Karthikeyan.G

Assignment-2

MTech - cybersecurity

1. Understand PING and document it, then answer the following question:

1a. Use ping on google.com and document your results on the output you received. [Find the IP address, Time to live value, and round-trip time value from the results you got].

```
C:\Users\guruk>ping brave.com

Pinging brave.com [2600:9000:264e:4800:6:d0d2:780:93a1] with 32 bytes of data:
Reply from 2600:9000:264e:4800:6:d0d2:780:93a1: time=37ms
Reply from 2600:9000:264e:4800:6:d0d2:780:93a1: time=48ms
Reply from 2600:9000:264e:4800:6:d0d2:780:93a1: time=41ms
Reply from 2600:9000:264e:4800:6:d0d2:780:93a1: time=67ms

Ping statistics for 2600:9000:264e:4800:6:d0d2:780:93a1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 37ms, Maximum = 67ms, Average = 48ms
```

1b. By default, ping will send 4 packets to check the details, here you must send 8 packets to check the output over google.com. Explain the purpose of doing it.

```
C:\Users\guruk>ping brave.com -n 8

Pinging brave.com [2600:9000:264e:fa00:6:d0d2:780:93a1] with 32 bytes of data:
Reply from 2600:9000:264e:fa00:6:d0d2:780:93a1: time=34ms
Reply from 2600:9000:264e:fa00:6:d0d2:780:93a1: time=55ms
Reply from 2600:9000:264e:fa00:6:d0d2:780:93a1: time=65ms
Reply from 2600:9000:264e:fa00:6:d0d2:780:93a1: time=88ms
Reply from 2600:9000:264e:fa00:6:d0d2:780:93a1: time=69ms
Reply from 2600:9000:264e:fa00:6:d0d2:780:93a1: time=82ms
Reply from 2600:9000:264e:fa00:6:d0d2:780:93a1: time=62ms
Reply from 2600:9000:264e:fa00:6:d0d2:780:93a1: time=64ms

Ping statistics for 2600:9000:264e:fa00:6:d0d2:780:93a1:
    Packets: Sent = 8, Received = 8, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 34ms, Maximum = 88ms, Average = 64ms
```

The PING -n command decides the how many numbers of echo requests to send. While commanding ping google.com -n 8, it gives the output of 8 echo packets with IP address and round trip

1c. Ping your local host. Explain the purpose of doing it.

```
C:\Users\guruk>ping localhost

Pinging MSI [::1] with 32 bytes of data:
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms

Ping statistics for ::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Ping local host is ensuring that our local network is operational, troubleshoot network problems, validate host name resolution. It provides a quick check of the networking functionality on your machine without involving external network configuration.

1. Read the Unix manual page for traceroute OR help for tracert. Experiment with the assorted options. Describe the three things that you found most useful in the result.

Tracert is a command line which helps to diagnosing network issues, understanding routing paths, and troubleshooting connectivity problems. It provides detailed information about the route packets take to reach a destination, helping network administrator and users pinpoint where problems might be occurring along the network path.

<u>The three useful things:</u>

1. Determining the route a packet takes
2. It can be useful for understanding the network infrastructure between points.
3. If you cannot reach a destination, this command helps us where the connection is failing.

a. Try tracert over Brave.com

```
C:\Users\guruk>tracert brave.com

Tracing route to brave.com [2600:9000:264e:2c00:6:d0d2:780:93a1]
over a maximum of 30 hops:

  1     3 ms     3 ms     3 ms  2409:40f4:201f:dd76::c2
  2    37 ms    28 ms    31 ms  2405:200:5218:24:3924:110:3:112
  3    17 ms    31 ms    25 ms  2405:200:5218:24:3925::1
  4    45 ms    26 ms    30 ms  2405:200:801:4f00::1ec
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10     *        *        *     Request timed out.
 11    62 ms    78 ms    32 ms  2600:9000:264e:2c00:6:d0d2:780:93a1

Trace complete.
```

b. Type tracert -d Brave.com

```
C:\Users\guruk>tracert -d brave.com

Tracing route to brave.com [2600:9000:2579:200:6:d0d2:780:93a1]
over a maximum of 30 hops:

  1     4 ms     3 ms     3 ms  2409:40f4:201f:dd76::c2
  2    39 ms    36 ms    26 ms  2405:200:5218:24:3924:110:3:112
  3    54 ms    36 ms    33 ms  2405:200:5218:24:3925::1
  4    16 ms    13 ms    18 ms  2405:200:801:4f00::1ec
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12    59 ms    37 ms    38 ms  2600:9000:2579:200:6:d0d2:780:93a1

Trace complete.
```

2.b.1) How many hops is your machine away from google.com? (Attach the output in the lab report)

My machine is 5 hops away from the google.com. refer question 2.a for screenshot.

2.b.2) Wait for a while and execute the same command again. Is the output the same as the first time? Observe and compare the difference and explain the reason

```
C:\Users\guruk>tracert -d brave.com

Tracing route to brave.com [2600:9000:2579:200:6:d0d2:780:93a1]
over a maximum of 30 hops:

