lo que se trata, tanto en el marco teórico como la introducción.
- Utilizar las herramientas tanto el framework LAF como ChirpStack para experimentaciones futuras en cuanto a lo que es el protocolo LPWAN y la tecnología LoRaWAN.
- Realizar adaptaciones al framework LAF, ya que este es de código abierto y se encuentra publicado en el repositorio de GitHub, siendo de libre acceso para la comunidad tanto de desarrolladores, como para investigaciones en lo que son redes LoRaWAN.

5 Bibliografía

[1] M. S. a. F. E.-M. D. Bastos, «A survey of technologies and security risks in smart home and city environments,» de *Internet of things*, 2018, pp. 1-7.

[2] B. A. M. H. a. J. S. Ruth Ande, «Internet of things,» de *Evolution and technologies from a security perspective. Sustainable Cities and So- ciety*, 2022, p. 54:101728.

[3] Iotech, «REdes LPWAN, SIGFOX; LoRa,» [En línea]. Available: https://iottech.com.co/redes-lpwan-sigfox-lora. [Último acceso: 26 08 2022].

[4] O. d. s. d. internauta, «OSI,» 21 08 2018. [En línea]. Available: https://www.osi.es/es/actualidad/blog/2018/08/21/que-son-los-ataques-dos-y-ddos. [Último acceso: 26 08 2022].

[5] A. B. a. D. R. R. Minerva, «Define IoT - IEEE Internet of Things,» [En línea]. Available: https://iot.ieee.org/definition.html. [Último acceso: 02 08 2022].

[6] E. A. Lee, «Cyber physical systems: Design challenges,,» de *Symposium on Object/Component/Service-Oriented Real-Time Distributed*, 208, pp. 363-369.

[7] J. W. H. Y. a. H. S. J. Shi, «A survey of Cyber-Physical Systems,» de *A survey of Cyber-Physical Systems*, 2011.

[8] B. M. a. D. G. J. Yick, «Wireless sensor network survey,» de *Comput Networks*, 2008, pp. 2292-2330.

[9] «IEEE Standard for Low-Rate Wireless Networks,» 2020. [En línea]. Available: https://ieeexplore.ieee.org/document/9144691. [Último acceso: 02 08 2022].

[10] E. S. Farrell, «Low-Power Wide Area Network (LPWAN) Overview,» 2018. [En línea]. Available: https://tools.ietf.org/html/rfc8376. [Último acceso: 02 08 2022].

[11] L.-A. Enma Parce, «LoRaAlliance,» [En línea]. Available: https://lora-alliance.org/about-lorawan/. [Último acceso: 27 07 2022].

[12] «The world's first search engine for Internet-connected devices,» [En línea]. Available: https://www.shodan.io/. [Último acceso: 08 08 2022].

[13] Gemalto, « Actility and Semtech. LoRaWANTM Security, full end-to-end enctryption for IoT application providers.,» [En línea]. Available: https://lora-alliance.org/sites/default/files/2019-05/lorawan_security_ whitepaper.pdf . [Último acceso: 08 08 2022].

[14] M. P. a. M. A. Cusumano, «Lean Software Development: A Tutorial,» 2012, pp. 26-32.

[15] Microsoft, «The STRIDE Threat Model,» [En línea]. Available: https://docs.microsoft.com/en-us/previous-versions/commerce-server/ ee823878(v%3dcs.20). [Último acceso: 16 08 2022].

[16] L. Constantino, «CSO Las fallas ed la implementación hacen que las redes LoRaWAN sean vulnerables a los ataques,» 28 01 2022. [En línea]. Available:

https://www.csoonline.com/article/3516318/implementation-flaws-make-lorawan-networks-vulnerable-to-attack.html. [Último acceso: 16 08 2022].

[17] M. S. -. Enfayo, «IOActive/LAF,» 07 01 2019. [En línea]. Available: https://github.com/IOActive/laf. [Último acceso: 16 08 2022].

[18] ChirpStrack, «Introduction - ChirpStack Application Server.,» [En línea]. Available: https://www.chirpstack.io/application-server/. [Último acceso: 16 08 2022].

[19] ChirpStack, «Introduction - ChirpStack Network Server.,» [En línea]. Available: https://www.chirpstack.io/network-server/. [Último acceso: 16 08 2022].

[20] ChirpStack, «Introduction - ChirpStack Concentratord,» [En línea]. Available: https://www.chirpstack.io/concentratord/. [Último acceso: 16 08 2022].

