भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

Computer Networks Lab (CS 342) Lab Assignment - 02

Student Details:

Name: AB Satyaprakash Roll Number: 180123062 Department: Mathematics and Computing

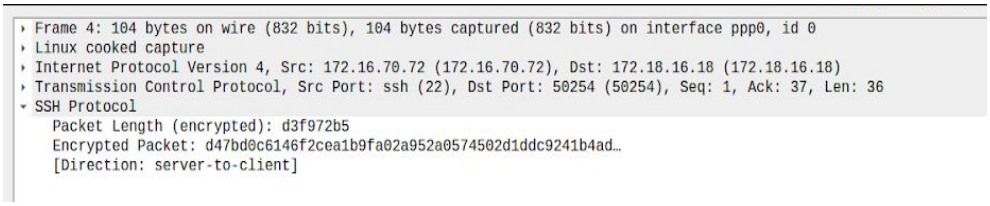
Question 1.

The application assigned to me was **FortiClient VPN (Desktop App)**. The protocols used by the application at the different layers are : (the physical layer protocols couldn't be figured out from the traces)

1. **Application Layer :**
   - **SSH** (Secure Shell Protocol)
   - **TLSv1.2** (Transport Layer Security Protocol)
   - **DNS** (Domain Network System Protocol)
2. **Transport Layer :**
   - **TCP** (Transmission Control Protocol)
   - **UDP** (User Datagram Protocol)
3. **Network Layer :**
   - **IPv4** (Internet Protocol Version 4)
4. **Link Layer :**
   - **Ethernet II**

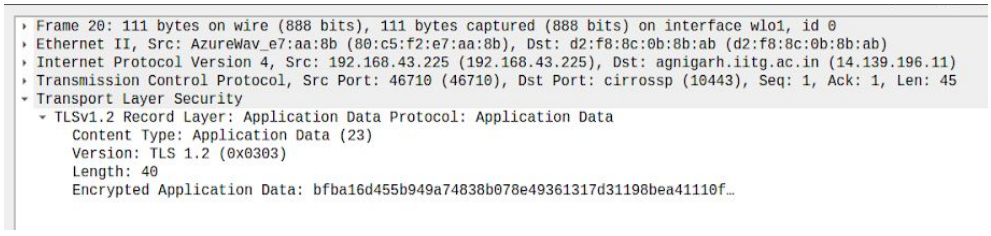
1. Application Layer

a. The **SSH** packet format has the following fields:



- **Packet Length** is the length of the packet in bytes, not including **HMAC** value or the **Packet Length field** itself.
- **Padding Length** is the length of random padding in bytes.
- **Payload** is the useful contents of the packet. If compression has been negotiated, this field is compressed.
- **Random Padding** is the arbitrary-length padding, such that the total length of (packet_length || padding_length || payload || random padding) is a multiple of the cipher block size or 8, whichever is larger.
- **MAC** (Message Authentication Code) : MAC (Message Authentication Code) field contains the MAC value, if the message authentication has been negotiated, this field contains the MAC bytes.

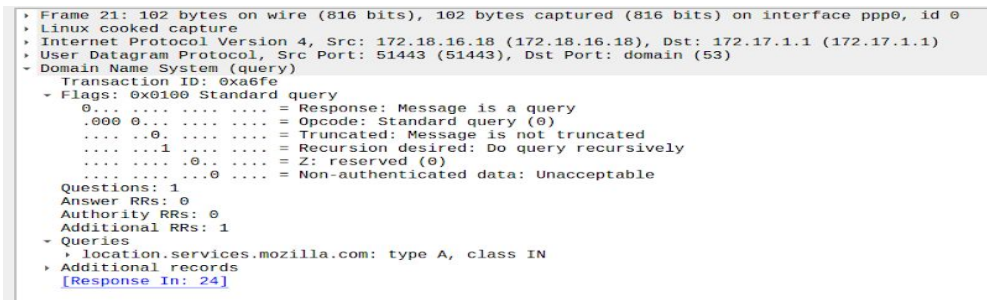
b. The **TLS (v1.2 or v1.3)** packet format has the following fields:



