Trabajo Fin de Grado en Ingeniería Informática

`WiFiGhost`: Herramienta de auditoría WiFi

Ismael Penacho Serhrouchni

Director: Ricardo J. Rodríguez Fernández
Co-Director: Daniel Uroz Hinarejos

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Junio de 2024
Curso 2023/2024

Agradecimientos

A mi madre y a mi hermana, sin su apoyo y amor incondicional nunca hubiera llegado a ser la persona que soy ha día de doy.

RESUMEN

La seguridad en redes WiFi es fundamental para proteger la privacidad y la integridad de la información transmitida a través de conexiones inalámbricas. El proceso de evaluación que busca identificar posibles vulnerabilidades en la red para prevenir accesos no autorizados se conoce como *auditoría de seguridad WiFi*. Durante estas auditorías, se realizan pruebas exhaustivas para evaluar la fortaleza de las contraseñas utilizadas y la configuración general de la red.

Los ataques más comunes incluyen pruebas de fuerza bruta, en las que se intentan múltiples combinaciones de contraseñas, y ataques de diccionario, que buscan coincidencias de las contraseñas con palabras comunes, claves predeterminadas, etcétera. El objetivo es descubrir debilidades que podrían ser explotadas por intrusos.

En este proyecto se desarrolla una herramienta, denominada `WiFiGhost`, que simplifica el proceso de auditoría al reducir la necesidad de conocimientos técnicos extensos por parte del usuario sobre auditorías WiFi. La herramienta dispone, además de una API REST, de una interfaz web. Esta interfaz no solo facilita la ejecución de comandos, como la monitorización de redes, sino que también mejora la presentación de los resultados de manera más amigable para el usuario. Además, se ha implementado un sistema de almacenamiento para gestionar de manera eficaz los diccionarios y los propios *hashes* y contraseñas obtenidas por el usuario, garantizando una gestión cómoda y organizada.

La herramienta permite acelerar tareas repetitivas durante la auditoría, como puede ser activar la tarjeta de red en modo monitor, cambios de la dirección de la tarjeta de red por motivos de seguridad y anonimato, o el escaneo de redes mostrando información relevante. Además también facilita el descifrado de los *hashes* obtenidos mediante el uso de tablas precomputadas.

ABSTRACT

Security in WiFi networks is fundamental to protect the privacy and integrity of information transmitted through wireless connections. The evaluation process that aims to identify possible vulnerabilities in the network to prevent unauthorized access is known as a *WiFi security audit*. During these audits, exhaustive tests are conducted to assess the strength of the passwords used and the overall network configuration.

The most common attacks include brute force tests, in which multiple password combinations are attempted, and dictionary attacks, which search for match passwords with common words, default keys, etc. The goal is to find weaknesses that can be exploited by intruders.

In this project, a tool called `WiFiGhost` is presented, which simplifies the audit process by reducing the need for extensive technical knowledge on the part of the user regarding WiFi audits. The tool has a web interface, as well as a REST API. The interface not only makes it easier to execute commands, such as network monitoring, but also improves the presentation of results in a more user-friendly way. Furthermore, a storage system has been implemented to effectively manage *wordlists*, *hashes* and passwords obtained by the user, guaranteeing comfortable and organized management.

The tool streamlines repetitive tasks during the audit, such as activating the network card in monitor mode, changing the physical network card address for security and anonymity, or scanning networks to display relevant information. It also facilitates the decryption of obtained hashes by using precomputed tables.

Índice general

Índice de figuras

Capítulo 1

Introducción

El proceso de auditar una red WiFi incluye técnicas avanzadas y requiere un conjunto de habilidades técnicas que puede ser una barrera para profesionales no especializados en entornos WiFi. Por lo tanto, la necesidad de abstraer y simplificar este proceso se vuelve evidente en la búsqueda de una solución que permita a profesionales de diversas disciplinas participar activamente en la evaluación de la seguridad de las redes WiFi.

El proyecto `WiFiGhost` se presenta como una respuesta estratégica, abriendo la puerta a una auditoría simplificada y accesible, eliminando la curva de aprendizaje y permitiendo la identificación rápida y precisa de vulnerabilidades en redes inalámbricas empresariales.

Al considerar el estado actual en este ámbito, dos aplicaciones relevantes son Fern-Wifi-Cracker [3] y Wifite [4]. Ambas buscan simplificar el proceso de auditoría al proporcionar funcionalidades automatizadas. Fern-Wifi-Cracker, aunque efectiva, presenta la limitación de depender de una interfaz no basada en web. Por otro lado, Wifite destaca por su enfoque en la simplicidad y la automatización, pero también se basa en una interfaz de línea de comandos. Estas aplicaciones representan una perspectiva actual del panorama de la auditoría WiFi automatizada, sirviendo como referentes en la búsqueda de soluciones más accesibles y completas.

El proceso de auditar las comunicaciones inalámbricas (*wireless*) se realiza con el objetivo de determinar el nivel de seguridad y confidencialidad que proporcionan las configuraciones de dicha red. Entre otros problemas, se suelen encontrar cifrados no óptimos o configuraciones erróneas en las redes inalámbricas empresariales, lo cual puede desembocar en una vía de entrada a la red de la organización o a la información de ésta.

Según las directrices del NIST SP 800-153 [11], la auditoría WiFi dispone de diferentes fases, entre ellas, una fase de reconocimiento del entorno y descubrimiento de la infraestructura. También se utiliza monitorización de red para visualizar los tipos de cifrado, el número de puntos de acceso que tiene la red de la organización, los canales por los que se emiten, el número de clientes conectados y a qué puntos de acceso, la calidad de la señal, etcétera. Posteriormente se emplean técnicas de análisis de vulnerabilidades sobre el *router*, técnicas de suplantación de identidad, envenenamiento de red y, por último, técnicas de ataques a contraseñas (por diccionario y fuerza bruta) [2]. La finalidad última de las técnicas mencionadas es detectar y verificar la existencia de vulnerabilidades.

1.1. Objetivo

Los objetivos fundamentales de este proyecto se derivan de simplificar y optimizar el proceso de auditoría de redes WiFi, abstrayendo conceptos teórico-prácticos y eliminando las barreras de aprendizaje asociadas.

El proyecto busca, en última instancia, hacer que la auditoría de redes WiFi sea una tarea más intuitiva y fácil de llevar a cabo, simplificando la tarea a llevar a cabo por parte del auditor. Esto se traduce en una experiencia más accesible y eficiente, eliminando complicaciones innecesarias. Para ello, en este TFG se implementa una herramienta denominada `WiFiGhost`.

Implicaciones éticas

Todas las pruebas y auditorías realizadas en el contexto de este proyecto se llevaron a cabo exclusivamente en entornos de laboratorio controlados. En ningún momento se afectaron redes de terceros sin su consentimiento explícito. Se garantiza que el propósito de estas pruebas es puramente educativo, destinado a proporcionar a los usuarios la oportunidad de practicar y mejorar sus habilidades en el ámbito de las auditorías de seguridad en redes WiFi.

Cabe resaltar que el usuario asume la responsabilidad de obtener los permisos necesarios y exime al desarrollador de `WiFiGhost` y cualquier entidad relacionada de cualquier responsabilidad derivada de un uso indebido o no autorizado.

1.2. Organización

Este documento se encuentra dividido en siete capítulos y un anexo. El Capítulo 2 introduce algunos conceptos útiles para facilitar la comprensión del resto del documento. En el Capítulo 3 se describen la arquitectura, diseño e implementación de la herramienta desarrollada. A continuación, en el Capítulo 4 se describe el uso del sistema implementado para la auditoría de redes WiFi. El Capítulo 5 concluye el trabajo y presenta el trabajo futuro.

Al final del documento se encuentra el Anexo A, que muestra detalladamente el tiempo invertido en este trabajo.

Capítulo 2

Conocimientos previos

En este capítulo se abordan los conceptos fundamentales necesarios para comprender la seguridad en redes WiFi y las herramientas de auditoría, así como UML. Estos conocimientos previos son esenciales para entender los métodos y técnicas utilizados a lo largo de este TFG.

2.1. Handshake

Handshake se refiere al intercambio de mensajes entre un dispositivo cliente y un punto de acceso Wi-Fi durante el proceso de autenticación. Cuando un dispositivo desea conectarse a una red protegida con *WPA* o *WPA2*, se inicia un *handshake*, donde el dispositivo solicita la conexión, el punto de acceso responde y ambas partes acuerdan una clave temporal para asegurar la comunicación. Este intercambio contiene información esencial para la seguridad de la red, incluyendo la confirmación de la identidad del dispositivo y la negociación de claves de cifrado. La captura de este *handshake* se vuelve crucial para realizar ataques de fuerza bruta y descifrar la contraseña de la red protegida con WPA o WPA2. En la figura 2.1 se muestra el proceso de autenticación.

