

# Assignment 3

## Anjishnu Mukherjee B05-511017020 (510517086)

3a. Analyse the packets (across all layers) exchanged with your computer while executing the following commands: (i) ping, (ii) traceroute, (iii) dig, (iv) arp, (v) wget.

(i) ping :

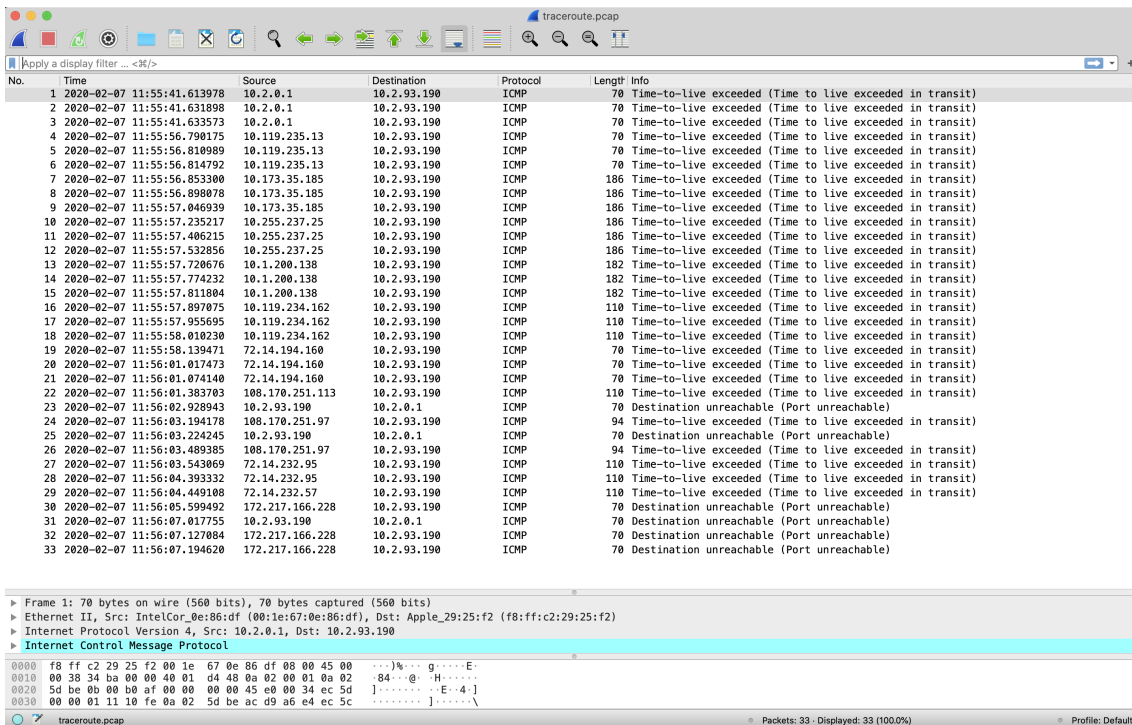
- We capture the relevant packets using the icmp filter.
- There are 2 types of packets : Ping Request and Response
- A request is made from my laptop's IP to the destination IP.
- Corresponding response is present in the next packet.

The screenshot displays the Wireshark network protocol analyzer interface. The top toolbar includes various icons for file operations, display filters, and packet analysis. The main window is divided into three panes:

- Packet List:** Shows a list of 12 captured packets. The first 6 are ICMP Echo (ping) requests, and the next 6 are ICMP Echo (ping) replies. The source and destination IP addresses are 192.168.1.8 and 8.8.8.8, respectively.
- Packet Details:** Shows the hierarchical structure of the selected packet (Frame 1). It includes Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol.
- Packet Bytes:** Shows the raw hex and ASCII data of the selected packet.