- **Content type** represents the type of data being encrypted
- **Version** represents the version of TLS (in my case TLSv1.2)
- **Length** represents the length of the packet in bytes
- **Encrypted Application Data** is the data that has been encrypted by the TLS

c. The **DNS** packet has the following fields:

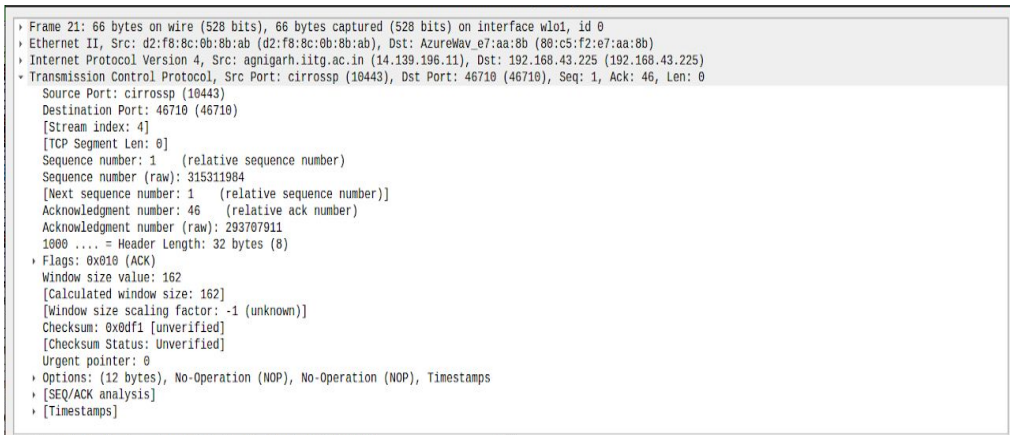
- **Transaction ID** is a 16 bit header section in a DNS message
- **DNS Header Flag** is a flag field in the 2nd 16 bit word of the query
- **Questions** represent the number of questions asked in the message in a specified format.
- **Answer RRs, Authority RRs and Additional RRs** are the various resource records.
- **Queries** represent the domain that is being queried.
- **Type** is the type of resource record
- **Class** represents class code.



2. Transport Layer

a. The **TCP** packet format has the following fields:

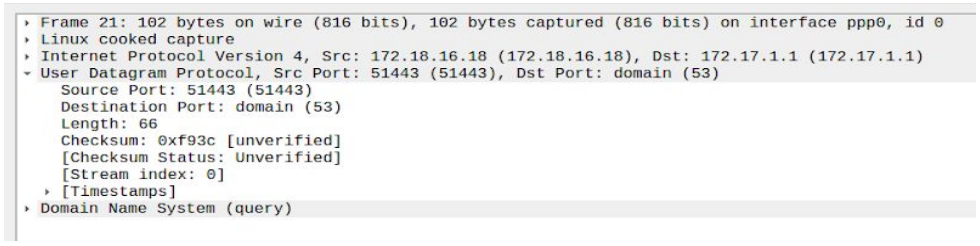
- **Source Port** and **Destination Port fields** (16 bits each) identify the end points of the connection.
- **Sequence Number field** (32 bits) specifies the number assigned to the first byte of data in the current message.
- **Acknowledgement Number field** (32 bits) contains the value of the next sequence number that the sender of the segment is expecting to receive, if the ACK control bit is set.
- **Data Offset (a.k.a. Header Length) field** (variable length) tells how many 32-bit words are contained in the TCP header.
- **Reserved field** (6 bits) must be zero. This is for future use.
- **Flags field** (6 bits) contains the various flags:
  URG—Indicates that some urgent data has been placed.
  ACK—Indicates that acknowledgement number is valid.
  PSH—Indicates that data should be passed to the application as soon as possible.
  RST—Resets the connection.
  SYN—Synchronizes sequence numbers to initiate a connection.
  FIN—Means that the sender of the flag has finished sending data.