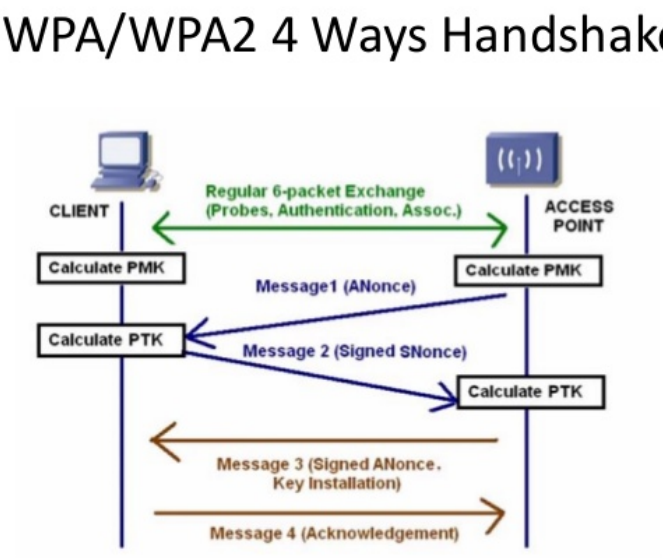

Figura 2.1: Concepto *Handshake*
Fuente: `https://blogs.protegerse.com/2017/10/17/krack-attack-rompe-el-cifrado-wpa2-y-ahora-que/`

Se debe tener en cuenta que al tratarse de una autenticación de claves precompartidas, se está haciendo uso de una contraseña única que de estar a disposición de cualquiera puede ser usada para llevar a cabo una asociación contra el punto de acceso wifi o lo que se conoce también como *Access Point* (AP). A la hora de llevar a cabo una asociación por una estación (cliente) contra el AP, se deja un rastro a nivel de paquetes, los cuales pueden ser capturados y tratados sin estar autenticados al punto de acceso para posteriormente extraer la contraseña de la red inalámbrica [1].

2.2. Tipos de cifrado en WiFi

Las redes WiFi utilizan diferentes protocolos de cifrado para proteger la confidencialidad y la integridad de la información transmitida. Se presentan a continuación los distintos cifrados actuales.

WEP (*Wired Equivalent Privacy*)

WEP fue uno de los primeros protocolos de cifrado utilizados en redes WiFi [12]. Emplea el algoritmo de cifrado de flujo RC4 para proteger la comunicación inalámbrica [13]. Las principales desventajas de este sistema incluyen debilidades en la implementación

del algoritmo RC4, el uso de claves estáticas que pueden ser vulnerables a ataques de fuerza bruta, y problemas relacionados con el vector de inicialización, lo que facilita la recuperación de la clave.

WPA (*Wi-Fi Protected Access*)

WPA se introdujo como una mejora de seguridad sobre WEP [12]. Utiliza el protocolo TKIP (*Temporal Key Integrity Protocol*) para mejorar la seguridad de las claves [15]. Las principales desventajas de este sistema incluyen vulnerabilidades en la implementación de TKIP, que abarcan ataques de inyección, así como el uso de claves precompartidas que pueden ser susceptibles a ataques de fuerza bruta.

WPA2 (*Wi-Fi Protected Access 2*)

WPA2 funciona en dos modos, modo personal o clave precompartida (WPA2-PSK), el cual se basa en un código de acceso compartido y en el que se centra este trabajo, y el modo empresarial (WPA2-EAP) que requiere un servidor de autenticación para gestionar las credenciales de los usuarios [12].

Ambos modos utilizan el protocolo de cifrado AES-CCMP (*Advanced Encryption Standard-Counter Mode Cipher Block Chaining Message Authentication Code Protocol*) [5]. El protocolo CCMP se basa en el algoritmo Estándar de cifrado avanzado (del inglés, *Advanced Encryption Standard*, AES [14]), que proporciona una verificación de la autenticidad e integridad de los mensajes. CCMP es más resistente y fiable que el TKIP original de WPA, que dificulta a los atacantes la detección de patrones.

Sin embargo, WPA2 también tiene sus inconvenientes. Por ejemplo, es vulnerable a los ataques de reinstalación de claves *KRACK* (*Key Reinstallation Attacks*) [16], que posibilita la interceptación de tráfico, junto con el uso de claves precompartidas que pueden ser susceptibles a ataques de fuerza bruta.

Existen dos modos principales de seguridad en WPA2. El modo personal o clave precompartida (WPA2-PSK) se basa en un código de acceso compartido y generalmente se usa en entornos domésticos. Por otro lado, el modo empresarial (WPA2-EAP) es más adecuado para uso en empresas u organizaciones, ya que proporciona un nivel de seguridad adicional mediante la utilización de un servidor de autenticación.

2.3. Tablas arcoiris

Las tablas arcoiris (o *rainbow tables*) son herramientas utilizadas en ciberseguridad para agilizar ataques de fuerza bruta contra contraseñas. En lugar de almacenar todos los posibles *hashes*, estas tablas precalculadas emplean una técnica denominada reducción para minimizar el espacio de almacenamiento. La reducción implica reducir la salida del *hash* a una longitud manejable. La estrategia central consiste en precalcular los *hashes*

de un conjunto de contraseñas comunes, almacenar estos pares en una tabla y, posteriormente, buscar los *hashes* capturados durante un ataque para descubrir las contraseñas originales correspondientes. La aplicación repetida de la reducción genera nuevas contraseñas y *hashes* en un proceso en cadena, dando lugar a un conjunto diverso de combinaciones conocido como "*Rainbow*". Este método se utiliza para optimizar la velocidad y eficiencia de los ataques, permitiendo a los atacantes recuperar contraseñas de manera más rápida [7].

2.4. Herramientas de auditoría WiFi

Suite Aircrack-ng

Aircrack-ng es una potente suite de herramientas de código abierto diseñada para evaluación de seguridad en redes inalámbricas. Incluye herramientas como Airodump-ng para la captura de paquetes, Aireplay-ng para la inyección de tráfico y la realización de ataques de deautenticación, y finalmente, Aircrack-ng para la recuperación de claves *WEP* y *WPA-PSK*. Además, Aircrack-ng es compatible con tarjetas inalámbricas que admiten el modo de monitorización, lo que facilita su integración en diversas plataformas [17].

MDK3

MDK3 es una herramienta de prueba de penetración incluida en la suite de herramientas Aircrack-ng, se destaca al ejecutar ataques específicos diseñados para poner a prueba la resistencia de una red en situaciones adversas, con un enfoque particular en la generación de tráfico malicioso y la degradación del rendimiento de la red. Su versatilidad abarca desde la saturación de la red hasta la generación de redes falsas, proporcionando así una evaluación exhaustiva de la capacidad de respuesta y seguridad de la infraestructura inalámbrica [6, 18].

HCXDumpTool

HCXDumpTool es una herramienta de código abierto, la cual tiene la capacidad para capturar información de redes Wi-Fi incluso cuando no hay clientes conectados. Esto se debe a su capacidad para forzar un nuevo *handshake* de autenticación. En situaciones donde no hay clientes conectados a la red WiFi objetivo, es difícil o imposible capturar el *handshake* convencionalmente [7, 19].

Pyrit

Es una herramienta de código abierto, que entre otras tiene dos funcionalidades cruciales para el objetivo principal. Permite la comprobación del *handhsake* capturado para

saber si este es correcto o es falso. Su enfoque principal es utilizar la potencia de procesamiento de unidades de procesamiento gráfico para acelerar significativamente el proceso de descifrado de contraseñas. Además permite usar *rainbow tables*, evitando tener que calcular los *hashes* de manera repetida durante un ataque de fuerza bruta [6, 7, 21].

2.5. UML

El lenguaje unificado de modelado (UML, en inglés *Unified Modeling Language* [22]) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos, etc. Es sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos.

UML usa elementos y los asocia de diferentes formas para realizar diagramas estructurales que representan aspectos estáticos de un sistema, diagramas de comportamiento, que captan los aspectos dinámicos de un sistema, y diagramas de interacción, que muestran cómo colaboran los distintos objetos del sistema de manera específica. Concretamente, en este trabajo se han usado diagrama de secuencia, que en UML muestran cómo los objetos interactúan entre sí y el orden en que se producen estas interacciones para un escenario en concreto.

Capítulo 3

Análisis, diseño e implementación

Este capítulo examina la arquitectura de `WiFiGhost`, así como se presentan diagramas UML de secuencia de su funcionamiento. Además, se abordan también las tecnologías utilizadas en la implementación.

3.1. Análisis e implementación

La arquitectura de `WiFiGhost`, como se muestra en la figura 3.1, sigue un enfoque de dos capas, distribuyendo la aplicación en un *frontend* y un *backend* para lograr una estructura eficiente y modular. La comunicación se realiza mediante HTTP en el puerto 8080 con el envío de peticiones al *backend*, además se utiliza el puerto 5000 para el envío de paquetes WEBSOCKET que permite una comunicación bidireccional en tiempo real entre el cliente (*frontend*) y el servidor (*backend*) utilizado en el escaneo de la red.