  1     4 ms     3 ms     3 ms  2409:40f4:201f:dd76::c2
  2    39 ms    36 ms    26 ms  2405:200:5218:24:3924:110:3:112
  3    54 ms    36 ms    33 ms  2405:200:5218:24:3925::1
  4    16 ms    13 ms    18 ms  2405:200:801:4f00::1ec
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12    59 ms    37 ms    38 ms  2600:9000:2579:200:6:d0d2:780:93a1

Trace complete.
```

No changes.

2. You must read about NETSTAT from the manual page or help, before answering the below questions:

Netstat is a command line that provides information about network connections, routing tables, network interface statistics. It is used for diagnosing network issues, monitoring network activity and issues, and understanding network configuration on both Unix- based and windows system.

3a. Use netstat to display information about the routing table

```
C:\Users\guruk>netstat -r
===========================================================================
Interface List
  4...0a 00 27 00 00 04 ......VirtualBox Host-Only Ethernet Adapter
  3...f0 ac 65 ce 64 5a ......Microsoft Wi-Fi Direct Virtual Adapter
 15...fa ac 65 ce 64 59 ......Microsoft Wi-Fi Direct Virtual Adapter #2
 12...56 b7 b7 41 4c 09 ......Intel(R) Wireless-AC 9560 160MHz
 20...f8 ac 65 ce 64 5d ......Bluetooth Device (Personal Area Network)
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0   192.168.253.41   192.168.253.90     35
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
      192.168.56.0    255.255.255.0         On-link      192.168.56.1    281
      192.168.56.1  255.255.255.255         On-link      192.168.56.1    281
    192.168.56.255  255.255.255.255         On-link      192.168.56.1    281
     192.168.253.0    255.255.255.0         On-link    192.168.253.90    291
    192.168.253.90  255.255.255.255         On-link    192.168.253.90    291
   192.168.253.255  255.255.255.255         On-link    192.168.253.90    291
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link      192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link    192.168.253.90    291
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link      192.168.56.1    281
  255.255.255.255  255.255.255.255         On-link    192.168.253.90    291
===========================================================================
Persistent Routes:
  None
```

```
IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
 12     51 ::/0                     fe80::c46:7bff:fe0c:5352
  1    331 ::1/128                  On-link
 12     51 2409:40f4:201f:dd76::/64 On-link
 12    291 2409:40f4:201f:dd76:712b:ce8c:c274:29cc/128
                                    On-link
 12    291 2409:40f4:201f:dd76:84ff:a836:d6b6:f066/128
                                    On-link
  4    281 fe80::/64                On-link
 12    291 fe80::/64                On-link
 12    291 fe80::5bc6:b991:5362:b26e/128
                                    On-link
  4    281 fe80::f903:971d:e3ac:524b/128
                                    On-link
  1    331 ff00::/8                 On-link
  4    281 ff00::/8                 On-link
 12    291 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None
```

3b. Use netstat to display about ethernet statistics.

```
C:\Users\guruk>netstat -e
Interface Statistics

                           Received            Sent

Bytes                     622570072       349816328
Unicast packets             21152480         896448
Non-unicast packets         50505688          96168
Discards                           0              0
Errors                             0              0
Unknown protocols                  0
```

4. What is the purpose of NSLOOKUP? Answer the following questions below:

nslookup is the command line is used to query DNS server and retrieve information about the DNS domain name and IP address. It helps in solving the DNS related issues and monitoring DNS records and resolving DNS server into IP addresses.

4a. Use nslookup to find out the internet address of the domain amrita.edu.

```
C:\Users\guruk>nslookup amrita.edu
Server:  UnKnown
Address:  192.168.253.41

Non-authoritative answer:
Name:    amrita.edu
Addresses:  64:ff9b::4cdf:533e
          64:ff9b::4b02:70a4
          75.2.112.164
          76.223.83.62
```

4b. What is the mail exchanger for the domain google.com.

```
C:\Users\guruk>nslookup -type=mx google.com
Server:  UnKnown
Address:  192.168.253.41

Non-authoritative answer:
google.com      MX preference = 10, mail exchanger = smtp.google.com
```

4c. What is the name server for amrita.edu.

```
C:\Users\guruk>nslookup -type=ns amrita.edu
Server:  UnKnown
Address:  192.168.118.5

Non-authoritative answer:
amrita.edu      nameserver = ns2.amrita.edu
amrita.edu      nameserver = ns4.amrita.edu
amrita.edu      nameserver = ns1.amrita.edu
amrita.edu      nameserver = ns3.amrita.edu

ns1.amrita.edu  internet address = 103.10.27.3
ns1.amrita.edu  internet address = 14.139.187.131
ns2.amrita.edu  internet address = 117.193.77.232
ns3.amrita.edu  internet address = 103.10.24.200
ns4.amrita.edu  internet address = 103.5.112.81
```

5.What is ARP and RARP? Answer the following questions below:

ARP and RARP are network protocols used to map addresses between the different layers of the Network layer protocols.

ARP - ARP is a protocol used to map IP address to the physical MAC address within a local network segment. When a device wants to communicate with another device on the same network, it needs to know the destination device's MAC address.

RARP – RARP is a reverse process of ARP protocol, where it is used to map physical MAC address to the IP address. This is used in situation where the device knew its MAC address and need to discover IP addresses.

5a. Use Arp command to find the gateway address and host systems hardware address.