- **Window field** (16 bits) specifies the size of the sender's receive window (that is, buffer space available for incoming data).
- **Checksum field** (16 bits) indicates whether the header was damaged in transit.
- **Urgent pointer field** (16 bits) points to the first urgent data byte in the packet.
- **Options field** (variable length) specifies various TCP options.
- **Data field** (variable length) contains upper-layer information.

b. The **UDP** packet format has the following fields:

- **Source port number** identifies the sender's port, when used, and should be assumed to be the port to reply to if needed.
- **Destination port number** identifies the receiver's port and is required.
- **Length** specifies the length in bytes of the UDP header and UDP data.

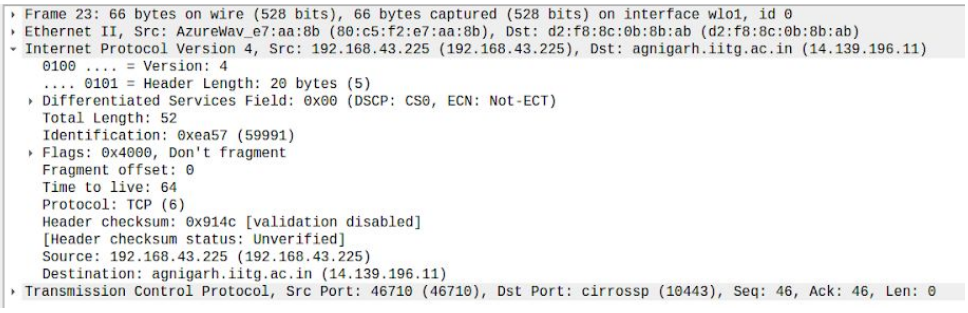


- **Checksum** field may be used for error-checking of the header and data.

3. Network Layer :

a. The **IPv4** packet has the following fields:

- **Version** is the version number of Internet Protocol used (e.g. IPv4).
- **IHL** is the length of the entire IP header.
- **DSCP** is Differentiated Services Code Point. This is a type of service.
- **ECN** is Explicit Congestion Notification. It carries information about the congestion seen in the route.

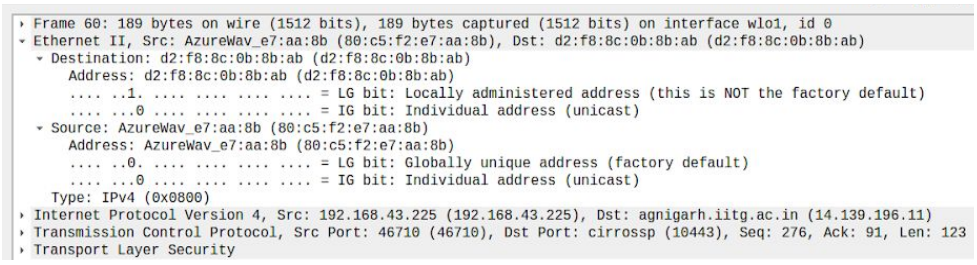


- **Total Length** is the length of the entire IP Packet (including IP header and IP Payload).

- **Identification** – If an IP packet is fragmented during the transmission, all the fragments contain the same identification number to identify the original IP packet they belong to.
- **Flags** are required by the network resources, if an IP Packet is too large to handle, these 'flags' tell if they can be fragmented or not. In this 3-bit flag, the MSB is always set to '0'.
- **Fragment Offset** tells the exact position of the fragment in the original IP Packet.
- **Time to Live** - To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross.
- **Protocol** tells the network layer at the destination host, to which protocol this packet belongs to, i.e. the next level Protocol.
- **Header Checksum** field is used to keep the checksum value of the entire header which is then used to check if the packet is received error-free.
- **Source Address** is a 32-bit address of the sender (or source) of the packet.
- **Destination Address** is a 32-bit address of the Receiver (or destination) of the packet.

4. Link Layer :

a. The **Ethernet II** packet has the following fields :
   - **Destination and Source IP** represent the IP address of endpoints
   - **Destination and Source MAC** represent the MAC address and manufacturer details of destination and source
   - **Type** tells about the IP version, i.e, IPv4 or IPv6



Question 2.