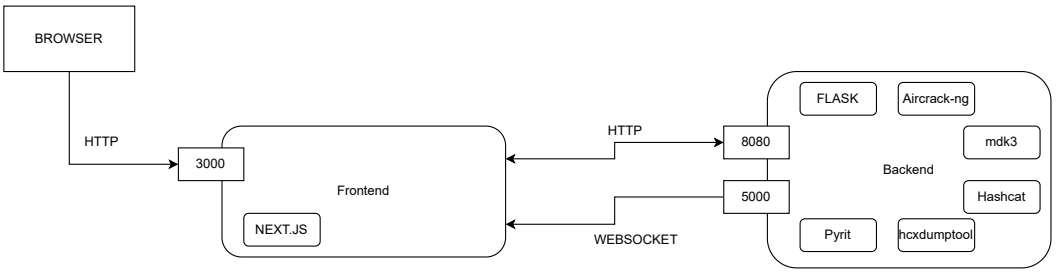

Figura 3.1: Arquitectura de la herramienta.

El *frontend*, basado en Next.js [10], se enfoca en la presentación y la interacción con el usuario final. Utilizando React, Next.js permite crear una interfaz de usuario moderna y receptiva. La elección de Next.js se fundamenta en su capacidad para construir aplicaciones de una sola página, proporcionando una experiencia de usuario fluida y evitando la recarga completa de la página en cada interacción. Además, la integración de componentes como MaterialUI (una biblioteca de componentes de interfaz de usuario para

React basada en Material Design) y keep-react (un conjunto de herramientas y componentes para la gestión del estado en aplicaciones React) garantiza un diseño atractivo y funcional. El *frontend* interactúa con el *backend* mediante solicitudes HTTP estándar, estableciendo así una comunicación efectiva entre ambas capas.

Respecto al *backend*, implementado con Flask [9], constituye la capa encargada de gestionar la lógica de ejecución de auditorías WiFi y la interacción con las herramientas esenciales para este propósito. Esta capa proporciona una interfaz de programación para la interacción con el *frontend* y asegura una gestión eficiente de los recursos del sistema.

Esta separación de *frontend* y *backend* en dos capas distintas facilita la escalabilidad, el mantenimiento y la modularidad del sistema, cumpliendo con los principios de una arquitectura eficiente y orientada a servicios.

3.2. Diseño

Inicialización

El proceso de *Inicialización* contempla el inicio de la aplicación, configuración de la estructura de almacenamiento de archivos temporales y la configuración necesaria del hardware para que la herramienta esté preparada para realizar tareas de escaneo de la red y ataques, lo que contempla finalizar procesos conflictivos para el uso de la tarjeta de red, activar el modo monitorización de la tarjeta de red y el cambio de la dirección MAC de origen. En la figura 3.2 se muestra el diagrama de secuencia que describe la interacción entre el *Frontend* y el *Backend* de `WifiGhost` durante el proceso de inicialización de la tarjeta de red.

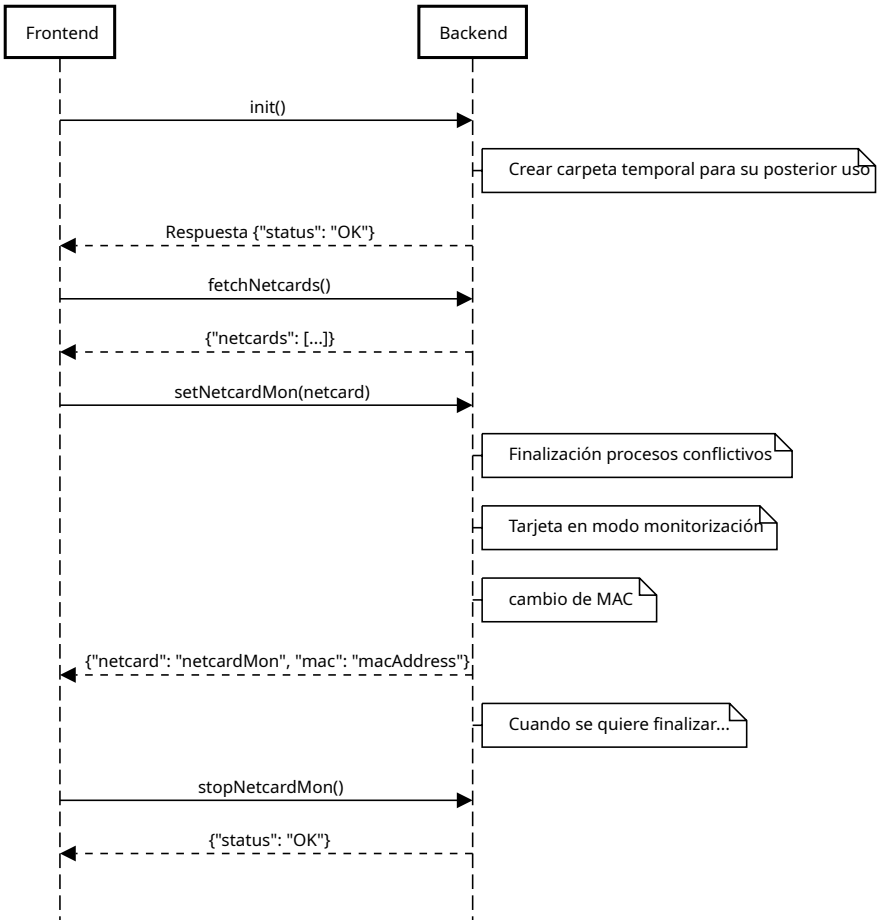

Figura 3.2: Diagrama de secuencia UML: Inicialización de la herramienta.

Escaneo y ataques

El proceso de *Escaneo y ataques* contempla la monitorización de la red escaneando diferente información relevante de los puntos de acceso disponibles en tiempo real [8] y los ataques correspondientes, en función de su tipo de seguridad. En la figura 3.3 se muestra el diagrama de secuencia que describe la interacción entre el *Frontend* y el *Backend* de `WifiGhost` durante el proceso de escaneo de la red, selección de punto de acceso y ataque contra el punto de acceso.

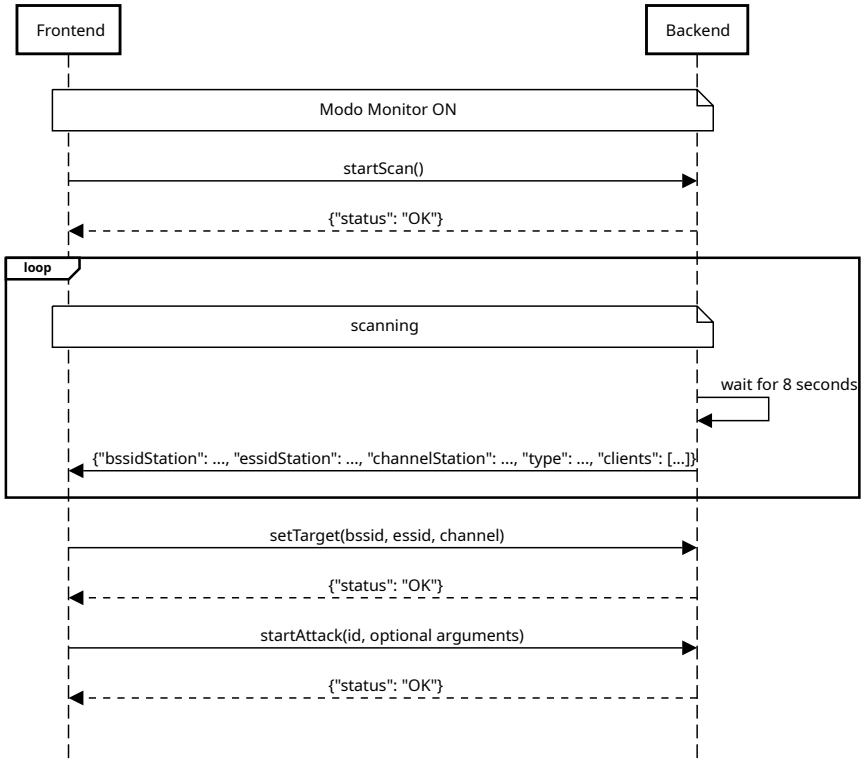

Figura 3.3: Diagrama de secuencia UML: Monitorización de la red y Ejecución de Ataque.

Cracking

El proceso de *Cracking* contempla el listado de *listas de palabras* y *hashes* disponibles y el *crackeo* de los datos de entrada que los generaron estos. En la figura 3.4 se muestra el diagrama de secuencia que describe la interacción entre el *Frontend* y el *Backend* de `WifiGhost` durante el proceso de *Cracking* del *hash* y *lista de palabras* seleccionados.

