Escola Politècnica Superior de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO DE FIN DE CARRERA

TÍTULO DEL TFC: Criptoanálisis práctico de WEP y WPA sobre WLAN 802.11

TITULACIÓN: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica

AUTOR: Carlos Hernández Chacón

DIRECTOR: Juan Hernández Serrano

FECHA: 12/11/2009

Título: Criptoanálisis práctico de WEP y WPA sobre WLAN 802.11

Autor: Carlos Hernández Chacón

Director: Juan Hernández Serrano

Fecha: 12/11/2009

Resumen

A día de hoy, es impensable imaginarse trabajadores o usuarios domésticos que no tengan conexión Wi-Fi en sus portátiles u ordenadores de sobremesa. De hecho, los ISP españoles en la actualidad proporcionan a sus clientes routers Wi-Fi por defecto. Estas redes tienen muchas ventajas a nivel de infraestructura, pero su despliegue presenta un grave inconveniente: la **seguridad**. En las comunicaciones inalámbricas, a diferencia de las cableadas, el medio de transmisión es accesible por cualquier atacante con una antena *wireless*, lo que facilita enormemente la interceptación o escucha de los datos que circulan por la red. Para evitar este problema nacieron los protocolos de seguridad WEP y más tarde WPA y 802.11i.

El objetivo de este TFC es realizar una auditoría de las redes 802.11 analizando sus vulnerabilidades y los defectos de los protocolos existentes.

Para conseguirlo, se realizan diferentes ataques activos y se plantean diferentes formas de evitarlos. Además, se realiza un estudio del despliegue 802.11 en un barrio de la población de Gavà, utilizando la técnica de *wardriving* para obtener datos estadísticos de los tipos de seguridad que implementan los AP (Puntos de Acceso).

El programa utilizado para la auditoria de redes es *aircrack-ng*, un paquete de software libre que dispone de varias herramientas: *airodump-ng* para la escucha o "*sniffing*" de redes, *aireplay-ng* para la inyección de paquetes, y el propio *aircrack-ng* para descifrar claves mediante fuerza bruta. También se ha usado InSSIDer para una recopilación más exhaustiva de datos acerca de las redes disponibles. La información recopilada ha permitido un mejor tratamiento estadístico y la evaluación de la seguridad inalámbrica en la zona.

Según los estudios realizados, se ha demostrado que el protocolo WEP es vulnerable, al igual que WPA si lo utilizamos con la clave que implementa el fabricante por defecto. La estadística también nos dice que la mayoría de usuarios de redes Wi-Fi utilizan WEP y muchos WPA con claves predefinidas. Con los resultados obtenidos tenemos la necesidad de concienciar a todos los usuarios de la correcta utilización de los protocolos de seguridad.

Title: Practical Cryptanalysis of WEP and WPA on 802.11 WLAN

Author: Carlos Hernández Chacón

Director: Juan Hernández Serrano

Date: November, 12th 2009

Overview

Nowadays, it is hard to find any workers or home users with no Wi-Fi connection at their laptop or desktop computers. In fact, by default, Spanish ISPs provide to their customers with Wi-Fi routers. Wireless networks have many advantages at the infrastructure level, but their deployment poses a serious drawback: **security**. In wireless communications, as opposed to wired ones, the transmission medium is accessible by any attacker with a wireless antenna, which greatly facilitates eavesdropping data flowing through the network. In order to avoid such risk, new security protocols like WEP, WPA and 802.11i have arisen.

The target of this TFC is conducting an audit of 802.11 networks by analyzing the vulnerabilities and weaknesses of existing security protocols specific to such networks.

To achieve this, we have implemented several active attacks and we have discussed or recommended different ways of solving them. Furthermore, a study is made of 802.11 deployment in a neighborhood of the city of Gavà. For this study, we have collected data related to the Wi-Fi access points within the neighborhood by means of wardriving techniques.

The program used for network auditing is aircrack-ng, a free software package with several tools: airodump-ng to listen or "sniffing" networks, aireplay-ng to inject packets, and aircrack-ng to crack keywords by brute force. InSSIDer has also been used to compile more information about available networks. The information gathered has enabled a better statistical treatment and evaluation of wireless security in the area evaluated.

According to studies, we conclude that WEP is vulnerable, and WPA is vulnerable as well if we use the keyword that indicates the default manufacturer. Statistics also tell us that most users of Wi-Fi networks use WEP and WPA with many default keywords. With the obtained results we need to make all users aware of the important use of security protocols.

Quisiera dedicar este TFC a todas las personas que han confiado en mí, en especial a mis padres, a mi hermano y a mi pareja, Aida. Sin su apoyo incondicional me hubiese resultado imposible sacarlo adelante. También quisiera agradecer el apoyo de compañeros como Cocos, Tonino, Sebas, Vanessa, entre otros. Y por supuesto a mi tutor Juan Hernández.

ÍNDICE

INTRODUCCIÓN 1

CAPÍTULO 1. ESTÁNDARES 802.11, WEP, WPA Y 802.11I 3

1.1. Breve historia de las redes inalámbricas de área local 3

1.2. Normativa IEEE 802.11 4
1.2.1. 802.11 Legacy 4
1.2.2. 802.11a – 54 Mbit/s, 5 Ghz estándar 4
1.2.3. 802.11b - Mejoras a 802.11 para soportar 5,5 y 11Mbit/s 4
1.2.4. 802.11g - 54 Mbit/s, estándar 2.4 GHz (compatible 802.11b) 5
1.2.5. 802.11n - Mejoras de rendimiento usando MIMO (Múltiples entradas – múltiples antenas de salida) y asignación de canales por región. 5
1.2.6. Tabla resumen de los estándares 802.11 6
1.2.7. Otros estándares 6
1.2.8. Estándares de seguridad 7

1.3. WEP (Wired Equivalent Privacy) 8
1.3.1. Cifrado WEP 8
1.3.2. Autenticación 10

1.4. WPA versión 1 13
1.4.1. Cifrado WPA 13
1.4.2. Autenticación WPA 14

1.5. WPA versión 2 y 802.11i 16
1.5.1. Cifrado en WPA v2 y 802.11i 17
1.5.2. Autenticación en WPA v2 y 802.11i 17

CAPÍTULO 2. VULNERABILIDADES DE WEP Y WPA 18

2.1 Defectos del protocolo WEP 18
2.1.1 Debilidad del vector de inicialización 18
2.1.2 Ataques pasivos 18
2.1.3 Ataques activos 19
2.1.4 Identificación de secuencias pseudoaleatorias iguales 19
2.1.5 Vulnerabilidad RC4 19

2.2 Debilidades del protocolo WPA 20

CAPÍTULO 3. ATACANDO 802.11 21

3.1 Suite aircrack-ng 21
3.1.1 Introducción 21
3.1.2 Airodump-ng 21
3.1.3 Aireplay-ng 24
3.1.4 Aircrack-ng 33

3.2 Caso práctico de ataque a WPA mediante aircrack-ng 35

CAPÍTULO 4. WARDRIVING 38

4.1 Definición wardriving 38

4.2 Analizando los datos de las redes Wi-Fi 38
4.2.1 Localización y equipo utilizado 38
4.2.2 Captura de datos durante el *wardriving* 39
4.2.3 Estadísticas a partir de los datos recopilados 40
4.2.4 Recomendaciones de seguridad 44

CAPÍTULO 5. CONCLUSIONES 45

5.1 Conclusiones 45

5.2 Líneas futuras 46

5.3 Estudio de ambientalización 46

REFERENCIAS 48

ANEXOS 49

Anexo 1 : Capturas de InSSIDer 49

INTRODUCCIÓN

La necesidad de movilidad, inmediatez e interconexión (entre dispositivos), la especificidad de algunos edificios (hospitales, hoteles, museos, etc.) y la imposibilidad de su cableado, han permitido que en los últimos cuatro años haya crecido exponencialmente el uso de redes Wi-Fi. Hecho al que ha contribuido también el cambio de estrategia experimentado por los fabricantes de ordenadores al integrar soluciones inalámbricas (infrarrojos, bluetooth, 3G y tarjetas Wi-Fi) en sus equipos, y ha empujado al mismo tiempo a los fabricantes de routers y módems ADSL a dotar a sus dispositivos de las correspondientes conexiones inalámbricas.

Las redes Wi-Fi presentan multitud de ventajas en términos de infraestructura: como por ejemplo la no necesidad de utilizar cables, la compatibilidad de todos los dispositivos y el reducido gasto de infraestructura para dotar de acceso a nuevos usuarios. Sin embargo, a pesar de las ventajas, el medio inalámbrico es fácilmente accesible por cualquier atacante, que sólo necesita disponer de una antena inalámbrica de la misma tecnología y a la misma frecuencia. En contrapartida, en las redes cableadas sería necesario “pinchar” el cable para atacarlas.

Hoy en día con la cantidad de información confidencial e importante que circula por la red: e-mails privados, transacciones bancarias, operaciones con tarjetas de crédito, historiales médicos y muchas otras, los usuarios de redes inalámbricas deben ser conscientes del papel que juega la **seguridad**. Para evitar que la información interceptada pueda ser interpretada por el atacante hay que dotar de seguridad a nivel de enlace (capa 2) a las comunicaciones. Con el objetivo de dar seguridad a la capa de 802.11, aparecen los protocolos de seguridad para Wi-Fi, inicialmente WEP y más tarde WPA y 802.11i.

Este TFC está encaminado al estudio de los protocolos de seguridad aplicados en la actualidad a las redes Wi-Fi. Se trata de un proyecto para evaluar las vulnerabilidades y defectos de los protocolos WEP, WPA y 802.11i, además se pretende analizar si estas vulnerabilidades están presentes en el despliegue Wi-Fi actual mediante la técnica del *wardriving*. De estar presentes, se presentan diferentes formas de evitarlas.

Para evaluar esta problemática lo más eficiente es poner a prueba la integridad de los protocolos WEP y WPA utilizando un programa como *aircrack-ng*. Esta suite Open Source, compuesta por diversas herramientas: airodump-ng para la escucha o “*sniffing*” de redes, aireplay-ng para la inyección de paquetes, y el propio *aircrack-ng* para descifrar claves mediante fuerza bruta, nos permite realizar una completa auditoria realizando diferentes tipos de ataques activos.

Este TFC se estructura como se detalla a continuación. El primer capítulo trata de introducir la tecnología utilizada, Wi-Fi. Se describen cronológicamente las características de los protocolos 802.11, entrando más en detalle en WEP, WPA y 802.11i, explicando autenticación y cifrado de los mismos. En el capítulo 2 se muestran las debilidades de WEP y WPA para así luego abordar

el tercer capítulo, que trata sobre los diferentes ataques activos a dichos protocolos. Una vez vulnerados, en el capítulo 4 se realiza un estudio mediante la técnica del *wardriving* para evaluar el tipo de seguridad utilizada por una parte de la población de Gavà, pudiendo luego ser extrapolado a nivel local. Finalmente en el último capítulo se extraen las conclusiones junto con posibles líneas futuras y un breve estudio medioambiental.

CAPÍTULO 1. Estándares 802.11, WEP, WPA y 802.11i

Para introducir y comprender el título de este TFC: “Criptoanálisis práctico de WEP y WPA sobre WLAN 802.11”, a continuación explicamos, a grandes rasgos, la historia de las redes inalámbricas de área local, la evolución de los diferentes protocolos de la normativa 802.11 y los protocolos WEP, WPA y 802.11i.

1.1. Breve historia de las redes inalámbricas de área local

En junio de 1997 se presentó el estándar para redes inalámbricas que se denominó IEEE 802.11, que especificaba una frecuencia de operación de 2,4 GHz con niveles de transmisión de datos de 1 a 2 Mbps. La norma IEEE 802.11 se diseñó para sustituir el equivalente a las capas físicas y MAC de la norma 802.3 (Ethernet): La distinción básica entre redes Ethernet y redes Wi-Fi radica en la forma en que se transmiten los paquetes de datos: las primeras por un medio guiado y las segundas por un medio inalámbrico.

La Alianza de Compatibilidad Ethernet Inalámbrica (*Wireless Ethernet Compatibility Alliance - WECA*), fue creada en 1999 entre otros por 3Com, Intersil, Lucent, Nokia, etc., siendo su principal objetivo normalizar la compatibilidad entre equipos (Interoperatividad y certificación), y fomentar el uso de la tecnología inalámbrica. En Abril del 2000 la WECA cumple uno de sus principales objetivos: la certificación de la interoperatividad de equipos según la norma IEEE 802.11b, que definía el estándar de comunicación para redes de área local inalámbricas (*Wireless Local Area Network – WLAN*). En el año 2003, la WECA constaba de más de 150 miembros, y en ese mismo año pasó a denominarse Wi-Fi Alliance, nombre que conserva hasta la actualidad.

Pese al pensamiento generalizado, Wi-Fi no es abreviatura de Wireless Fidelity. No es un acrónimo. No tiene significado. El término “Wi-Fi” y el logotipo (**Fig. 1.1**) fueron creados por la agencia de publicidad Interbrand Corporation a petición de la WECA, utilizando una palabra que fuese corta, comercial y fácil de recordar (elegida entre 10 posibles candidatas propuestas por Interbrand) y un logo al estilo ying-yang, que es un concepto basado en la dualidad de todo lo existente en el universo.

Fig. 1.1 Logotipo Wi-Fi

A nivel de seguridad la Wi-Fi Alliance después de mucho trabajo, primero certificó el estándar 802.1x basado en servidores RADIUS y más tarde el estándar 802.11i basado en WPA y WPA2.

1.2. Normativa IEEE 802.11

Existen diversos estándares de la norma IEEE 802.11 aprobados. A continuación se resumen los que se han considerado más relevantes para el desarrollo de este TFC:

1.2.1. 802.11 Legacy

La versión original del estándar IEEE 802.11 [1] publicada en 1997 define el uso de CSMA/CA (Múltiple acceso por detección de portadora evitando colisiones) como método de acceso al medio y especifica dos velocidades de transmisión teóricas de 1 y 2 Mbit/s que se transmiten por señales infrarrojas (IR), las cuales siguen siendo parte del estándar, si bien no hay implementaciones disponibles. Todo esto a una frecuencia de 2,4GHz.

1.2.2. 802.11a – 54 Mbit/s, 5 Ghz estándar

La última corrección del estándar 802.11a [2] fue ratificada en 1999. Este estándar utiliza los mismos protocolos de base que el estándar original, opera en la banda de 5 Ghz y utiliza 52 subportadoras OFDM (*Orthogonal Frequency-Division Multiplexing* – Multiplexación por Division de Frecuencias Ortogonales) la velocidad máxima es de 54 Mbit/s, lo que lo hace en la práctica un estándar muy útil para redes inalámbricas con velocidades reales de aproximadamente 20 Mbit/s. Además 802.11a tiene 12 canales no solapados, 8 para red inalámbrica y 4 para conexiones punto a punto. Una desventaja de este estándar es que no puede interoperar con equipos del estándar 802.11b, que detallamos más adelante, a no ser que se disponga de equipos que implementen ambos estándares.

Puesto que la banda de 2.4 Ghz tiene un gran uso (es la misma banda usada por los teléfonos inalámbricos y los hornos de microondas, entre otros aparatos), el utilizar la banda de 5 GHz representa una ventaja del estándar 802.11a, dado que se producen menos interferencias. Sin embargo, la utilización de esta banda también tiene sus desventajas, ya que restringe el uso de los equipos 802.11a a únicamente a aquellos que tengan visión directa entre ellos, lo que conlleva la necesidad de la instalación de un mayor número de puntos de acceso, esto significa también que los equipos que trabajan con este estándar no pueden penetrar tan lejos como los del estándar 802.11b dado que sus ondas son más fácilmente absorbidas.

1.2.3. 802.11b - Mejoras a 802.11 para soportar 5,5 y 11Mbit/s

La revisión 802.11b del estándar original fue ratificada en 1999. 802.11b tiene una velocidad máxima de transmisión de 11 Mbit/s y utiliza el mismo método de acceso CSMA/CA definido en el estándar original. El estándar 802.11b funciona en la banda de 2.4 GHz. Debido al espacio ocupado por la

codificación del protocolo CSMA/CA, en la práctica, la velocidad máxima de transmisión con este estándar es de aproximadamente 5.9 Mbit/s sobre TCP y 7.1 Mbit/s sobre UDP.

1.2.4. 802.11g - 54 Mbit/s, estándar 2.4 GHz (compatible 802.11b)

En junio de 2003, se ratificó un tercer estándar de modulación: 802.11g [3]. Que es la evolución del estándar 802.11b, éste utiliza la banda de 2.4 Ghz (al igual que el estándar 802.11b) pero opera a una velocidad teórica máxima de 54 Mbit/s, que en promedio es de 22.0 Mbit/s de velocidad real de transferencia, similar a la del estándar 802.11a. Es compatible con el estándar b y utiliza las mismas frecuencias. La necesidad de hacer compatibles los dos estándares obligó a dedicar al tema de la compatibilidad buena parte del proceso de diseño del nuevo estándar. Sin embargo, en redes bajo el estándar g la presencia de nodos bajo el estándar b reduce significativamente la velocidad de transmisión, prevaleciendo el estándar b sobre el g.

Los equipos que trabajan bajo el estándar 802.11g llegaron al mercado muy rápidamente, incluso antes de su ratificación que fue el 20 de junio del 2003. Fue una ventaja el que los fabricantes pudieran aprovechar los equipos con estándar b para construir equipos bajo este nuevo estándar.

Actualmente se venden equipos con esta especificación, con potencias de hasta medio vatio, que permite hacer comunicaciones de centenares de metros con una antena adecuada.

1.2.5. 802.11n - Mejoras de rendimiento usando MIMO (Múltiples entradas – múltiples antenas de salida) y asignación de canales por región.

En enero de 2004, la IEEE anunció la formación de un grupo de trabajo para desarrollar una nueva revisión del estándar 802.11. La velocidad real de transmisión podría llegar a los 600 Mbps (lo que significa que las velocidades teóricas de transmisión serían aún mayores), y debería ser hasta 10 veces más rápida que una red bajo los estándares 802.11a y 802.11g, y cerca de 40 veces más rápida que una red bajo el estándar 802.11b. También se espera que el alcance de operación de las redes sea mayor con este nuevo estándar gracias a la tecnología MIMO (*Multiple Input – Multiple Output*), que permite utilizar varios canales a la vez para enviar y recibir datos gracias a la incorporación de varias antenas. El estándar 802.11n ha sido aprobado definitivamente en Septiembre de 2009.

A diferencia de las otras versiones de Wi-Fi, 802.11n puede trabajar en dos bandas de frecuencias: 2,4 GHz (la que emplean 802.11b y 802.11g) y 5 GHz (la que usa 802.11a). Gracias a ello, 802.11n es compatible con dispositivos basados en todas las ediciones anteriores de Wi-Fi. Además, es útil que trabaje en la banda de 5 GHz, ya que está menos congestionada, y con 802.11n permite alcanzar un mayor rendimiento, una vez resueltos los inconvenientes de 802.11a.

