INDEX

Practical No: 1

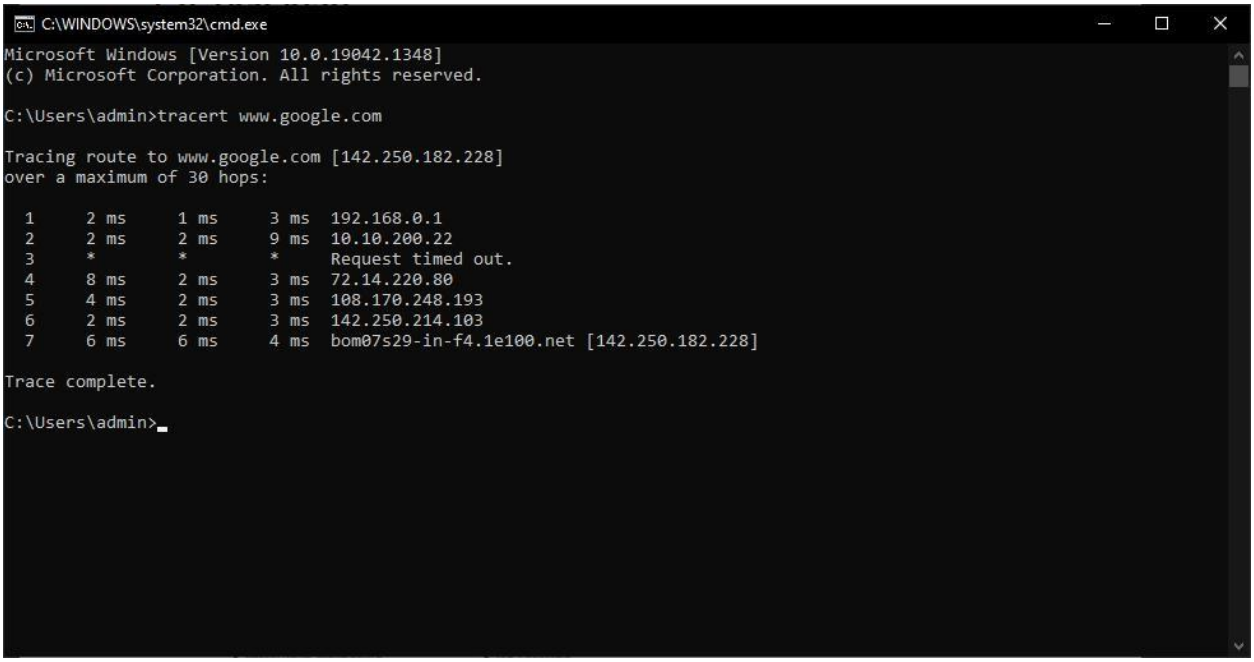
Ip Tracing:

An IP address (Internet Protocol address) is a unique numerical label assigned to a device. It provides the location of the device in a network and a route on how to get there. The internet uses an IP address to send IP packets from a source to a destination. It is a building block that lets the internet function.

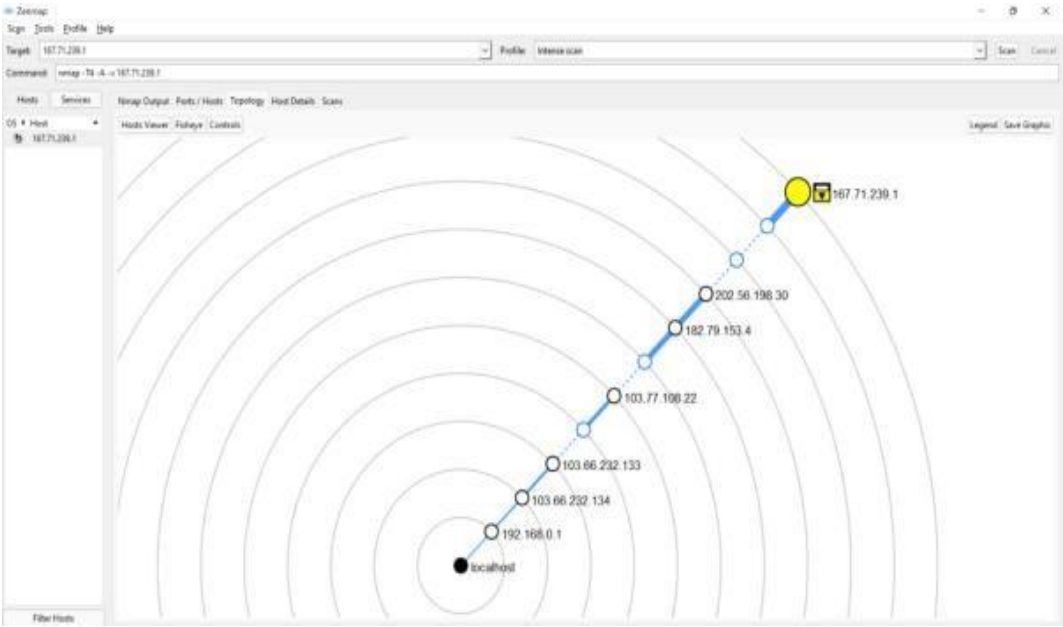
An ip address does not reveal personal information (like a name, social security number or physical address). Millions of devices, like modems and routers keep logs of ip addresses. Your modem at home, or the 4G antennae you connect to with your phone are logging your ip addresses. Logs are necessary to maintain the internet. Logs with IP addresses are everywhere!

Our IP tracker uses the IP address to identify and collect online details based on the IP number. Combined with cookies, Opentracker lets you enrich, view, download and process IP tracker data.

Opentracker records each unique user and their IP address. Our IP tracer maps where an IP address (and the visitor behind it) originates from, and enriches this data with different sources.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>tracert www.google.com

Tracing route to www.google.com [142.250.182.228]
over a maximum of 30 hops:

  1     2 ms     1 ms     3 ms  192.168.0.1
  2     2 ms     2 ms     9 ms  10.10.200.22
  3     *        *        *     Request timed out.
  4     8 ms     2 ms     3 ms  72.14.220.80
  5     4 ms     2 ms     3 ms  108.170.248.193
  6     2 ms     2 ms     3 ms  142.250.214.103
  7     6 ms     6 ms     4 ms  bom07s29-in-f4.1e100.net [142.250.182.228]

Trace complete.