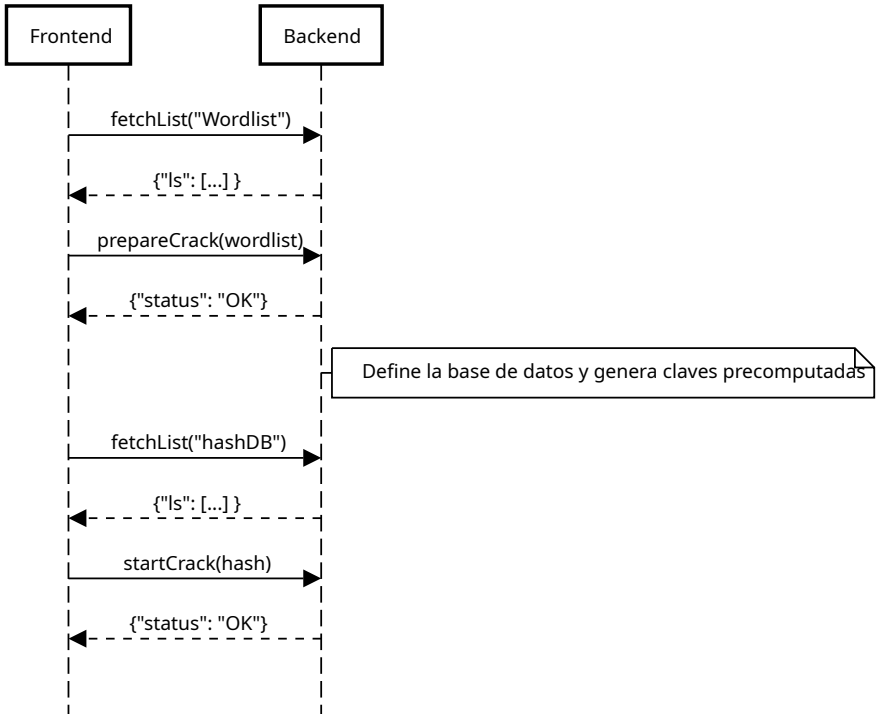

Figura 3.4: Diagrama de secuencia UML: *Cracking*.

Gestión archivos

El proceso de *Gestión de archivos*, que se describe en la figura 3.5 y como su propio nombre indica, contempla todo tipo de interacción con los ficheros del sistema ya sean *contraseñas*, *Hashes*, *listas de palabras* o *redes falsas*.

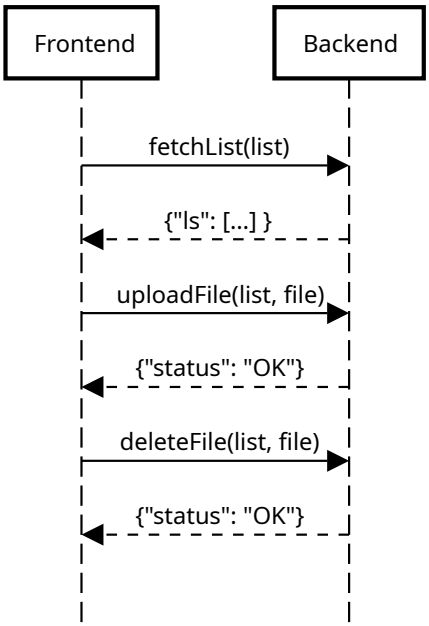

Figura 3.5: Diagrama de secuencia UML: Gestión de archivos.

Licencia

El código está disponible en GitHub [25], bajo la licencia pública general de GNU versión 3 (GNU GPLv3) [23].

Capítulo 4

Proceso de auditoría con WiFiGhost

En este capítulo se explicará detalladamente la usabilidad de la herramienta, proporcionando un enfoque acerca de los diferentes procesos de auditoría WiFi que realiza, en concreto: la monitorización de la red, los tipos de ataques en base al protocolo de encriptación y la gestión de archivos.

4.1. Monitorización de la red

La monitorización de la red es una parte esencial del proceso de auditoría WiFi. La herramienta desarrollada permite a los usuarios analizar las redes WiFi disponibles en su entorno. Utilizando el modo monitor de la tarjeta de red (como se muestra en la figura 4.1), `WiFiGhost` captura datos relevantes sobre las redes, incluyendo información sobre los puntos de acceso y clientes conectados tal y como se muestra en la figura 4.2. Cuenta también con la capacidad de modificar temporalmente la dirección MAC para evitar la detección por parte de los consultores de ciberseguridad de la organización bajo auditoría en los casos donde se ha encargado hacer una auditoría sin comunicarlo a ciertas partes. Esta función contribuye a mantener un perfil discreto durante la realización de las auditorías, permitiendo un análisis más efectivo de la seguridad de la red sin despertar alertas innecesarias.

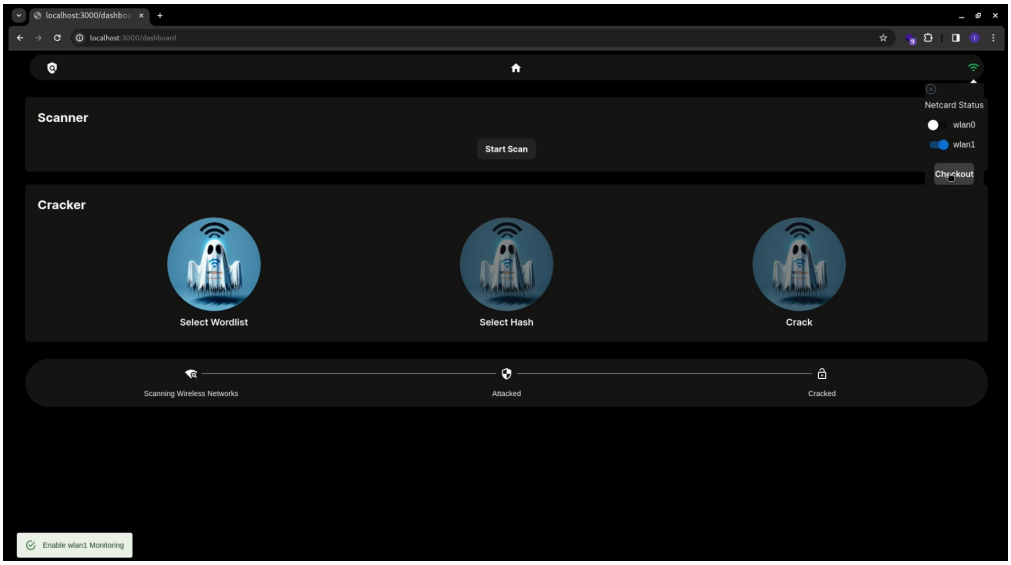

Figura 4.1: Selección y activación de modo monitor.

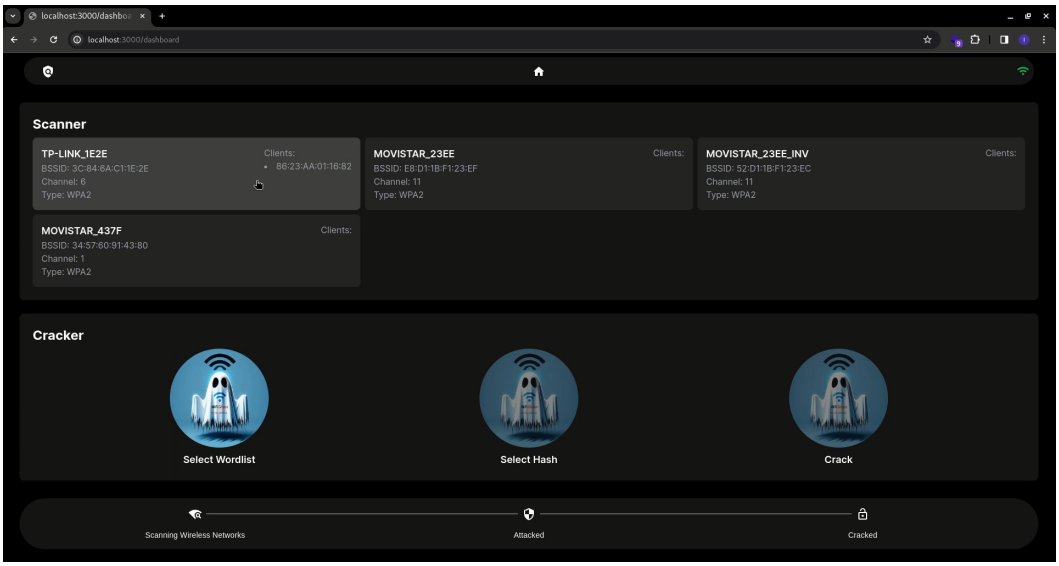

Figura 4.2: Monitorización de redes disponibles.