1.2.6. Tabla resumen de los estándares 802.11

Una vez descritos los considerados estándares más importantes hasta la fecha, podemos observar en la **Tabla 1.1** las diferencias más relevantes entre ellos:

ESTÁNDARES	AÑO RAFITIFACIÓN	VELOCIDAD TRANSMISIÓN	BANDA DE FRECUENCIA	PRINCIPAL INNOVACIÓN
802.11 Legacy	1997	1 Mbit/s – 2 Mbit/s	2,4 GHz	- CSMA/CA - Señales IR
802.11a	1999	20 Mbit/s – 54 Mbit/s	5 GHz	- Redes inal. - Banda 5 GHz
802.11b	1999	11 Mbit/s	2,4 GHz	- Banda 2,4 GHz - Soporte vel 5,5-11Mbit/s
802.11g	2003	54 Mbit/s	2,4 GHz	- Mayor Vtx - Compatible con 802.11b
802.11n	2004	600 Mbit/s	2,4 GHz y 5 GHz	- Mayor Vtx - Mayor alcance - Compat. 2 frec.

Tabla 1.1 Estándares 802.11

1.2.7. Otros estándares

Además de todos los estándares mencionados anteriormente cabe mencionar que también existen otros, como por ejemplo:

1.2.7.1 802.11e – QoS

El objetivo de este estándar es introducir nuevos mecanismos a nivel de capa MAC para soportar los servicios que requieren garantías de Calidad de Servicio (*Quality of Service - QoS*). Para cumplir con su objetivo IEEE 802.11e [4] introduce un nuevo elemento llamado *Hybrid Coordination Function* (HCF) con dos tipos de acceso:

- EDCA - *Enhanced Distributed Channel Access*: Donde tiene mayor probabilidad de ser enviado el tráfico de alta prioridad, que el tráfico de menor prioridad: Una estación con tráfico de alta prioridad esperara menos para enviar sus paquetes que una estación con paquetes de baja prioridad.

- HCCA - *Hybrid Controlled Channel Access*: Se trata del control de acceso al medio.

1.2.7.2 802.11h – Espectro 802.11a (5 GHz) para compatibilidad europea

La especificación 802.11h es una modificación sobre el estándar 802.11 para WLAN desarrollado por el grupo de trabajo 11 del comité de estándares LAN/MAN del IEEE (IEEE 802) y que se hizo público en octubre de 2003. 802.11h intenta resolver problemas derivados de la coexistencia de las redes 802.11 con sistemas de Radares y Satélite.

1.2.8. Estándares de seguridad

A continuación se introducen los estándares de seguridad de la normativa 802.11, que, para una mejor comprensión de este TFC, se explican más en detalle en las secciones 1.3, 1.4 y 1.5.

1.2.8.1 WEP

(Wired Equivalent Privacy) WEP es un algoritmo de seguridad que sirve para ofrecer protección a las redes inalámbricas, incluido en la primera versión del estándar IEEE 802.11, mantenido sin cambios en las nuevas 802,11a y 802.11b, por la necesidad de garantizar compatibilidad entre los distintos fabricantes. Además es un sistema de encriptación estándar implementado en la MAC y soportado por la mayoría de soluciones inalámbricas como veremos más adelante en una explicación más detallada.

1.2.8.2 802.11i - Mejora de seguridad

El estándar 802.11i elimina muchas de las debilidades y problemas de sus predecesores tanto en lo que se refiere a autenticación de usuarios como a la robustez de los métodos de cifrado. El estándar viene a ratificar las mejoras de seguridad propuestas por la Wi-Fi Alliance en el sistema *Wi-Fi Protected Access* (WPA), que se detalla más adelante.

Las mejoras más importantes en la seguridad radican en su capacidad para trabajar en colaboración con 802.1X y en la incorporación de encriptación *Advanced Encryption Standard* (AES). Aparte de incrementar de manera más que significativa la seguridad de los entornos WLAN, también reduce considerablemente la complejidad y el tiempo de *roaming* de los usuarios de un punto de acceso a otro.

1.3. WEP (Wired Equivalent Privacy)

El sistema WEP (*Wired Equivalent Privacy* - Privacidad Equivalente al Cable), incluido en la primera versión del estándar IEEE 802.11, se utiliza para dotar de privacidad a las redes inalámbricas. La intención principal de WEP es garantizar que los sistemas WLAN dispongan de un nivel de confidencialidad equivalente al de las redes LAN cableadas, mediante el cifrado de los datos que son transportados por las señales de radio. Un propósito secundario de WEP es el de evitar que usuarios no autorizados puedan acceder a las redes WLAN (es decir, proporcionar autenticación). Este propósito secundario no está enunciado de manera explícita en el estándar 802.11, pero se considera una importante característica del algoritmo WEP. El estándar WEP se implementa sobre la capa MAC (*Medium Access Control* – Control de Acceso al Medio) y dota de seguridad a las comunicaciones en redes 802.11 con un coste de recursos muy bajo.

1.3.1. Cifrado WEP

Todo sistema WLAN puede ser sometido a escuchas pasivas por parte de usuarios ajenos a la comunicación. La única forma efectiva de prevenir que alguien pueda comprometer los datos transmitidos consiste en utilizar mecanismos de cifrado. La mayor vulnerabilidad del medio inalámbrico son las escuchas de los datos transmitidos, un término que se conoce en el argot de las telecomunicaciones como *esnifar*.

WEP emplea el algoritmo RC4 de *RSA Data Security*, y es utilizado para cifrar las transmisiones realizadas a través del aire. El estándar define el uso de RC4 con claves semillas (*seeds*) de 64 y/ó 128 bits, de los cuales 24 bits corresponden al vector de inicialización (IV – *Inicialization Vector*) y el resto, 40 ó 104 bits, a la clave secreta compartida entre emisor y receptor. El IV se genera dinámicamente y debe ser diferente para cada trama. El objetivo perseguido con el IV es cifrar con claves diferentes, para impedir que un posible atacante pueda capturar suficiente tráfico cifrado con la misma clave y terminar finalmente deduciendo ésta. Como es lógico, ambos extremos deben conocer tanto la clave como el IV. Lo primero sabemos ya que es conocido puesto que está almacenado en la configuración de cada elemento de red. El IV, en cambio, se genera en un extremo y se envía en claro dentro de la propia trama al otro extremo, por lo que también será conocido.

1.3.1.1 El algoritmo de cifrado y descifrado WEP

A la hora de cifrar los mensajes mediante WEP debemos seguir los pasos que detallamos a continuación:

1. Se calcula el CRC de 32 bits de los datos. Este CRC-32 es el método que propone WEP para garantizar la integridad de los mensajes (ICV, *Integrity Check Value*).

2. Se concatena el ICV al mensaje que queremos enviar. Tal y como podemos observar en la **Fig. 1.2**.

3. Por otra parte se concatena la clave secreta a continuación del IV formado la semilla (*seed)*.

4. El generador pseudoaleatorio de RC4 genera una secuencia de caracteres (*keystream – Secuencia RC4*), de la misma longitud que los bits obtenidos en el punto 2 a partir del *seed*.

5. Se calcula la OR exclusiva (XOR) byte a byte de los caracteres del punto 2 con los del punto 4. El resultado es el mensaje cifrado.

6. Le añadimos el ICV al mensaje cifrado resultante del punto anterior.

7. Finalmente se envía el IV (sin cifrar) y el mensaje cifrado dentro del campo de datos (*frame body*) de la trama IEEE 802.11.

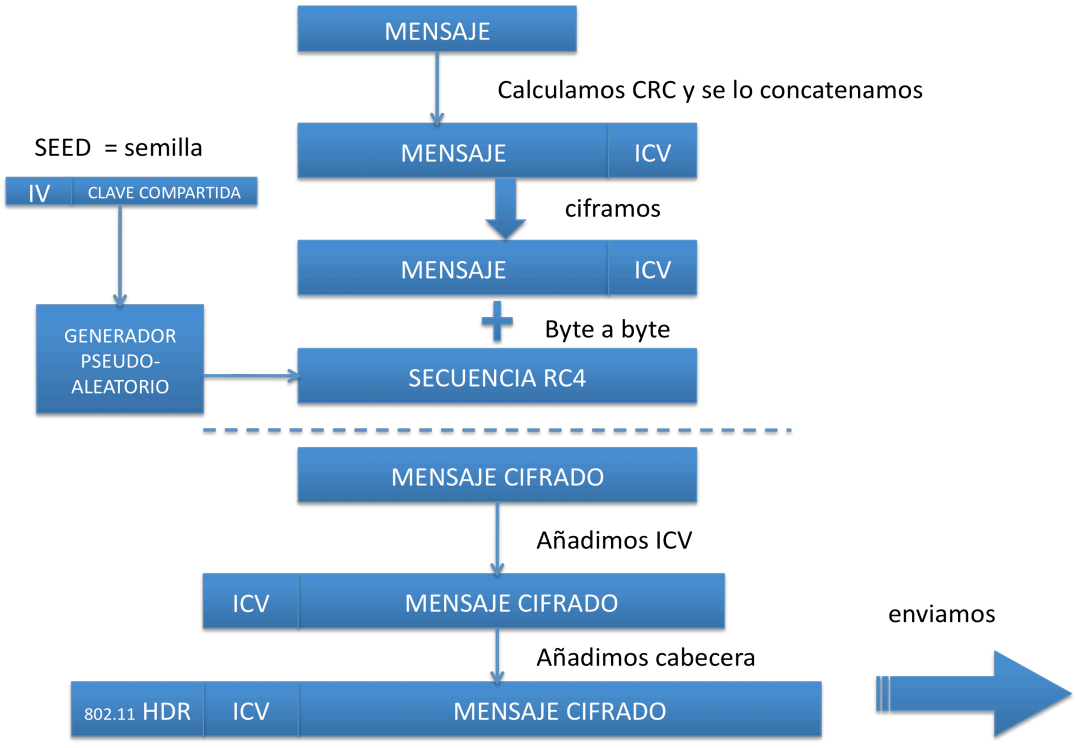


Fig. 1.2 Cifrado WEP

Una vez el receptor recibe el mensaje cifrado, lo que debe hacer es descifrarlo como se muestra en la **Fig. 1.3** y siguiendo los pasos detallados a continuación:

1. Se utiliza el IV y la clave compartida para descifrar el *payload* (carga del mensaje) y el ICV.

2. Se vuelve a calcular el ICV y se compara con el original.

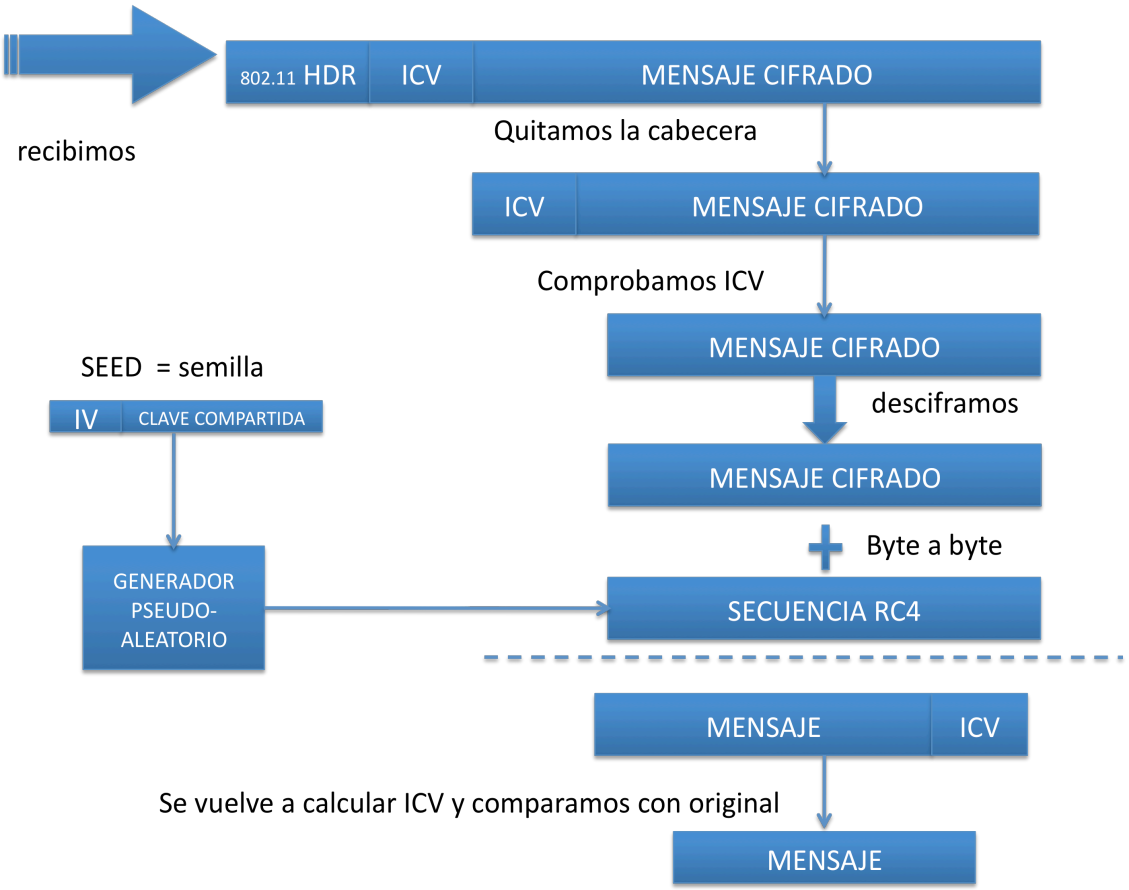


Fig. 1.3 Descifrado WEP

1.3.2. Autenticación

En temas de seguridad y más concretamente en una comunicación, la autenticación es el proceso de verificación de la identidad del remitente. En nuestro caso, WEP proporciona dos tipos de autenticación:

a) Un sistema abierto, en el que todos los usuarios tienen permiso para acceder a la WLAN (*Open Authentication*).

b) Una autenticación mediante clave compartida, que controla el acceso a la WLAN y evita accesos no autorizados a la red (*Shared Key Authentication*). En él se utiliza una clave secreta compartida entre todas las estaciones y puntos de acceso del sistema WLAN.

La autenticación mediante clave compartida es muy insegura, por lo tanto todas las implementaciones de 802.11 utilizan *Open Authentication* por defecto en la cual todos los clientes que inician el proceso de autenticación reciben el OK por parte del AP (ver **Fig. 1.4**).

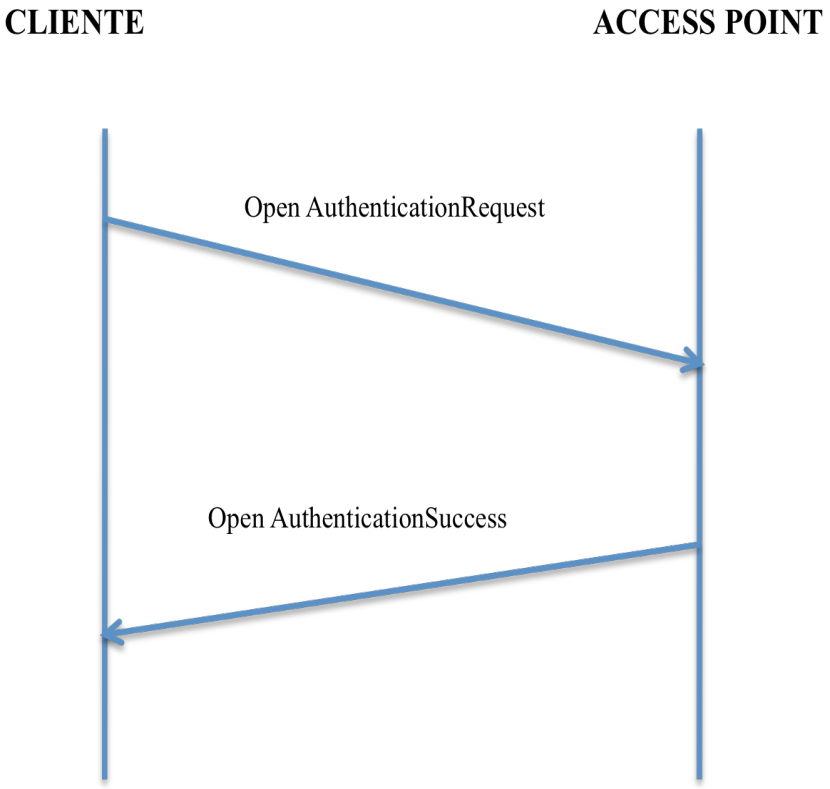


Fig. 1.4 Open Authentication

A continuación mostramos las fases del procedimiento de autenticación mediante clave compartida (ver **Fig. 1.5**):

1. La estación cliente envía una petición de autenticación al Punto de Acceso (*Access Point*).
2. El punto de acceso envía de vuelta un texto modelo: un reto.
3. El cliente tiene que cifrar el texto modelo usando la clave WEP ya configurada, y reenviarlo al Punto de Acceso en otra petición de autenticación.
4. El Punto de Acceso descifra el texto codificado, y lo compara con el

texto modelo que había enviado. Dependiendo del éxito de esta comparación, el Access Point envía una confirmación o una denegación. Después de la autenticación y la asociación, WEP puede ser usado para cifrar los paquetes de datos.

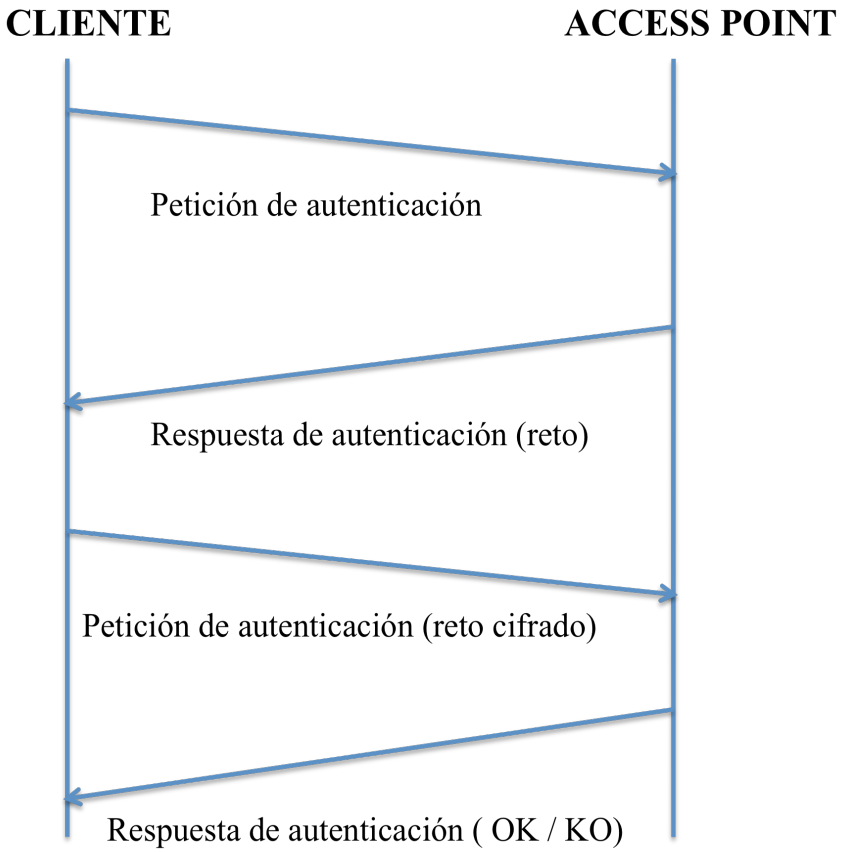


Fig. 1.5 Shared Key Authentication

1.4. WPA versión 1

A continuación explicamos las características de *Wi-Fi Protected Access* (WPA) versión 1 que fue creado por la Wi-Fi Alliance provisionalmente para ocupar el lugar de WEP, mientras 802.11i era finalizado.