C:\Users\admin>
```

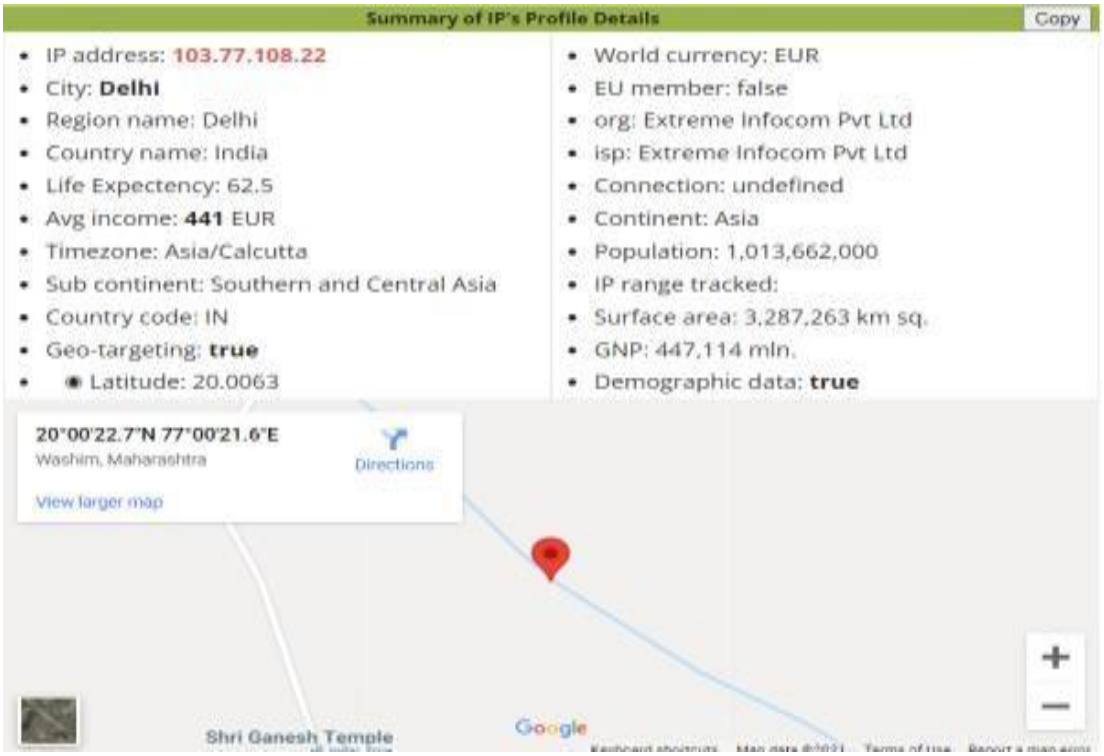


1. First we track ip of 192.168.0.1

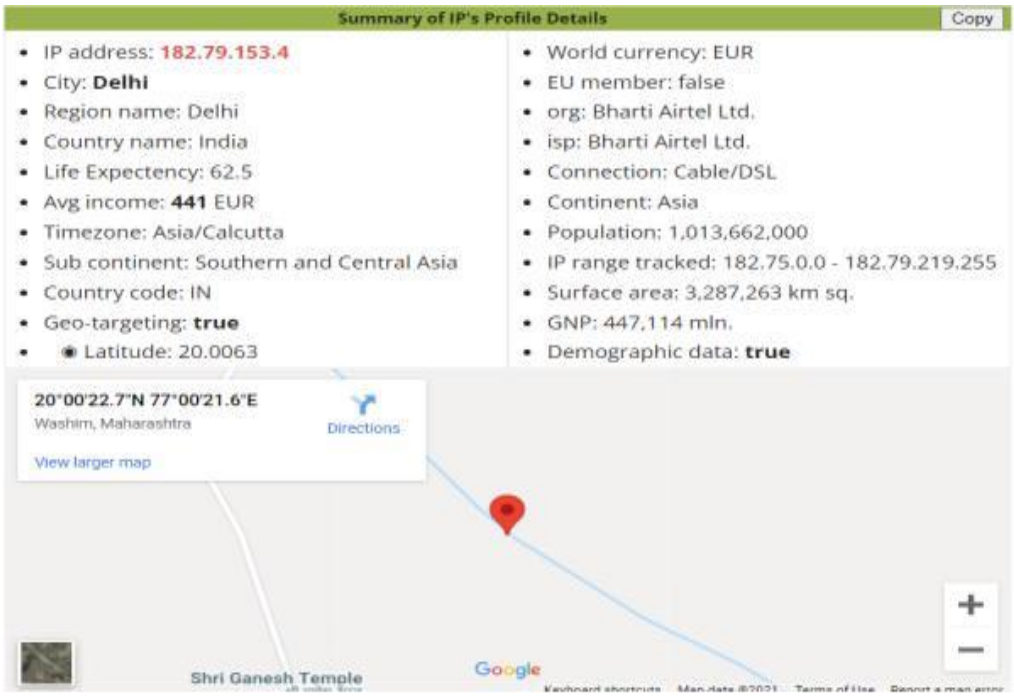
We cannot get details because it private.

2. Now track 103.66.232.134 and 103.66.232.133

3. Now track 103.77.108.22



4. Now track 182.79.153.4



5. Now track 202.56.198.30

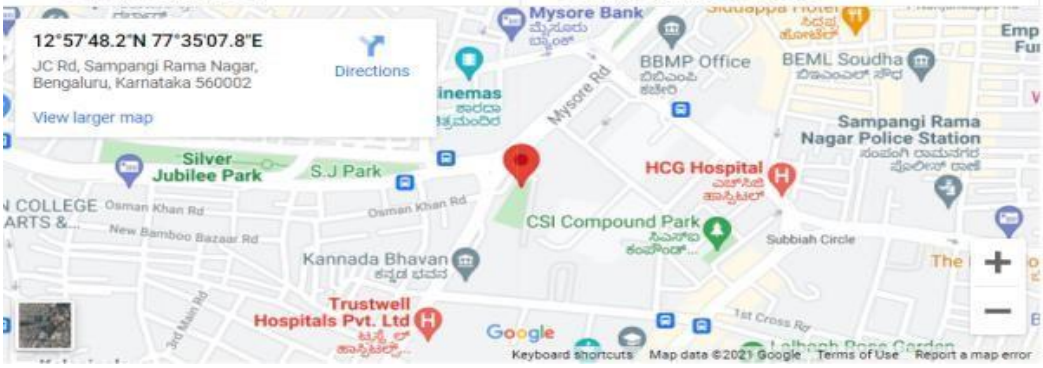
Summary of IP's Profile Details Copy

- IP address: **202.56.198.30**
- City: **Coimbatore**
- Region name: Tamil Nadu
- Country name: India
- Life Expectency: 62.5
- Avg income: **441** EUR
- Timezone: Asia/Calcutta
- Sub continent: Southern and Central Asia
- Country code: IN
- Geo-targeting: **true**
- Latitude: 11.0142
- World currency: EUR
- EU member: false
- org: Airtel
- isp: Airtel
- Connection: Cable/DSL
- Continent: Asia
- Population: 1,013,662,000
- IP range tracked: 202.56.197.0 - 202.56.198.255
- Surface area: 3,287,263 km sq.
- GNP: 447,114 mln.

6. Now the final destination 167.71.239.1

Summary of IP's Profile Details Copy

- IP address: **167.71.239.1**
- City: **Bengaluru**
- Region name: Karnataka
- Country name: India
- Life Expectency: 62.5
- Avg income: **441** EUR
- Timezone: Asia/Calcutta
- Sub continent: Southern and Central Asia
- Country code: IN
- Geo-targeting: **true**
- Latitude: 12.9634
- World currency: EUR
- EU member: false
- org: Digital Ocean
- isp: Digital Ocean
- Connection: undefined
- Continent: Asia
- Population: 1,013,662,000
- IP range tracked: 167.71.0.0 - 167.71.255.255
- Surface area: 3,287,263 km sq.
- GNP: 447,114 mln.
- Demographic data: **true**



Practical No: 2

Vulnerability scanning with NMAP :

1. – TCP Connect Port Scan

TCP connect scan is the default TCP scan type when SYN scan is not an option. This is the case when a user does not have raw packet privileges or is scanning IPv6 networks. The TCP Connect Scan is a simple probe that attempts to directly connect to the remote system without using any stealth.

SYNTAX: nmap -sT <IP Address>

```
oem@linux: ~

Govind@linux:~$ sudo nmap -sT 104.21.71.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-08 15:34 IST
Nmap scan report for 104.21.71.20
Host is up (0.0049s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
443/tcp  open  https
8080/tcp open  http-proxy
8443/tcp open  https-alt

Nmap done: 1 IP address (1 host up) scanned in 4.87 seconds
```

2. – TCP SYN Port Scan

This type of scan won't establish a TCP connection. It will scan by sending a SYN flag packet and if the port is open, then a SYN/ACK will be send back as a response by the target machine, thus result in a half embryo connection. Since a full connection won't establish, the connection info will not be logged by the Firewalls/IDSs and hence it is widely known as Stealth scan. If RST pack is received as a response, then probably the post is closed.

SYNTAX: nmap -sS <IP Address>

```
Govind@linux:~$ sudo nmap -sS 104.21.71.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-08 15:35 IST
Nmap scan report for 104.21.71.20
Host is up (0.0033s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
443/tcp  open  https
8080/tcp open  http-proxy
8443/tcp open  https-alt

Nmap done: 1 IP address (1 host up) scanned in 4.68 seconds
```

3. – Version Scan

Version Detection collects information about the specific service running on an open port, including the product name and version number. This information can be used in determining an

entry point for an attack. The –sV option enables version detection, and the -A option enables both OS fingerprinting and version detection.

SYNTAX: nmap -sV <IP Address>

```
Govind@linux:~$ sudo nmap -sV 104.21.71.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-08 15:35 IST
NSOCK ERROR [21.3610s] mksock_bind_addr(): Bind to 0.0.0.0:631
Nmap scan report for 104.21.71.20
Host is up (0.0043s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE       VERSION
80/tcp   open  http          cloudflare
443/tcp  open  ssl/https     cloudflare
8080/tcp open  http-proxy    cloudflare
8443/tcp open  ssl/https-alt cloudflare

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.41 seconds
```

4. – UDP Port Scan

UDP scan works by sending a UDP packet to the targeted port. If noresponse is received, then the port will be considered as Open | filtered. Filtered because some firewalls won't respond to the blocked UDP ports. If the port is closed, then an ICMP response (ICMP port unreachable error type 3, code 3) will be send by the target device.

SYNTAX: nmap -sU <IP Address>

```
Govind@linux:~$ sudo nmap -sU 104.21.71.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-08 15:36 IST
Nmap scan report for 104.21.71.20
Host is up (0.0037s latency).
Not shown: 999 open|filtered ports
PORT      STATE  SERVICE
33459/udp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 11.30 seconds
```

5. – OS Fingerprinting

With -O (Capital O) or –osscan-guess, you can easily detect the target Operating System behind it using TCP/IP stack fingerprinting. Nmap sends a series of TCP and UDP packets to the remote host and examines the responses. After performing dozens of tests, Nmap compares the results to its database and prints out the OS details if there is a match.

SYNTAX: nmap – O <IP Address>

```
Govind@linux:~$ sudo nmap -O 104.21.71.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-08 15:37 IST
Nmap scan report for 104.21.71.20
Host is up (0.0043s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
443/tcp  open  https
8080/tcp open  http-proxy
8443/tcp open  https-alt
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: phone
Running (JUST GUESSING): Google Android 6.X|7.X (85%), Linux 3.X|4.X (85%)
OS CPE: cpe:/o:google:android:6 cpe:/o:google:android:7 cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
Aggressive OS guesses: Android 6.0 - 7.1.2 (Linux 3.18 - 4.4.1) (85%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.11 seconds
```

6. – Scan OS Information

With Nmap, you can detect which OS and version is running on the remote host. To enable OS & version detection, script scanning and traceroute, you can use "-A" option with NMAP. This type of scan uses the ACK flags. Unlike other scans, ACK scan is not used to determine whether the port is Open or Closed. It is used to map out firewall rulesets, determining whether they are stateful or not and which ports are filtered. Stateful Firewalls, will respond with RST packet as the sequence is not in order.

SYNTAX: nmap -A <IP Address>

```
Govind@linux:~$ sudo nmap -A 104.21.71.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-08 15:37 IST
Nmap scan report for 104.21.71.20
Host is up (0.0039s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE       VERSION
80/tcp   open  http          cloudflare
|_http-server-header: cloudflare
|_http-title: Site doesn't have a title (text/plain; charset=UTF-8).
443/tcp  open  ssl/https     cloudflare
|_http-server-header: cloudflare
|_http-title: 400 The plain HTTP request was sent to HTTPS port
8080/tcp open  http-proxy    cloudflare
|_http-server-header: cloudflare
|_http-title: Site doesn't have a title (text/plain; charset=UTF-8).
8443/tcp open  ssl/https-alt cloudflare
|_http-server-header: cloudflare
|_http-title: 400 The plain HTTP request was sent to HTTPS port
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: phone
Running (JUST GUESSING): Google Android 6.X|7.X (85%), Linux 3.X|4.X (85%)
OS CPE: cpe:/o:google:android:6 cpe:/o:google:android:7 cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
Aggressive OS guesses: Android 6.0 - 7.1.2 (Linux 3.18 - 4.4.1) (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 6 hops