FortiClient VPN (Desktop App) has numerous functionalities, some which are freely (and readily) available -

a. Establish Connection with Remote Server
b. Disconnecting from Remote Server
c. Establish SSH connection to a remote PC

However other functionalities like **Endpoint Status (realtime), Endpoint Quarantine (due to security reasons), Software Inventory and Dashboard management** are not available for free. We'll look at the three available features and attempt to explain those.

a. **Establish Connection with a Remote Server** (using **IPv4** over **Ethernet II**)
   - A **DNS** query is sent to **IP addr(destination)** via **UDP**. The destination in our case is **agnigarh.iitg.ac.in**
   - On receipt of a response, a **TCP handshake** is done using **ACK and SYN** methods
   - The **TLS** then shares encrypted keys and source data between endpoints. The **TCP** packets are also exchanged.
   - Other protocols involved are **HTTP** (for text files), **OCSP** (for certificates like SHA certificates), **IGMPv3** (for multicasting).

b. **Disconnecting from a Remote Server** (using **IPv4** over **Ethernet II**)
   - **TCP** and **TLS** do their respective jobs before we disconnect, i.e, **TCP** exchanges packets, while **TLS** exchanges security keys
   - **NTP** protocol helps in time synchronization and **IGMPv3** helps to disconnect multiple destinations simultaneously.

c. **Establishing a SSH connection to a remote PC**
   - After **TCP** handshake is done between source and remote PCs, sensitive information like passwords and application packets are shared over **TLS** and **TCP** connections respectively.
   - **NTP** in this case ensures that the endpoint PCs are synchronised to measure the time delays effectively.
   - **HTTP GET** verifies connectivity between the end points.

(For question 3 please find the .._evening files for both types attached)

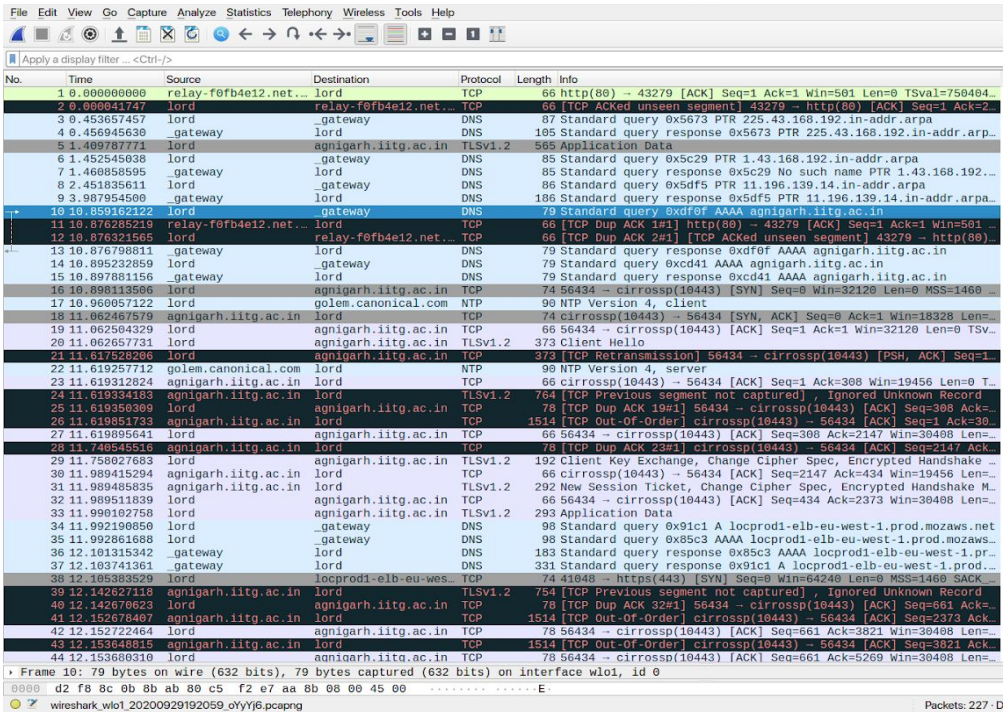
Question 3.