Este protocolo fue implementado para corregir las deficiencias del sistema previo WEP y no requería un cambio de hardware. En cambio, como veremos más adelante esto no ocurre con WPA versión 2.

1.4.1. Cifrado WPA

Los datos utilizan el algoritmo RC4 con una clave de 128 bits y un vector de inicialización de 48 bits. Una de las mejoras más sobresalientes sobre su predecesor, WEP, es TKIP (*Temporal Key Integrity Protocol*, o Protocolo de integridad de clave temporal), el cual explicaremos en el siguiente apartado.

Además de proporcionar autenticación y cifrado, WPA proporciona mejor integridad de la carga útil. La verificación de redundancia cíclica (CRC) utilizada en WEP es insegura porque permite alterar la carga útil y actualizar el mensaje de verificación de redundancia cíclica sin necesidad de conocer la clave WEP como demostraremos más adelante en el apartado 2.2. En cambio WPA utiliza un Código de Integridad de Mensaje (*Message Integrity Code* - MIC) que es en realidad un algoritmo denominado “Michael” [5]. El MIC de WPA incluye un mecanismo que contrarresta los intentos de ataque para vulnerar TKIP y bloques temporales.

En resumen, WPA hace más difícil vulnerar las redes inalámbricas al incrementar los tamaños de las claves e IVs, reduciendo el número de paquetes enviados con claves relacionadas y añadiendo un sistema de verificación de mensajes.

1.4.1.1 TKIP (Temporal Key Integrity Protocol)

TKIP [6] es un protocolo de gestión de claves dinámicas aceptado por cualquier adaptador que permite utilizar una clave diferente para cada paquete transmitido. La clave se construye a partir de la clave base, la dirección MAC de la estación emisora y del número de serie del paquete como IV.

Cada paquete que se transmite utilizando TKIP incluye un número de serie único de 48 bits que se incrementa en cada nueva transmisión para asegurar que todas las claves son distintas. Esto evita "ataques de colisión" o de *replay* que se basan en la retransmisión de paquetes cifrados con la misma clave ya que si se inyectara un paquete con una clave temporal que se hubiese podido detectar, el paquete estaría fuera de secuencia y sería descartado. Como se explica más adelante en el apartado 2.2.3, los ataques activos a WEP, al no existir control de secuencia, se basan precisamente en el reenvío de paquetes.

Por otro lado, al utilizar el número de serie del paquete como vector de inicialización (IV), también evitamos IV duplicados.

1.4.2. Autenticación WPA

En WPA podemos encontrarnos con dos tipos de autentificación tal y como se detalla a continuación.

1.4.2.1 PSK (Pre-Shared Key)

De un modo similar al WEP, el método PSK requiere introducir la misma clave compartida en todos los equipos de la red tal y como se observa en la **Fig. 1.6**. Con la PSK se genera, a través del intercambio de dos números aleatorios entre AP y cliente, una clave de cifrado para cada proceso de autenticación llamada PTK.

Una vez el cliente está autenticado, el protocolo TKIP utiliza 6 claves de cifrado por cada sesión (4 de ellas para comunicaciones unicast y 2 broadcast) que son generadas a partir de las direcciones MAC, ESSID y la PTK antes mencionada.

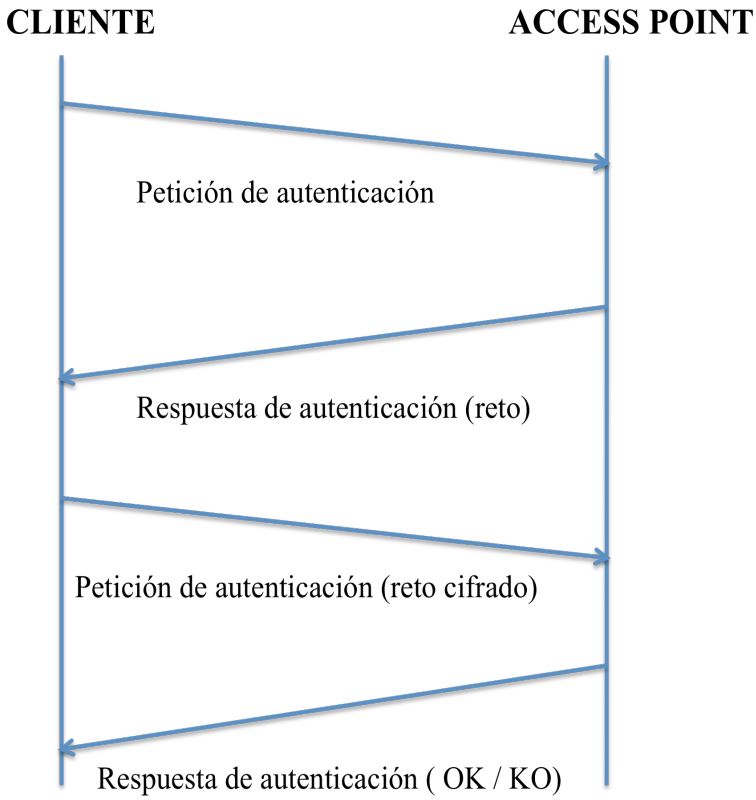


Fig. 1.6 Authentication WPA

1.4.2.2 Basado en 802.1x

El método de autenticación basado en 802.1x (protocolo que se explicará en el siguiente apartado *1.4.2.2.1*) se lleva a cabo mediante el uso de un servidor, donde se almacenan las credenciales y contraseñas de los usuarios de la red.

Además de poder utilizar una clave compartida (PSK), lo cual suple la complejidad de implementación de un servidor de autenticación 802.1x en hogares y oficinas pequeñas, WPA puede utilizar Protocolos Extensibles de Autenticación (EAP o *Extensible Authentication Protocol*), como los mencionados a continuación:

- TLS (*Transport Layer Security*): Requiere autenticación tanto para el cliente como para el AP, creando así una conexión cifrada entre ambas partes.

- TTLS (*Tunneled TLS)*: Protocolo desarrollado por Funk Software y orientado a trabajar con servidores RADIUS. Certifica un AP para establecer un túnel para la autenticación del cliente encriptado.

- EAP-LEAP (*Lightweight EAP*): Se trata de un protocolo propiedad de CISCO y que requiere su infraestructura para poder ser utilizado. En éste las credenciales del usuario se envían sin cifrar.

- PEAP (*Protected EAP)*: Similar a TLS pero solamente se utiliza certificación para el AP. Cabe resaltar que las versiones de Microsoft y Cisco son incompatibles.

Entre los diversos servidores que pueden utilizarse para este tipo de implementaciones, están los basados en RADIUS [7] (*Remote Authentication Dial-In User Server*), un protocolo de AAA (Authentication, *Authorization and Acounting)* para aplicaciones como acceso a redes o movilidad IP.

1.4.2.2.1 IEEE 802.1x

El IEEE 802.1x es un estándar para el control de acceso a red de nivel 2 basado en puertos para la autenticación y distribución de claves. Éste traduce las tramas enviadas por un algoritmo de autenticación en el formato necesario para que estas sean entendidas por el sistema de autenticación que utilice la red.

IEEE 802.1x no es por sí mismo un método de autenticación y debe emplearse de forma conjunta con protocolos de autenticación para llevar a cabo la verificación de las credenciales de usuario, como por ejemplo cualquier tipo de EAP.

Los tres elementos que intervienen en un sistema 802.1x son los siguientes:

- *Autenticador*: generalmente es un AP y su función es forzar el proceso de autenticación y enrutar en tráfico a los dispositivos adecuados de la red.
- *Solicitante*: es el usuario que solicita el acceso a la red.
- *Servidor de autenticación*: lleva a cabo la autenticación de las credenciales de usuario. Generalmente se suele emplear como servidor de autenticación remota de usuarios servidores RADIUS. Adicionalmente, el servidor puede contener políticas para ese usuario concreto que podría aplicar el AP (como por ejemplo, priorizar ciertos tráficos o descartar otros).

En la siguiente figura (**Fig. 1.7)** mostramos la arquitectura típica de un sistema de autenticación 802.1x/EAP. En la figura se observa que entre el cliente y el AP (Autenticador) el protocolo empleado es IEEE 802.1x. El protocolo entre el AP y el servidor de autenticación no está definido en el estándar IEEE 802.1x ni en el estándar IEEE 802.11, en nuestro caso se utiliza RADIUS. Cuando el cliente se conecta a un AP que soporta 802.1x comienza el intercambio de mensajes de autenticación EAP entre ambos para llevar a cabo la autenticación de usuario contra el servidor de autenticación.

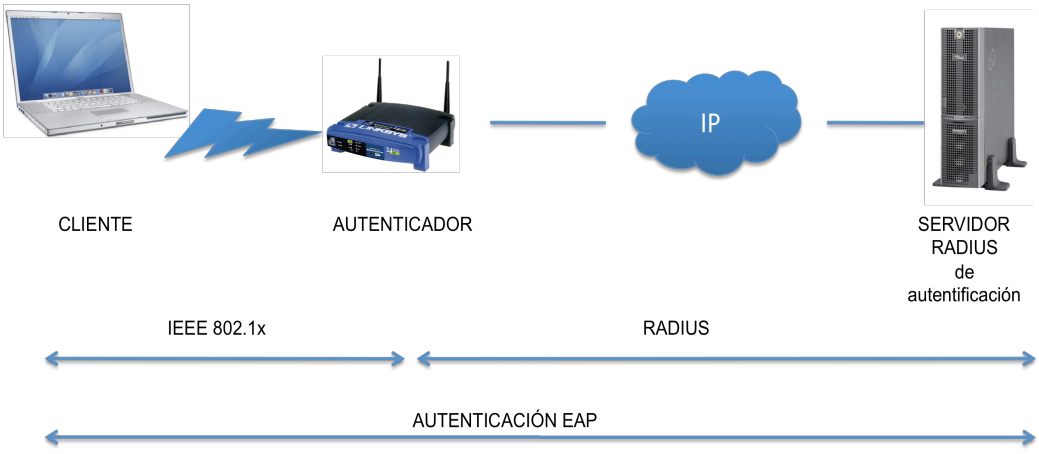


Fig. 1.7 Arquitectura IEEE 802.1x

1.5. WPA versión 2 y 802.11i

La Wi-Fi Alliance lanzó en septiembre de 2004, WPA2, que es la versión certificada de la especificación completa del estándar IEEE 802.11i, que fue certificado en junio de 2004.

Las principales diferencias entre las dos versiones de WPA las veremos en la tabla mostrada a continuación (**Tabla 1.2**). Ésta nos muestra el modo personal, para uso casero y que utiliza autenticación mediante clave compartida (como WEP) y el modo empresarial, utilizado a nivel de empresa y con mejoras en la autenticación (802.1x / EAP como en WPA):

		WPAv1	**WPAv2**
MODO PERSONAL	AUTENTICACIÓN	PSK	PSK
	CIFRADO	TKIP(RC4) / MIC	CCMP(AES) / CBC-MAC
MODO EMPRESARIAL	AUTENTICACIÓN	802.1x / EAP	802.1x / EAP
	CIFRADO	TKIP (RC4) / MIC	CCMP(AES) / CBC-MAC

Tabla 1.2 Diferencias entre WPAv1 y WPAv2

1.5.1. Cifrado en WPA v2 y 802.11i

Una de las diferencias más destacables entre WPAv1 y WPAv2, es que esta última substituye el cifrado RC4 por CCMP (*Counter Mode with Cipher Block Chaining Message Authentication Code Protocol*) que utiliza AES (*Advanced Encryption Standard*), tal y como hemos podido ver en la **Tabla 1.2**.

AES es un esquema de cifrado de datos por bloques que utiliza una estrategia de red de sustitución-permutación. Este utiliza 128 bits como tamaño de bloque y 128, 192 o 256 como tamaño de llave, si bien en su versión original el algoritmo podía usar indistintamente 128, 160, 192, 224 o 256 bits.

CCMP es protocolo que utiliza AES (*Advanced Encryption Standard*) como algoritmo criptográfico y proporciona integridad y confidencialidad. Se basa en el modo CCM (*Counter with CBC-MAC*) del algoritmo de cifrado AES y utiliza llaves de 128 bits con IVs de 48 bits.

CCMP consta del algoritmo de privacidad que es el "*Counter Mode*" (CM) y del algoritmo de integridad y autenticidad que es el "*Cipher Block Chaining Message Authentication Code*" (CBC-MAC).

Además del cambio de cifrado, WPAv2 requiere un hardware totalmente nuevo para poder llevar a cabo el cifrado con normalidad.

1.5.2. Autenticación en WPA v2 y 802.11i

Los mecanismos encargados de la autenticación y la gestión de las claves en WPA versión 2 y 802.11i son EAP y 802.1x, ya mencionados anteriormente en el WPA versión 1.

CAPÍTULO 2. Vulnerabilidades de WEP y WPA

Después de todo lo visto en el Capítulo 1 a continuación tratamos las debilidades de ambos protocolos y los posibles ataques.

2.1 Defectos del protocolo WEP

Mostramos a continuación las debilidades del protocolo WEP y el porqué de la creación de protocolos como WPA o 802.11i para aumentar la seguridad en las comunicaciones entre dispositivos inalámbricos.

2.1.1 Debilidad del vector de inicialización

La implementación del vector de inicialización (IV) en el algoritmo WEP tiene varios problemas de seguridad. Recordemos que el IV es la parte que varía de la semilla (*seed*) para impedir que un posible atacante recopile suficiente información cifrada con una misma clave.

El estándar 802.11 no especifica cómo manejar el IV. En principio se debería cambiar en cada trama para mejorar la privacidad, pero no es obligatorio, por lo tanto cada fabricante elije como tratar el IV y la mayoría de ellos optan por fijar el IV a 0 cada vez que arrancamos la tarjeta e incrementarlo a 1 para cada trama enviada. Esto provoca que las primeras combinaciones de IVs y clave secreta se repitan muy a menudo.

Además de todo esto, sabemos que el número de IVs diferentes no es demasiado elevado (2^{24}=16 millones aprox.), por lo que terminarán también repitiéndose al cabo de horas o días dependiendo del tráfico de la red.

Con esto llegamos a la conclusión de que la repetición tarde o temprano se producirá. La cantidad de repeticiones dependerá de la implementación que elija cada fabricante para variar el IV (aleatoria, secuencial, etc.) y de la carga de la red.

2.1.2 Ataques pasivos

Un ataque pasivo es aquel en el que sólo se escucha la información de la red y no se deja rastro alguno del *sniffing.* Al atacante no le hace falta participar de manera activa en la comunicación para poder recopilar paquetes con contraseñas u otra información relevante para la resolución de la clave.

2.1.3 Ataques activos

Los ataques activos son aquellos en los cuales el atacante no sólo escucha la información que circula a través de la red sino que también participa de manera activa interactuando con el tráfico tal y como veremos en el Capítulo 3. Atacando 802.11.

2.1.4 Identificación de secuencias pseudoaleatorias iguales

El mensaje cifrado que queremos enviar es el resultado de realizar un XOR entre el generador pseudoaleatorio (*Keystream* - Secuencia RC4) y el texto, como hemos visto en el apartado *1.3.1.1*. Si tenemos que los mensajes cifrados son:

$$C1 = M1 \oplus RC4\ (\ IV\ ,\ k\ )$$
$$C2 = M2 \oplus RC4\ (\ IV\ ,\ k\ )$$

teniendo en cuenta esto y si se realiza un XOR de dos mensajes cifrados con el mismo IV y *keystream*:

$$C1 \oplus C2 = [\ M1 \oplus \cancel{RC4\ (\ IV\ ,\ k\ )}] \oplus [M2 \oplus \cancel{RC4\ (\ IV\ ,\ k\ )}] =$$

$$= M1 \oplus M2$$

se obtiene el XOR de los dos textos en claro ya que el *keystream* se cancela.

$$\mathbf{C1 \oplus C2 = M1 \oplus M2}$$

M1 y M2 los podremos recuperar debido a la redundancia que habitualmente tienen los textos planos. Buscaremos dos textos sobre los que, aplicados un XOR, resulten en el valor dado M1 ⊕ M2.

2.1.5 Vulnerabilidad RC4

El sistema de cifrado RC4 básicamente tiene dos vulnerabilidades: lo predecibles que pueden llegar a ser los primeros bytes generados por RC4 y la concatenación de la clave compartida con un IV público y conocido, que hacen vulnerable la clave del algoritmo de cifrado.

En cuanto a lo predecibles que pueden llegar a ser los primeros bytes, tenemos que comentar que se debe a que el primer byte del *keystream* es del estilo:

$$S\ [\ S(1) + S[S(1)]\ ]$$

ya que se concatena un vector inicial de 3 bytes a la clave. También sabemos que el primer byte de texto en claro que se envía es la cabecera *LLC (SNAP)* para TCP/IP es **0xAA**.

Este problema se podría solucionar descartando estos primeros bytes de salida del RC4. Es decir, empezando a utilizar desde el byte “n”.

La segunda vulnerabilidad que tiene el sistema de generación de claves RC4 es la concatenación de la clave compartida con el IV. Si un atacante puede obtener la primera palabra de salida del algoritmo RC4 correspondiente a cada IV, podría reconstruir la clave secreta con poco trabajo.

Como posible solución a esta segunda vulnerabilidad está la de utilizar una función de *hash* para substituir al concatenado. Esto produciría un considerable retardo a la hora de la inicialización del RC4 ya que se trata de funciones lentas comparadas con el estándar.

2.2 Debilidades del protocolo WPA

Así como en el protocolo WEP hemos analizado los posibles caminos por los cuales vulnerarlo, WPA se considera computacionalmente irrompible. De hecho sólo es vulnerable la versión WPA-PSK (*Pre-Shared Key*), que se basa en una clave compartida entre usuario y Access Point. Tal y como se explicó en el apartado *1.4.2 Autenticación WPA*, la clave compartida sólo se utiliza durante el *handshake* inicial para negociar claves de sesión. Si conseguimos capturar dicho *handshake*, a partir de las MACs (AP y cliente), del SSID y de los números aleatorios intercambiados, podemos conseguir el secreto compartido tal y como se observa en la **Fig. 2.8**.

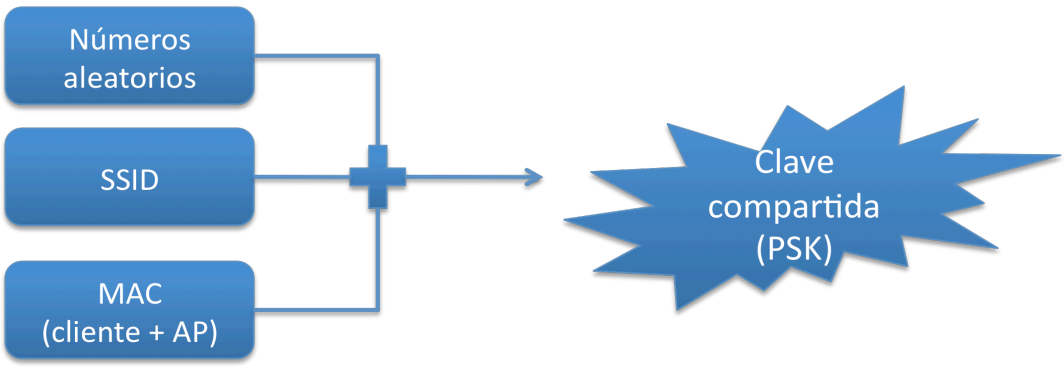


Fig. 2.8 Ataque WPA

Una vez capturado todo lo necesario y sólo a falta de saber la PSK, utilizaremos la fuerza bruta o el uso de un diccionario para sacar la clave utilizada en el cifrado.