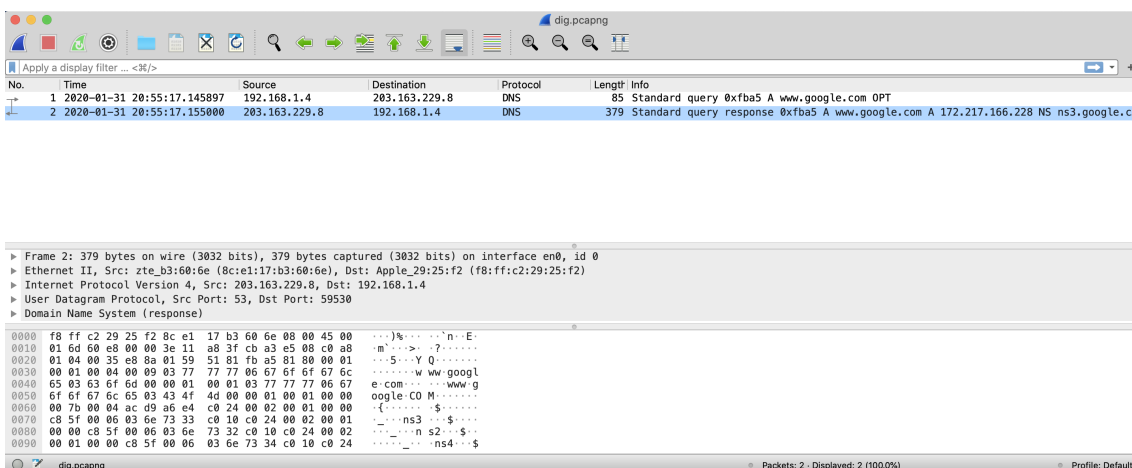
(ii) traceroute :

- Makes use of TTL mechanism limiting the time to live for each packet to some value.
- Devices send back an ICMP message when dropping the packet.
- Thus we filter using ICMP filter.
- The protocol used for the probes in a UNIX system is UDP and responses are send back using ICMP.



(iii) dig :

- We filter using dns protocol.
- The info for the request packet shows 'A' because we are querying for a IP address.
- The DNS response packet shows the IP address corresponding to the name specified in the request.

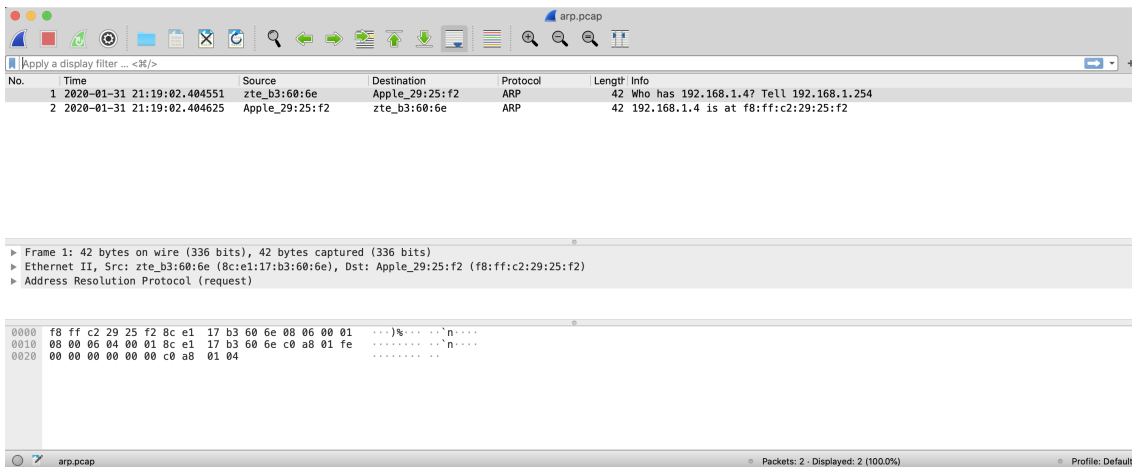


(iv) arp :

- We filter using the ARP protocol.
- 2 packets are captured - ARP request and ARP reply.
- The request packet asks for the MAC address corresponding to the