```
C:\Users\guruk>arp -a

Interface: 192.168.56.1 --- 0x4
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static

Interface: 192.168.253.90 --- 0xc
  Internet Address      Physical Address      Type
  192.168.253.41        0e-46-7b-0c-53-52     dynamic
  192.168.253.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

5b. How do you find the Arp entries for a particular interface?

```
C:\Users\guruk>arp -a 192.168.253.255

Interface: 192.168.253.90 --- 0xc
  Internet Address      Physical Address      Type
  192.168.253.255       ff-ff-ff-ff-ff-ff     static
```

5c. How do delete an Arp entry?

```
C:\Windows\System32>arp -d 192.168.56.255

C:\Windows\System32>
```

After command

```
C:\Windows\System32>arp -a 192.168.56.255
No ARP Entries Found.
```

5d. How do you add and Arp entry in Arp cache?

```
Interface: 192.168.56.1 --- 0x4
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static

Interface: 192.168.118.54 --- 0xc
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
```

6. Read about TCPDUMP tool [use manual page]. Answer the questions below: Tcpudmp is a packet

<u>Tcpdump</u> is a packet traffic analyser that captures and displays it in readable format. It is often used for debugging network issues, analysing network performance, and understanding network protocols. It works by capturing packets from the network interface and interpreting them according to various protocol standards.