[21] C. Requiements, «Requirements,» [En línea]. Available: https://www.chirpstack.io/project/install/requirements/. [Último acceso: 17 08 2022].

[22] Gateway-Bridge-intallation. [En línea]. Available: https://www.chirpstack.io/gateway-bridge/install/debian/. [Último acceso: 17 08 2022].

[23] U. I. 18.04, « Debian Ubuntu installation,» [En línea]. Available: https://www.chirpstack.io/network-server/install/debian/. [Último acceso: 17 08 2022].

[24] Aplication-Server-Install, «Chirpstack-debian / ubuntu / aplication server,» [En línea]. Available: https://www.chirpstack.io/application-server/install/debian/. [Último acceso: 17 08 2022].

[25] P. kubernetes, 31 07 2021. [En línea]. Available: https://gochronicles.com/minikube/. [Último acceso: 26 08 2022].

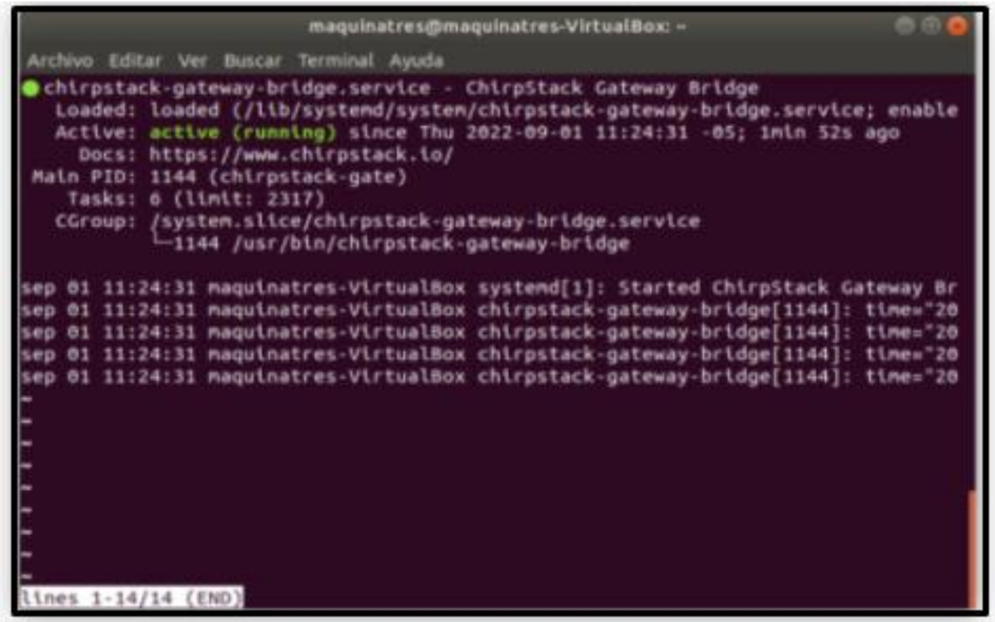
[26] minikube, «minikube start,» 21 06 2022. [En línea]. Available: https://minikube.sigs.k8s.io/docs/start/. [Último acceso: 17 08 2022].

[27] g. p. d. AWS, «aws,» [En línea]. Available: https://docs.aws.amazon.com/es_es/iot/latest/developerguide/iot-dg.pdf#connect-iot-lorawan-uplink-metadata-format. [Último acceso: 28 08 2022].

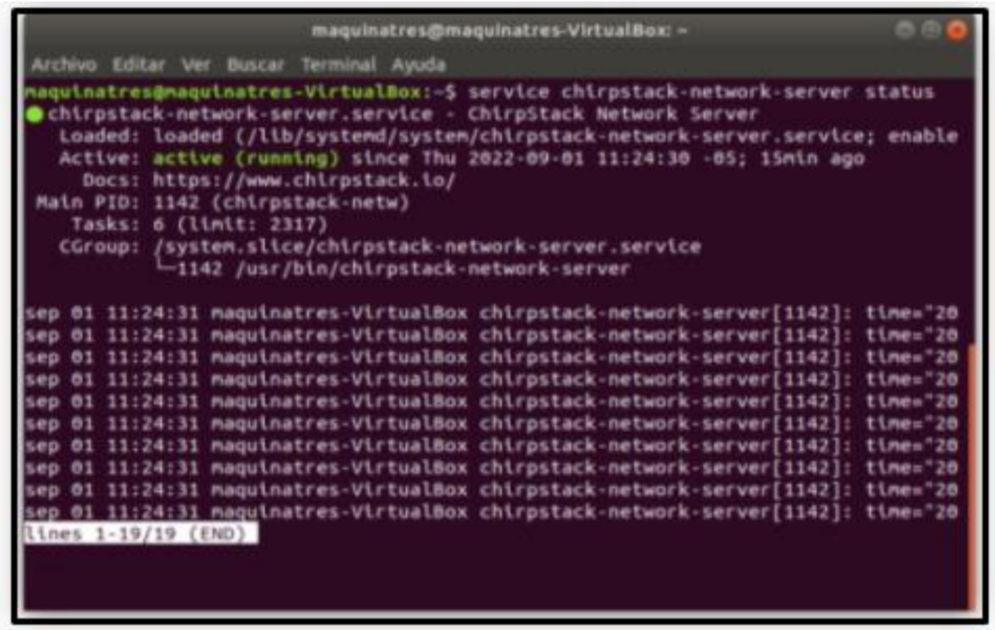
[28] S. Ukraini, «Cryptii,» [En línea]. Available: https://cryptii.com/pipes/base64-to-hex. [Último acceso: 30 08 2022].