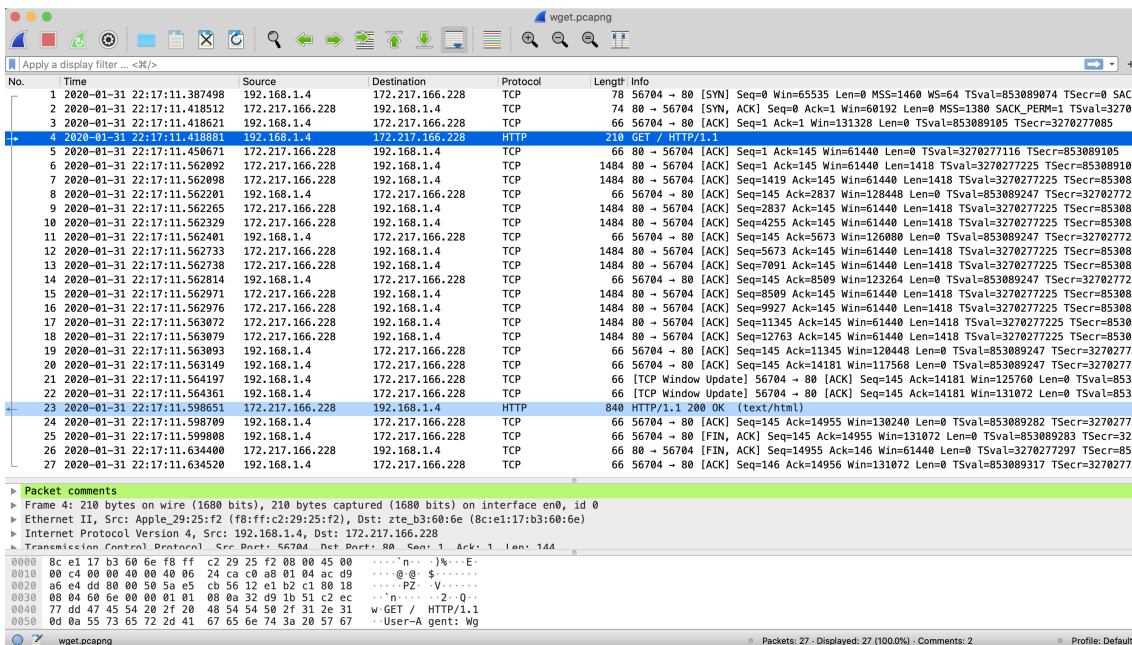
IP address of en0.

- en0 is the active promiscuous mode interface I am using in wireshark.
- The reply packet gives the MAC address.



(v) wget :

- We filter using tcp.port == 80 ( I am trying to get a http page using wget.)
- Many packets get captured, but only 2 are of interest. (GET/HTTP and HTTP OK)
- GET corresponds to sending the request and OK is the response on successful completion.