TRACEROUTE (using port 443/tcp)
HOP RTT      ADDRESS
1   3.09 ms  _gateway (192.168.0.1)
2   1.42 ms  10.10.200.22
3   ...
4   2.95 ms  114.79.130.57.dvois.com (114.79.130.57)
5   17.72 ms 103.27.170.48
6   2.78 ms  104.21.71.20

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 206.81 seconds
```

7. – Scan Top Ports

Instead of scanning as many ports as the default scan does, the fast scan only scans a few.

SYNTAX: nmap -F <IP Address>

```
Govind@linux:~$ sudo nmap -F 104.21.71.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-08 15:42 IST
Nmap scan report for 104.21.71.20
Host is up (0.0043s latency).
Not shown: 96 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
443/tcp  open  https
8080/tcp open  http-proxy
8443/tcp open  https-alt

Nmap done: 1 IP address (1 host up) scanned in 1.84 seconds
```

9. – TCP ACK Port Scan

This type of scan uses the ACK flags. Unlike other scans, ACK scan is not used to determine whether the port is Open or Closed. It is used to map out firewall rule-sets, determining whether they are stateful or not and which ports are filtered. Stateful Firewalls, will respond with a RST packet as the sequence is not in order.

SYNTAX: nmap -sA <IP Address>

```
Govind@linux:~$ sudo nmap -sA 104.21.71.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-08 15:46 IST
Nmap scan report for 104.21.71.20
Host is up (0.023s latency).
Not shown: 996 filtered ports
PORT     STATE      SERVICE
80/tcp   unfiltered http
443/tcp  unfiltered https
8080/tcp unfiltered http-proxy
8443/tcp unfiltered https-alt

Nmap done: 1 IP address (1 host up) scanned in 9.75 seconds
```

10. – Ping Scan

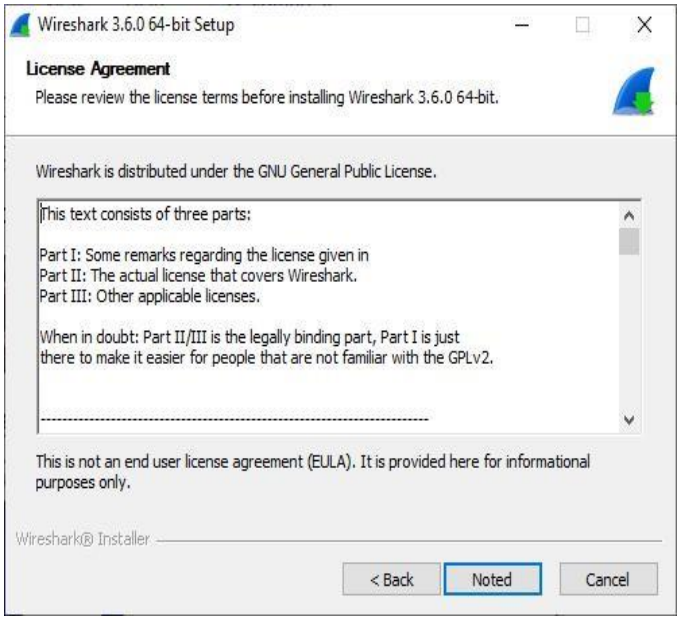
This type of scan is used to detect which computers or devices are online, rather than which ports are open. In this, Nmap sends an ICMP ECHO REQUEST packet to the destination system. If an ICMP ECHO REPLY is received, the system is considered as up, and ICMP packets are not blocked. If there is no response to the ICMP ping request, Nmap will try a “TCP Ping”, to determine whether ICMP is blocked, or if the host is really not online.

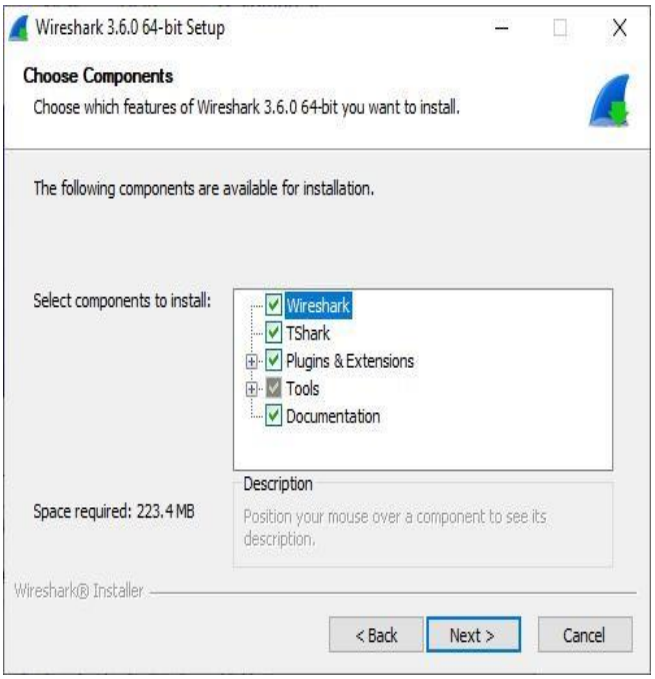
SYNTAX: nmap -sP <IP Address>

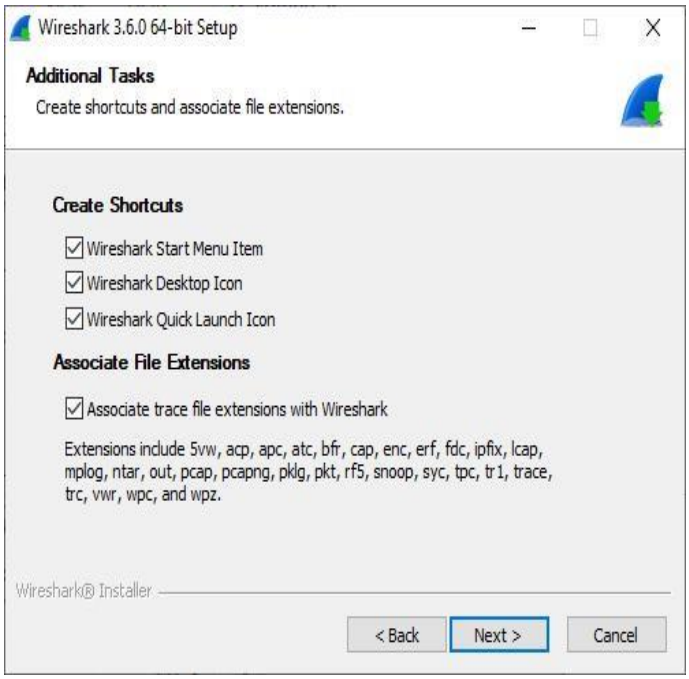
```
Govind@linux:~$ sudo nmap -sP 104.21.71.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-08 15:47 IST
Nmap scan report for 104.21.71.20
Host is up (0.0056s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

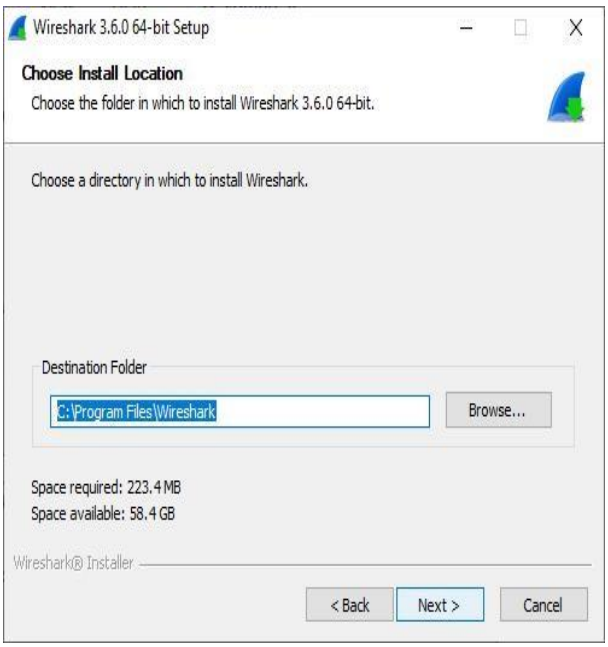
Practical No: 3

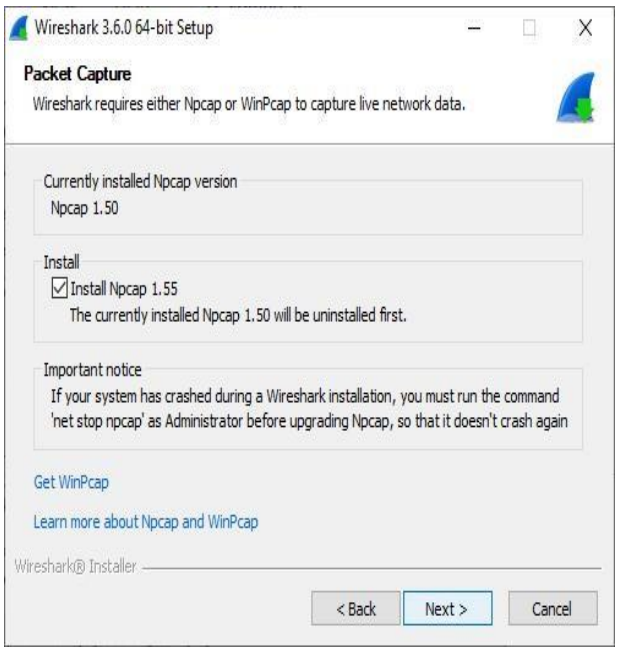
WireShark:

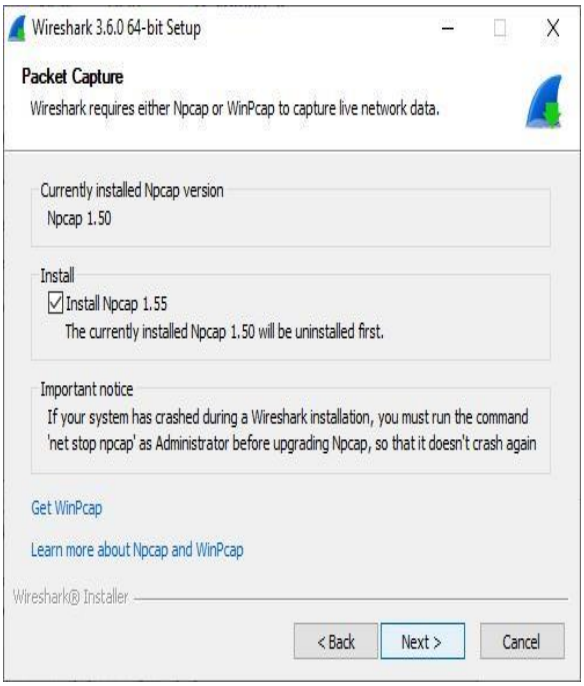


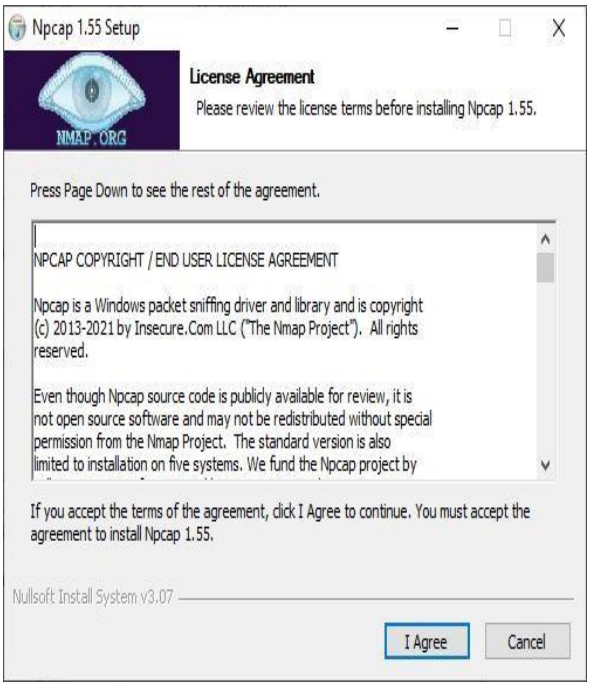


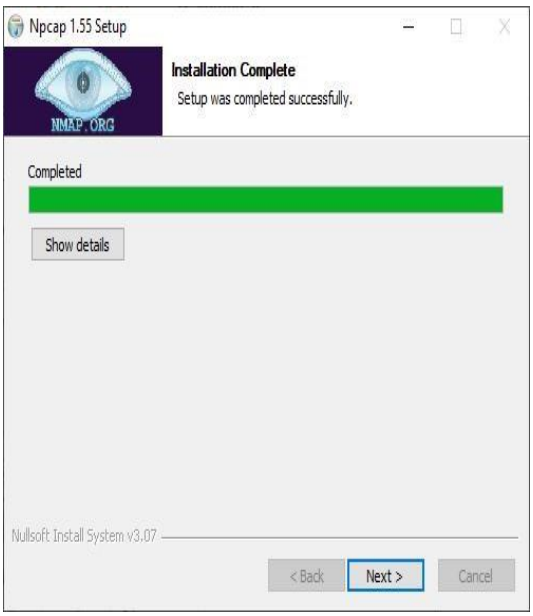


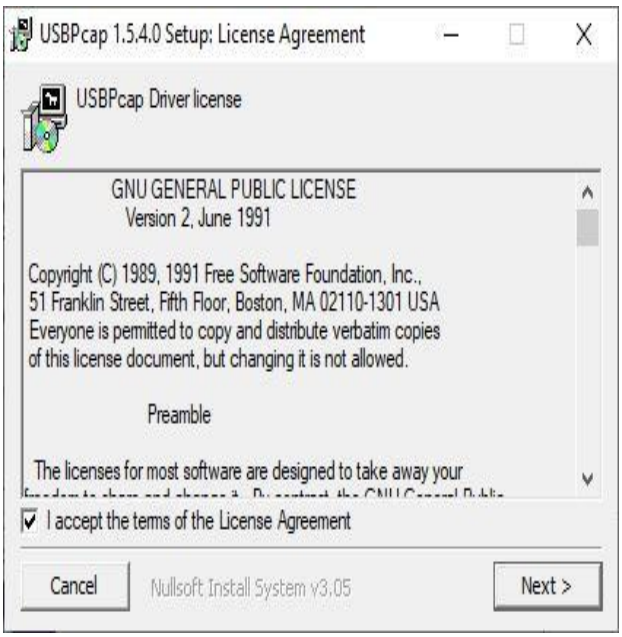

Wireshark 3.6.0 64-bit Setup
Choose Install Location
Choose the folder in which to install Wireshark 3.6.0 64-bit.
Choose a directory in which to install Wireshark.
Destination Folder
C:\Program Files\Wireshark
Browse...
Space required: 223.4 MB
Space available: 58.4 GB
Wireshark® Installer
< Back
Next >
Cancel

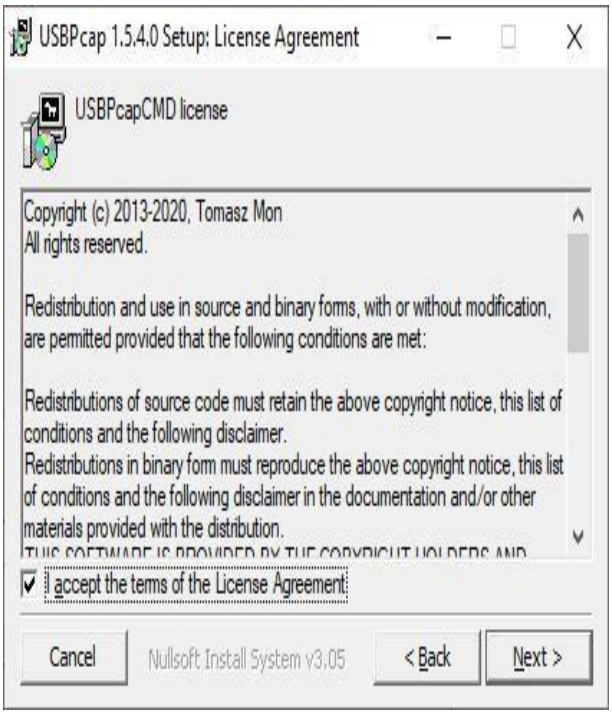

Wireshark 3.6.0 64-bit Setup
Packet Capture
Wireshark requires either Npcap or WinPcap to capture live network data.
Currently installed Npcap version
Npcap 1.50
Install
Install Npcap 1.55
The currently installed Npcap 1.50 will be uninstalled first.
Important notice
If your system has crashed during a Wireshark installation, you must run the command 'net stop npcap' as Administrator before upgrading Npcap, so that it doesn't crash again
Get WinPcap
Learn more about Npcap and WinPcap
Wireshark® Installer
< Back
Next >
Cancel

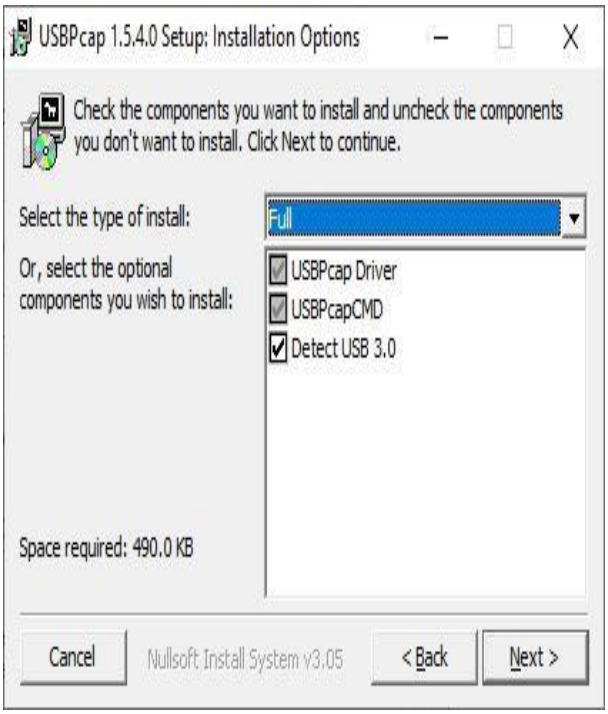

Wireshark 3.6.0 64-bit Setup
Packet Capture
Wireshark requires either Npcap or WinPcap to capture live network data.
Currently installed Npcap version
Npcap 1.50
Install
Install Npcap 1.55
The currently installed Npcap 1.50 will be uninstalled first.
Important notice
If your system has crashed during a Wireshark installation, you must run the command 'net stop npcap' as Administrator before upgrading Npcap, so that it doesn't crash again
Get WinPcap
Learn more about Npcap and WinPcap
Wireshark® Installer
< Back
Next >
Cancel