6a. Using tcpdump, get the information about the general incoming network traffic with domain names.

```
┌──(root㉿kali)-[/home/karthikeyan]
└─# tcpdump -i eth0 -n -v port 53
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
20:55:12.715324 IP (tos 0x0, ttl 64, id 31815, offset 0, flags [DF], proto UDP (17), length 61)
    10.0.2.15.59837 > 192.168.253.41.53: 64281+ A? www.youtube.com. (33)
20:55:12.715414 IP (tos 0x0, ttl 64, id 31816, offset 0, flags [DF], proto UDP (17), length 61)
    10.0.2.15.59837 > 192.168.253.41.53: 34820+ AAAA? www.youtube.com. (33)
20:55:12.822915 IP (tos 0x0, ttl 64, id 2184, offset 0, flags [none], proto UDP (17), length 351)
    192.168.253.41.53 > 10.0.2.15.59837: 64281 17/0/0 www.youtube.com. CNAME youtube-ui.l.google.com., youtube-ui.l.google.com. A 142.250.182.142, youtube-ui.l.google.com. A 142.250.183.238, youtube-ui.l.google.com. A 142.250.193.110, you
tube-ui.l.google.com. A 142.250.193.142, youtube-ui.l.google.com. A 142.250.193.174, youtube-ui.l.google.com. A 142.250.205.238, youtube-ui.l.google.com. A 172.217.167.142, youtube-ui.l.google.com. A 172.217.160.142, youtube-ui.l.google.c
om. A 142.250.71.14, youtube-ui.l.google.com. A 142.250.76.46, youtube-ui.l.google.com. A 142.250.77.142, youtube-ui.l.google.com. A 142.250.77.174, youtube-ui.l.google.com. A 142.250.182.14, youtube-ui.l.google.com. A 142.250.182.46, you
tube-ui.l.google.com. A 142.250.182.78, youtube-ui.l.google.com. A 142.250.182.110 (323)
20:55:12.825216 IP (tos 0x0, ttl 64, id 2185, offset 0, flags [none], proto UDP (17), length 207)
    192.168.253.41.53 > 10.0.2.15.59837: 34820 5/0/0 www.youtube.com. CNAME youtube-ui.l.google.com., youtube-ui.l.google.com. AAAA 2404:6800:4007:81a::200e, youtube-ui.l.google.com. AAAA 2404:6800:4007:817::200e, youtube-ui.l.google.com.
 AAAA 2404:6800:4007:818::200e, youtube-ui.l.google.com. AAAA 2404:6800:4007:819::200e (179)
20:55:14.213101 IP (tos 0x0, ttl 64, id 65372, offset 0, flags [DF], proto UDP (17), length 57)
    10.0.2.15.57375 > 192.168.253.41.53: 55219+ A? i.ytimg.com. (29)
20:55:14.213179 IP (tos 0x0, ttl 64, id 65373, offset 0, flags [DF], proto UDP (17), length 57)
    10.0.2.15.57375 > 192.168.253.41.53: 54206+ AAAA? i.ytimg.com. (29)
20:55:14.239279 IP (tos 0x0, ttl 64, id 48597, offset 0, flags [DF], proto UDP (17), length 66)
    10.0.2.15.36759 > 192.168.253.41.53: 20154+ A? fonts.googleapis.com. (38)
20:55:14.240673 IP (tos 0x0, ttl 64, id 48598, offset 0, flags [DF], proto UDP (17), length 66)
    10.0.2.15.36759 > 192.168.253.41.53: 5543+ AAAA? fonts.googleapis.com. (38)
20:55:14.249515 IP (tos 0x0, ttl 64, id 2290, offset 0, flags [none], proto UDP (17), length 313)
    192.168.253.41.53 > 10.0.2.15.57375: 55219 10/0/0 i.ytimg.com. A 172.217.31.214, i.ytimg.com. A 142.250.71.22, i.ytimg.com. A 142.250.71.54, i.ytimg.com. A 142.250.195.54, i.ytimg.com. A 142.250.77.182, i.ytimg.com. A 142.250.182.22,
i.ytimg.com. A 142.250.182.54, i.ytimg.com. A 142.250.182.86, i.ytimg.com. A 142.250.182.118, i.ytimg.com. A 142.250.182.150, i.ytimg.com. A 142.250.183.246, i.ytimg.com. A 142.250.193.118, i.ytimg.com. A 142.250.193.150, i.ytimg.com. A 1
42.250.193.182, i.ytimg.com. A 142.250.205.246, i.ytimg.com. A 172.217.167.150 (285)
20:55:14.309339 IP (tos 0x0, ttl 64, id 2293, offset 0, flags [none], proto UDP (17), length 169)
    192.168.253.41.53 > 10.0.2.15.57375: 54206 4/0/0 i.ytimg.com. AAAA 2404:6800:4007:81a::2016, i.ytimg.com. AAAA 2404:6800:4007:81b::2016, i.ytimg.com. AAAA 2404:6800:4007:818::2016, i.ytimg.com. AAAA 2404:6800:4007:819::2016 (141)
20:55:14.892999 IP (tos 0x0, ttl 64, id 2751, offset 0, flags [none], proto UDP (17), length 82)
    192.168.253.41.53 > 10.0.2.15.36759: 20154 1/0/0 fonts.googleapis.com. A 142.250.77.170 (54)
20:55:15.129382 IP (tos 0x0, ttl 64, id 3071, offset 0, flags [none], proto UDP (17), length 94)
    192.168.253.41.53 > 10.0.2.15.36759: 5543 1/0/0 fonts.googleapis.com. AAAA 2404:6800:4007:818::200a (66)
20:55:15.750605 IP (tos 0x0, ttl 64, id 46023, offset 0, flags [DF], proto UDP (17), length 56)
    10.0.2.15.34130 > 192.168.253.41.53: 46939+ A? o.pki.goog. (28)
20:55:15.910012 IP (tos 0x0, ttl 64, id 3740, offset 0, flags [none], proto UDP (17), length 107)
    192.168.253.41.53 > 10.0.2.15.34130: 46939 2/0/0 o.pki.goog. CNAME pki-goog.l.google.com., pki-goog.l.google.com. A 142.250.183.131 (79)
20:55:15.916013 IP (tos 0x0, ttl 64, id 21106, offset 0, flags [DF], proto UDP (17), length 56)
    10.0.2.15.42366 > 192.168.253.41.53: 1128+ A? o.pki.goog. (28)
20:55:15.920762 IP (tos 0x0, ttl 64, id 3741, offset 0, flags [none], proto UDP (17), length 107)
```


```
20:55:15.129382 IP (tos 0x0, ttl 64, id 3071, offset 0, flags [none], proto UDP (17), length 94)
    192.168.253.41.53 > 10.0.2.15.36759: 5543 1/0/0 fonts.googleapis.com. AAAA 2404:6800:4007:818::200a (66)
20:55:15.750605 IP (tos 0x0, ttl 64, id 46023, offset 0, flags [DF], proto UDP (17), length 56)
    10.0.2.15.34130 > 192.168.253.41.53: 46939+ A? o.pki.goog. (28)
20:55:15.910012 IP (tos 0x0, ttl 64, id 3740, offset 0, flags [none], proto UDP (17), length 107)
    192.168.253.41.53 > 10.0.2.15.34130: 46939 2/0/0 o.pki.goog. CNAME pki-goog.l.google.com., pki-goog.l.google.com. A 142.250.183.131 (79)
20:55:15.916013 IP (tos 0x0, ttl 64, id 21106, offset 0, flags [DF], proto UDP (17), length 56)
    10.0.2.15.42366 > 192.168.253.41.53: 1128+ A? o.pki.goog. (28)
20:55:15.920762 IP (tos 0x0, ttl 64, id 3741, offset 0, flags [none], proto UDP (17), length 107)
```