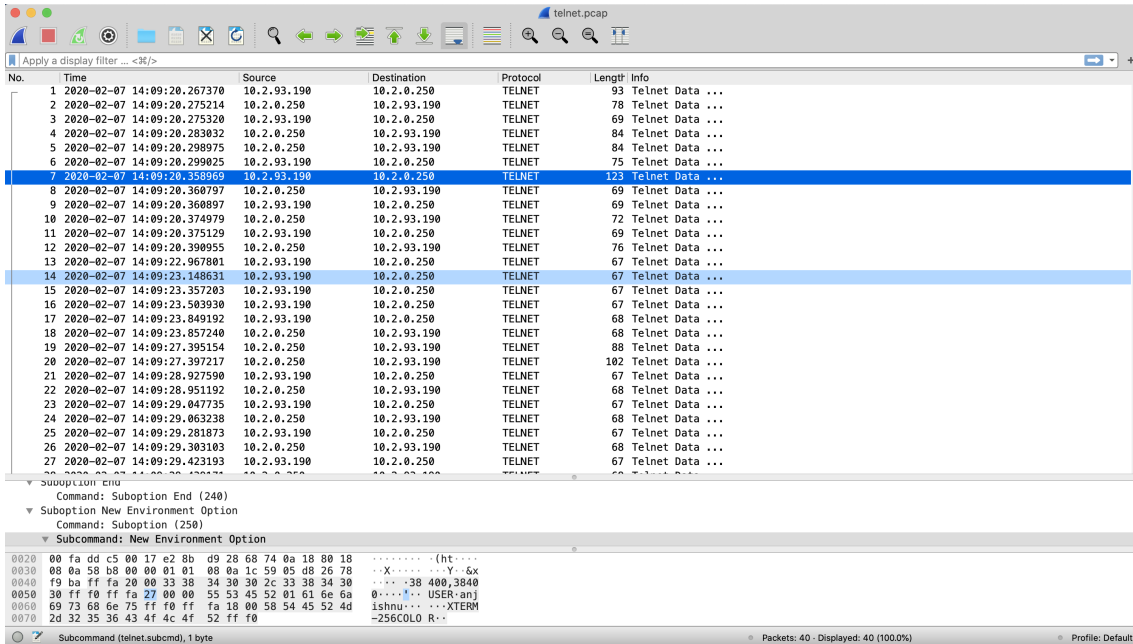


3b. Capture the packets while sending/receiving telnet request/response between your computer and a custom server running the telnet daemon. What is your observation while analysing the application layer data?

- telnet 10.2.0.250 (Username/Password : test/test)
- Packets between client and server are NOT encrypted using SSH

or any other protocols.

- The only protocol being used is telnet which is not secured in any form.

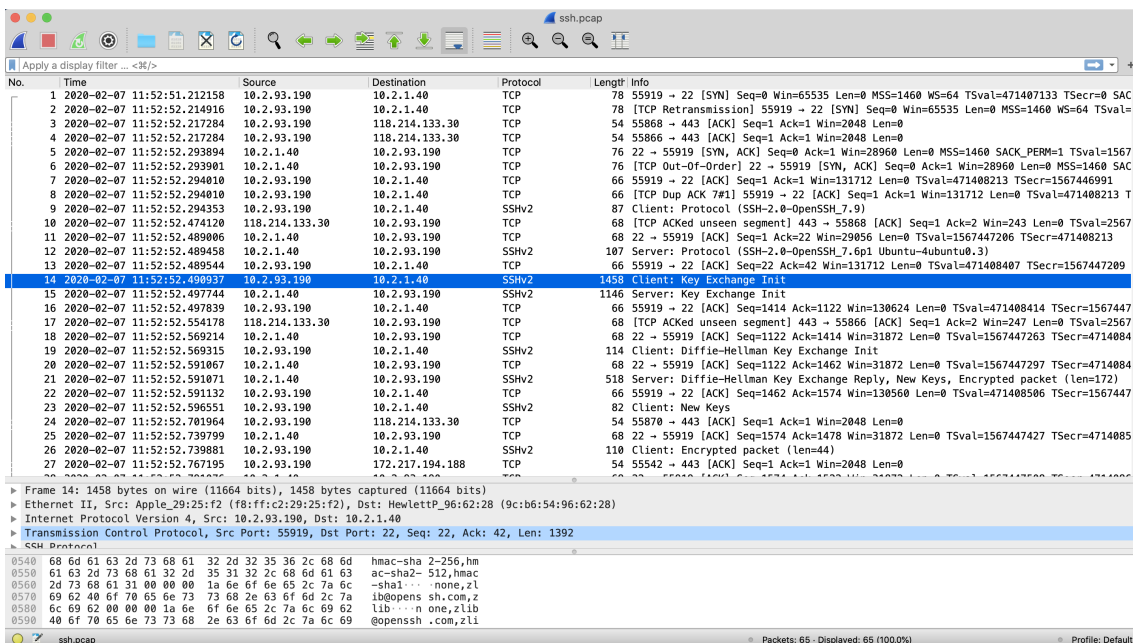


The image shows a Wireshark packet capture titled 'telnet.pcap'. The packet list on the left shows 28 packets, all of which are Telnet data. The selected packet (No. 7) is expanded, showing the 'Suboption End' command (248) and the 'Suboption New Environment Option' command (250). The packet details pane shows the 'Subcommand: New Environment Option' and the raw data bytes.