4.2. Ataques

`WiFiGhost` ofrece una variedad de ataques controlados en redes WiFi. Estos ataques están diseñados para evaluar la seguridad de la red y detectar posibles vulnerabilidades

en función del tipo de cifrado. A continuación, se describe cada ataque que soporta la herramienta de manera particular.

4.2.1. WPA/WPA2 con clientes

En la figura 4.3 se muestra un ejemplo de la herramienta donde muestra los tipos de ataque disponibles para redes WPA/WPA2 con clientes conectados.

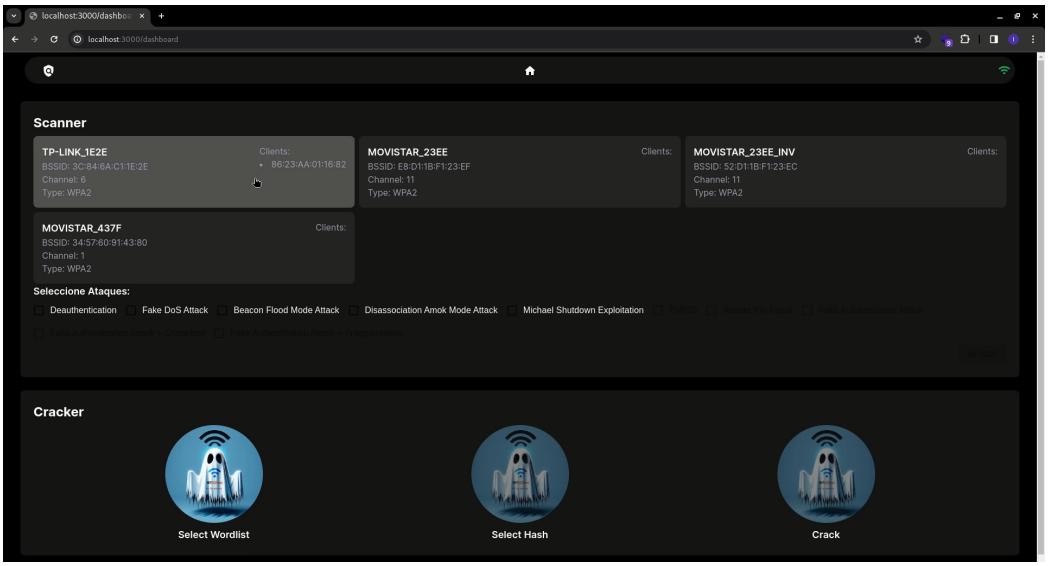

Figura 4.3: Selección de ataques disponibles para redes WPA/WPA2 con clientes conectados.

- **Desautenticación:** Se aprovecha la provisión del protocolo IEEE 802.11 (WiFi) para enviar un marco de desautenticación al punto de acceso inalámbrico (véase figura 4.4). Este marco puede dirigirse específicamente a la dirección MAC del cliente que se pretende desconectar o emitirse en modo *broadcast*, asegurando el éxito del ataque [2].

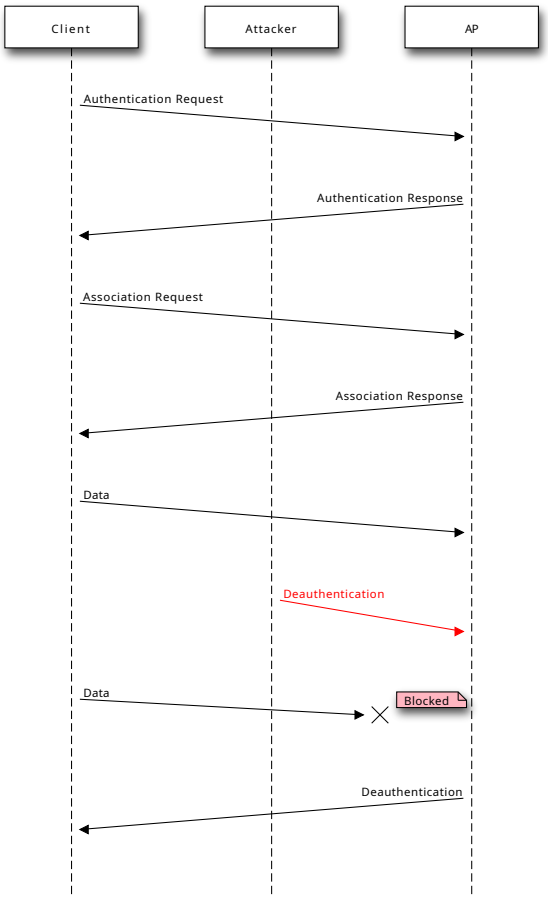

Figura 4.4: Proceso de desautenticación.
Fuente: `https://en.wikipedia.org/wiki/Wi-Fi_deauthentication_attack`

Al concluir el ataque, el dispositivo objetivo se reconectará automáticamente al punto de acceso, sin intervención adicional por parte del atacante. Este proceso automático es el momento en el cual se genera el *handshake*.

- ***Fake DoS Attack:*** Se emplea el modo de ataque "*Authentication DoS Mode*" de la herramienta mdk3 [18], el cual se encarga de asociar a miles de clientes al punto de acceso objetivo. Este proceso tiene como consecuencia inmediata la ralentización y temporal inoperabilidad de la red, llegando a expulsar a clientes distantes o con débil señal WiFi [2, 7].

- ***Beacon Flood Mode Attack:*** Los *beacons* son paquetes que contienen información crucial sobre el punto de acceso, como el canal, el tipo de cifrado y el nombre de la red. Estos paquetes se transmiten en claro para permitir que tarjetas de red

y otros dispositivos recojan la información necesaria para la conexión [1, 2, 6, 7]. A través de la herramienta mdk3 [18], se realiza un ataque denominado "*Beacon Flood Attack*", generando múltiples paquetes *beacon* con información falsa. Este ataque busca generar numerosos puntos de acceso en el mismo canal que el punto de acceso objetivo, perturbando el espectro de onda y dejando la red no operativa e invisible para los usuarios [1].

- ***Disassociation Amok Mode Attack:*** Aunque comparte similitudes con un ataque de desautenticación dirigido, el modo de operación de tipo lista denegada/lista permitida en mdk3 [18] posibilita especificar qué clientes deben permanecer autenticados al punto de acceso al agregarlos a una lista permitida y, en contraste, cuáles deben ser desautenticados al añadirlos a una lista denegada [6, 7].

- ***Michael Shutdown Explotation:*** Este ataque puede desactivar puntos de acceso que utilizan cifrado *TKIP* y extensiones de calidad de servicio (QoS) con 1 paquete de datos QoS esnifado y 2 paquetes de datos QoS inyectados [6, 7].

4.2.2. *WPA/WPA2 sin clientes:*

En la figura 4.5 se muestra un ejemplo de la herramienta donde muestra los tipos de ataque disponibles para redes WPA/WPA2 sin clientes conectados.

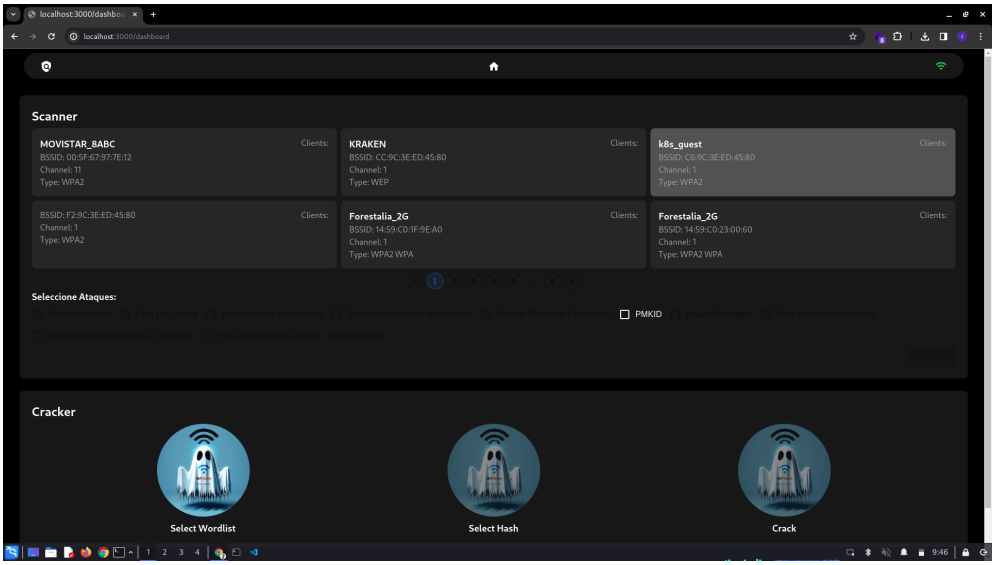

Figura 4.5: Selección de ataques disponibles para redes WPA/WPA2 sin clientes conectados.