CAPÍTULO 3. Atacando 802.11

Una vez vistas las debilidades de WEP y WPA lo que debemos hacer ahora es comprobar de manera práctica dichas vulnerabilidades. Para esto utilizaremos la suite *aircrack-ng* [8] detallada a continuación.

3.1 Suite aircrack-ng

Para la realización de este TFC se ha utilizado un Live CD Linux llamado Backtrack 3 en el cual podemos encontrar la suite *aircrack-ng* y una tarjeta Wi-Fi con un chipset Broadcom bcm43xx.

3.1.1 Introducción

La suite *aircrack-ng* es una herramienta para crackear claves WEP y WPA/WPA2-PSK. Para crackear el primer tipo de claves se utilizarán diversos métodos tal y como veremos en el apartado 3.1.3. Para el descifrado de claves WPA y WPA/PSK será necesario la utilización de un diccionario, tal y como se verá en el apartado 3.2.

Esta suite se compone de 3 herramientas necesarias para llevar a cabo el descifrado de claves: *airodump-ng*, *aireplay-ng* y *aircrack-ng*. Las cuales se detallan a continuación.

3.1.2 Airodump-ng

A continuación se hace un análisis de la herramienta *airodump-ng*, explicando para qué y cómo se utiliza.

3.1.2.1 Descripción

La herramienta *airodump-ng* se utiliza principalmente para capturar paquetes 802.11 que circulan por la red de una manera pasiva. Ésta captura nos permitirá más adelante mediante la herramienta *aircrack-ng* descifrar la clave, gracias a la recopilación de todos los IVs que contienen dichos paquetes.

3.1.2.2 Utilización

La utilización de esta herramienta requiere antes una serie de pasos a seguir para poder sacarle el máximo provecho.

Antes de empezar a capturar paquetes debemos configurar nuestra tarjeta *wireless* en modo monitor para que pueda detectar todo el tráfico que circula a

nuestro alrededor. Esto se consigue tecleando los siguientes comandos en la consola de nuestra distribución Linux:

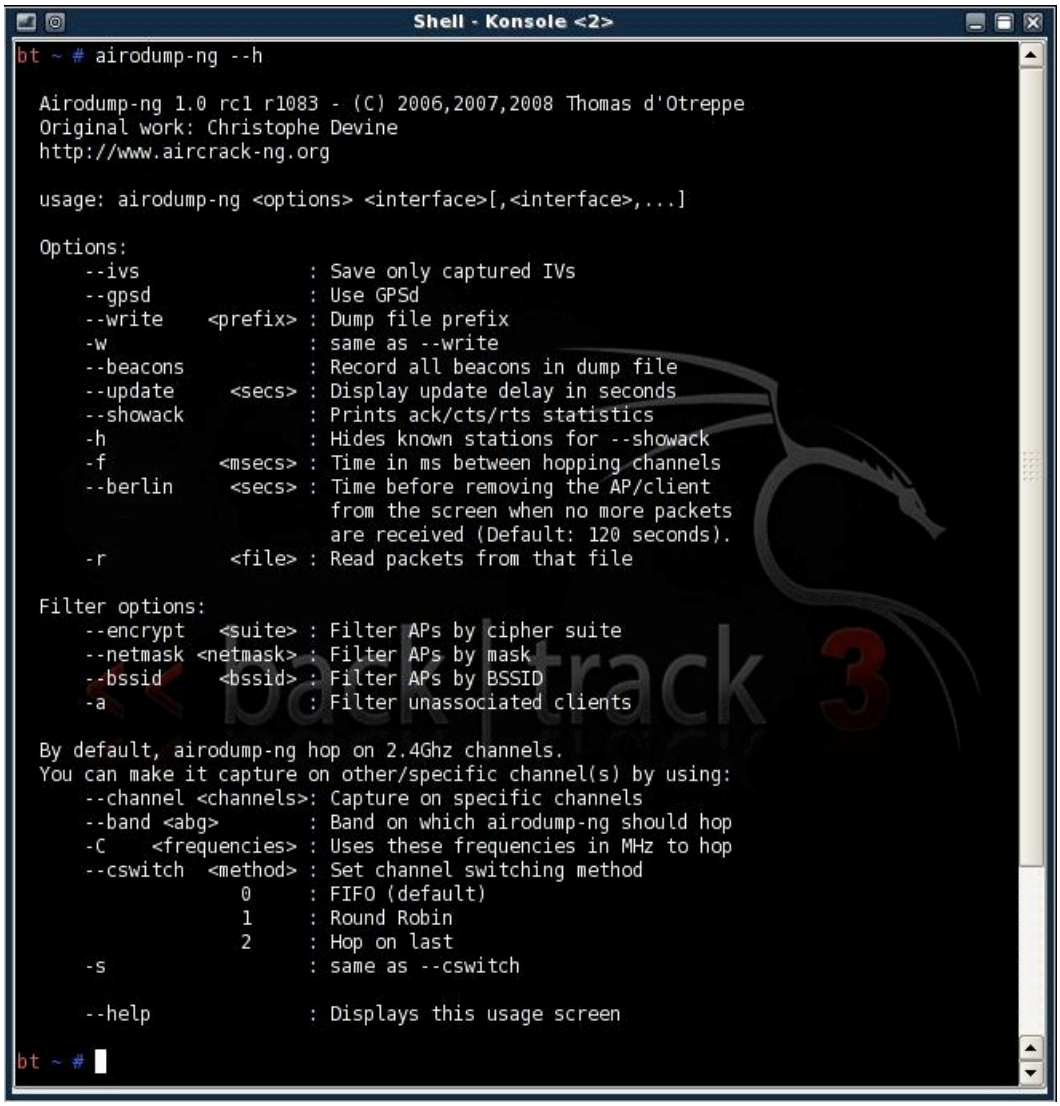
```
airmon-ng <start | stop><interface> [canal]
```

- *<start | stop>*: indica si queremos activar o desactivar el modo monitor.
- *<interface>*: debemos indicar el nombre de nuestra interface.
- *[canal]*: podemos indicar el número de canal (no obligatorio).

Una vez activado el modo monitor de nuestra tarjeta Wi-Fi, podremos empezar a detectar paquetes y capturarlos mediante *airodump-ng*, tecleando lo siguiente:

```
airodump-ng <opciones><interface>
```

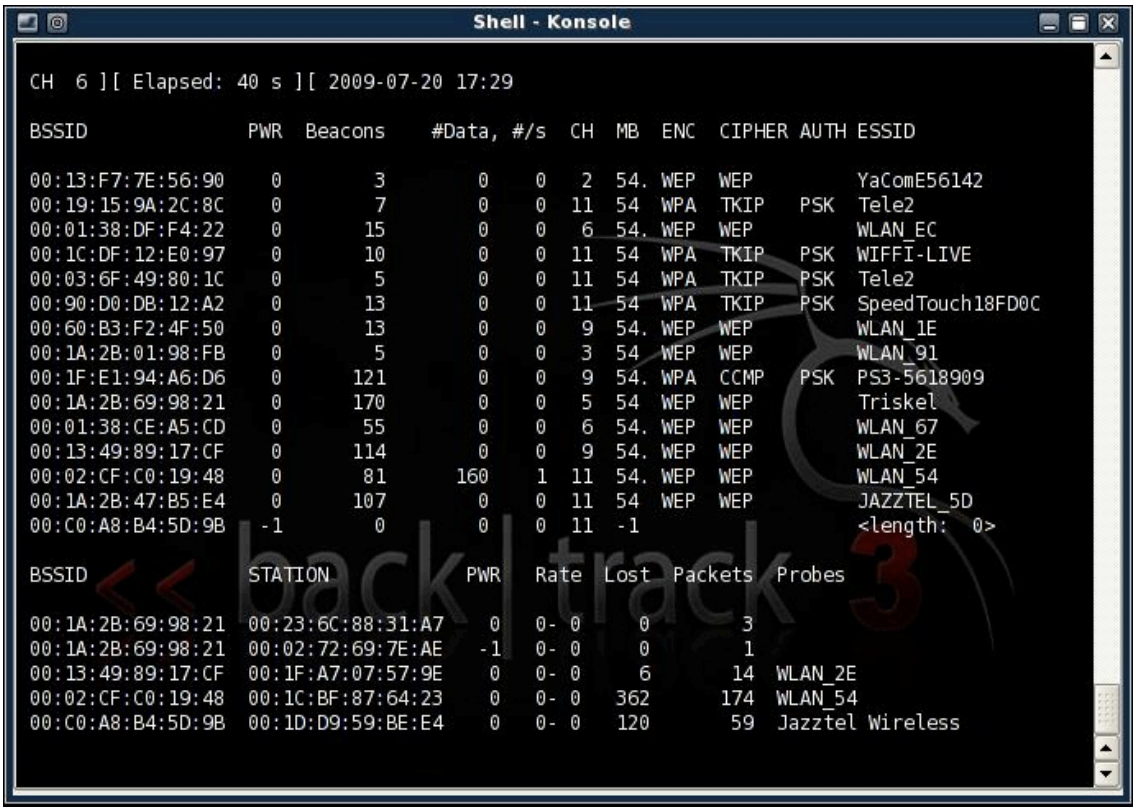
- *<opciones> :* las opciones que podemos introducir son las que se muestran a continuación en la **Captura 3.1**:



Captura 3.1 Opciones airodump-ng

- *<interface>:* nombre de nuestra interface.

Una vez ejecutado el comando, analizaremos los datos mostrados (**Captura 3.2**):



Captura 3.2 Analizando redes con airodump-ng

BSSID	Nos indica la MAC del Access Point (AP)
PWR	Nivel de señal. Nos indica la potencia con la que nos llega la señal del AP. A mayor PWR, más cerca estaremos del AP o del cliente. Si nos indica -1 quiere decir que: o el *driver* no soporta la detección de nivel de señal ó el PWR es -1 para algunos clientes (*stations*) es porque los paquetes proceden del AP hacia el cliente pero las transmisiones del cliente se encuentran fuera del rango de cobertura de la tarjeta.
Beacons	Número de "paquetes anuncio" o *beacons* enviados por el AP. Cada punto de acceso envía alrededor de diez *beacons* por segundo cuando el *rate* o velocidad es de 1M, (la más baja) de tal forma que se pueden recibir desde muy lejos.
#Data	Número de paquetes de datos capturados (si tiene clave WEP, equivale también al número de IVs), incluyendo paquetes de datos *broadcast* (dirigidos a todos los clientes).
#/s	Número de paquetes de datos capturados por segundo calculando la media de los últimos 10 segundos.
CH	Número de canal (obtenido de los *beacons*).

MB	Velocidad máxima soportada por el AP. Si MB = 11, es 802.11b, si MB = 22 es 802.11b+ y velocidades mayores son 802.11g. El punto (después del 54) indica que esa red soporta un preámbulo corto o "*short preamble*".
ENC	Algoritmo de encriptación que se usa. OPN = no existe encriptación (abierta),"WEP?" = WEP u otra (no se han capturado suficientes paquetes de datos para saber si es WEP o WPA/WPA2), WEP (sin el interrogante) indica WEP estática o dinámica, y WPA o WPA2 en el caso de que se use TKIP o CCMP.
CIPHER	Detector del cifrado. Puede ser CCMP, WRAP, TKIP, WEP, WEP40, o WEP104.
AUTH	El protocolo de autenticación usado. Puede ser MGT, PSK (clave compartida), u OPN (abierta).
ESSID	Nos indica el nombre del AP. También llamado *SSID* (*Service Set IDentifier*), que puede estar en blanco si la ocultación del SSID está activada en el AP. En este caso, airodump-ng intentará averiguar el SSID analizando paquetes "*probe responses*" y "*association requests*" (son paquetes enviados desde un cliente al AP).
STATION	Dirección MAC de cada cliente asociado.
Lost	El número de paquetes perdidos en los últimos 10 segundos.
Packets	El número de paquetes de datos enviados por el cliente.
Probes	Los ESSID's a los cuales ha intentado conectarse el cliente.

3.1.3 Aireplay-ng

La herramienta *aireplay-ng* es una herramienta fundamental para descifrar contraseñas WEP/WPA-PSK, tal y como se explica a continuación.

3.1.3.1 Descripción

Cuando estamos analizando una red y no disponemos de suficientes paquetes capturados de manera pasiva (*airodump-ng*) para después descifrar la clave, es necesaria la herramienta *aireplay-ng*, que nos permite actuar de manera activa en la red, es decir, inyectar paquetes. Con esto conseguiremos que el AP o un cliente asociado nos respondan y así conseguir más IVs.

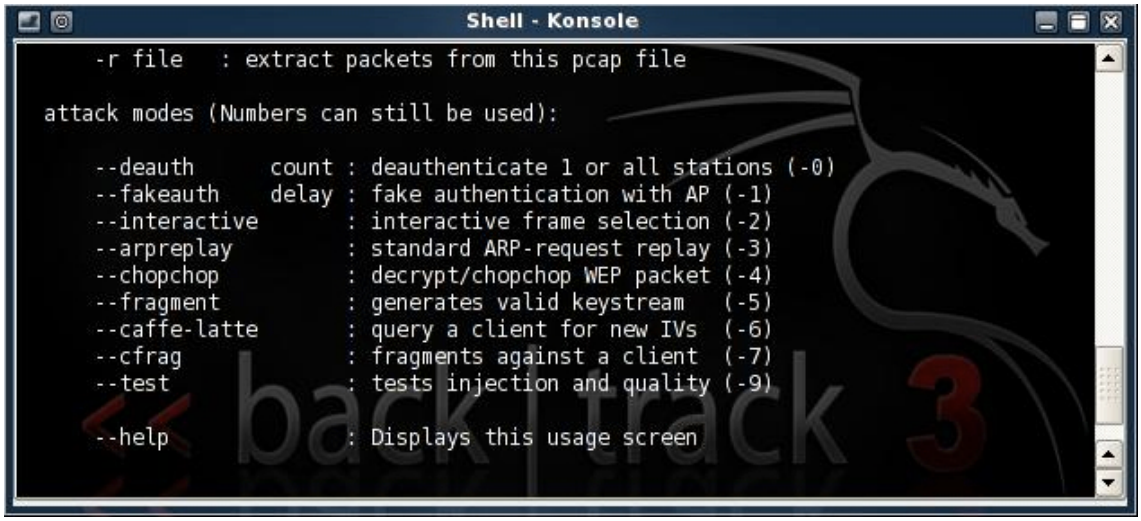
Acto seguido describimos los diferentes tipos de inyección de paquetes y su utilización en cada caso.

3.1.3.2 Utilización y métodos de inyección

Antes de poder utilizar *aireplay-ng* tenemos que asegurarnos que el *driver* de nuestra tarjeta *wireless* es compatible con la inyección de tráfico en la red, consultando la siguiente web:

http://www.aircrack-ng.org/doku.php?id=compatibility_drivers&s[]=drivers&s[]=injection.

Una vez comprobado, y en caso de no tener ningún problema, disponemos de varios métodos para la inyección que se detallan a continuación en la **Captura 3.3**.



Captura 3.3 Ataques aireplay-ng

3.1.3.2.1 Deautenticación (-0)

El requisito mínimo para este ataque es que el AP que queremos atacar tenga al menos un cliente asociado.

El ataque deautenticación (-0) envía paquetes de desasociación a uno o más clientes que están asociados en ese momento a un AP. Las razones por las que es útil desasociar o deautenticar clientes pueden ser:

- Recuperar o desvelar un ESSID oculto que no es divulgado o anunciado por el AP, aprovechando que el cliente vuelve a conectarse al AP y este le pasa todos los parámetros incluido el identificador.
- Capturar *handshakes* WPA/WPA2 forzando a los clientes a volverse a autenticar como veremos más adelante en el ataque a dichos protocolos.
- Generar peticiones ARP para así poder reinyectarlas y aumentar la captura de IVs.

Los comandos a utilizar para ejecutar este ataque son los siguientes:

```
aireplay-ng -0 1 -a XX:XX:XX:XX:XX:XX -c 00:00:00:00:00:00
                                    eth1
```

donde:

- *-0* significa deautenticación.

- *1* es el número de deautenticaciones a enviar (0 significa enviarlas sin intervalos, es decir, continuamente).
- *-aXX:XX:XX:XX:XX:XX* es la MAC del AP.
- *-c00:00:00:00:00:00* es la MAC del cliente asociado. Si no se especifica este campo se deautenticarían a todos los clientes.
- *eth1* es el nombre de nuestra interface.

3.1.3.2.2 Fake authentication with AP (-1)

Este ataque es clave en el descifrado de claves WEP ya que su utilización es vital para luego ejecutar los ataques de *ARP-request replay (-3)* y *decrypt chop-chop(-4)*, descritos en los apartados 3.1.3.2.4 y 3.1.3.2.5 sucesivamente.

Fake authentication with AP consiste en crear una falsa autenticación para que nuestra MAC quede registrada en la tabla de asociación del AP. Este ataque resulta útil cuando no tenemos ningún cliente asociado al AP para poder utilizar su MAC, siempre y cuando el AP no tenga filtrado de MACs. Si se da el caso, deberemos esperar a que se conecte un cliente y este ataque quedará inutilizado.

La ejecución de este ataque se lleva a cabo de la siguiente manera:

```
aireplay-ng -1 0 –e WLAN3 –a 00:00:00:00:00:00 –h
             11:11:11:11:11:11 eth1
```

donde:

- *-1* significa *Fake authentication* (Autenticación falsa).
- *0* tiempo que transcurre para volverse a asociar con el AP.
- *-e WLAN3* es el nombre del AP.
- *-a 00:00:00:00:00:00* es la MAC del AP.
- *-h 11:11:11:11:11:11* es la MAC de nuestra tarjeta.
- *eth1* es nuestra interface.