No.	Time	Source	Destination	Protocol	Length	Info
1	2020-02-07 14:09:20.267370	10.2.93.190	10.2.0.250	TELNET	93	Telnet Data ...
2	2020-02-07 14:09:20.275214	10.2.0.250	10.2.93.190	TELNET	78	Telnet Data ...
3	2020-02-07 14:09:20.275320	10.2.93.190	10.2.0.250	TELNET	69	Telnet Data ...
4	2020-02-07 14:09:20.283832	10.2.0.250	10.2.93.190	TELNET	84	Telnet Data ...
5	2020-02-07 14:09:20.298975	10.2.0.250	10.2.93.190	TELNET	84	Telnet Data ...
6	2020-02-07 14:09:20.299025	10.2.93.190	10.2.0.250	TELNET	75	Telnet Data ...
7	2020-02-07 14:09:20.358909	10.2.93.190	10.2.0.250	TELNET	123	Telnet Data ...
8	2020-02-07 14:09:20.360997	10.2.0.250	10.2.93.190	TELNET	69	Telnet Data ...
9	2020-02-07 14:09:20.360997	10.2.93.190	10.2.0.250	TELNET	69	Telnet Data ...
10	2020-02-07 14:09:20.374979	10.2.0.250	10.2.93.190	TELNET	72	Telnet Data ...
11	2020-02-07 14:09:20.375129	10.2.93.190	10.2.0.250	TELNET	69	Telnet Data ...
12	2020-02-07 14:09:20.390955	10.2.0.250	10.2.93.190	TELNET	76	Telnet Data ...
13	2020-02-07 14:09:22.967801	10.2.93.190	10.2.0.250	TELNET	67	Telnet Data ...
14	2020-02-07 14:09:23.148631	10.2.93.190	10.2.0.250	TELNET	67	Telnet Data ...
15	2020-02-07 14:09:23.357203	10.2.93.190	10.2.0.250	TELNET	67	Telnet Data ...
16	2020-02-07 14:09:23.503930	10.2.93.190	10.2.0.250	TELNET	67	Telnet Data ...
17	2020-02-07 14:09:23.849192	10.2.93.190	10.2.0.250	TELNET	68	Telnet Data ...
18	2020-02-07 14:09:23.857240	10.2.0.250	10.2.93.190	TELNET	68	Telnet Data ...
19	2020-02-07 14:09:27.395154	10.2.0.250	10.2.93.190	TELNET	88	Telnet Data ...
20	2020-02-07 14:09:27.397217	10.2.0.250	10.2.93.190	TELNET	102	Telnet Data ...
21	2020-02-07 14:09:28.927590	10.2.93.190	10.2.0.250	TELNET	67	Telnet Data ...
22	2020-02-07 14:09:28.951192	10.2.0.250	10.2.93.190	TELNET	68	Telnet Data ...
23	2020-02-07 14:09:29.047735	10.2.93.190	10.2.0.250	TELNET	67	Telnet Data ...
24	2020-02-07 14:09:29.063238	10.2.0.250	10.2.93.190	TELNET	68	Telnet Data ...
25	2020-02-07 14:09:29.281873	10.2.93.190	10.2.0.250	TELNET	67	Telnet Data ...
26	2020-02-07 14:09:29.383103	10.2.0.250	10.2.93.190	TELNET	68	Telnet Data ...
27	2020-02-07 14:09:29.423193	10.2.93.190	10.2.0.250	TELNET	67	Telnet Data ...
28	2020-02-07 14:09:29.423193	10.2.93.190	10.2.0.250	TELNET	67	Telnet Data ...

3c. Capture the packets while sending/receiving ssh request/response between your computer and one of the department servers. What is your observation while analysing the application layer data?

- Packets exchanged between the client and server are encrypted using SSHv2 based on OpenSSH 7.9
- The TLS protocol is used in the application layer.



The image shows a Wireshark packet capture titled 'ssh.pcap'. The packet list on the left shows 28 packets, all of which are SSHv2 data. The selected packet (No. 14) is expanded, showing the 'SSHv2 Client: Key Exchange Init' and the raw data bytes.