ANEXOS

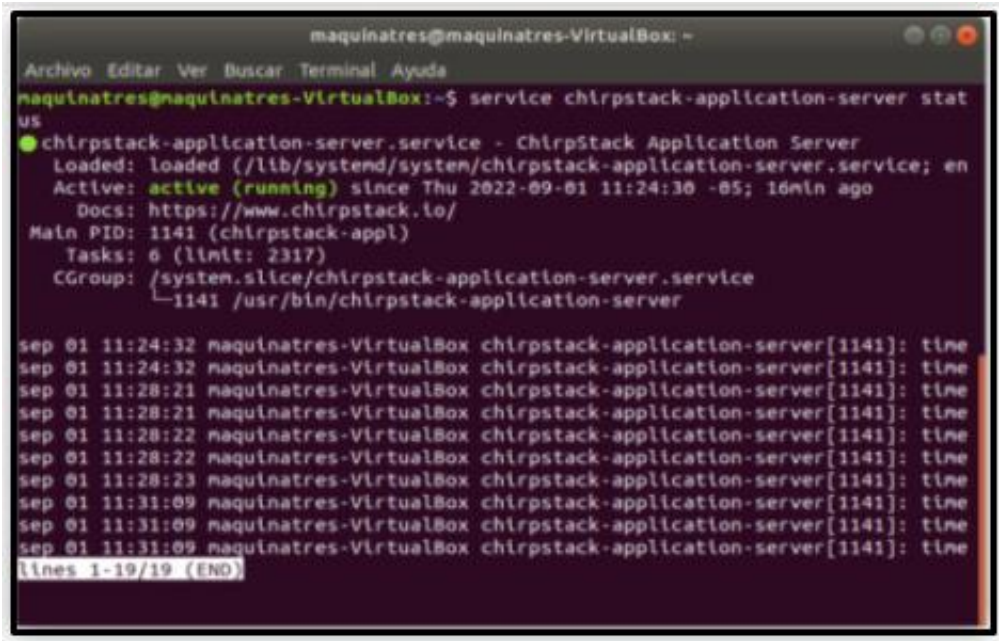
ANEXO 1 ESTADO DEL SERVICIO GATEWAY BRIDGE (GW)