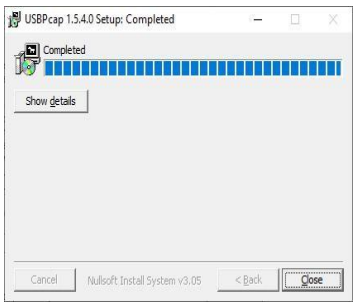

Npcap 1.55 Setup
NMAP.ORG
License Agreement
Please review the license terms before installing Npcap 1.55.
Press Page Down to see the rest of the agreement.
NPCAP COPYRIGHT / END USER LICENSE AGREEMENT
Npcap is a Windows packet sniffing driver and library and is copyright (c) 2013-2021 by Insecure.Com LLC ("The Nmap Project"). All rights reserved.
Even though Npcap source code is publicly available for review, it is not open source software and may not be redistributed without special permission from the Nmap Project. The standard version is also limited to installation on five systems. We fund the Npcap project by
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Npcap 1.55.
Nullsoft Install System v3.07
I Agree
Cancel

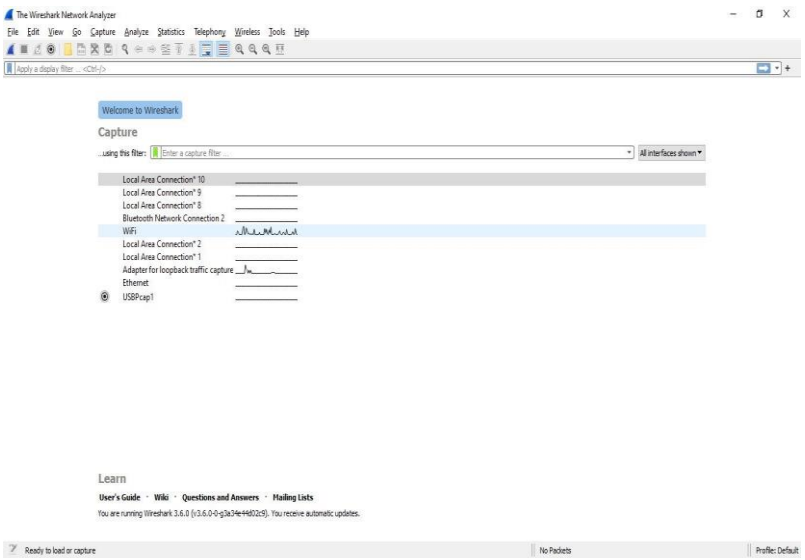

Npcap 1.55 Setup
NMAP.ORG
Installation Complete
Setup was completed successfully.
Completed
Show details
Nullsoft Install System v3.07
< Back
Next >
Cancel

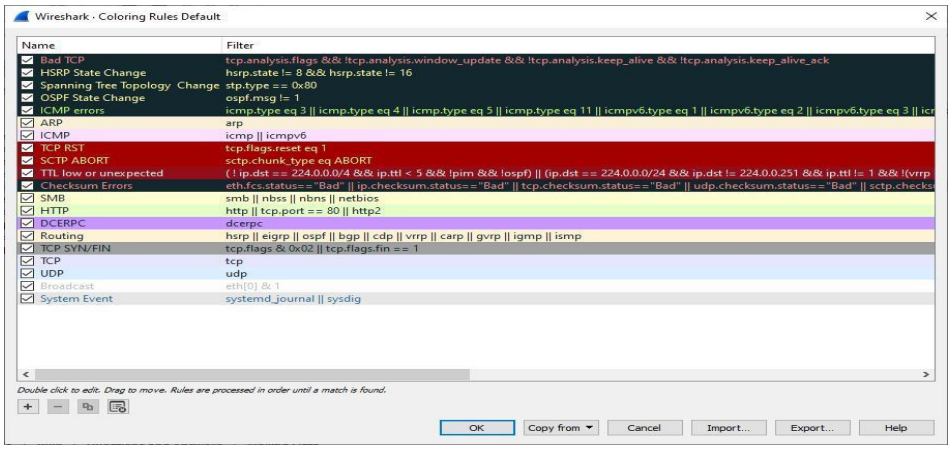

USBPcap 1.5.4.0 Setup: License Agreement
USBPcap Driver license
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
The licenses for most software are designed to take away your
I accept the terms of the License Agreement
Cancel
Nullsoft Install System v3.05
Next >

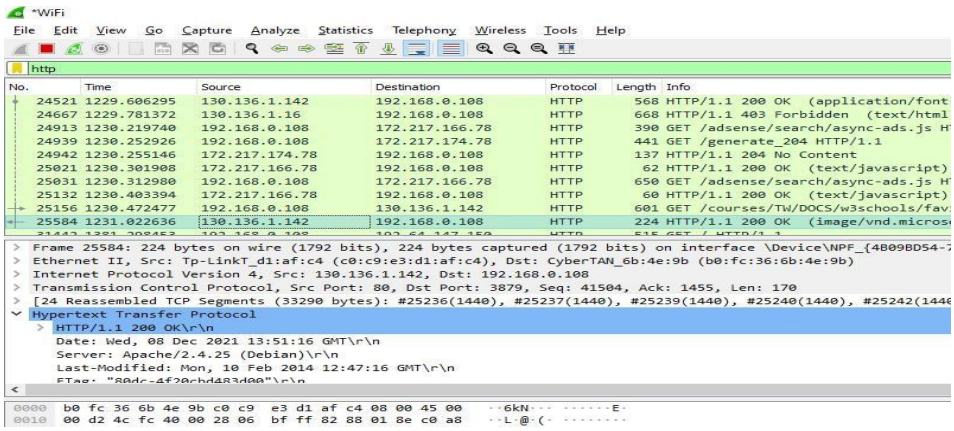

USBPcap 1.5.4.0 Setup: License Agreement
USBPcapCMD license
Copyright (c) 2013-2020, Tomasz Mon
All rights reserved.
Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:
Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other
materials provided with the distribution.
I accept the terms of the License Agreement
Cancel
Nullsoft Install System v3.05
< Back
Next >


USBPcap 1.5.4.0 Setup: Installation Options
Check the components you want to install and uncheck the components
you don't want to install. Click Next to continue.
Select the type of install:
Full
Or, select the optional
components you wish to install:
USBPcap Driver
USBPcapCMD
Detect USB 3.0
Space required: 490.0 KB
Cancel
Nullsoft Install System v3.05
< Back
Next >



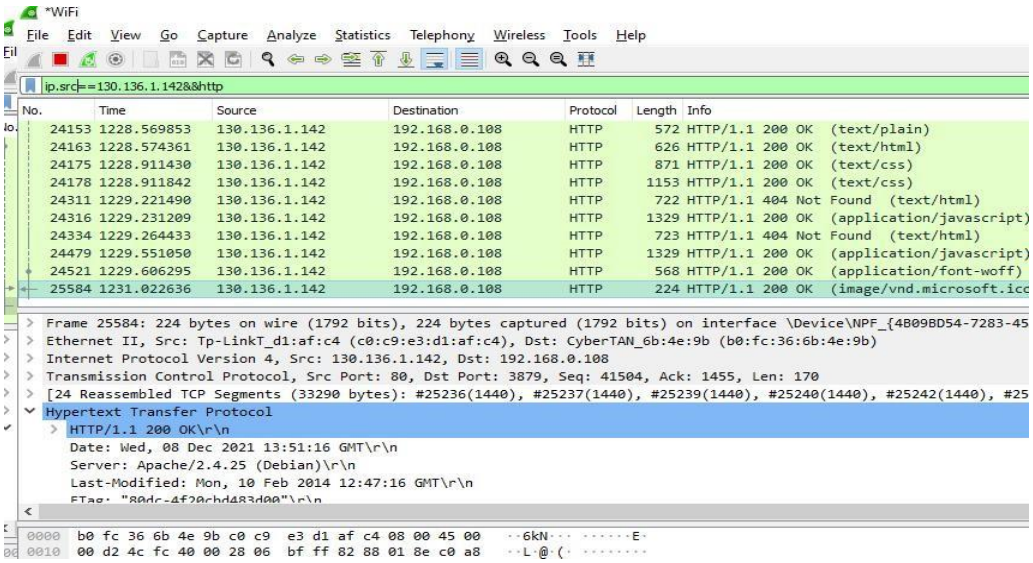






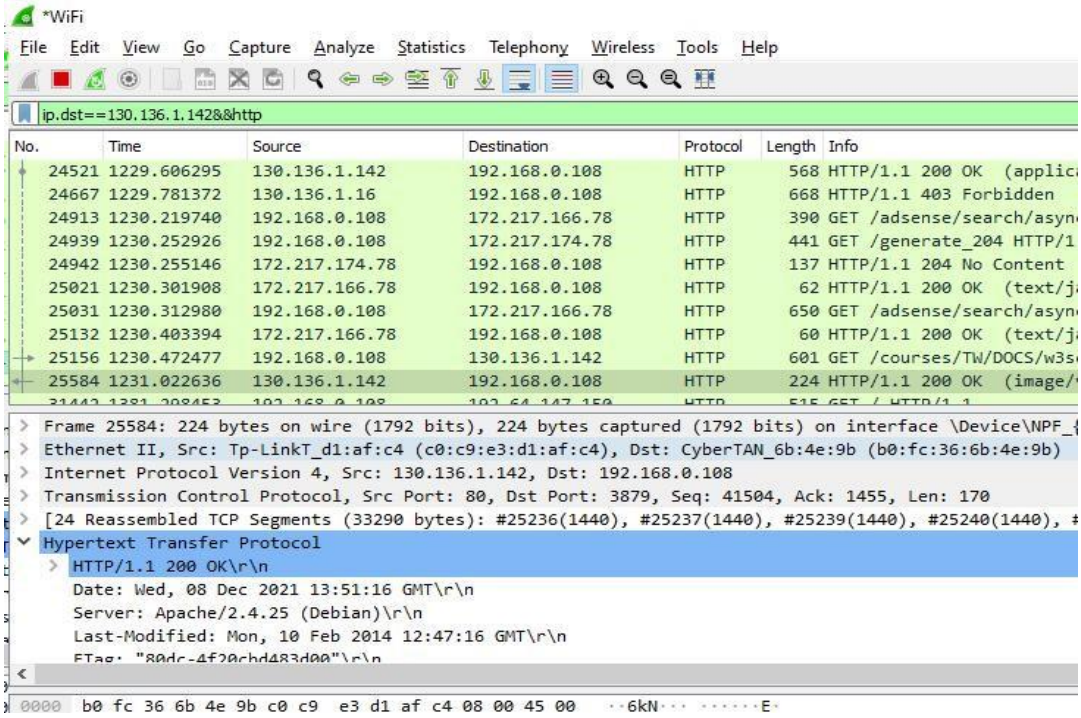
Inspect HTTP Traffic from a Given IP Address