No.	Time	Source	Destination	Protocol	Length	Info
1	2020-02-07 11:52:51.212158	10.2.93.190	10.2.1.40	TCP	78	55919 → 22 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=471407133 TSecr=0 SAC=
2	2020-02-07 11:52:52.214916	10.2.93.190	10.2.1.40	TCP	78	[TCP Retransmission] 55919 → 22 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=
3	2020-02-07 11:52:52.217284	10.2.93.190	118.214.133.30	TCP	54	55868 → 443 [ACK] Seq=1 Ack=1 Win=2048 Len=0
4	2020-02-07 11:52:52.217284	10.2.93.190	118.214.133.30	TCP	54	55866 → 443 [ACK] Seq=1 Ack=1 Win=2048 Len=0
5	2020-02-07 11:52:52.293894	10.2.1.40	10.2.93.190	TCP	76	22 → 55919 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1567
6	2020-02-07 11:52:52.293901	10.2.1.40	10.2.93.190	TCP	76	[TCP Out-of-Order] 22 → 55919 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SAC
7	2020-02-07 11:52:52.294010	10.2.93.190	10.2.1.40	TCP	66	55919 → 22 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=471408213 TSecr=1567446991
8	2020-02-07 11:52:52.294010	10.2.93.190	10.2.1.40	TCP	66	[TCP Dup ACK #71] 55919 → 22 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=471408213 T
9	2020-02-07 11:52:52.294353	10.2.93.190	10.2.1.40	SSHv2	87	Client: Protocol (SSH-2.0-OpenSSH_7.9)
10	2020-02-07 11:52:52.474120	118.214.133.30	10.2.93.190	TCP	68	[TCP ACKed unseen segment] 443 → 55868 [ACK] Seq=1 Ack=2 Win=243 Len=0 TSval=2567
11	2020-02-07 11:52:52.489006	10.2.1.40	10.2.93.190	TCP	68	22 → 55919 [ACK] Seq=1 Ack=2 Win=29056 Len=0 TSval=1567447206 TSecr=471408213
12	2020-02-07 11:52:52.489458	10.2.1.40	10.2.93.190	SSHv2	107	Server: Protocol (SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3)
13	2020-02-07 11:52:52.489544	10.2.93.190	10.2.1.40	TCP	66	55919 → 22 [ACK] Seq=22 Ack=42 Win=131712 Len=0 TSval=471408407 TSecr=1567447209
14	2020-02-07 11:52:52.490937	10.2.93.190	10.2.1.40	SSHv2	1458	Client: Key Exchange Init
15	2020-02-07 11:52:52.497744	10.2.1.40	10.2.93.190	SSHv2	1146	Server: Key Exchange Init
16	2020-02-07 11:52:52.497839	10.2.93.190	10.2.1.40	TCP	66	55919 → 22 [ACK] Seq=1414 Ack=1122 Win=130624 Len=0 TSval=471408414 TSecr=1567447
17	2020-02-07 11:52:52.554178	118.214.133.30	10.2.93.190	TCP	68	[TCP ACKed unseen segment] 443 → 55866 [ACK] Seq=1 Ack=2 Win=247 Len=0 TSval=2567
18	2020-02-07 11:52:52.569214	10.2.1.40	10.2.93.190	TCP	68	22 → 55919 [ACK] Seq=1122 Ack=1414 Win=31872 Len=0 TSval=1567447263 TSecr=4714084
19	2020-02-07 11:52:52.569315	10.2.93.190	10.2.1.40	SSHv2	114	Client: Diffie-Hellman Key Exchange Init
20	2020-02-07 11:52:52.591067	10.2.1.40	10.2.93.190	TCP	68	22 → 55919 [ACK] Seq=1122 Ack=1462 Win=31872 Len=0 TSval=1567447297 TSecr=4714084
21	2020-02-07 11:52:52.591071	10.2.1.40	10.2.93.190	SSHv2	518	Server: Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=172)
22	2020-02-07 11:52:52.591132	10.2.93.190	10.2.1.40	TCP	66	55919 → 22 [ACK] Seq=1462 Ack=1574 Win=130560 Len=0 TSval=471408506 TSecr=1567447
23	2020-02-07 11:52:52.596551	10.2.93.190	10.2.1.40	SSHv2	82	Client: New Keys
24	2020-02-07 11:52:52.701964	10.2.93.190	118.214.133.30	TCP	54	55870 → 443 [ACK] Seq=1 Ack=1 Win=2048 Len=0
25	2020-02-07 11:52:52.739799	10.2.1.40	10.2.93.190	TCP	68	22 → 55919 [ACK] Seq=1574 Ack=1478 Win=31872 Len=0 TSval=1567447427 TSecr=4714085
26	2020-02-07 11:52:52.739891	10.2.93.190	10.2.1.40	SSHv2	110	Client: Encrypted packet (len=44)
27	2020-02-07 11:52:52.767195	10.2.93.190	172.217.194.188	TCP	54	55542 → 443 [ACK] Seq=1 Ack=1 Win=2048 Len=0
28	2020-02-07 11:52:52.767195	10.2.93.190	10.2.0.250	TCP	66	55542 → 443 [ACK] Seq=1 Ack=1 Win=2048 Len=0