- ***Clientless PKMID Attack:*** Este ataque brinda la capacidad de vulnerar la seguridad de *WPA* y *WPA2* mediante el *Pairwise Master Key Identifier* o *PMKID*,

una característica de *roaming* habilitada en muchos dispositivos. En contraste con ataques existentes, la principal diferencia radica en que en este ataque no es necesario capturar un *EAPOL* o un saludo de 4-vías, como en casos anteriores. La nueva técnica se lleva a cabo con el *RSN IE* (Elemento de Información de Red Robusta) de una simple trama *EAPOL* [2].

4.2.3. *WPS*

En la figura 4.6 se muestra un ejemplo de la herramienta donde muestra los tipos de ataque disponibles para redes WPS.

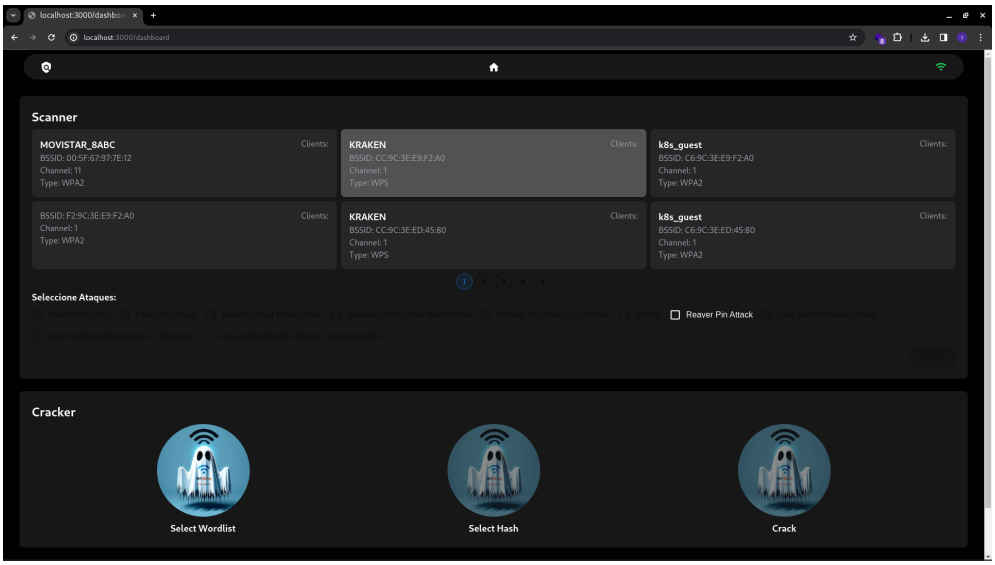

Figura 4.6: Selección de ataques disponibles para redes WPS.

- ***Reaver Pin Attack:*** El Sistema de configuración protegida de WiFi (del inglés, *WiFi Protected Setup* o WPS), comúnmente pasado por alto en *routers* y puntos de acceso WiFi, es una función conveniente que facilita la configuración de dispositivos clientes con una red inalámbrica al presionar simultáneamente un botón en el punto de acceso y otro en el dispositivo cliente. Aunque aparenta ser una característica ingeniosa para que usuarios menos experimentados establezcan rápidamente conexiones seguras, su seguridad parece relativa al requerir acceso físico al hardware. A pesar de que algunos dispositivos más recientes integran protección contra estos ataques, el exploit *WPS* de *Reaver* [26] sigue siendo efectivo en muchas redes. Es crucial señalar que, en este caso, el sistema vulnerable es *WPS*, no *WPA*. Es decir, redes *WPA/WPA2* con *WPS* desactivado serán inmunes a este tipo de ataque [1,2].

4.2.4. *WEP*

En la figura 4.7 se muestra un ejemplo de la herramienta donde muestra los tipos de ataque disponibles para redes WEP.

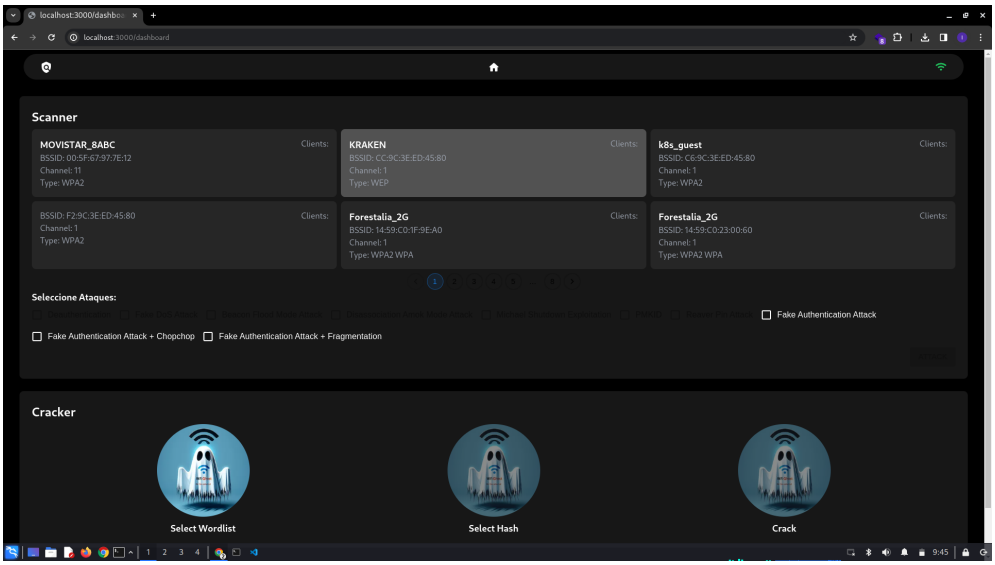

Figura 4.7: Selección de ataques disponibles para redes WEP.

- **Ataque de Autenticación Falsa:** La realización del ataque de autenticación falsa posibilita la ejecución de ambos tipos de autenticación *WEP* y la asociación con el punto de acceso. Este procedimiento resulta útil cuando se requiere una dirección *MAC* asociada (un cliente) en ataques y no hay clientes previamente asociados. Es importante destacar que este ataque no genera paquetes *ARP* [1, 7].

- **Ataque de Autenticación Falsa + Chopchop:** Cuando este ataque tiene éxito, puede desencriptar un paquete de datos *WEP* sin necesidad de conocer la clave [1, 2, 6, 7].

- **Ataque de Autenticación Falsa + Fragmentación:** El objetivo no es recuperar la clave *WEP* en sí, sino simplemente obtener la semilla del algoritmo pseudo-aleatorio utilizado por WEP (del inglés, *Pseudo-Random Generation Algorithm*, PRGA). Posteriormente, el PRGA se utiliza para generar paquetes con contenido falsificado que, a su vez, se emplean en varios ataques de inyección. Es esencial recibir al menos un paquete de datos del punto de acceso para iniciar el ataque. El programa extrae una pequeña cantidad de material de claves del paquete y luego intenta enviar paquetes *ARP* y/o *LLC* con contenido conocido al punto de acceso [1, 17].

4.3. Obtención de la contraseña

`WiFiGhost` ofrece capacidades avanzadas de obtención de contraseñas en redes WiFi, aprovechando eficientemente las tablas arcoiris. Esta característica se especializa en la evaluación de la fortaleza de las contraseñas, empleando diccionarios para identificar y resaltar posibles debilidades en la seguridad de la red.

Para el caso de paquetes PMKID, el *crackeo* del *hash* obtenido se realiza mediante *Hashcat* , que hace uso de la GPU, en caso de tener una, para calcular las contraseñas [24].

La figura 4.8 muestra el estado inicial de la sección de *crackeo*, en la figura 4.9 se muestra el estado final después de haber llevado a cabo el proceso de obtención de contraseña.