3.1.3.2.3 Interactive frame selection (-2)

Este ataque (Reenvío interactivo de paquetes) nos permite escoger el paquete a reenviar en el proceso de inyección. Se pueden obtener los paquetes para inyectar de 2 formas:

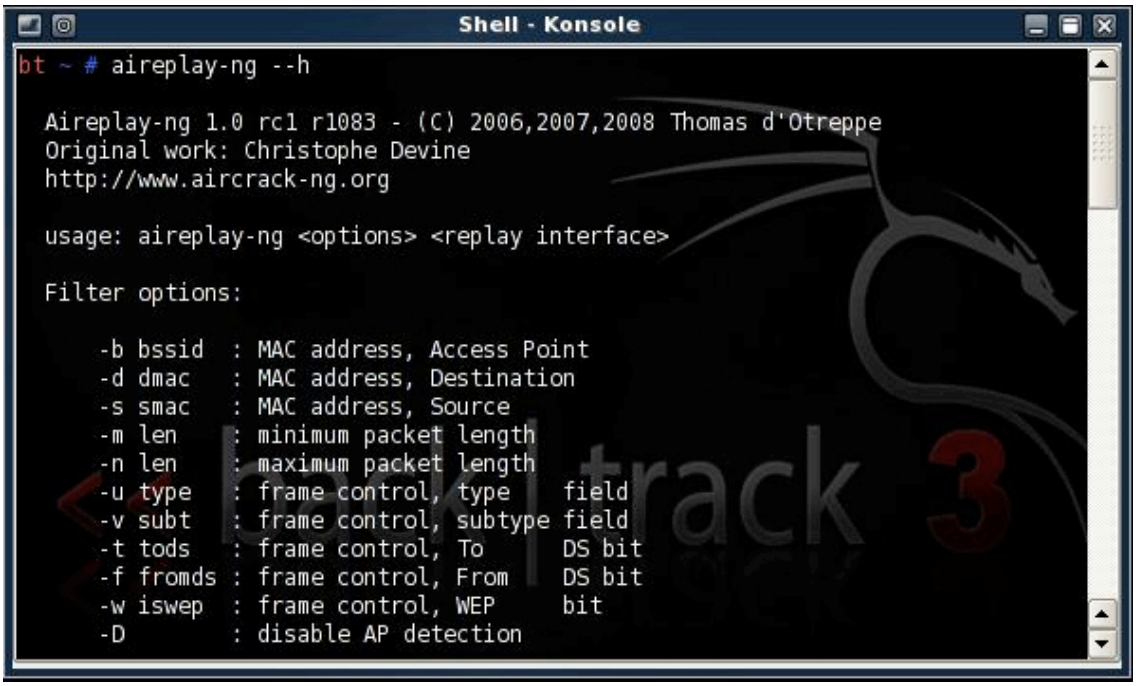
- Capturando paquetes al instante con nuestra tarjeta Wi-Fi mediante *airodump-ng* como hemos visto en el apartado 3.1.2.
- La segunda es utilizando un archivo con formato estándar *.cap* o .Pcap (Packet CAPture) que permite leer paquetes capturados en sesiones anteriores, o que han sido obtenidos con otros ataques.

Para ejecutar este ataque teclearemos en nuestra consola:

```
aireplay-ng -2 <opc. de filtro><opc. de reenv> -r <nom.
                archivo><interface>
```

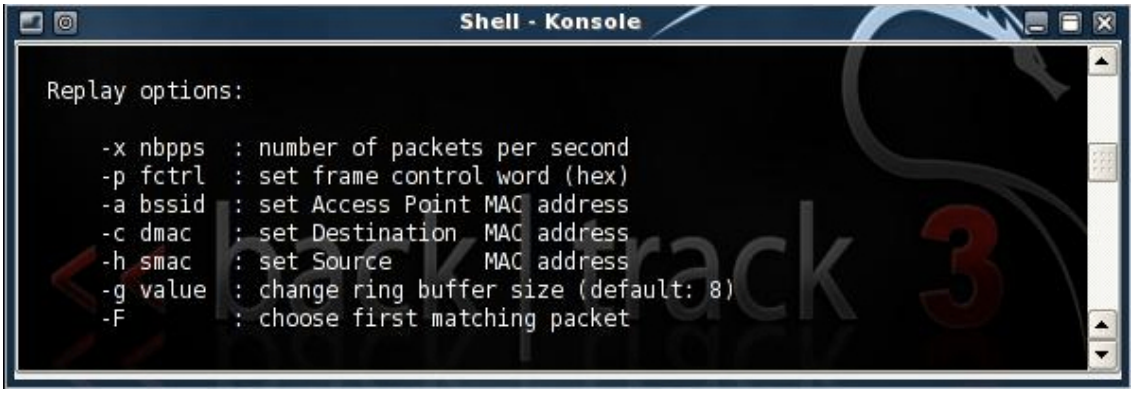
donde:

- *-2* significa *Interactive Frame Selection* (Reenvío interactivo de paquetes).
- *<opc. de filtro>* son las siguientes (**Captura 3.4**) :



Captura 3.4 Opciones de filtro en aireplay-ng

- *<opc. de reenv>* son las siguientes (**Captura 3.5**):



Captura 3.5 Opciones de reenvío de aireplay-ng

- *-r <nom. archivo>* es el nombre del archivo *.cap* del que queremos leer los paquetes para inyectar.
- *<interface>* el nombre de nuestra interface.

Un ejemplo de este ataque podría ser el siguiente, donde se fija como dirección de destino FF:FF:FF:FF:FF:FF (*broadcast*) para así poder captar más IVs generados por el AP:

```
aireplay-ng -2 -p 0841 -c FF:FF:FF:FF:FF:FF -b <MAC AP> -h
                    <MAC Wi-Fi>  eth1
```

donde:

- *-2* es el número de ataque.
- *-p 0841* fija el "*Frame Control*" en el paquete para que parezca que está siendo enviado desde un cliente.
- *-c FF:FF:FF:FF:FF:FF* se fija como MAC destino cualquiera (broadcast).
- *-b <MAC AP>* es la MAC del punto de acceso (BSSID).
- *-h <MAC Wi-Fi>* MAC de los paquetes que se están transmitiendo, que tiene que coincidir con la MAC de nuestra tarjeta.
- *eth1* el nombre de nuestra interface.

3.1.3.2.4 Standard ARP-request replay (-3)

El ataque de reenvío de petición ARP o "*ARP request*" es el modo más efectivo y eficaz para generar nuevos IVs.

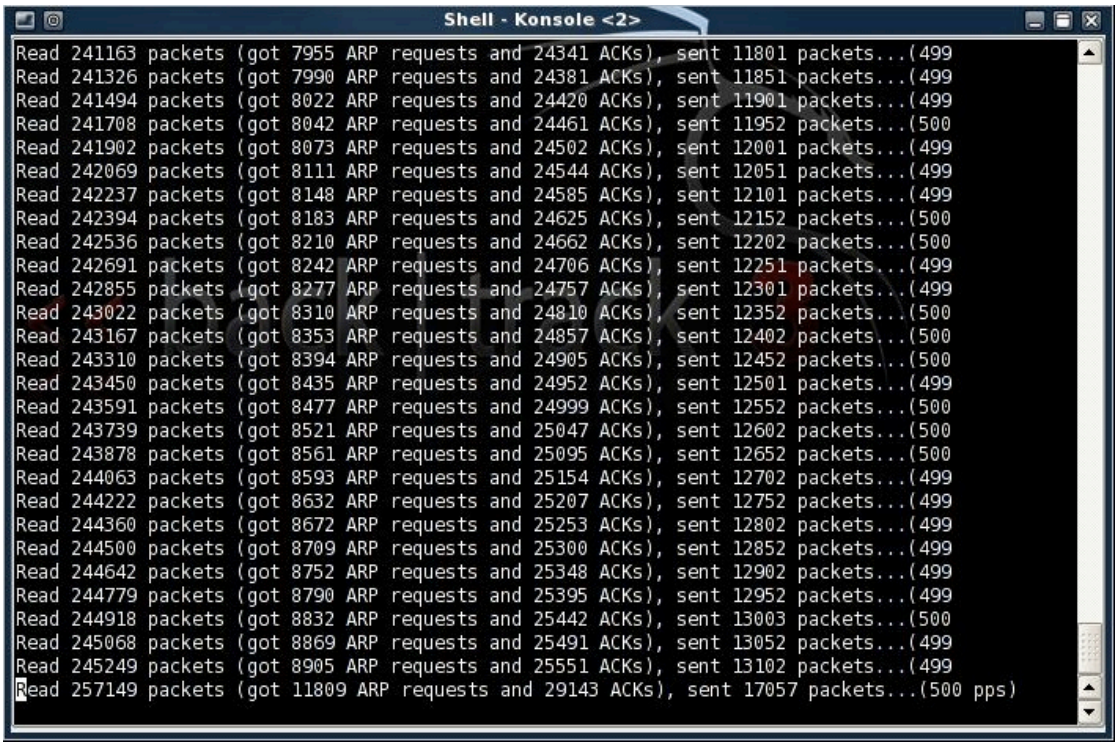
Como antes hemos comentado en el apartado 3.1.3.2.2, después de ejecutar el ataque *Fake Authentication with AP*(-1), lanzamos el *Standard ARP-request replay*(-3) que escucha el medio hasta encontrar un paquete ARP. Una vez encontrado lo retransmite hacia el AP, lo que provoca que éste tenga que repetir el paquete ARP con un IV nuevo. El programa retransmite el mismo paquete ARP una y otra vez (**Captura 3.6**), con lo que conseguiremos nuevos IVs por cada paquete ARP repetido por el AP. Todos estos nuevos IVs nos permitirán averiguar la clave WEP.

Los comandos necesarios para ejecutar este ataque son los siguientes:

```
aireplay-ng -3 -b 00:00:00:00:00:00 -h 11:11:11:11:11:11
                        eth0
```

donde:

- *-3* es el ataque *Standard ARP-request replay*.
- *-b 00:00:00:00:00:00* es la MAC del AP.
- *-h 11:11:11:11:11:11* MAC de los paquetes que se están transmitiendo, que tiene que coincidir con la MAC de nuestra tarjeta.
- *eth1* es el nombre de nuestra interface.



Captura 3.6 Envio de paquetes

3.1.3.2.5 Decrypt chop-chop WEP packet (-4)

En este y en el siguiente apartado explicaremos dos ataques que no consiguen la clave por sí mismos sino que obtienen otros parámetros para luego descifrar la clave con la ayuda de otros ataques. Ambos necesitan un paquete de datos WEP como mínimo.

El correspondiente a este apartado, el ataque *Decrypt chop-chop WEP packet* (Descifrado chop-chop de un paquete WEP) puede revelar el texto plano de un paquete WEP (estática y dinámica) sin necesidad de conocer la clave.

Algunos APs no son vulnerables a este tipo de ataques ya que descartan los paquetes menores de 60 bytes. En el caso de que el AP deseche paquetes menores de 42 bytes, *aireplay-ng* comprueba el *checksum* del encabezado (*header*) tan pronto como este sea predecible y acto seguido trata de adivinar las partes restantes de datos del paquete.

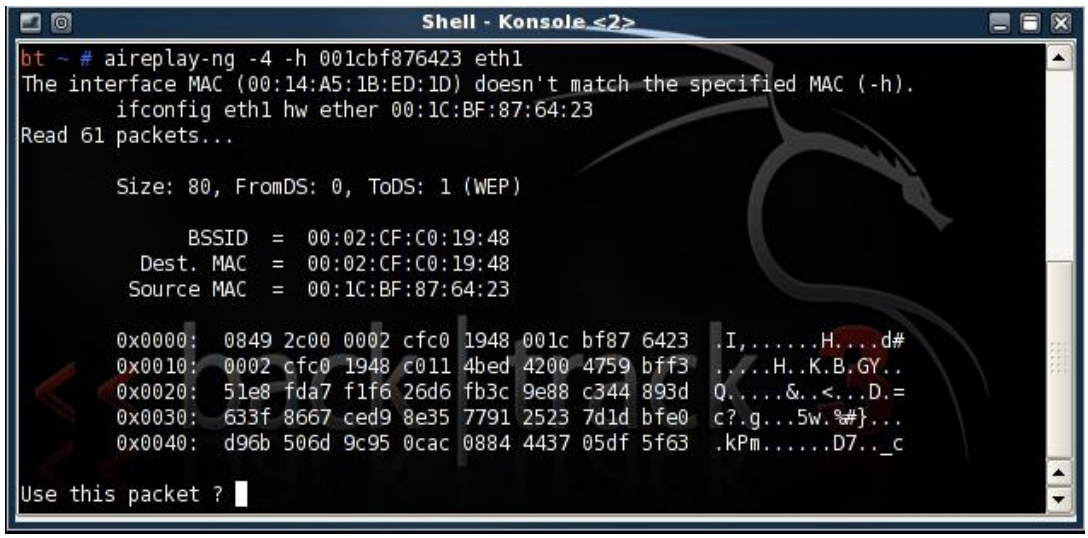
En el caso de que no tengamos paquetes *ARP Request* circulando, *aireplay-ng* puede capturar paquetes de enrutamiento RIPv2, los cuales también nos servirán como vemos acto seguido en la ejecución del ataque.

La ejecución de este ataque se realiza de la siguiente manera:

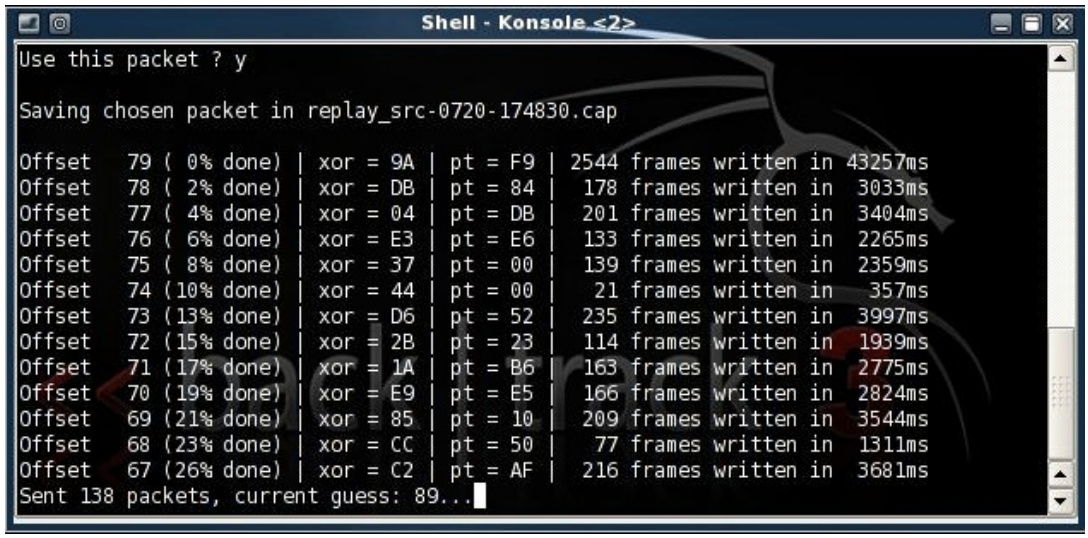
1. Primero de todo desciframos un paquete que circule por la red, utilizando la MAC de un cliente asociado al AP. Se selecciona el paquete

a enviar (**Captura 3.7**) y se guarda el paquete elegido (**Captura 3.8** y **Captura 3.9**).

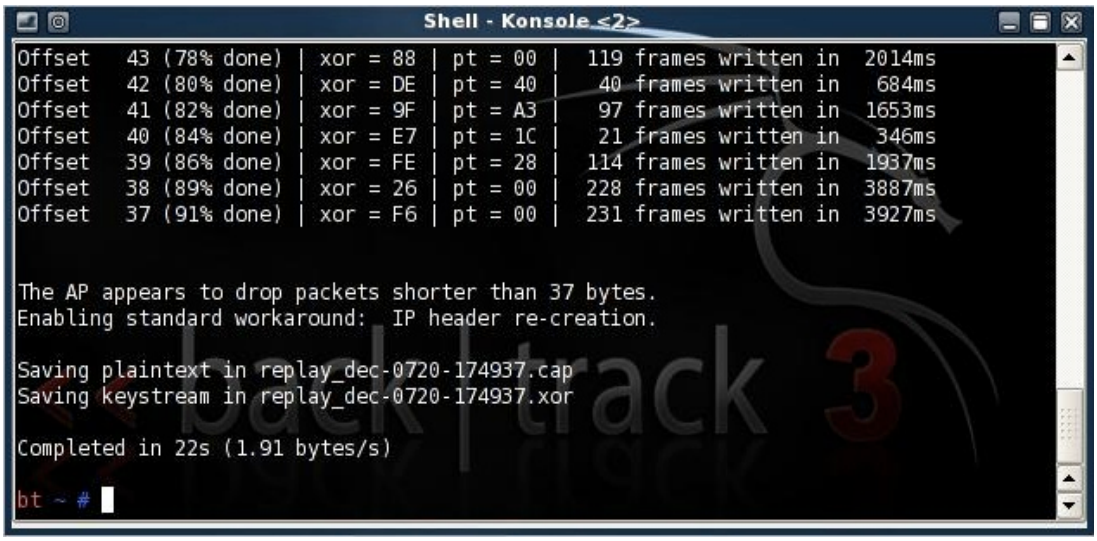
```
aireplay -4 -h 11:11:11:11:11:11 eth1
```



Captura 3.7 Elección del paquete



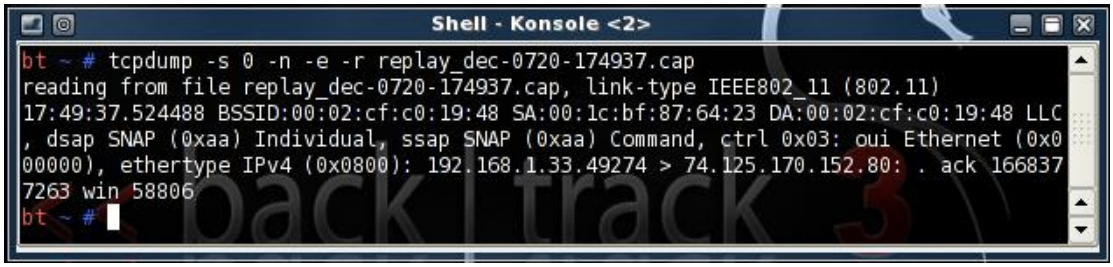
Captura 3.8 Guardando el paquete elegido



Captura 3.9 Paquete guardado y listo para forjar

2. Usamos el comando *tcpdump* para explorar los diferentes campos del paquete (en nuestro caso es un ACK pero también podría ser un paquete de enrutamiento RIP) y así poder escoger la IP que nos interesa (**Captura 3.10**).

```
tcpdump -s 0 -n -e -r <archivo .cap>
```



Captura 3.10 Exploramos el paquete con tcpdump

3. Ejecutamos el comando *packetforge-ng* para crear nuestro paquete ARP, que debe ser respondido con *ARP Request* por el AP. En caso de que exista filtrado MAC, es muy importante especificar la IP destino y la dirección MAC de origen, ya que tiene que ser la de un cliente asociado.

```
packetforge-ng <archivo .xor> 1 00:13:10:30:24:9C
00:09:5B:EB:C5:2B 192.168.1.100 192.168.1.2 < nuestra
petición arp.cap>
```

4. Una vez creada nuestra petición ARP ya podemos enviarla mediante el siguiente comando.

```
aireplay-ng -2 -r <nuestra petición arp.cap> eth1
```

3.1.3.2.6 Generates valid keystream (-5)

Como hemos comentado en el apartado anterior este ataque no recupera la clave WEP por sí mismo y necesita como mínimo un paquete WEP recibido del AP para poder llevarlo a cabo.

Con *Generates valid keystream* o ataque de fragmentación (genera un [IV + contraseña] válido) podemos obtener 1500 bytes de un PRGA (*Pseudo Random Generation Algorithm*) para después poder generar paquetes con *packetforge-ng* como hemos visto en el ataque anterior.