3d. Enter the URL: <http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html> and capture packets using Wireshark. After your browser has displayed the INTRO-wireshark-file1.html page (it is a simple one line of congratulations), stop Wireshark packet capture. Answer the following from the packets captured:

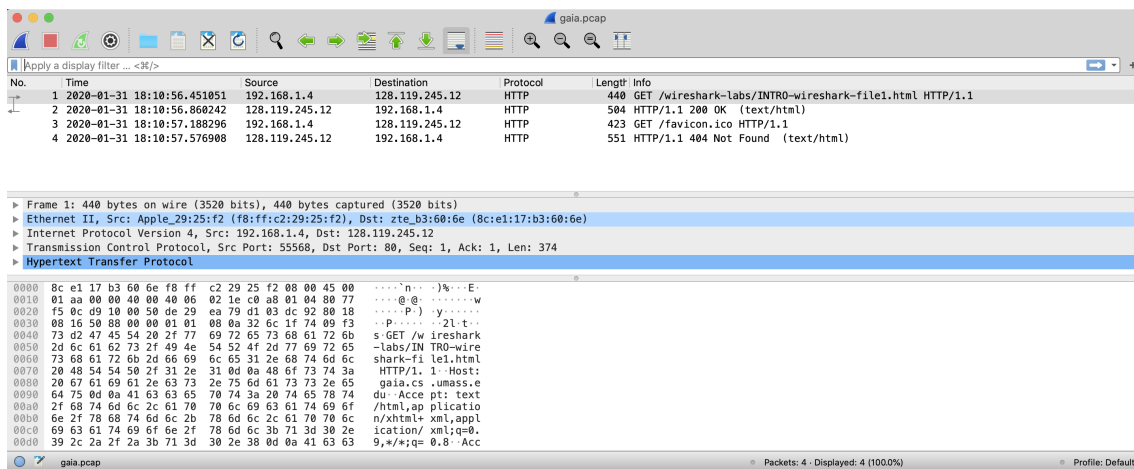
- We use http filter for Wireshark capture.

(i) How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received?

HTTP GET timestamp - Jan 31, 2020 18:10:56.451051000 IST

HTTP OK timestamp - Jan 31, 2020 18:10:56.860242000 IST

Time delta - 0.409191000 seconds



(ii) What is the Internet address of the gaia.cs.umass.edu?

What is the Internet address of your computer?

Support your answer with an appropriate screenshot from your computer.

IP of gaia.cs.umass.edu : 128.119.245.12

Verified further using nslookup.

---

(base) anjishnu@mymacpro ~ % nslookup 128.119.245.12

;; Got recursion not available from 89.207.131.21, trying next server

Server: 8.8.8.8

Address: 8.8.8.8#53

Non-authoritative answer:

12.245.119.128.in-addr.arpa name = gaia.cs.umass.edu.