For this question we'll analyse the 2 functionalities given in the question doc (as sample functionalities), i.e,

- **Establish Connection (**with **agnigarh.iitg.ac.in)**

I have attached a screenshot from my laptop (and also added a file **q3_establish_connection_evening.pcapng**), to show my work. As can be seen from the image below, these are the observations :

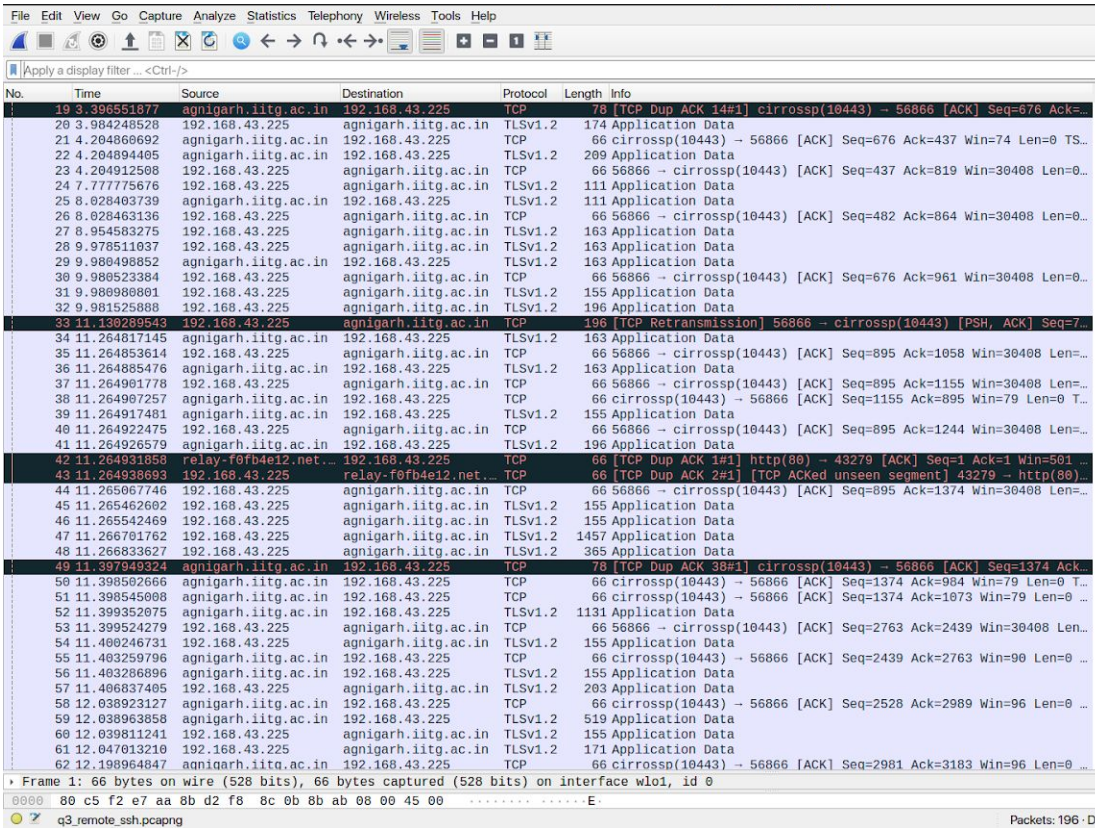
a. In line number 10 (highlighted in blue), the first **DNS** messages are sent to the remote server, to check connectivity.
b. Then after a standard query response is received (line 15), the **TCP** starts handshake via **ACK** and **SYN** methods (line 16 and 18 highlighted in gray).
c. Then **TLSv1.2** does a **'Client Hello'**, and exchanges encrypted and sensitive messages like Client Keys, Cipher Specs etc. (line 29)



d. After this **TCP** application packets are exchanged from line number 30 onwards.

e. Thereafter **DNS** and **UDP** make a connection with the destination (agnigarh.iitg.ac.in) over IPv4 whose IP address is 14.139.196.11.