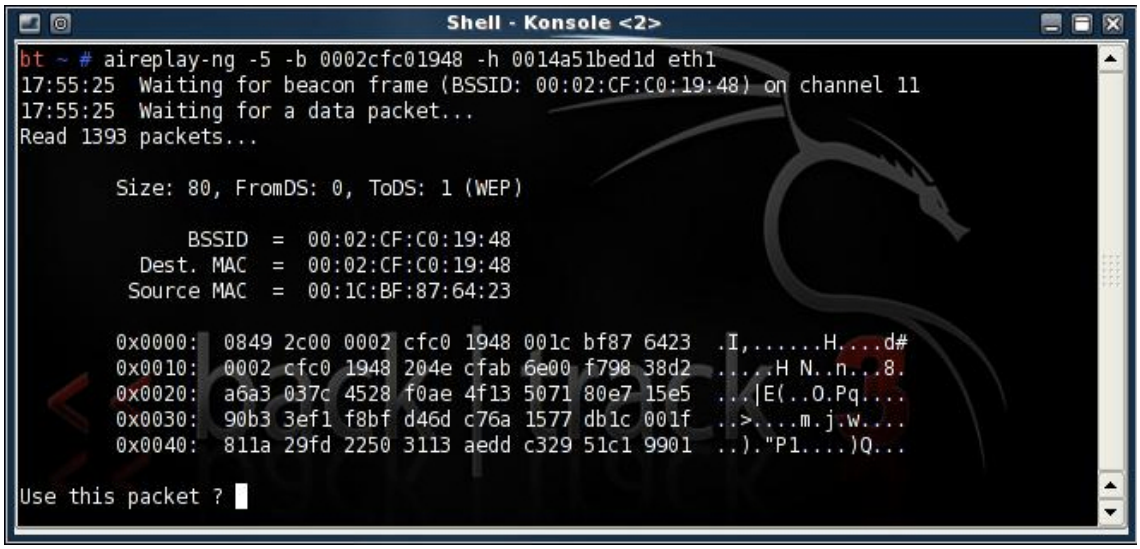
El PRGA es texto plano + texto cifrado y se utiliza para poder cifrar los paquetes que queremos enviar. Si conseguimos interceptar texto cifrado y sabemos cuál es el texto plano podemos averiguar el PRGA y una vez averiguado, usarlo para cifrar cualquier paquete y así poder generar tráfico.

La misión de este ataque es obtener una pequeña cantidad de información sobre la clave de un paquete e intenta enviar un ARP y/o paquetes LLC al AP. Si el AP recibe el paquete y contesta entonces se podrá obtener un poco más de información sobre la clave de ese nuevo paquete enviado por el AP. Este ciclo se repite varias veces hasta que obtenemos los 1500 bytes del PRGA.

Los pasos a seguir para ejecutar este ataque son los siguientes:

1. Iniciamos el ataque con el siguiente comando y acto seguido seleccionamos el paquete (**Captura 3.11**) que queremos probar de manera similar al ataque *Decrypt chop-chop WEP packet (-4)*.

```
aireplay-ng -5 -b <MAC AP> -h <nuestra MAC> eth1
```



Captura 3.11 Selección del paquete

Una vez elegido el paquete que queremos utilizar, el programa se encarga de enviar el paquete seleccionado e ir recopilando partes del PRGA. Acto seguido pasamos al paso 2.

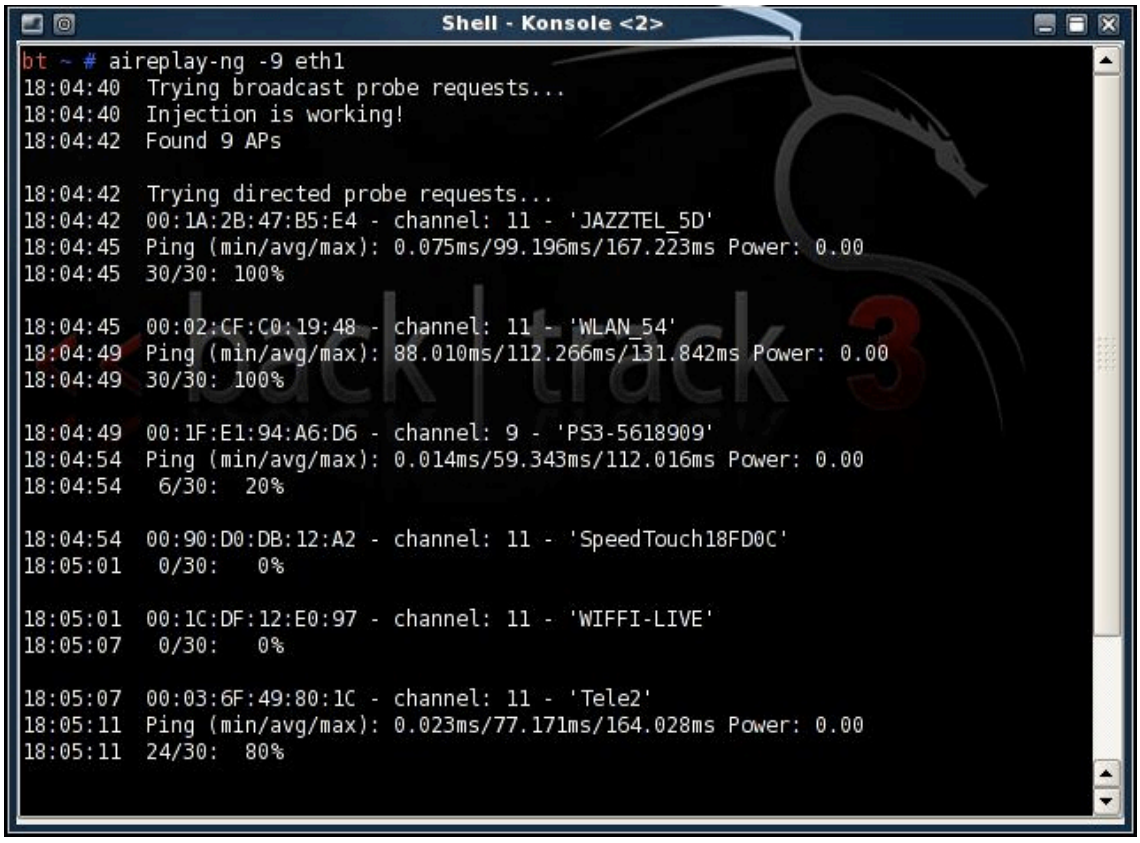
2. Una vez obtenido los 1500 bytes del PRGA que se guarda en el archivo fragment-XXX-XXXX.xor ejecutamos el *packetforge-ng* para generar uno o más paquetes y usarlos con alguno de los ataques de inyección.

3.1.3.2.7 Test injection and quality (-9)

Una vez explicados los ataques más importantes para testear la vulnerabilidad del protocolo WEP, debemos mencionar que antes de realizar cualquier ataque, la suite *aircrack-ng* nos proporciona una herramienta con la cual podemos comprobar la calidad y el porcentaje de inyección sobre todos los AP alcanzables (**Captura 3.12**).

Los comandos para ejecutar este test son los siguientes:

```
aireplay-ng -9 eth1
```



Captura 3.12 Test de inyección de paquetes

3.1.4 Aircrack-ng

Como hemos visto en los apartados 3.1.2 y 3.1.3, ya tenemos los paquetes necesarios para descifrar la clave WEP. ¿Cómo podemos averiguarla? Muy fácil. Ejecutando la herramienta *aircrack-ng*.

3.1.4.1 Descripción

Aircrack-ng es una herramienta de criptoanálisis que nos permite recuperar la clave a partir de la captura mediante *airodump-ng* de paquetes cifrados, combinando ataques estadísticos con ataques de fuerza bruta.

Para el caso de las claves WPA/WPA2-PSK, como veremos más adelante, será necesario la utilización de un diccionario.

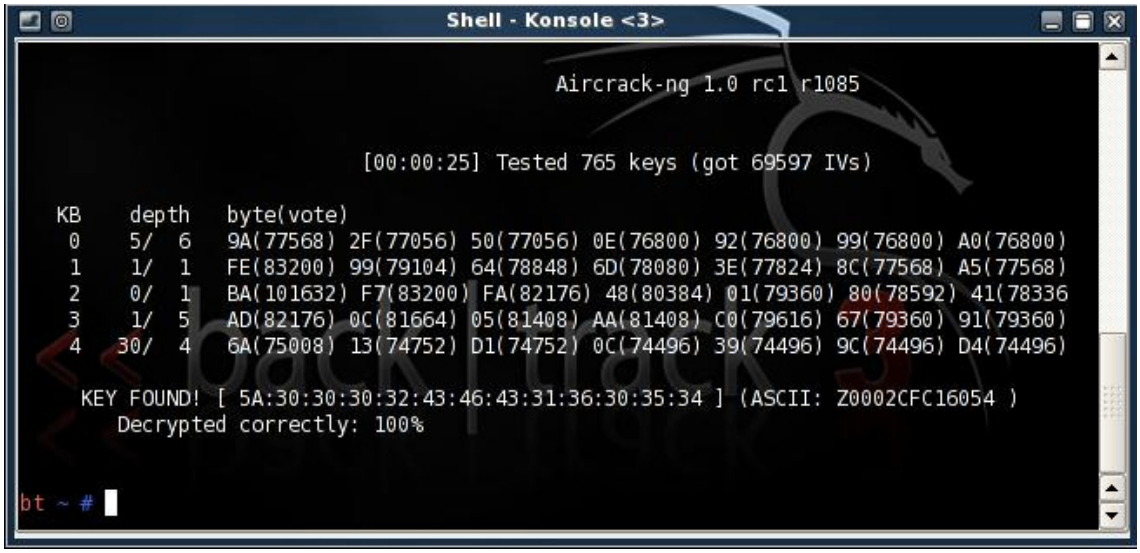
3.1.4.2 Utilización

Para utilizar esta herramienta simplemente tenemos que teclear los siguientes comandos y utilizar las diferentes opciones disponibles (**Captura 3.13**):

```
aircrack-ng [opciones] <archivo(s) de captura>
```

Captura 3.13 Opciones de *aircrack-ng*

Una vez ejecutado el comando y hemos esperado hasta hallar la clave, obtendremos una captura de pantalla similar a esta (**Captura 3.14**):



Captura 3.14 Clave descifrada

donde:

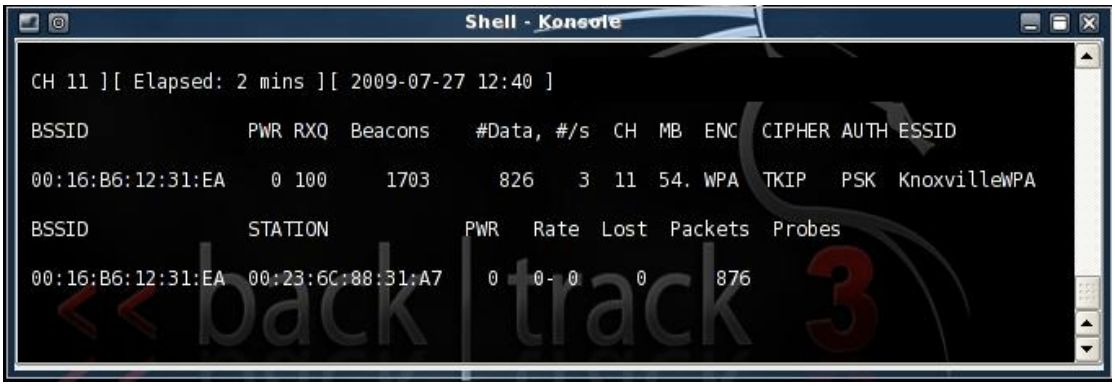
- *KB (KeyByte)* es el número de cada uno de los bytes o caracteres de la clave.
- *Depth* es la profundidad de la actual búsqueda de la clave.
- *Byte* o carácter que se está probando.
- *Vote* son los votos o número de probabilidades que sea correcto ese byte.

3.2 Caso práctico de ataque a WPA mediante aircrack-ng

Al igual que hemos visto en el apartado 3.1 donde se explicaba como atacar al protocolo WEP, ahora vamos a ver como lo haríamos con WPA paso a paso.

1. El primer paso a seguir es seleccionar mediante *airodump-ng* el AP que queremos atacar, filtrando el canal en el que emite y su MAC. En nuestro caso es el AP con ESSID KnoxvilleWPA tal y como observamos en la **Captura 3.15**.

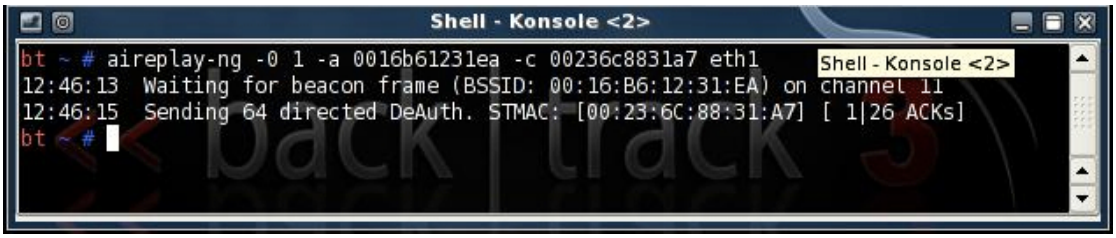
```
Airodump-ng –c 11 --bssid 00:16:b6:12:31:EA –w wpa
                          eth1
```



Captura 3.15 Selección AP mediante *airodump-ng*

2. Es importante que haya un cliente asociado al AP que hemos seleccionado anteriormente para así poder llevar a cabo este paso en el cual mediante la herramienta *aireplay-ng*, deautenticaremos al cliente para forzarlo a que vuelva así a conectarse al AP. Esto lo observamos en la **Captura 3.16**.

```
Aireplay-ng -0 1 –a 00:16:B6:12:31:EA –c
        00:23:6C:88:31:A7 eth1
```

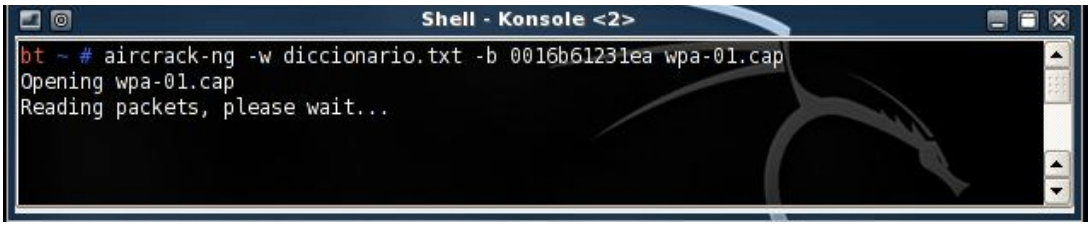


Captura 3.16 Deautenticación del cliente

3. En el paso anterior nos preguntamos el porqué de la deautenticación del cliente con el AP para que de manera automática intente la reconexión. Pues bien, esto se debe a que al forzar la reconexión del cliente con el AP, aprovechamos para capturar el *handshake* (intercambio de datos). Una vez capturado el intercambio, tal y como vemos en la **Captura 3.17**, nos aparece en la parte superior derecha de la ventana *airodump-ng* que tenemos activa.

Captura 3.17 Captura del handshake

4. Disponer del *handshake* entre cliente y AP quiere decir que tenemos el SSID del AP, la MAC de las dos partes y el intercambio de números aleatorios. Una vez recopilada toda esta información, *aircrack-ng* se encarga de analizar los paquetes del intercambio (**Captura 3.18**)



Captura 3.18 Analizando handshake

y mediante un diccionario previamente confeccionado (**Fig. 3.9**), se va comparando palabra por palabra para así poder averiguar el secreto compartido (**Captura 3.19**).

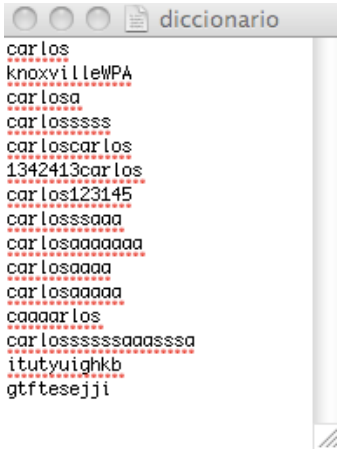
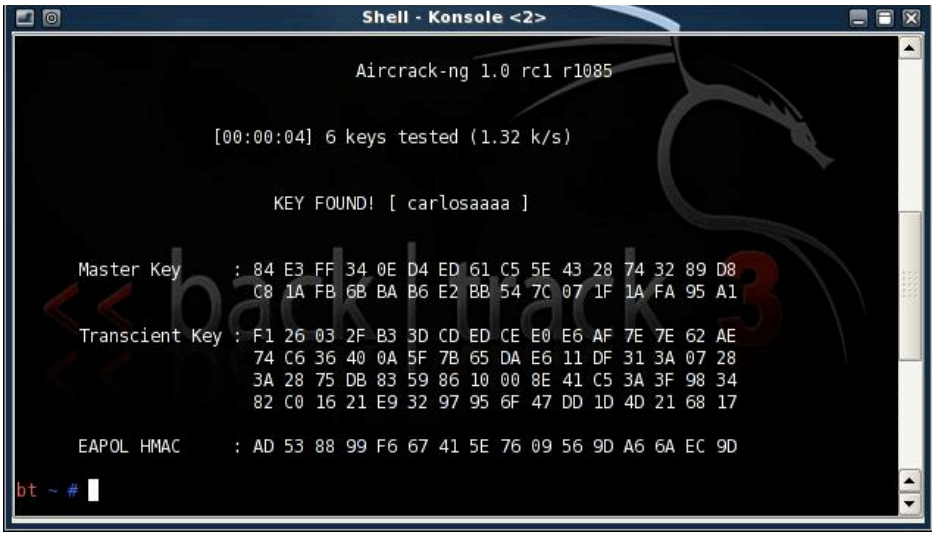


Fig. 3.9 Diccionario



Captura 3.19 Clave compartida encontrada

CAPÍTULO 4. Wardriving

En este último capítulo hablaremos sobre la técnica de auditoria Wi-Fi denominada *wardriving* con el propósito de analizar los protocolos de seguridad que utilizan los usuarios de redes *wireless* del barrio de Can Ribes en Gavà.

4.1 Definición *wardriving*

Los inicios del *wardriving* se remontan a la película Juegos de Guerra (*War Games*) donde un joven utilizaba su módem para llamar a números de teléfono aleatorios y poder acceder a redes, este método se conocía como *wardialing*.

El *wardialing* con la aparición de las redes Wi-Fi pasó a denominarse *wardriving*. Técnica que consiste en la búsqueda de redes *wireless* a bordo de un vehículo en movimiento con la ayuda de un portátil o una PDA para la detección de éstas y un GPS para su localización física.

4.2 Analizando los datos de las redes Wi-Fi

Una vez hemos visto la definición del término *wardriving*, es hora de realizar el estudio de seguridad del cual sacaremos las conclusiones oportunas.