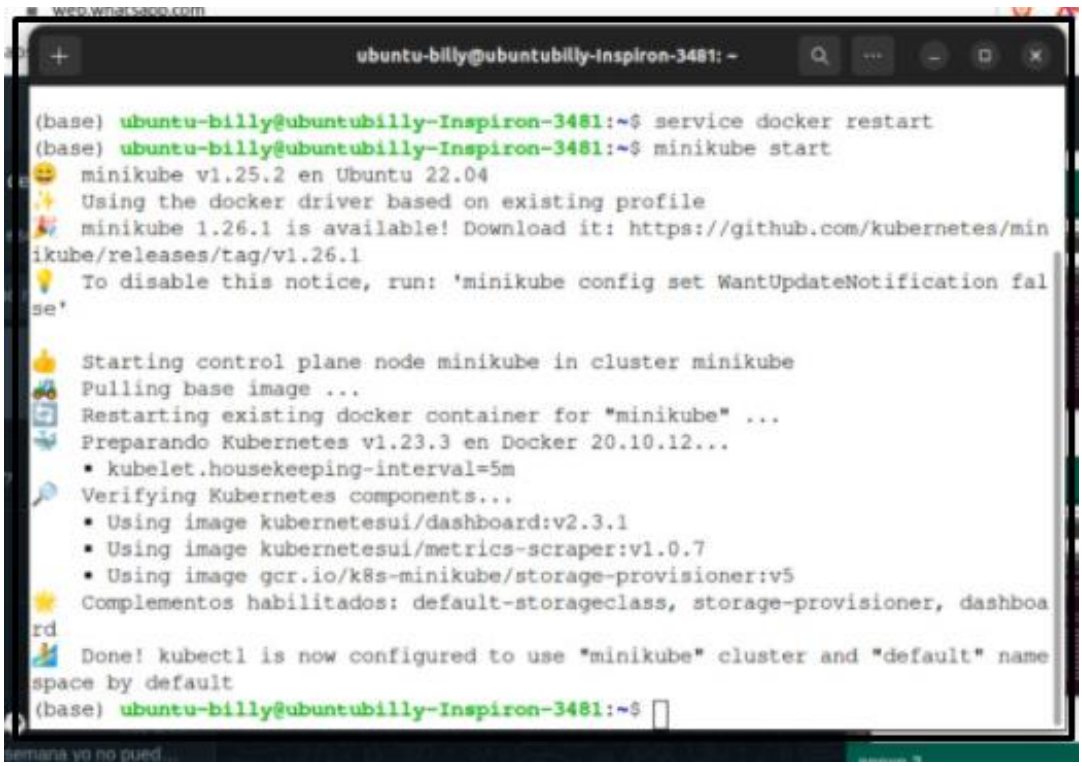
ANEXO 2 ESTADO DEL SERVICION NETWORK SERVER (NS)



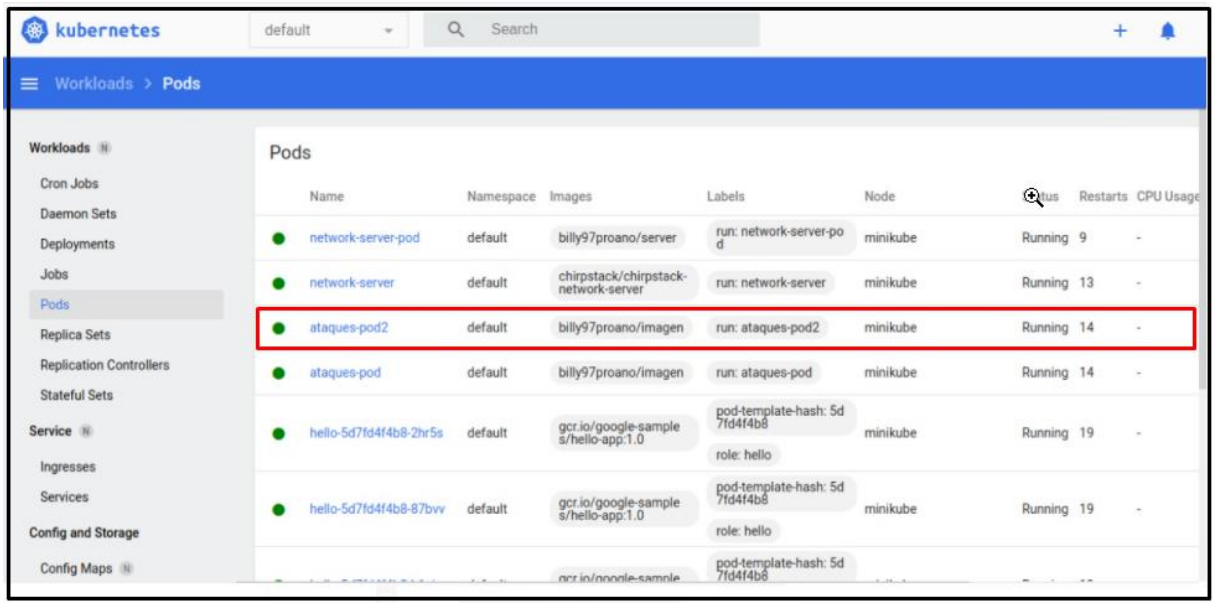
ANEXO 3 ESTADO DEL SERVICIO DEL APPLICATION SERVER (AS)



ANEXO 4 SERVICIO DE KUBECTL (MINIKUBE) EJECUTANDO



ANEXO 5 EJECUCION DE KUBERNETES VERIFICACION DE PODS CORRIENDO



ANEXO 6 SERVICIOS Y PUERTOS DE LA ARQUITECTURA CHIRPSTACK (GW – NS – AS)