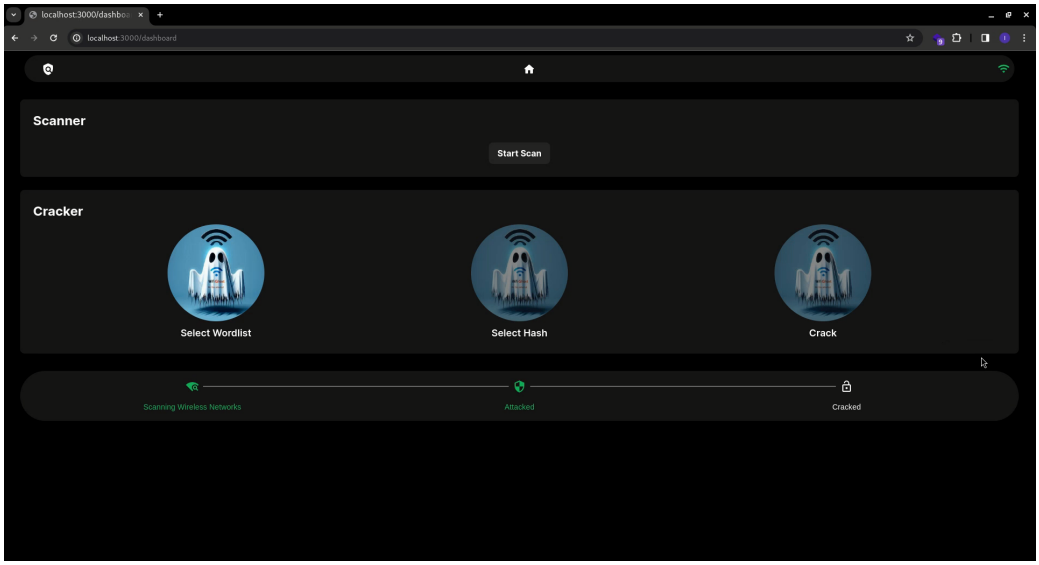

Figura 4.8: Estado inicial *Cracking*.

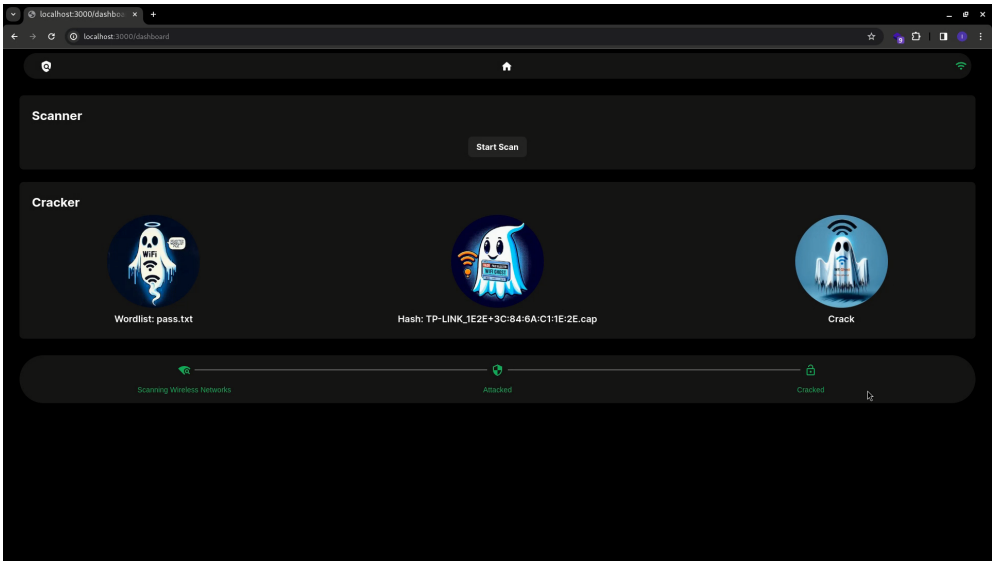

Figura 4.9: Estado final *Cracking*.

4.4. Gestión de archivos

La gestión de archivos en `WiFiGhost` facilita la administración de redes falsas utilizados en el *Beacon Flood Mode Attack*, así como los diccionarios empleados en el descifrado de contraseñas. Los usuarios pueden cargar y eliminar ambos. Todo esto se muestra en la figura 4.10

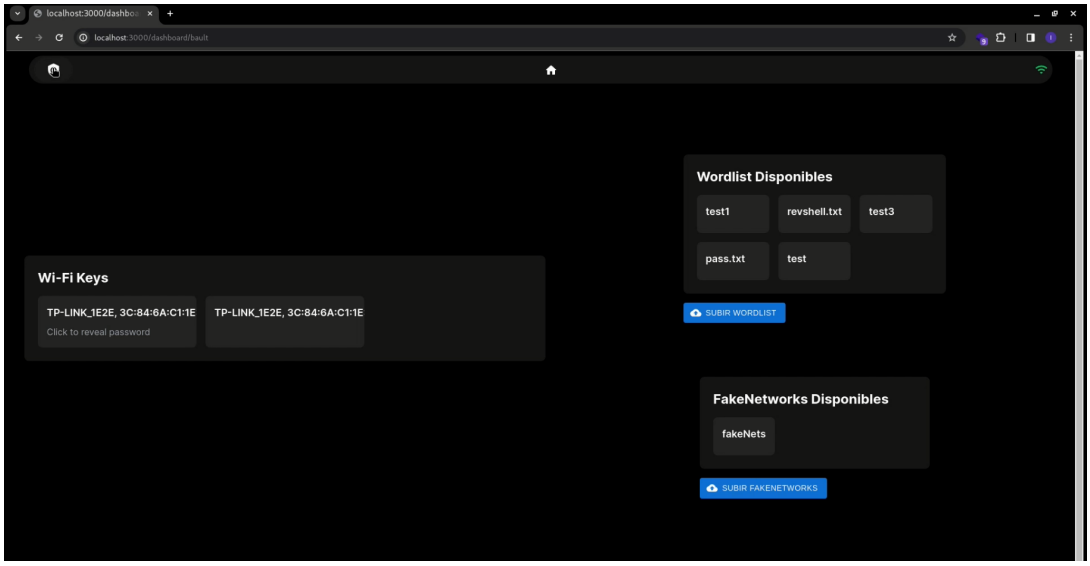

Figura 4.10: Sobre la gestión de archivos.

4.4.1. Almacenamiento de *handshakes* y contraseñas:

`WiFiGhost` almacena automáticamente los *handshakes* capturados y las contraseñas asociadas en su formato identificativo de "*ESSID,BSSID*". Además, se proporciona una vista detallada de las contraseñas correspondientes, brindando una gestión eficiente de los datos recopilados durante las auditorías WiFi.

Capítulo 5

Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones principales, destacando las bondades e implicaciones. Además, se consideran aquí futuras mejoras y extensiones a `WiFiGhost`.

5.1. Conclusiones

`WiFiGhost` es una herramienta eficaz y completa en el ámbito de las auditorías de ciberseguridad WiFi. La interfaz web de `WiFiGhost` proporciona una experiencia de usuario intuitiva y accesible, y mejora significativamente la comprensión y el análisis de los resultados por parte de los usuarios. Otro aspecto notable es la automatización y eficiencia que ofrece `WiFiGhost`. La herramienta automatiza tareas repetitivas, lo cual acelera significativamente el proceso de auditoría.

5.2. Trabajo futuro

Se podría explorar la incorporación de nuevas técnicas de auditoría, la optimización de la velocidad de ejecución y la mejora continua de la interfaz de usuario. Como trabajo futuro se plantean diferentes líneas de acción:

- Extender las técnicas de ataque con diversas técnicas de ataque a puntos de acceso inalámbrico que buscan comprometer la seguridad de la red, como los ataques de suplantación. Entre ellas se encuentra el *Evil Twin*, una técnica de suplantación de puntos de acceso donde el atacante crea una red falsa que imita a una legítima para engañar a los usuarios y hacer que se conecten a ella, exponiéndolos a riesgos como la interceptación de datos y el robo de credenciales [1, 27].
- Extender `WiFiGhost` a otros modos de WPA2, como WPA2-EAP. Se propone realizar una investigación detallada sobre el funcionamiento de redes WiFi que emplean el protocolo WPA2-EAP (*WiFi Protected Access 2 con Extensible Authentication Protocol*) [1, 20] para su posterior auditoría y automatización. Este enfoque se centra en redes que, en lugar de utilizar claves precompartidas, confían en certificados para la autenticación, siendo un ejemplo prominente el entorno *eduroam*.

Bibliografía

[1] Verdés Castelló, F. “Hacking redes WiFi: Tecnología, Auditorías y Fortificación”. 0xWord. 2020.

[2] OffSec. https://www.offsec.com/courses/pen-210/ [Online, Accedido por última vez 06-Feb-2024]

[3] Fern-Wifi-Cracker. *https://github.com/savio-code/fern-wifi-cracker* [Online, Accedido por última vez 06-Feb-2024]

[4] Wifite. https://github.com/savio-code/fern-wifi-cracker/ [Online, Accedido por última vez 06-Feb-2024]

[5] National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality", SP 800-38C, May 2004.