Yes there are handshaking messages in the sequence, which is obvious because before every TCP application data (packet) sharing, there has to be a handshake done to ensure that there is a smooth data transfer from source to destination and vice versa.



- **<u>Establishing a SSH connection to my remote PC (satyapra@172.16.70.72)</u>**

I have attached a screenshot in this case as well, showing my work (and as before provided a file **q3_remote_ssh_evening.pcapng**). These are the observations which can be made by considering the image below.

a. From the image we see that there is no **DNS** in this case, but only **TCP** and **TLS**. This is mostly because we opened Wireshark after the connection was established over **VPN**, and then we used **SSH** to connect to my remote machine.

b. As is expected **TLSv1.2** and **TCP** are doing their respective jobs in sharing secure data (labelled as application data), and sending packets of information between endpoints.

c. The two IP addresses in this case that communicate with each other are **192.168.43.225** (IP of my wifi network) and **agnigarh.iitg.ac.in**, i.e the source and destination.

d. One peculiar thing in this case is the absence of handshakes which can be attributed to the already complete VPN connection and only SSH remote information sharing.

(For questions 4 and 5 please find the attached 6 trace files and nomenclature is .._morning, .._afternoon and .._evening)

Question 4.

For **establish connection** functionality

<u>Time of the day(↦) / Data(↧)</u>	**<u>Morning</u>**	**<u>Afternoon</u>**	**<u>Evening</u>**
a) Throughput	1848 bytes/s	1552 bytes/s	2106 bytes/s

b) RTT (timespan/total packets)	104 ms (avg)	118.7789 ms (avg)	105.5859 ms (avg)
c) Packet Size	192 bytes	184	222 bytes
d) No of packets lost	0	0	0
e) TCP and UDP packets	TCP -> 199 UDP -> 18	TCP -> 171 UDP -> 22	TCP -> 207 UDP -> 18
f) No of responses wrt to 1 request	0.8389	0.8380	0.8907

For **ssh remote connection** functionality

Time of the day(↦) / Data(↧)	**Morning**	**Afternoon**	**Evening**
a) Throughput	1605 bytes/s	1726 bytes/s	1046 bytes/s
b) RTT (timespan/total packets)	107.1984 ms (avg)	98.4317 ms (avg)	165.1224 ms (avg)
c) Packet Size	172 bytes	170 bytes	173 bytes
d) No of packets lost	0	0	0
e) TCP and UDP packets	TCP -> 235 UDP -> 20	TCP -> 221 UDP -> 2	TCP -> 190 UDP -> 0
f) No of responses wrt to 1 request	0.9318	1.0458	0.9191

We found these details as follows :

- **Throughput, RTT, Packet Size (average), Number of packets lost** were found in *Wireshark->Statistics->CaptureFile*
- TCP and UDP packets number was found in *Wireshark->Statistics->Protocol Hierarchy*
- Number of requests wrt to 1 request was found by dividing (after applying filter ***ip.src==192.168.43.225 and ip.dst==192.168.43.225)*** the total number of packets coming to (dest) and from (src) host, and taking the ratio. It came close to 1 (which is expected ideally).

Question 5.

2 types of IP addresses were always found and remained constant in both **SSH** and **Establish Connection** traces. They were the host IP and the destination IP addresses corresponding to ***192.168.43.225 (host)*** and ***14.139.196.11(agnigarh.iitg.ac.in).***

Apart from this several other IP addresses were found particularly the ones related to AWS (amazon web services) and that IP was changing with time. The IP addresses were found in *Wireshark->Statistics->Resolved Addresses->(apply filter hosts).*

The IPs at different times of day were :

a. **52.48.132.232 (compute.amazonaws.com) -> afternoon ssh and establish connection trace**
b. **54.228.140.0 (prod.mozaws.com) -> evening and morning establish connection trace**

This can be explained because the **AWS** server is a cloud server and thus has a dynamic location (not placed at a single geographical location). However **agnigarh** and my **PC** were geographically static and thus had the same IP all day long. Apart from **AWS**, I also found **Canonical (Ubuntu)** servers behaving similarly, because of the same reason.