4.2.1 Localización y equipo utilizado

Para realizar este estudio hemos elegido el barrio de Can Ribes, en Gavà. Partiendo de la calle Lluís Solé i Sabarís y recorriendo ≈ 1km^2 tal y como se observa en la **Fig. 4.10.**

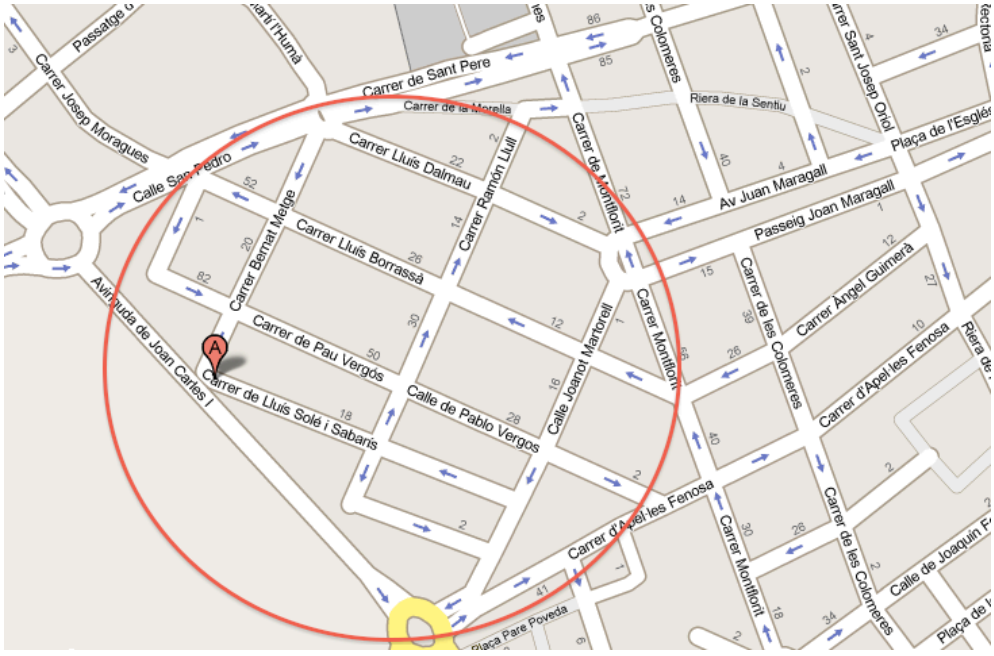


Fig. 4.10 Mapa de acción del *wardriving*

Hemos utilizado el siguiente equipo tal y como se observa en la **Foto 4.1**:

- Portátil con Windows XP.
- Adaptador USB Wi-Fi Edimax EW-7318USg.
- Antena omnidireccional OvisLink WAI-100-B de 10dBi.
- Programa Open Source InSSIDer.

Foto 4.1 Equipo utilizado para *wardriving*

4.2.2 Captura de datos durante el *wardriving*

Para realizar la captura de datos de las redes Wi-Fi, nos montamos en el coche con nuestro portátil y el adaptador USB Wi-Fi + antena omnidireccional con conector SMA. Vamos recorriendo a una velocidad aproximada de 30 Km/h las calles del radio de acción para así poder capturar la señal de todos los APs alcanzables.

La información que nos proporciona el programa *Open Source* InSSIDer[9], es la siguiente (**Fig 4.11**):

Dirección MAC	Vendor	SSID	Canal	RSSI	Seguridad	Tipo de red	Velocida	Primera vista	Último visto
00:22:2d:0b:8a:53	SMC Network...	WLANB8A689	1	-72	WEP	Access Point	54	15:12:50	15:13:31

Fig. 4.11 Información proporcionada por InSSIDer

En este ejemplo se nos muestra:

- Dirección MAC del AP: 00:22:2d:0b:8a:53
- Vendor (Fabricante del AP): SMC Networks
- SSID (Identificador del AP): WLANB8A689
- Canal utilizado por el AP: Canal 1
- RSSI (Fuerza de la señal recibida) : -72 dBm (cuanto más alta mejor)

- Seguridad utilizado por el AP: WEP
- Tipo de red: Access Point
- Velocidad del AP: 54 Mbps

El programa nos va capturando y acumulando todos los APs detectados (hasta un total de 120) como se puede comprobar en el **Anexo 1**.

4.2.3 Estadísticas a partir de los datos recopilados

A continuación presentamos las estadísticas elaboradas a partir de los datos acumulados durante el *wardriving*.

Empezaremos con los protocolos de seguridad (**Gráfico 4.1**) que utilizan los usuarios de redes Wi-Fi del barrio de Can Ribes, observando que la gran mayoría (84%) utilizan WEP, un 12% de ellos utiliza WPA, mientras que hay un 4% que no utiliza ningún tipo de seguridad.

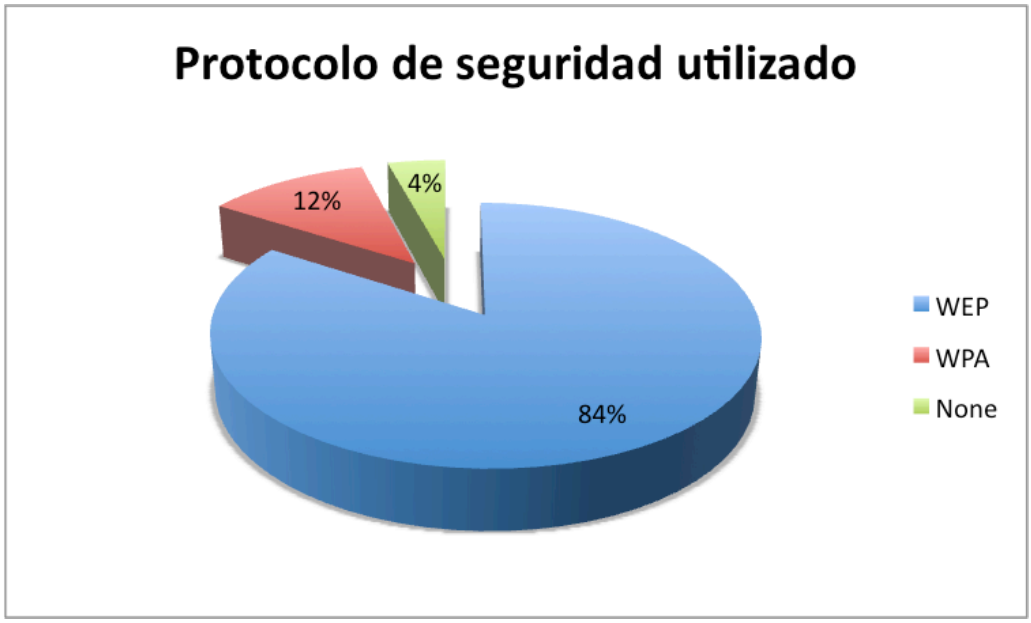


Gráfico 4.1 Seguridad utilizada en APs

De este estudio podemos deducir que tan sólo un 12% de los analizados (los que utilizan WPA) saben realmente, o al menos su ISP o instalador, lo crucial que resulta la seguridad en las conexiones inalámbricas. El valor de la información que puede llegar a circular por ellas debería de concienciar de la utilización de este protocolo o del WPA2. En la figura se observa claramente como el 88% de los usuarios no es consciente de la vulnerabilidad de sus APs al utilizar el protocolo WEP (como hemos demostrado anteriormente en el Capítulo 3) o peor aún operan sin seguridad alguna.

Acto seguido tratamos el SSID de los APs del estudio, es decir, el identificador o nombre del AP de cada usuario. Según podemos observar en el **Gráfico 4.2**, la mayoría de usuarios detectados (62%) utilizan el identificador por defecto de

su operador (en este caso el WLAN_XX que es propio de Telefónica). Esto supone un importante peligro para su seguridad, ya que los APs con el SSID WLAN_XX dan muchas pistas para averiguar la clave WEP si es esta la implementada.

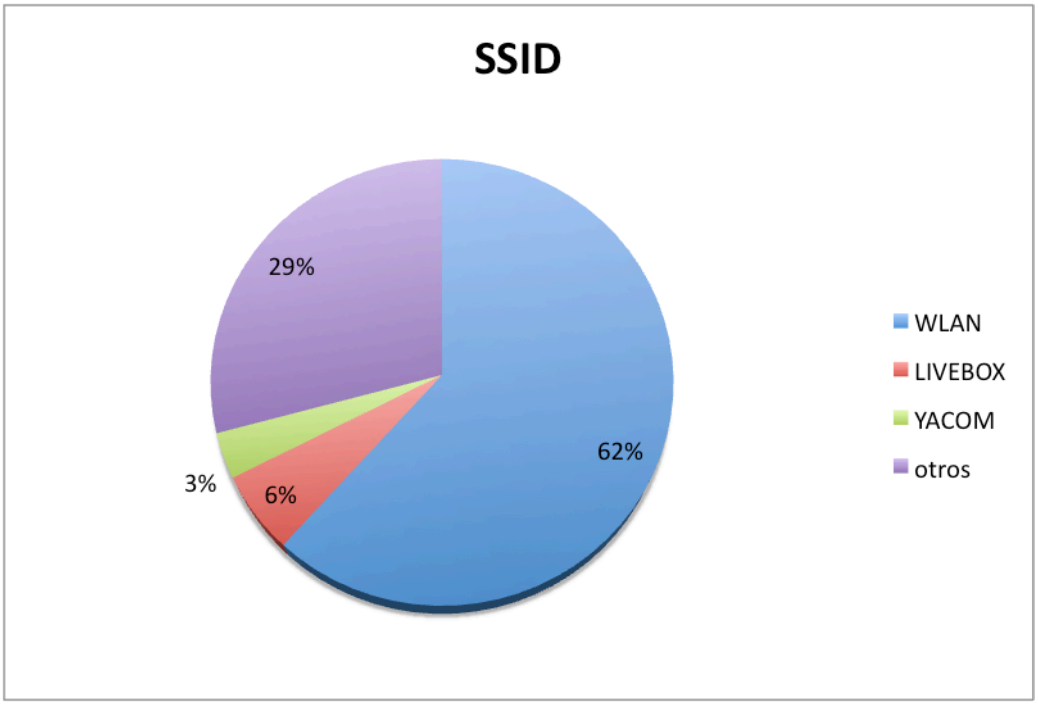


Gráfico 4.2 SSID utilizado por los usuarios

Si por ejemplo tuviésemos los siguientes datos:

- MAC del AP: 00:01:38:c6:44:a1
- Fabricante del router: Xavi Technologies
- SSID: WLAN_4C

Con tan sólo estos datos ya tendríamos el 69,23% de la clave WEP (9 caracteres de 13) que sería la siguiente:

X000138abcd4C

Donde:

X : Es la inicial del fabricante del router (X →Xavi , Z→Zyxel).
000138: Son los 3 primeros valores de la MAC, que corresponden al fabricante Xavi (Para cada fabricante son únicos).
abcd: Son los únicos caracteres que tendríamos que resolver mediante fuerza bruta.
4C: Se corresponden con los dos últimos caracteres del SSID.

Para descubrir los 4 caracteres que nos faltan, seria suficiente con la captura de varios paquetes y probar a descifrarlos recorriendo todas las combinaciones posibles. Con un simple ataque de fuerza bruta bastaría para sacarlos.

Para terminar con el estudio, evaluaremos el uso de los diferentes canales de frecuencia que utilizan los usuarios detectados (**Gráfico 4.3**).

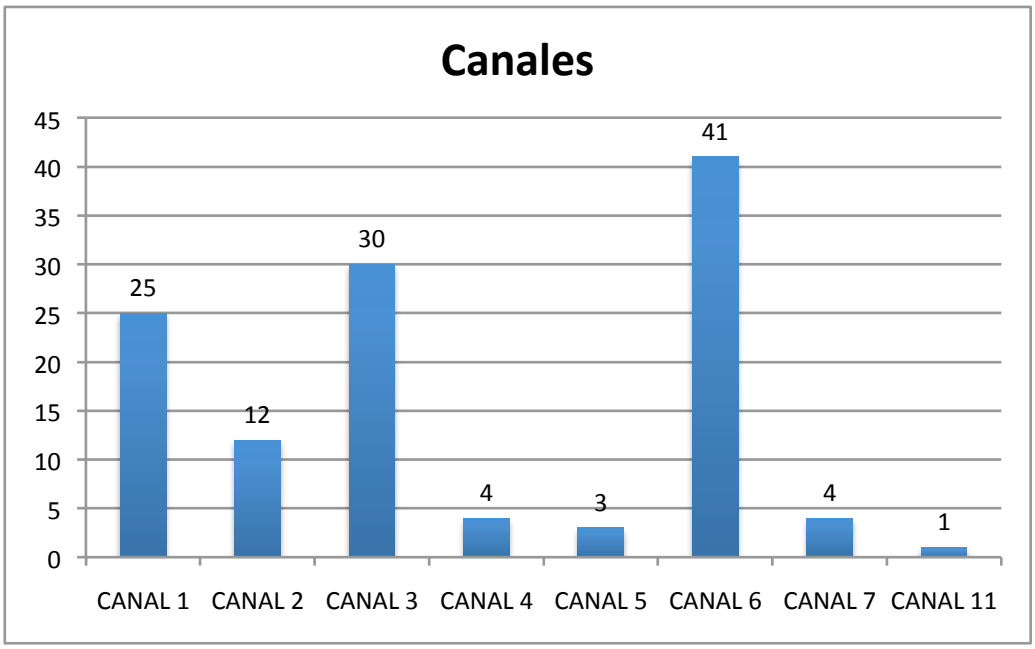


Gráfico 4.3 Canales utilizados por los usuarios

Como podemos observar en el **Gráfico 4.3** los canales más utilizados por los usuarios analizados a lo largo de todo el *wardriving* son el 1, el 3 y el 6. Con estos resultados podemos llegar a deducir que muchos usuarios utilizan el canal por defecto de su router Wi-Fi (la mayoría el canal 1 o el canal 6), aunque lo realmente interesante de analizar a partir de estos resultados son las interferencias que se puedan llegar a provocar entre usuarios que utilizan canales contiguos.

En el estándar 802.11 la banda de frecuencias de transmisión empieza en 2,4 GHz y admite un total de 13 canales en Europa, separados 5MHz entre ellos. Tal y como vemos en la **Fig. 4.12** cada canal tiene un lóbulo principal (donde se concentra la mayor parte de potencia de la señal) a frecuencia *fc* de 22MHz de ancho.

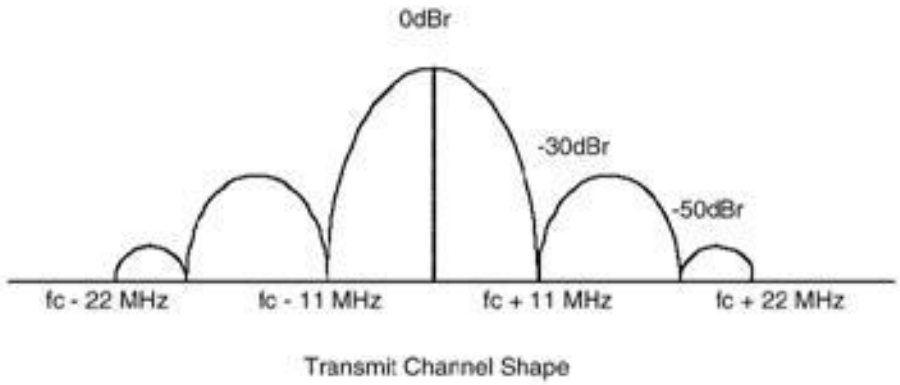


Fig. 4.12 Modulación canal Wi-Fi.

Teniendo en cuenta lo explicado, deducimos que los canales adyacentes presentan interferencias entre ellos y que los canales óptimos para transmitir son el 1, el 6 y el 11, separados 25 MHz entre ellos tal y como vemos en la **Fig 4.13**.

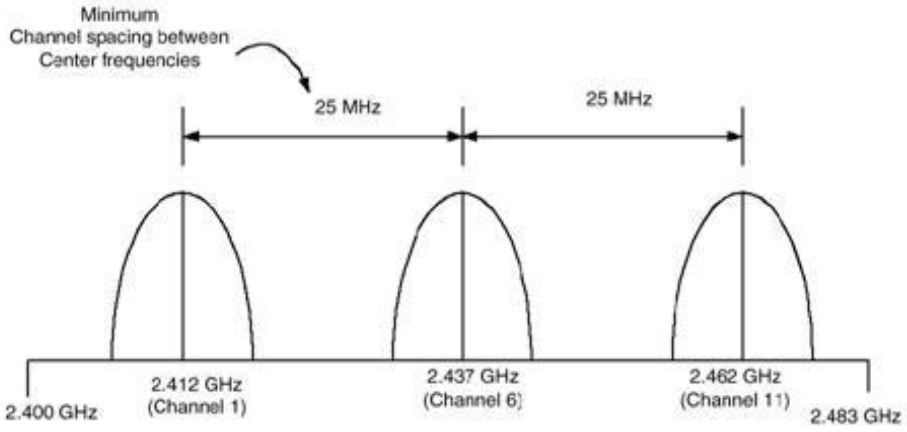


Fig. 4.13 Canales óptimos para la transmisión

Lo ideal en cuanto a la distribución de canales sería que los usuarios que estuviesen en el mismo radio de cobertura utilizasen canales lo más separados posibles entre ellos o en su defecto el mismo canal. Pese a la creencia popular, el uso de un canal libre puede ser mucho peor en términos de eficiencia que compartir canal. En el primer caso, podríamos tener interferencias de los canales adyacentes mientras que en el segundo el mecanismo de control de acceso al medio facilitará a las dos redes el uso compartido del canal. Por ejemplo, si encontramos libre el canal 2 y el 1 está ocupado, el rendimiento en el 2 se verá tan afectado por la interferencia (ver **Fig. 4.12**) que sería mejor compartir el canal 1. De todas formas, lo ideal es utilizar un canal libre lo más separado posible tal y como se indica en la **Fig. 4.13**.

4.2.4 Recomendaciones de seguridad

A partir del análisis de los protocolos de seguridad presentados en este TFC y de los datos obtenidos del despliegue WLAN en el barrio de Can Ribes de Gavà, creemos necesario establecer las siguientes recomendaciones de seguridad para cualquier usuario:

1. Utilizar el protocolo WPA en cualquier caso (es el más seguro como hemos visto con anterioridad).

2. Cambiar la clave WPA que viene por defecto en el AP. Utilizando una clave con caracteres, números y símbolos con tal de hacerla criptográficamente imposible de vulnerar por fuerza bruta.

Siguiendo estas dos recomendaciones básicas, se puede garantizar al usuario que su red Wi-Fi es al fin “segura”, o al menos tan segura como lo sería una red cableada.

CAPÍTULO 5. Conclusiones

Finalmente, vamos a concluir este TFC extrayendo conclusiones sobre los resultados obtenidos, mencionando posibles líneas futuras y realizando un pequeño estudio sobre el impacto de nuestro proyecto en el medio ambiente.

5.1 Conclusiones

En este TFC hemos llevado a cabo una auditoria de los protocolos de seguridad Wi-Fi: WEP y WPA, haciendo uso de programas Open Source.

Como ya hemos comentado a lo largo del TFC, la seguridad en las conexiones *wireless* juega un papel crucial cual paraguas en pleno diluvio. Y no hay mejor manera de evaluar la seguridad en redes *wireless* que intentar vulnerarla por todos los medios. Igual que no hay mejor forma de probar un buen paraguas que salir con él cuando cae un buen chaparrón.

En el capítulo 1 y 2 hemos abordado temas teóricos sobre la evolución de los estándares 802.11 y más concretamente WEP, WPA y 802.11i. Acto seguido se han estudiado las vulnerabilidades de dichos protocolos para después pasar a la parte práctica, el tercer capítulo, donde hemos realizado diferentes tipos de ataques activos para tratar de sacar claves WEP y ataques por fuerza bruta para WPA. El programa elegido para esta evaluación ha sido *aircrack-ng*, una suite muy completa para realizar auditorias Wi-Fi que forma parte de un *live CD* de Linux llamado *Backtrack 3* (software libre).