In this particular tip, we will prepend ip==130.136.1.142&& to the filter stanza to monitor HTTP traffic between the local computer and 130.136.1.142



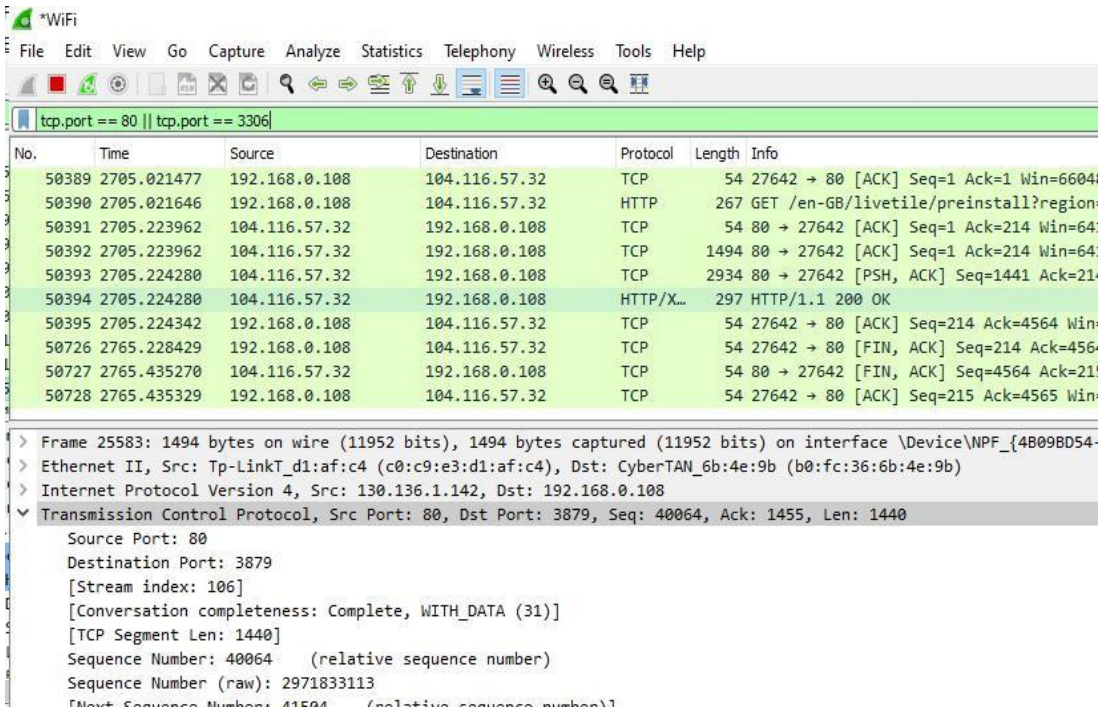
Inspect HTTP Traffic to a Given IP Address

In this case, we will use ip.dst as part of the capture filter as follows:



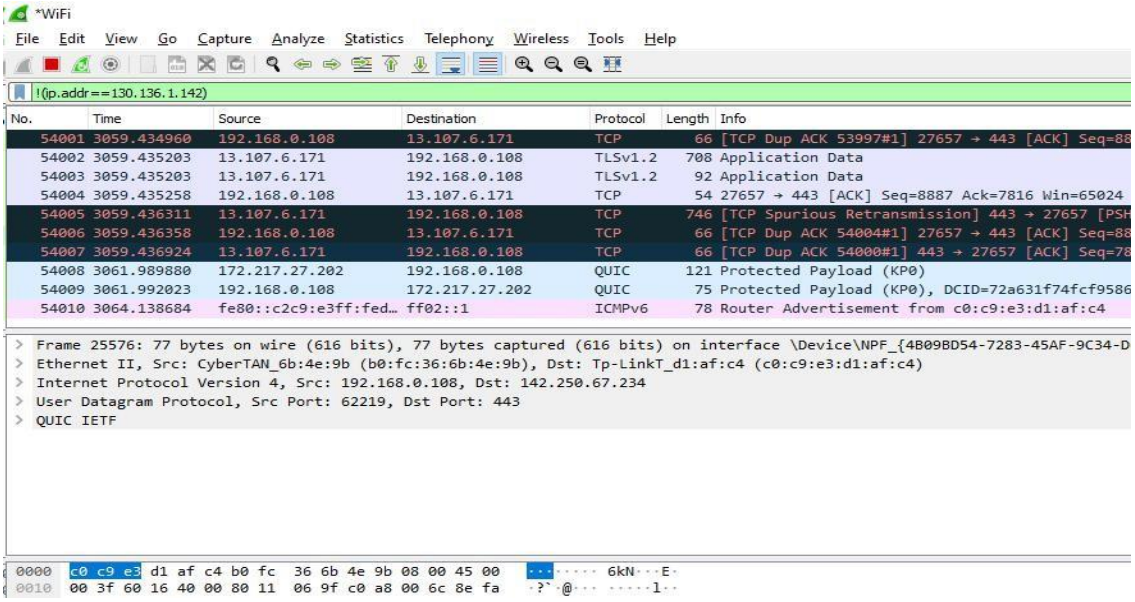
Monitor Apache and MySQL Network Traffic

Sometimes you will be interested in inspecting traffic that matches either (or both) conditions whatsoever. For example, to monitor traffic on TCP ports 80 (webserver) and 3306 (MySQL / MariaDB database server), you can use an OR condition in the capture filter:



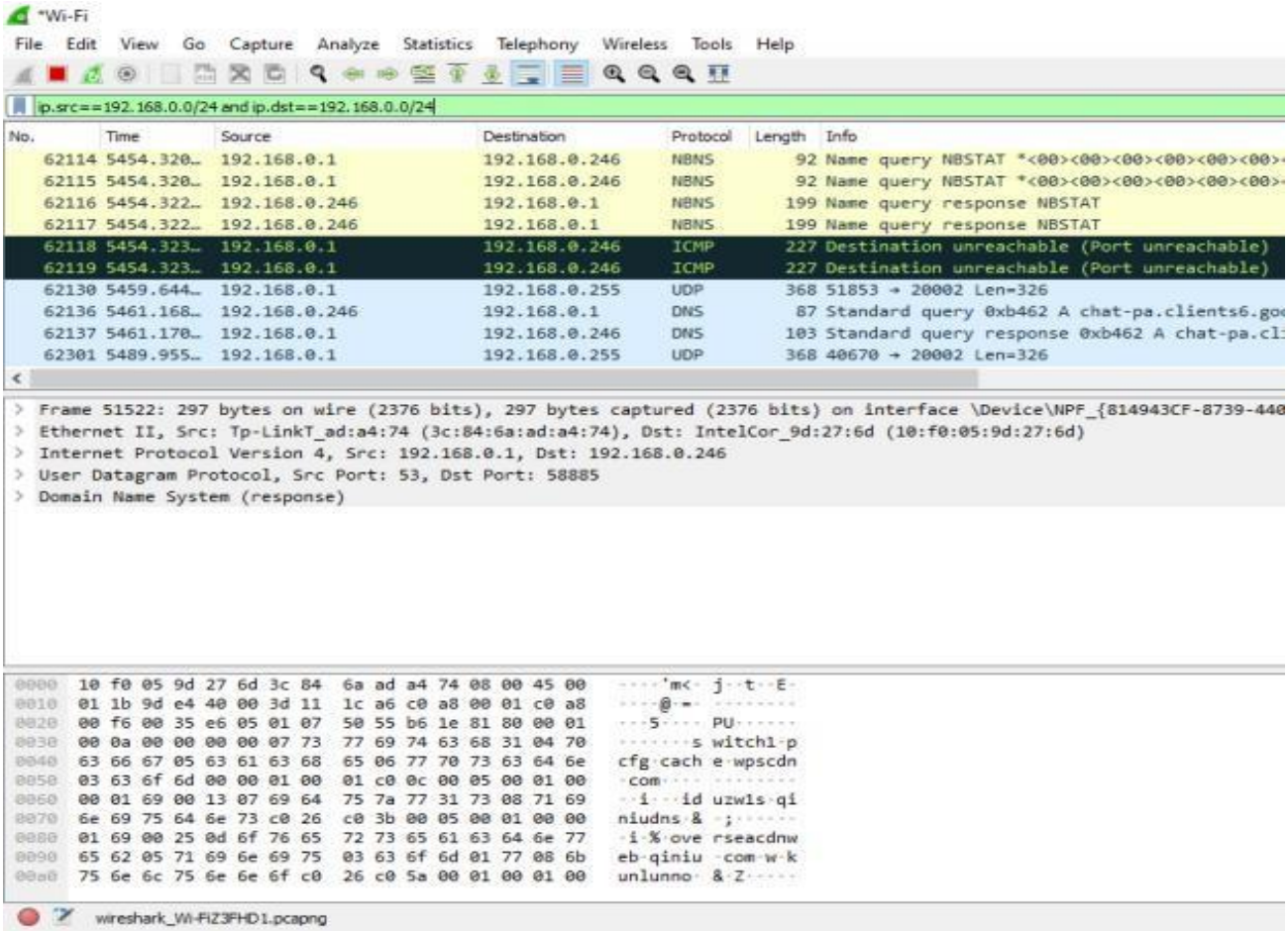
Reject Packets to Given IP Address

To exclude packets not matching the filter rule, use ! and enclose the rule within parentheses. For example, to exclude packages originating from or being directed to a given IP address, you can use:



Monitor Local Network Traffic (192.168.0.0/24)

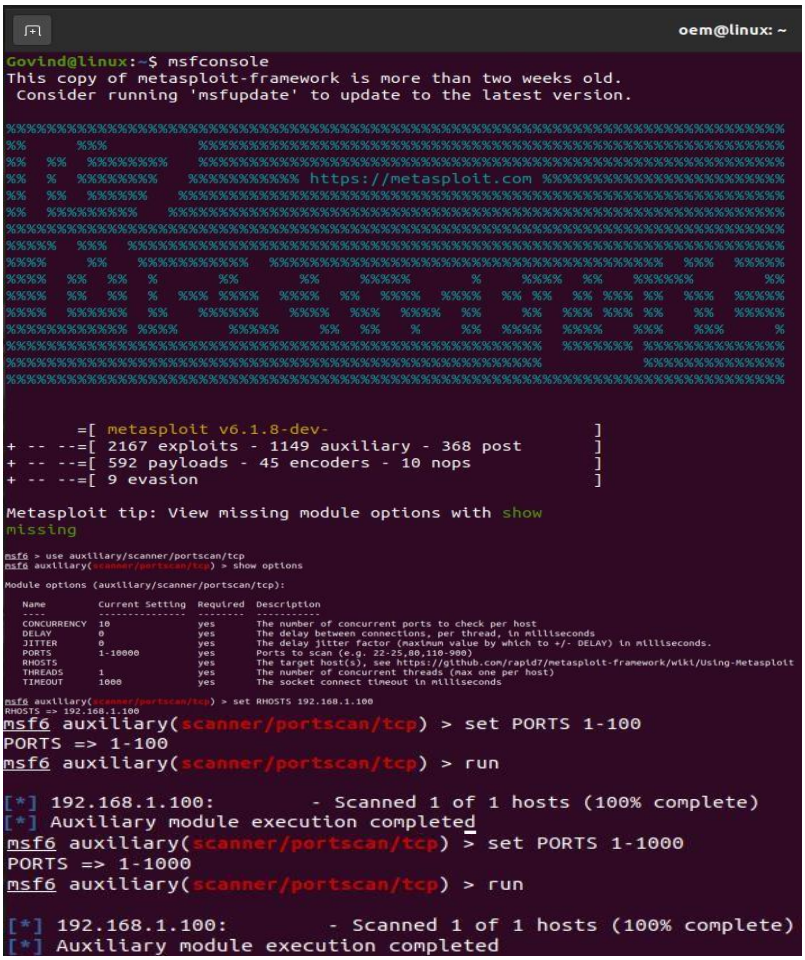
The following filter rule will display only local traffic and exclude packets going to and coming from the Internet:


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.src==192.168.0.0/24 and ip.dst==192.168.0.0/24
No. Time Source Destination Protocol Length Info
62114 5454.320… 192.168.0.1 192.168.0.246 NBNS 92 Name query NBSTAT *<00><00><00><00><00><00>
62115 5454.320… 192.168.0.1 192.168.0.246 NBNS 92 Name query NBSTAT *<00><00><00><00><00><00>
62116 5454.322… 192.168.0.246 192.168.0.1 NBNS 199 Name query response NBSTAT
62117 5454.322… 192.168.0.246 192.168.0.1 NBNS 199 Name query response NBSTAT
62118 5454.323… 192.168.0.1 192.168.0.246 ICMP 227 Destination unreachable (Port unreachable)
62119 5454.323… 192.168.0.1 192.168.0.246 ICMP 227 Destination unreachable (Port unreachable)
62130 5459.644… 192.168.0.1 192.168.0.255 UDP 368 51853 → 20002 Len=326
62136 5461.168… 192.168.0.246 192.168.0.1 DNS 87 Standard query 0xb462 A chat-pa.clients6.goo
62137 5461.170… 192.168.0.1 192.168.0.246 DNS 103 Standard query response 0xb462 A chat-pa.cl
62301 5489.955… 192.168.0.1 192.168.0.255 UDP 368 40670 → 20002 Len=326
Frame 51522: 297 bytes on wire (2376 bits), 297 bytes captured (2376 bits) on interface \Device\NPF_{814943CF-8739-440
Ethernet II, Src: Tp-LinkT_ad:a4:74 (3c:84:6a:ad:a4:74), Dst: IntelCor_9d:27:6d (10:f0:05:9d:27:6d)
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.246
User Datagram Protocol, Src Port: 53, Dst Port: 58885
Domain Name System (response)
0000 10 f0 05 9d 27 6d 3c 84 6a ad a4 74 08 00 45 00
0010 01 1b 9d e4 40 00 3d 11 1c a6 c0 a8 00 01 c0 a8
0020 00 f6 00 35 e6 05 01 07 50 55 b6 1e 81 80 00 01
0030 00 0a 00 00 00 00 07 73 77 69 74 63 68 31 04 70
0040 63 66 67 05 63 61 63 68 65 06 77 70 73 63 64 6e
0050 03 63 6f 6d 00 00 01 00 01 c0 0c 00 05 00 01 00
0060 00 01 69 00 13 07 69 64 75 7a 77 31 73 08 71 69
0070 6e 69 75 64 6e 73 c0 26 c0 3b 00 05 00 01 00 00
0080 01 69 00 25 0d 6f 76 65 72 73 65 61 63 64 6e 77
0090 65 62 05 71 69 6e 69 75 03 63 6f 6d 01 77 08 6b
00a0 75 6e 6c 75 6e 6e 6f c0 26 c0 5a 00 01 00 01 00
wireshark_Wi-FiZ3FHD1.pcapng

Practical No: 4

Metasploit

1. Open up the terminal window and start Metasploit using the command msfconsole .
2. Select the auxiliary module portscan/tcp to perform a port scan against a target system.
3. Using the show command, list down all parameters that need to be configured in order to run this auxiliary module.
4. Using the set RHOSTS command, set the IP address of our target system.
5. Using the set PORTS command, select the port range you want to scan on your target system.
6. Using the run command, execute the auxiliary module with the parameters configured earlier.
7. You can see the use of all the previously mentioned commands in the following screenshot:



Similarly, payloads in the Metasploit Framework let us decide what action is to be performed on the target system once the exploit is successful. The following screenshot shows a sample payload that can be used to obtain a reverse TCP shell from a compromised Windows system:

```
msf6 > use payload/windows/shell/reverse_tcp
msf6 payload(windows/shell/reverse_tcp) > show options

Module options (payload/windows/shell/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port

msf6 payload(windows/shell/reverse_tcp) > set LHOST 192.168.1.2
LHOST => 192.168.1.2
msf6 payload(windows/shell/reverse_tcp) > set LPORT 4455
LPORT => 4455
msf6 payload(windows/shell/reverse_tcp) >
```

The banner command: The banner command is a very simple command used to display the Metasploit Framework banner information. This information typically includes its version details and the number of exploits, auxiliaries, payloads, encoders, and nops generators available in the currently installed version. ts syntax is msf6> banner . The following screenshot shows the use of the banner command:

```
oem@linux: ~
msf6 > banner
                                                  ___          ____
                                             ,-""   `.      < HONK >
                                           ,'  _   e )`-._ /  ----
                                          /  ,' `-._<.===-'
                                         /  /
                                        /  ;
                                       /   ;
                   _                  |
 (`-.        _.-"" ""--..__,'         |
 <_  `-""                             \
  <`-                                  :
   (__   <__.                          ;
     `-.   '-.__.      _.'            /
        \       `-.__,-'         _,'
         `-.  ,     /__,-'
           ""._\__,'< <____
                | |  `-----.`.
                | |        \ `.
                ; |___      \-``
                \   --<
                 `.`.<
                   `-'


       =[ metasploit v6.1.8-dev-                          ]
+ -- --=[ 2167 exploits - 1149 auxiliary - 368 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View all productivity tips with the
tips command
```

The version command: The version command is used to check the version of the current Metasploit Framework installation. You can visit the following site in order to check the latest version officially released by Metasploit:

Its syntax is msf6> version . The following screenshot shows the use of the version command:

```
msf6 > version
Framework: 6.1.8-dev-
Console  : 6.1.8-dev-
```

he connect command: The connect command present in the Metasploit Framework gives similar functionality to that of a putty client or netcat. You can use this feature for a quick port scan or for port banner grabbing.