[6] iPhantasmic. Gihub.https://github.com/iPhantasmic/OSWP?tab=readme-ov-file [Online, Accedido por última vez 06-Feb-2024]

[7] S4vitar. Gihub. https://gist.github.com/s4vitar/3b42532d7d78bafc824fb28a95c8a5eb [Online, Accedido por última vez 06-Feb-2024]

[8] Adrianhuber17. Medium. “How to build a simple real-time application using Flask, React and Socket.io” https://medium.com/@adrianhuber17/how-to-build-a-simple-real-time-application-using-flask-react-and-socket-io-7ec2ce2da977 [Online, Accedido por última vez 06-Feb-2024]

[9] Flask. https://flask.palletsprojects.com/en/3.0.x/ [Online, Accedido por última vez 06-Feb-2024]

[10] Next.js 14. https://nextjs.org/learn [Online, Accedido por última vez 06-Feb-2024]

[11] NIST SP 800-153 Guidelines for Securing Wireless Local Area Networks (WLANs) https://csrc.nist.gov/pubs/sp/800/153/final [Online, Accedido por última vez 20-May-2024]

[12] WEP, WPA, WPA2 y WPA3. https://www.kaspersky.es/resource-center/definitions/wep-vs-wpa [Online, Accedido por última vez 20-May-2024]

[13] RC-4 - SISTEMA DE CIFRADO DE SECRETO COMPARTIDO. https://www.ccn-cert.cni.es/publico/seriesCCN-STIC/series/400-Guias_Generales/401-glosario_abreviaturas/index.html?n=756.html [Online, Accedido por última vez 20-May-2024]

[14] FIPS PUB 197, Advanced Encryption Standard (AES), National Institute of Standards and Technology, U.S. Department of Commerce, November 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[15] Funcionamiento del protocolo de seguridad TKIP. https://abcxperts.com/como-funciona-el-protocolo-de-seguridad-tkip/ [Online, Accedido por última vez 20-May-2024]

[16] Key Reinstallation Attacks. https://www.krackattacks.com/ [Online, Accedido por última vez 20-May-2024]

[17] Aircrack-ng. https://www.aircrack-ng.org/ [Online, Accedido por última vez 20-May-2024]

[18] MDK3. https://www.kali.org/tools/mdk3/ [Online, Accedido por última vez 20-May-2024]

[19] HCXDumpTool. https://github.com/ZerBea/hcxdumptool [Online, Accedido por última vez 20-May-2024]

[20] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., .ExtensibleAuthentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, http://www.rfc-editor.org/info/rfc3748

[21] Pyrit. https://github.com/JPaulMora/Pyrit [Online, Accedido por última vez 20-May-2024]

[22] UML. https://www.uml.org/ [Online, Accedido por última vez 20-May-2024]

[23] GNU GPLv3. https://www.gnu.org/licenses/gpl-3.0.html [Online, Accedido por última vez 10-Jun-2024]

[24] Hashcat. https://hashcat.net/hashcat/ [Online, Accedido por última vez 10-Jun-2024]

[25] WiFiGhost. https://github.com/p3n4x0/WiFiGhost [Online, Accedido por última vez 10-Jun-2024]

[26] Reaver. https://github.com/p3n4x0/WiFiGhost [Online, Accedido por última vez 10-Jun-2024]

[27] C. Modi, V.;Parekh. Detection of Rogue Access Point to Prevent Evil Twin Attack in Wireless Network. International Journal of Engineering Research and Technology, 6(4), 2014

Apéndice A

Horas de Trabajo

En este capítulo se detallan las horas de trabajo invertidas en las diversas tareas realizadas a lo largo del proyecto.En la Figura A.1 se muestra una gráfica detallada que representa el tiempo invertido en cada tarea específica. La Figura A.2 contiene un diagrama de Gantt que ilustra la planificación temporal de este trabajo.

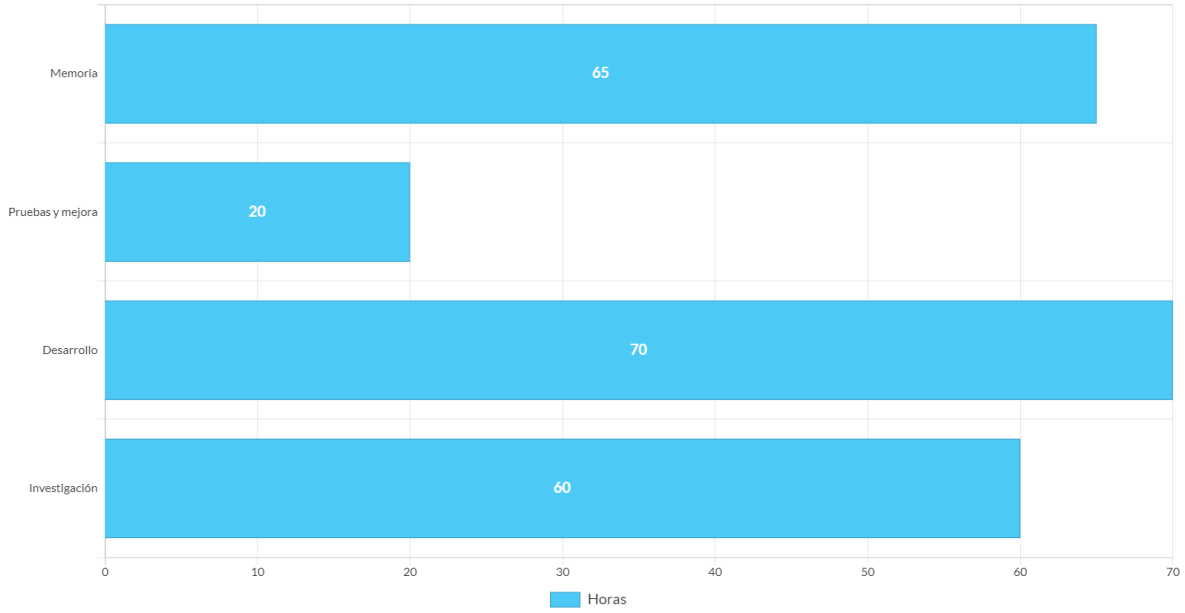

Figura A.1: Desglose de horas empleadas por tarea.

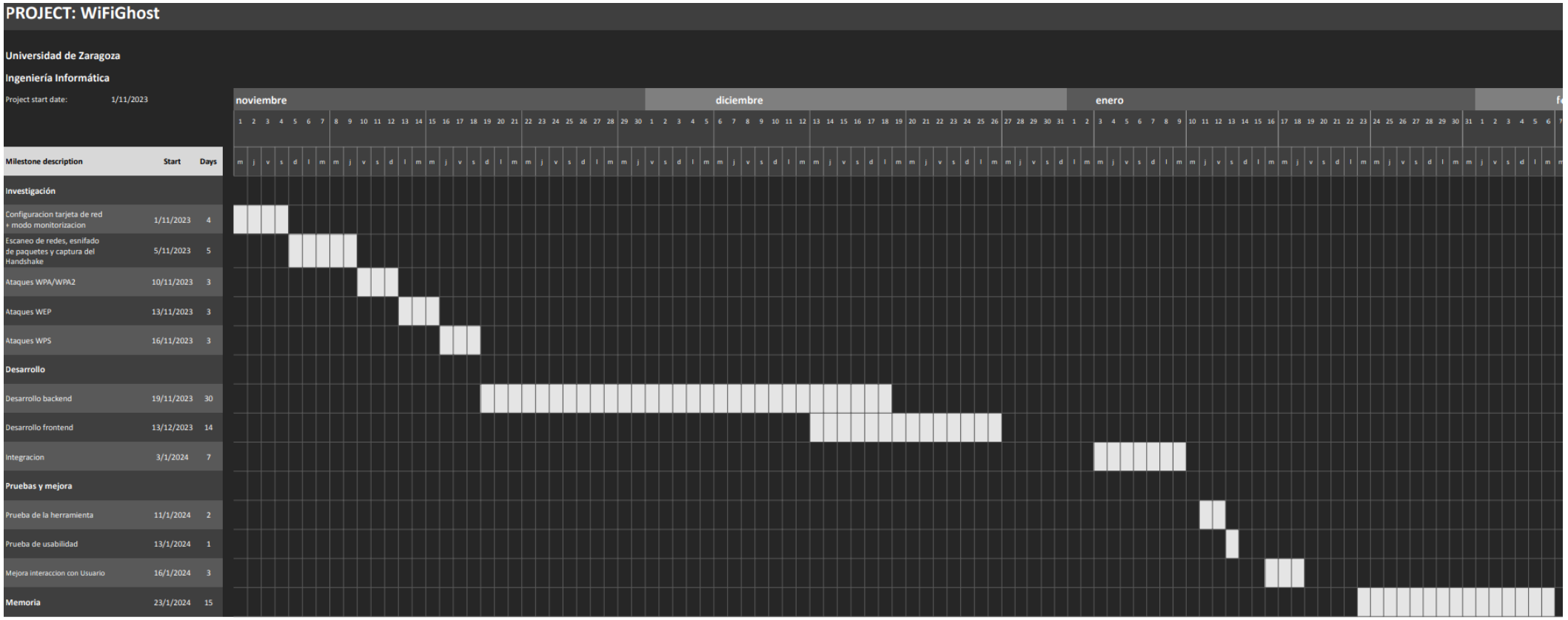

Figura A.2: Diagrama de Gantt.