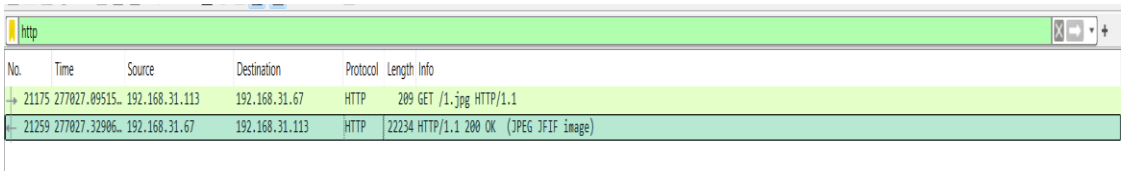

```
    192.168.253.41.53 > 10.0.2.15.42366: 1128 2/0/0 o.pki.goog. CNAME pki-goog.l.google.com., pki-goog.l.google.com. A 142.250.183.131 (79)
20:55:16.135084 IP (tos 0x0, ttl 64, id 55900, offset 0, flags [DF], proto UDP (17), length 56)
    10.0.2.15.58107 > 192.168.253.41.53: 15314+ A? o.pki.goog. (28)
20:55:16.140486 IP (tos 0x0, ttl 64, id 3754, offset 0, flags [none], proto UDP (17), length 107)
    192.168.253.41.53 > 10.0.2.15.58107: 15314 2/0/0 o.pki.goog. CNAME pki-goog.l.google.com., pki-goog.l.google.com. A 142.250.183.131 (79)
20:55:16.891344 IP (tos 0x0, ttl 64, id 47895, offset 0, flags [DF], proto UDP (17), length 63)
    10.0.2.15.33105 > 192.168.253.41.53: 16465+ A? fonts.gstatic.com. (35)
20:55:16.891837 IP (tos 0x0, ttl 64, id 47896, offset 0, flags [DF], proto UDP (17), length 63)
    10.0.2.15.33105 > 192.168.253.41.53: 9565+ AAAA? fonts.gstatic.com. (35)
20:55:17.030500 IP (tos 0x0, ttl 64, id 54663, offset 0, flags [DF], proto UDP (17), length 65)
    10.0.2.15.54221 > 192.168.253.41.53: 51628+ A? accounts.google.com. (37)
20:55:17.030591 IP (tos 0x0, ttl 64, id 54664, offset 0, flags [DF], proto UDP (17), length 65)
    10.0.2.15.54221 > 192.168.253.41.53: 63911+ AAAA? accounts.google.com. (37)
20:55:17.053509 IP (tos 0x0, ttl 64, id 3802, offset 0, flags [none], proto UDP (17), length 79)
    192.168.253.41.53 > 10.0.2.15.33105: 16465 1/0/0 fonts.gstatic.com. A 142.250.196.163 (51)
20:55:17.076582 IP (tos 0x0, ttl 64, id 3805, offset 0, flags [none], proto UDP (17), length 81)
    192.168.253.41.53 > 10.0.2.15.54221: 51628 1/0/0 accounts.google.com. A 74.125.200.84 (53)
20:55:17.076582 IP (tos 0x0, ttl 64, id 3806, offset 0, flags [none], proto UDP (17), length 91)
    192.168.253.41.53 > 10.0.2.15.33105: 9565 1/0/0 fonts.gstatic.com. AAAA 2404:6800:4007:82c::2003 (63)
20:55:17.158436 IP (tos 0x0, ttl 64, id 3807, offset 0, flags [none], proto UDP (17), length 93)
    192.168.253.41.53 > 10.0.2.15.54221: 63911 1/0/0 accounts.google.com. AAAA 2404:6800:4003:c00::54 (65)
20:55:18.152884 IP (tos 0x0, ttl 64, id 37659, offset 0, flags [DF], proto UDP (17), length 73)
    10.0.2.15.60592 > 192.168.253.41.53: 1094+ A? googleads.g.doubleclick.net. (45)
20:55:18.153334 IP (tos 0x0, ttl 64, id 37660, offset 0, flags [DF], proto UDP (17), length 73)
    10.0.2.15.60592 > 192.168.253.41.53: 29003+ AAAA? googleads.g.doubleclick.net. (45)
20:55:18.231929 IP (tos 0x0, ttl 64, id 3859, offset 0, flags [none], proto UDP (17), length 89)
    192.168.253.41.53 > 10.0.2.15.60592: 1094 1/0/0 googleads.g.doubleclick.net. A 142.250.195.162 (61)
20:55:18.231929 IP (tos 0x0, ttl 64, id 3860, offset 0, flags [none], proto UDP (17), length 101)
    192.168.253.41.53 > 10.0.2.15.60592: 29003 1/0/0 googleads.g.doubleclick.net. AAAA 2404:6800:4007:825::2002 (73)
^C
30 packets captured
30 packets received by filter
0 packets dropped by kernel
```