De los ataques, se concluye que WEP es un protocolo totalmente ineficiente debido sobre todo a la debilidad del vector de inicialización. Por otro lado WPA es mucho menos vulnerable siempre y cuando no se utilice una clave por defecto.

A parte de los ataques realizados, hemos querido realizar un estudio estadístico del tipo de seguridad que se utiliza en las redes Wi-Fi, para el cual se ha escogido un barrio de la población de Gavà. La herramienta elegida para obtener la información acerca de estas redes ha sido InSSIDer (*Open Source*), herramienta muy útil que nos ha proporcionado todos los datos necesarios para realizar el análisis.

El estudio realizado mediante la técnica del *wardriving* nos ha mostrado cifras muy curiosas: el 84% utilizaba WEP, el 12% WPA y el resto, el 4% ningún tipo de protección. A parte de analizar los protocolos utilizados, también hemos creído conveniente analizar otros datos como por ejemplo los SSIDs, de los cuales un 62% correspondían al SSID por defecto WLAN_XX, muy fácil de vulnerar.

Una vez realizado el estudio, se llega a la conclusión de que la mayor parte de usuarios de redes Wi-Fi no son conscientes de la importancia de una buena utilización de la seguridad en sus APs. De ella depende que posibles atacantes puedan o no, interpretar la información interceptada, y a través de ésta acceder

a: transacciones bancarias, contraseñas, e-mails, números de tarjetas de crédito, etc.

Durante la realización de este TFC, nos hemos encontrado con algunas dificultades. *Aircrack-ng* es una suite muy compleja con un sinfín de opciones a la hora de realizar los ataques activos, para atacar cada AP necesitábamos parámetros diferentes dependiendo de la configuración del router. También nos encontramos con la problemática de que al hacer el *wardriving,* la antena de serie que llevaba el adaptador USB Wi-Fi no era lo suficientemente potente para detectar un mínimo de APs para hacer una estadística convincente, así que decidimos colocarle una antena de 10dBi para ampliar la cobertura.

A pesar de las pequeñas dificultades sufridas, se han cumplido los objetivos de este TFC: la auditoria de seguridad de las redes Wi-Fi y la concienciación de la necesidad de una buena implementación y correcto uso de los mecanismos de seguridad en las redes inalámbricas.

5.2 Líneas futuras

Una vez finalizado este TFC y realizadas todas las pruebas de vulnerabilidad de los protocolos de seguridad existentes hoy en día, lo único ampliable a nivel de cálculo computacional para vulnerar WPA, es hablar de CUDA [10].

CUDA (*Compute Unified Device Architecture*), tecnología creada por NVIDIA, es un entorno basado en lenguaje C que permite resolver problemas computacionales complejos (fuerza bruta en WPA y WPA2) en menos tiempo aprovechando la gran capacidad de procesamiento de las GPUs (Unidad de Procesamiento Gráfico) de las tarjetas gráficas NVIDIA.

En el caso de WPA, como hemos visto en el apartado *3.2 Caso práctico de ataque a WPA mediante aircrack-ng*, el único método para adivinar la clave es la fuerza bruta o el ataque de diccionario. Con la ayuda de CUDA, y en especial de las tarjetas gráficas NVIDA (mediante la utilización de sus procesadores – GPUs) este proceso puede llegar a ser 10.000 veces más rápido que con la utilización única de CPU del propio PC.

Otra posible ampliación de este TFC hubiese sido, como ya hemos comentado en el apartado de *wardriving*, la utilización de un GPS bluetooth para situar en un mapa todos los APs auditados gracias a los archivos *.gpx* que genera InSSIDer para luego exportarlos a Google Earth.

5.3 Estudio de ambientalización

El único punto negro que podemos encontrar en este TFC, referente a contaminación medioambiental, es la polución electromagnética producida por la transmisión Wi-Fi. Todos los dispositivos con los que hemos trabajado no

han superado los 100mW de potencia, por lo tanto no podemos considerar que su radiación sea dañina o peligrosa para el medio ambiente.

Podemos asegurar que estamos ante una tecnología utilizada a nivel mundial por millones de personas que no reviste peligro alguno en términos de radiación y que, por lo tanto, no refleja ninguna problemática destacada en el campo medioambiental.

REFERENCIAS

[1] Enciclopedia Wikipedia : 802.11 legacy
Web http://es.wikipedia.org/wiki/IEEE_802.11#802.11_legacy

[2] Página Código Unix sección Wireless : 802.11a
Web http://codigounix.com.ar/?p=247

[3] Definición e historia de 802.11g
Web http://es.wikipedia.org/wiki/802.11g#802.11g

[4] Enciclopedia Wikipedia : 802.11e
Web http://es.wikipedia.org/wiki/802.11e#802.11e

[5] Enciclopedia Wikipedia : algoritmo MIC o “Michael”
Web http://en.wikipedia.org/wiki/Message_authentication_code

[6] Protocolo de seguridad TKIP
Web http://en.wikipedia.org/wiki/Temporal_Key_Integrity_Protocol

[7] Explicación servidores RADIUS
Web http://en.wikipedia.org/wiki/RADIUS

[8] Página ofial de la suite *aircrack-ng*
Web http://www.aircrack-ng.org/

[9] Página oficial programa Open Source InSSIDer
Web http://www.metageek.net/products/inssider

[10] Página oficial de CUDA
Web http://www.nvidia.es/object/cuda_what_is_es.html

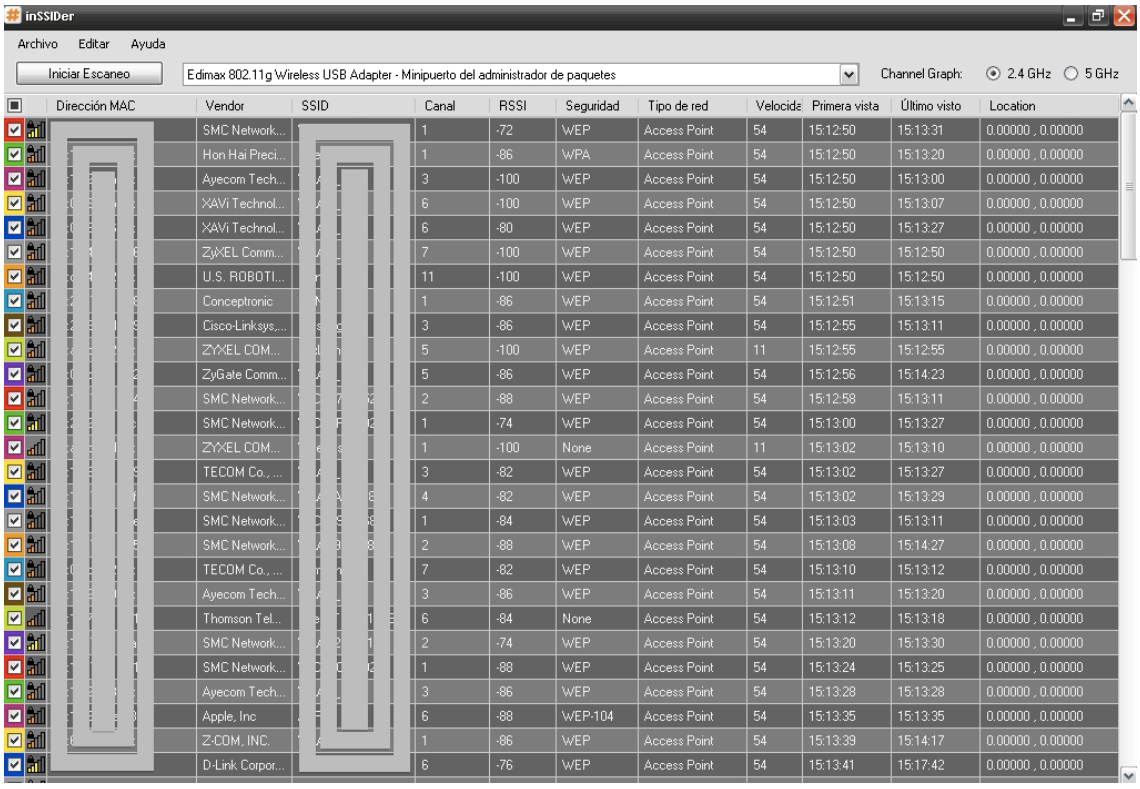
ANEXOS

A continuación se muestran las capturas de pantalla de InSSIDer durante el wardriving.

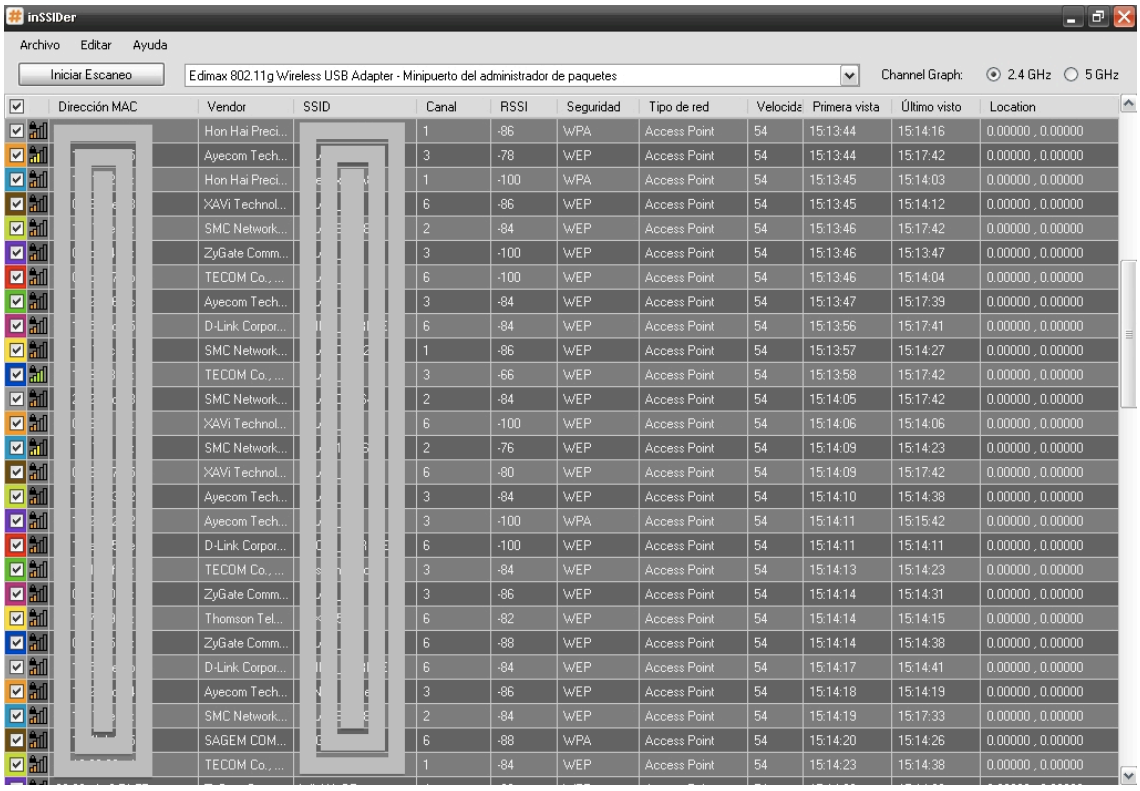
Anexo 1 : Capturas de InSSIDer

Las capturas que vemos a continuación (**Captura 1.1**, **Captura 1.2**, **Captura 1.3**, **Captura 1.4** y **Captura 1.5**) nos muestran todos los datos (hasta un total de 120 AP's) recopilados durante el *wardriving* por el barrio de Can Ribes de Gavà. Estos datos mostrados son: la dirección MAC del AP, el fabricante del AP, el SSID, el canal utilizado, la potencia de la señal del AP (RSSI), el tipo de seguridad, la velocidad, la localización etc.

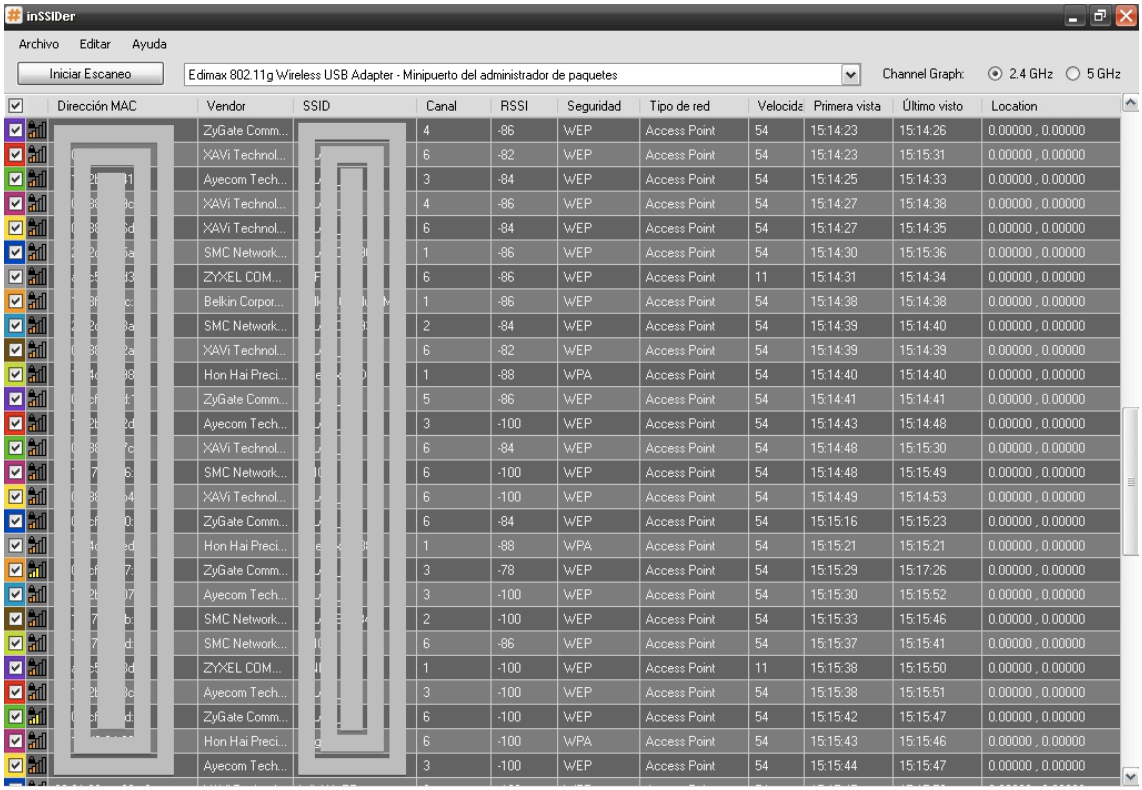
Las pruebas realizadas antes del estudio definitivo se realizaron con el equipo mostrado en el *4.2.1 Localización y equipo utilizado* y un GPS bluetooth. Para salvaguardar la privacidad de la posición de los AP's decidimos a última hora no utilizar este artilugio, así como borrar las direcciones MAC y el SSID de los mismos.



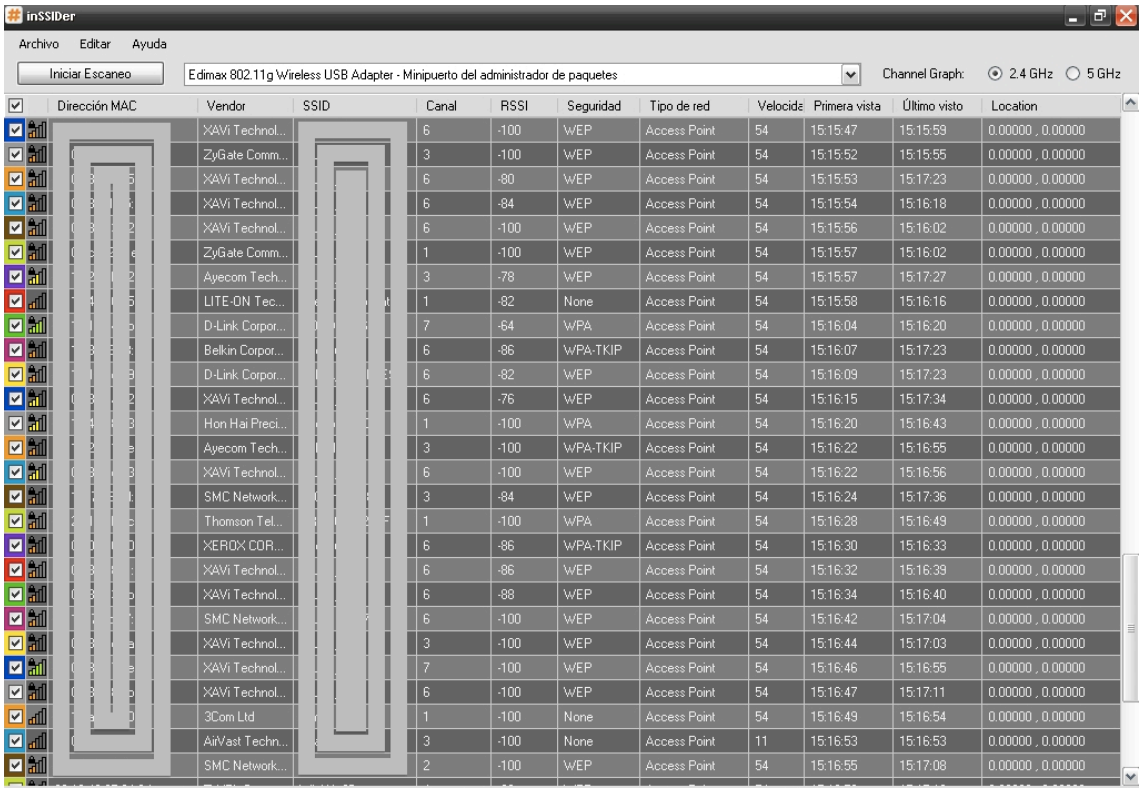
Captura 1.1 Datos recopilados 1



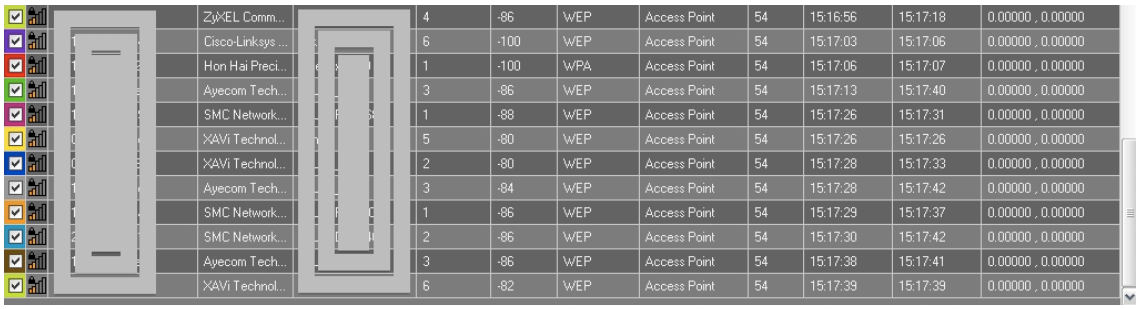
Captura 1.2 Datos recopilados 2



Captura 1.3 Datos recopilados 3



Captura 1.4 Datos recopilados 4



Captura 1.5 Datos recopilados 5