---

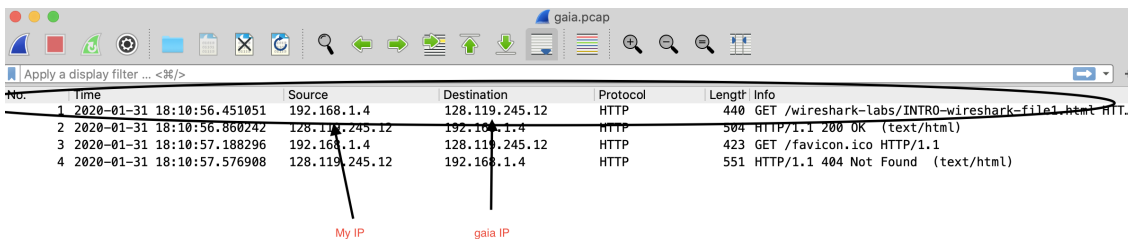
```
Default (zsh)
(base) anjishnu@mymacpro ~ % nslookup 128.119.245.12
;; Got recursion not available from 89.207.131.21, trying next server
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
12.245.119.128.in-addr.arpa      name = gaia.cs.umass.edu.

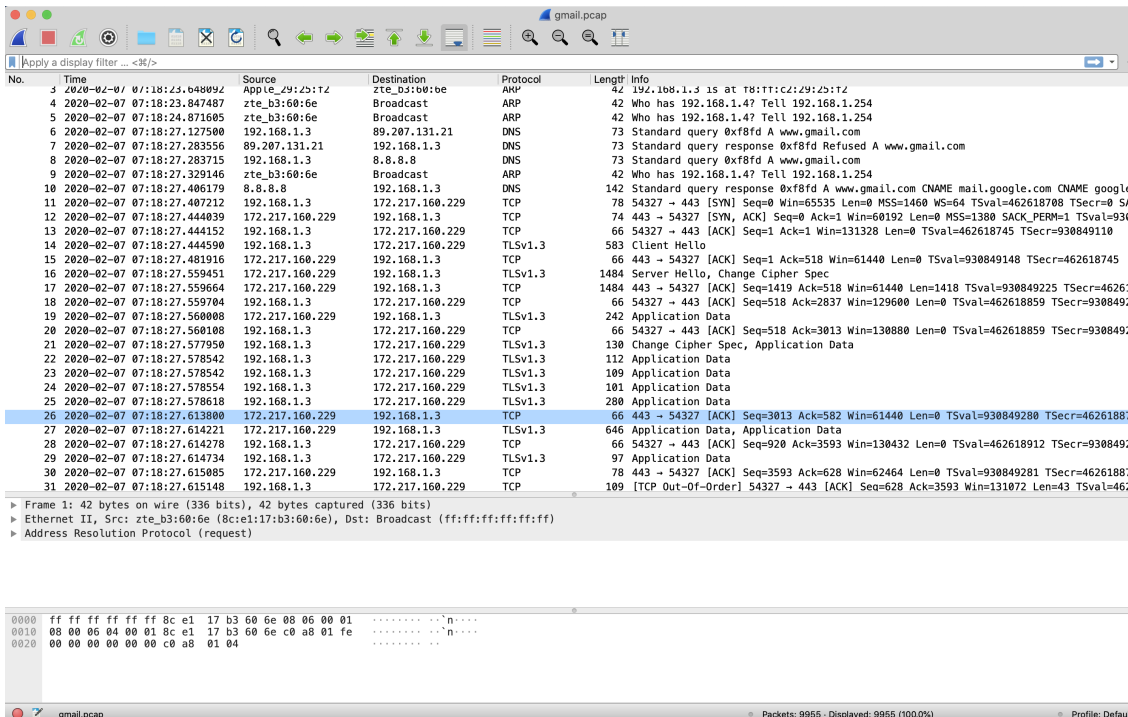
Authoritative answers can be found from:
```

IP of active promiscuous mode interface of my machine is 192.168.1.4

(Relevant screenshots attached with this document in submission email.)



3e. Start the Wireshark packet capturing service. Enter the URL: <https://www.gmail.com> on your browser and sign-in to your gmail account by providing credentials (Username/Password). Answer the following from the captured packets:



i. Is there any difference in the application layer protocol?

This URL uses HTTPS whereas for the previous question the URL used HTTP. Thus Gmail application data is encrypted whereas for the previous question, we can directly see all the data captured from the packets as it was not secure.

ii. How it is different from the HTTP data you analysed in the above problem?

The application data is encrypted using TLS protocol version 1.3 in this case. Whereas for the previous question, there was no encryption involved.