Its syntax is msf6> connect <ip:port> . The following screenshot shows the use of the connect command:

```
oem@linux: ~
msf6 > connect <ip:port>
Usage: connect [options] <host> <port>

Communicate with a host, similar to interacting via netcat, taking advantage of
any configured session pivoting.

OPTIONS:

    -C        Try to use CRLF for EOL sequence.
    -P <opt>  Specify source port.
    -S <opt>  Specify source address.
    -c <opt>  Specify which Comm to use.
    -h        Help banner.
    -i <opt>  Send the contents of a file.
    -p <opt>  List of proxies to use.
    -s        Connect with SSL.
    -u        Switch to a UDP socket.
    -w <opt>  Specify connect timeout.
    -z        Just try to connect, then return.
msf6 >
msf6 >
msf6 > connect google.com 80
[*] Connected to google.com:80 (via: 192.168.0.108:45471)
```

The help command: As the name suggests, the help command offers additional information on the usage of any of the commands within the Metasploit Framework.

Its syntax is msf6> help . The following screenshot shows the use of the help command:

```
oem@linux: ~
msf6 > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    color         Toggle color
    connect       Communicate with a host
    debug         Display information useful for debugging
    exit          Exit the console
    features      Display the list of not yet released features that can be opte
    get           Gets the value of a context-specific variable
    getg          Gets the value of a global variable
    grep          Grep the output of another command
    help          Help menu
    history       Show command history
    load          Load a framework plugin
    quit          Exit the console
    repeat        Repeat a list of commands
    route         Route traffic through a session
    save          Saves the active datastores
    sessions      Dump session listings and display information about sessions
    set           Sets a context-specific variable to a value
    setg          Sets a global variable to a value
    sleep         Do nothing for the specified number of seconds
    spool         Write console output into a file as well the screen
    threads       View and manipulate background threads
    tips          Show a list of useful productivity tips
    unload        Unload a framework plugin
    unset         Unsets one or more context-specific variables
    unsetg        Unsets one or more global variables
    version       Show the framework and console library version numbers
```

The save command: At times, when performing a penetration test on a complex target environment, a lot of configuration changes are made in the Metasploit Framework. Now, if the penetration test needs to be resumed again at a later point of time, it would be really painful to configure the Metasploit Framework again from scratch. The save command saves all the configurations to a file and it gets loaded upon the next startup, saving all the reconfiguration efforts.

Its syntax is msf6>save . The following screenshot shows the use of the save command:

```
msf6 > save
Saved configuration to: /home/oem/.msf4/config
msf6 >
```

The sessions command: Once our target is exploited successfully, we normally get a shell session on the target system. If we are working on multiple targets simultaneously, then there might be multiple sessions actively open at the same time. The Metasploit Framework allows us to switch between multiple sessions as and when required. The sessions command lists down all the currently active sessions established with various target systems.

Its syntax is msf6>sessions . The following screenshot shows the use of the sessions command:

```
msf6 > sessions

Active sessions
===============

No active sessions.
```

The spool command: Just like any application has debug logs that help out in debugging errors, the spool command prints out all the output to a user-defined file along with the console. The output file can later be analyzed based on the requirement.

Its syntax is msf6>spool . The following screenshot shows the use of the spool command:

```
msf6 > spool
Usage: spool <off>|<filename>

Example:
  spool /tmp/console.log
```

The show command: The show command is used to display the available modules within the Metasploit Framework or to display additional information while using a particular module.

Its syntax is msf6> show . The following screenshot shows the use of the show command:

```
oem@linux: ~
msf6 > show -h
[*] Valid parameters for the "show" command are: all, encoders, nops, exploits,
[*] Additional module-specific parameters are: missing, advanced, evasion, targ
msf6 > show nops

NOP Generators
==============

   #  Name                  Disclosure Date  Rank    Check  Description
   -  ----                  ---------------  ----    -----  -----------
   0  nop/aarch64/simple                      normal  No     Simple
   1  nop/armle/simple                        normal  No     Simple
   2  nop/mipsbe/better                       normal  No     Better
   3  nop/php/generic                         normal  No     PHP Nop Generator
   4  nop/ppc/simple                          normal  No     Simple
   5  nop/sparc/random                        normal  No     SPARC NOP Generator
   6  nop/tty/generic                         normal  No     TTY Nop Generator
   7  nop/x64/simple                          normal  No     Simple
   8  nop/x86/opty2                           normal  No     Opty2
   9  nop/x86/single_byte                     normal  No     Single Byte
```

The info command: The info command is used to display details about a particular module within the Metasploit Framework. For example, you might want to view information on meterpreter payload, such as what the supported architecture ia and what the options required in order to execute this are:

Its syntax is msf6> info . The following screenshot shows the use of the info command:

```
oem@linux: ~
msf6 > info -h
Usage: info <module name> [mod2 mod3 ...]

Options:
* The flag '-j' will print the data in json format
* The flag '-d' will show the markdown version with a browser. More info, but
show info for the currently active module.

       Name: Windows Meterpreter (Reflective Injection), Reverse TCP Stager
     Module: payload/windows/meterpreter/reverse_tcp
   Platform: Windows
       Arch: x86
Needs Admin: No
 Total size: 296
       Rank: Normal

Provided by:
  skape <mmiller@hick.org>
  sf <stephen_fewer@harmonysecurity.com>
  OJ Reeves
  hdm <x@hdm.io>

Basic options:
Name      Current Setting  Required  Description
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread
LHOST                      yes       The listen address (an interface may be s
LPORT     4444             yes       The listen port

Description:
  Inject the Meterpreter server DLL via the Reflective Dll Injection
  payload (staged). Requires Windows XP SP2 or newer. Connect back to
  the attacker
```

The irb command: The irb command invokes the interactive Ruby platform from within the Metasploit Framework. The interactive Ruby platform can be used for creating and invoking custom scripts typically during the post- exploitation phase.

Its syntax is msf6>irb . The following screenshot shows the use of the irb command:

```
msf6 > irb
[*] Starting IRB shell...
[*] You are in the "framework" object

irb: warn: can't alias jobs from irb_jobs.
>> puts "MetaSploit is Working"
=> nil
>>
```

The makerc command: When we use the Metasploit Framework for pen testing a target, we fire a lot many commands. At end of the assignment or that particular session, we might want to review what all activities we performed through Metasploit. The makerc command simply writes out all the command history for a particular session to a user defined output file.

Its syntax is msf6>makerc . The following screenshot shows the use of the makerc command:

```
oem@linux: ~
>> exit
msf6 > makerc
Usage: makerc <output rc file>

Save the commands executed since startup to the specified file.

msf6 > makerc /home/govind/Desktop/msfcommands.txt
[*] Saving last 17 commands to /home/govind/Desktop/msfcommands.txt ...
msf6 >
msf6 >
msf6 > get
Usage: get var1 [var2 ...]

The get command is used to get the value of one or more variables.
```

The get command: The get command is used to retrieve the value contained in a particular local variable within the Metasploit Framework. For example, you might want to view what is the IP address of the target system that you have set for a particular exploit.

Its syntax is msf6>get . The following screenshot shows the use of the msf6> get command:

```
msf6 > get
Usage: get var1 [var2 ...]

The get command is used to get the value of one or more variables.

msf6 > get RHOST
RHOST =>
msf6 >
```

The getg command: The getg command is very similar to the get command, except it returns the value contained in the global variable.

Its syntax is msf6> getg . The following screenshot shows the use of the msf6> getg command:

```
msf6 > getg
Usage: getg var1 [var2 ...]

Exactly like get -g, get global variables

msf6 > getg RHOSTS
RHOSTS =>
msf6 >
```

The set and setg commands: The set command assigns a new value to one of the (local) variables (such as RHOST , RPORT , LHOST , and LPPORT) within the Metasploit Framework. However, the set command assigns a value to the variable that is valid for a limited session/instance. The setg command assigns a new value to the (global) variable on a permanent basis so that it can be used repeatedly whenever required.

Its syntax is:

msf6> set <VARIABLE> <VALUE> msf6> setg

<VARIABLE> <VALUE>

We can see the set and setg commands in the following screenshot:

```
msf6 > set RHOST 192.168.1.30
RHOST => 192.168.1.30
msf6 > setg RHOST 192.168.1.30
RHOST => 192.168.1.30
msf6 >
```

The unset and unsetg commands: The unset command simply clears the value previously stored in a (local) variable through the set command. The unsetg command clears the value previously stored in a (global) variable through the setg command: syntax is:

msf6> unset<VARIABLE> msf6>

unsetg <VARIABLE>

We can see the unset and unsetg commands in the following screenshot:

```
msf6 > unset RHOST
Unsetting RHOST...
msf6 > unsetg RHOST
Unsetting RHOST...
msf6 >
```

The Metasploit Framework offers a simple utility called msfupdate that connects to the respective online repository and fetches the updates:

```
oem@linux: ~
Govind@linux:~$ msfupdate
Switching to root user to update the package
[sudo] password for oem:
Adding metasploit-framework to your repository list..OK
Updating package cache..OK
Checking for and installing update..
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.8.0-43-generic linux-hwe-5.8-headers-5.8.0-43 linux-image-5.8.0-43-generic linux-modules-5.8.0-4
  linux-modules-extra-5.8.0-43-generic
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  metasploit-framework
1 upgraded, 0 newly installed, 0 to remove and 240 not upgraded.
Need to get 265 MB of archives.
After this operation, 25.8 MB of additional disk space will be used.
Get:1 http://downloads.metasploit.com/data/releases/metasploit-framework/apt lucid/main amd64 metasploit-framework
~1rapid7-1 [265 MB]
Fetched 265 MB in 1min 28s (3,002 kB/s)
(Reading database ... 266890 files and directories currently installed.)
Preparing to unpack .../metasploit-framework_6.1.19+20211207112548~1rapid7-1_amd64.deb ...
Unpacking metasploit-framework (6.1.19+20211207112548~1rapid7-1) over (6.1.8+20210927102556~1rapid7-1) ...
Setting up metasploit-framework (6.1.19+20211207112548~1rapid7-1) ...
Run msfconsole to get started
```