6b. Using tcpdump, get the information about the general incoming network traffic with ip address on specific interface.

```
┌──(root㉿kali)-[/home/karthikeyan]
└─# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
20:58:40.170194 IP 10.0.2.15.35550 > 142.250.182.142.443: Flags [P.], seq 2146859738:2146859777, ack 69033695, win 65535, length 39
20:58:40.171375 IP 142.250.182.142.443 > 10.0.2.15.35550: Flags [.], ack 39, win 65535, length 0
20:58:40.261259 IP 142.250.182.142.443 > 10.0.2.15.35550: Flags [P.], seq 1:40, ack 39, win 65535, length 39
20:58:40.261311 IP 10.0.2.15.35550 > 142.250.182.142.443: Flags [.], ack 40, win 65535, length 0
20:58:55.142014 IP6 fe80::a00:27ff:fe97:ae59 > ff02::2: ICMP6, router solicitation, length 8
20:59:20.176325 IP 10.0.2.15.56554 > 142.250.195.162.443: Flags [P.], seq 4016197073:4016197112, ack 67527136, win 30660, length 39
20:59:20.177600 IP 142.250.195.162.443 > 10.0.2.15.56554: Flags [.], ack 39, win 65535, length 0
20:59:20.236897 IP 142.250.195.162.443 > 10.0.2.15.56554: Flags [P.], seq 1:40, ack 39, win 65535, length 39
20:59:20.236936 IP 10.0.2.15.56554 > 142.250.195.162.443: Flags [.], ack 40, win 30660, length 0
20:59:25.352009 ARP, Request who-has 10.0.2.2 tell 10.0.2.15, length 28
20:59:25.353597 ARP, Reply 10.0.2.2 is-at 52:54:00:12:35:02, length 46
20:59:32.535892 IP 142.250.182.142.443 > 10.0.2.15.35550: Flags [P.], seq 40:113, ack 39, win 65535, length 73
20:59:32.535936 IP 10.0.2.15.35550 > 142.250.182.142.443: Flags [.], ack 113, win 65535, length 0
20:59:32.537251 IP 10.0.2.15.35550 > 142.250.182.142.443: Flags [P.], seq 39:78, ack 113, win 65535, length 39
20:59:32.537957 IP 142.250.182.142.443 > 10.0.2.15.35550: Flags [.], ack 78, win 65535, length 0
20:59:32.540294 IP 10.0.2.15.35550 > 142.250.182.142.443: Flags [P.], seq 78:102, ack 113, win 65535, length 24
20:59:32.541738 IP 142.250.182.142.443 > 10.0.2.15.35550: Flags [.], ack 102, win 65535, length 0
20:59:32.545002 IP 10.0.2.15.35550 > 142.250.182.142.443: Flags [F.], seq 102, ack 113, win 65535, length 0
20:59:32.546004 IP 142.250.182.142.443 > 10.0.2.15.35550: Flags [.], ack 103, win 65535, length 0
20:59:32.604556 IP 142.250.182.142.443 > 10.0.2.15.35550: Flags [F.], seq 113, ack 103, win 65535, length 0
20:59:32.604603 IP 10.0.2.15.35550 > 142.250.182.142.443: Flags [.], ack 114, win 43996, length 0
```

7. Use Wireshark (Latest version) to solve the below scenarios

A. Find the data transferred



No.	Time	Source	Destination	Protocol	Length	Info
21175	277027.09515..	192.168.31.113	192.168.31.67	HTTP	209	GET /1.jpg HTTP/1.1
21259	277027.32906..	192.168.31.67	192.168.31.113	HTTP	22234	HTTP/1.1 200 OK (JPEG JFIF image)

B. Find the source and destination IP of that log.

The source IP is 92.168.31.113 and the destination IP is 192.168.31.67

C. Find the Data length (Bytes) and verify the checksum status on destination.

```
Window: 235
[Calculated window size: 30080]
[Window size scaling factor: 128]
Checksum: 0x16c4 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
```

2. Now you have found that some kind of file has been downloaded by insider in unencrypted web traffic. Your task is to

A. Find the name and type of file.

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
  Date: Sat, 14 Oct 2017 11:04:49 GMT\r\n
  Server: Apache/2.4.25 (Debian)\r\n
  Last-Modified: Sat, 14 Oct 2017 11:01:40 GMT\r\n
  ETag: "60e6d-55b7fb2ae006a"\r\n
  Accept-Ranges: bytes\r\n
  Content-Length: 396909\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: image/jpeg\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.233905409 seconds]
  [Request in frame: 21175]
  [Request URI: http://192.168.31.67/1.jpg]
  File Data: 396909 bytes
```

The name of the file is .jpg and the content type is image/jpeg.

B. Export that file from that web traffic, then analyse the file for any secret information.

The file has the secret information on the image called anthem.

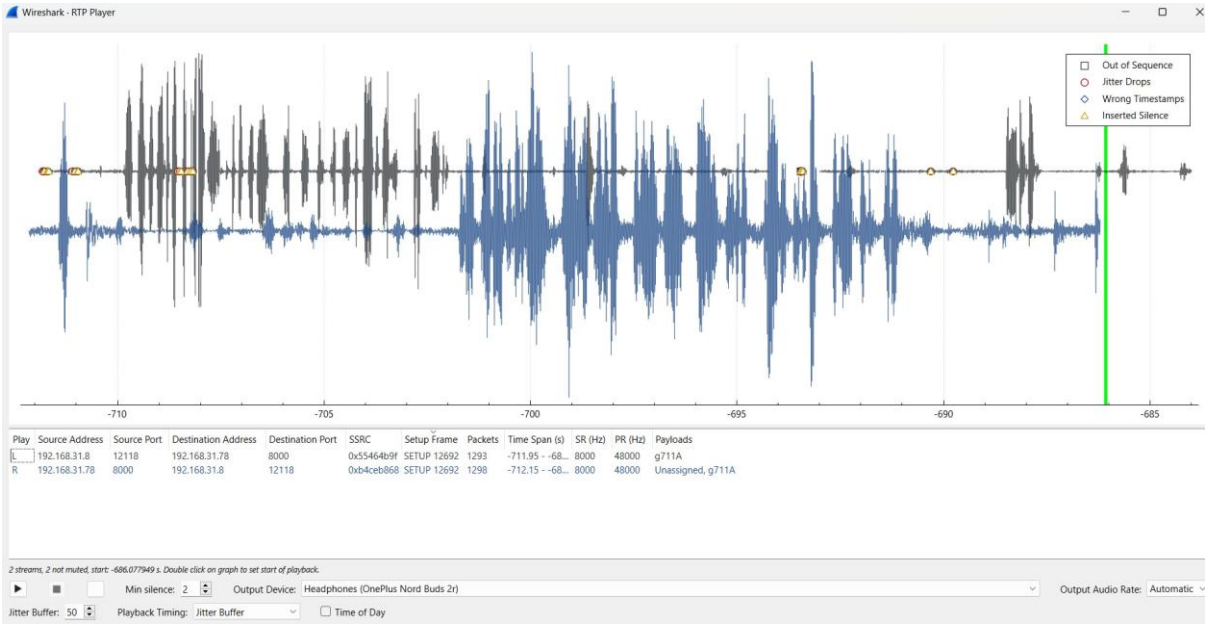
C. Find the hostname in which the file is stored.

```
GET /1.jpg HTTP/1.1
User-Agent: Wget/1.18 (linux-gnu)
Accept: */*
Accept-Encoding: identity
Host: 192.168.31.67
Connection: Keep-Alive
```

The hostname is 192.168.31.67.

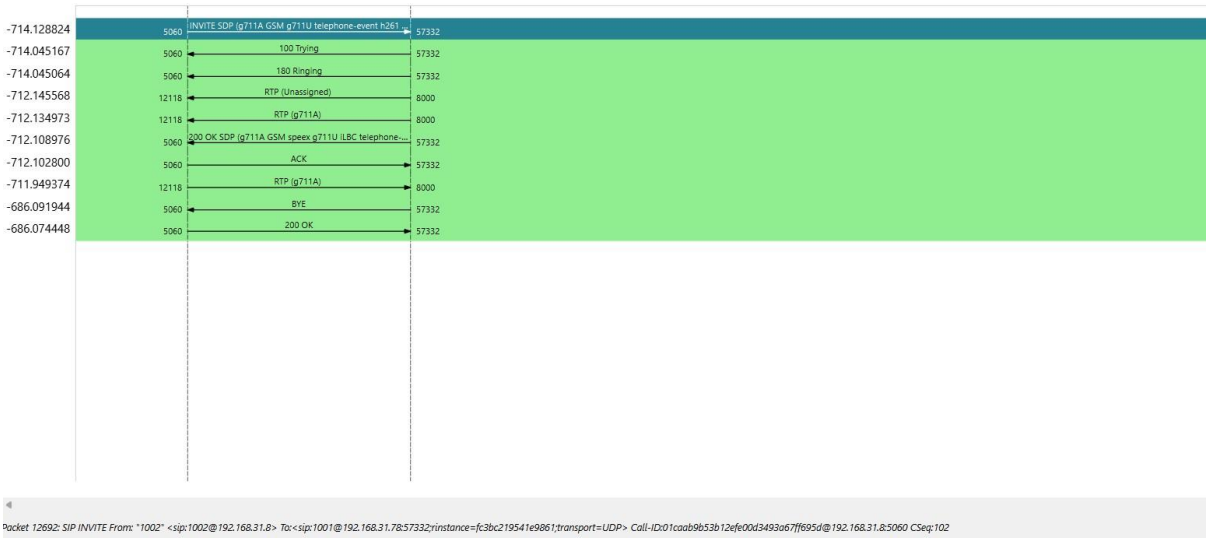
3. Based upon their activities, auditing team has started investigation against them and found that the insider passed some sensitive information via call to someone. The traffic has been captured.

3a. Analyse the traffic and find those conversations and extract the sensitive information in it.



We are also able to listen to the VOIP call in which a password is shared. in the call, we can find out that the password is **LIMBO**

3b. Find the call-ID when the status of the call is ringing.



call ID: 01caab953b12efe00d3493a67ff695d@192.168.31.8:5060.

3c. On further investigation, you have a suspect on some wireless device communications. List out the Bluetooth devices communications from this traffic and find the details about native Bluetooth adapter.

Bluetooth Devices

BD_ADDR	OUI	Name	LMP Version	LMP Subversion	Manufacturer	HCI Version	HCI Revision	Is Local Adapter
00:00:00:00:00:00	00:00:00							
30:21:88:70:9c:18		ZEB-INFINITY V2	2.1 + EDR	256	Unknown 0x%04x			
30:22:00:33:ff:2b		KETTLE	2.1 + EDR	256	Unknown 0x%04x			
3c:bb:fd:a7:07:c1	SamsungElect	Galaxy On5	2.1 + EDR	256	Unknown 0x%04x			
4c:bb:58:43:35:be	ChiconyElect	Virtual Bluetooth Adapter	2.1 + EDR	256	Unknown 0x%04x	2.1 + EDR	256	true
a0:21:95:87:4d:7d	SamsungElect	Vinayakar thunai	2.1 + EDR	256	Unknown 0x%04x			
a0:32:99:3c:65:52	Lenovo	Lenovo VIBE X3	2.1 + EDR	256	Unknown 0x%04x			
dc:e8:38:3e:54:6d	CKTelecom	LS-4505	2.1 + EDR	256	Unknown 0x%04x			
fc:58:fa:28:0d:c2	ShenZhenShiX	HP S6500	2.1 + EDR	256	Unknown 0x%04x			

Bluetooth Device - 4c:bb:58:43:35:be (Virtual Bluetooth Ad...

	Value	Changes
BD_ADDR	4c:bb:58:43:35:be	1
OUI	ChiconyElect	1
Name	Virtual Bluetooth Adapter	1
Class of Device	000100	1
LMP Version	2.1 + EDR	1
LMP Subversion	256	1
Manufacturer	Unknown 0x%04x	1
HCI Version	2.1 + EDR	1
HCI Revision	256	1
Scan		
Authentication		
Encryption		
ACL MTU	8192	1
ACL Total Packets	128	1

16 changes

Close

The information about the native adapter is shown above

4a. Analyse the captured WPA handshake from this traffic and report in detail about it to your administrator.

eapol

No.	Time	Source	Destination	Protocol	Length	Info
8757	5.713306	CKTelecom_41:59…	SamsungElect_a7…	EAPOL	133	Key (Message 1 of 4)
8774	5.718932	SamsungElect_a7…	CKTelecom_41:59…	EAPOL	155	Key (Message 2 of 4)
8776	5.728146	SamsungElect_a7…	CKTelecom_41:59…	EAPOL	133	Key (Message 4 of 4)

Three messages in the WPA handshakes have been captured. It looks like the handshake was incomplete due to incomplete packet deliveries

4b. Geo locate all the endpoint of wireless devices.

Bluetooth · 10 | Ethernet · 16 | IEEE 802.11 · 8 | IEEE 802.15.4 | IPv4 · 23 | IPv6 · 4 | TCP · 32 | UDP · 25

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	Latitude	Longitude	AS Number	AS Organization
52.216.131.91	4,568	5 MB	2,988	4 MB	1,580	92 kB	United States	Ashburn	39.0469°	-77.4903°	16509	AMAZON-02
54.84.236.82	4	571 bytes	3	517 bytes	1	54 bytes	United States	Ashburn	39.0469°	-77.4903°	14618	AMAZON-AES
74.125.68.125	4	259 bytes	2	120 bytes	2	139 bytes	United States		37.751°	-97.822°	15169	GOOGLE
74.125.130.188	2	121 bytes	1	66 bytes	1	55 bytes	United States		37.751°	-97.822°	15169	GOOGLE
74.125.200.189	13	1 kB	6	940 bytes	7	457 bytes	United States		37.751°	-97.822°	15169	GOOGLE
172.217.24.106	5	333 bytes	3	225 bytes	2	108 bytes	United States		37.751°	-97.822°	15169	GOOGLE
172.217.26.206	45	15 kB	23	8 kB	22	6 kB	United States		37.751°	-97.822°	15169	GOOGLE
192.168.2.100	3	497 bytes	3	497 bytes	0	0 bytes						
192.168.31.1	16	6 kB	13	5 kB	3	242 bytes						
192.168.31.8	2,628	571 kB	1,310	284 kB	1,318	287 kB						
192.168.31.16	8	792 bytes	3	159 bytes	5	633 bytes						
192.168.31.25	4	553 bytes	4	553 bytes	0	0 bytes						
192.168.31.67	92	403 kB	45	400 kB	47	3 kB						
192.168.31.78	7,327	5 MB	2,966	396 kB	4,361	5 MB						
192.168.31.89	2	100 bytes	1	50 bytes	1	50 bytes						
192.168.31.113	92	403 kB	47	3 kB	45	400 kB						
192.168.31.120	3	276 bytes	3	276 bytes	0	0 bytes						
192.168.31.255	3	276 bytes	0	0 bytes	3	276 bytes						
216.58.197.46	44	18 kB	23	9 kB	21	9 kB	United States		37.751°	-97.822°	15169	GOOGLE
216.58.197.67	9	499 bytes	1	66 bytes	8	433 bytes	United States		37.751°	-97.822°	15169	GOOGLE
216.58.220.37	2	121 bytes	1	66 bytes	1	55 bytes	United States		37.751°	-97.822°	15169	GOOGLE
217.10.68.152	4	400 bytes	2	260 bytes	2	140 bytes	Germany		51.2993°	9.491°	15594	netzquadrat GmbH
239.255.255.250	18	6 kB	0	0 bytes	18	6 kB						

4c. Analyse the protocol level information transfer between wireless devices.

Wireshark · Wireless LAN Statistics · evidence (2).pcapng

BSSID	Channel	SSID	Percent Packets	Percent Retry	Retry	Beacons	Data Pkts	Probe Reqs	Probe Resp	Auths	Deauths	Other	Protection
dc:e8:38:41:59:dd	6	Rogue	100.0	0.0	0	1	18	0	5	2	0	4	Unknown
33:33:00:00:00:02			24.1	0.0	0	0	7	0	0	0	0	0	
33:33:00:00:00:16			6.9	0.0	0	0	2	0	0	0	0	0	
33:33:ff:a7:07:c2			6.9	0.0	0	0	2	0	0	0	0	0	
3c:bb:fd:a7:07:c2			75.9	0.0	0	14	1	0	1	2	0	4	
4c:eb:42:70:ac:4c			6.9	0.0	0	0	0	0	2	0	0	0	
c0:c1:21:2d:95:b1			6.9	0.0	0	0	0	0	2	0	0	0	
dc:e8:38:41:59:dd			58.6	0.0	0	4	2	0	5	2	0	4	Base station
ff:ff:ff:ff:ff:ff			13.8	0.0	0	0	4	0	0	0	0	0	