**Computer Networks Labsheet2**

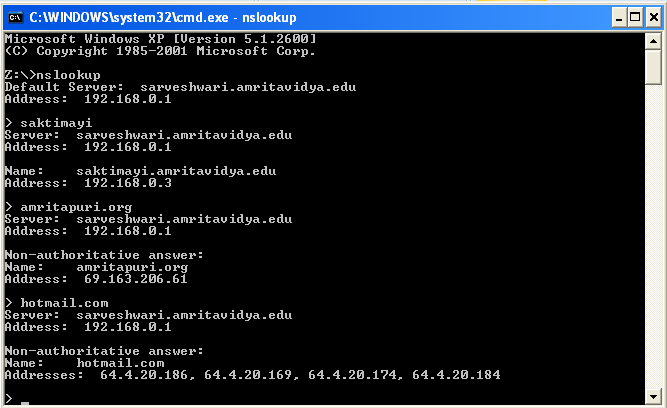
Analyzing the Web pages accessed using Wireshark

**Part1: Mapping the given URL to IP address using nslookup**

To run *nslookup* in Linux/Unix, you just type the *nslookup* command on the command line. To run it in Windows, open the Command Prompt and run *nslookup* on the command line.

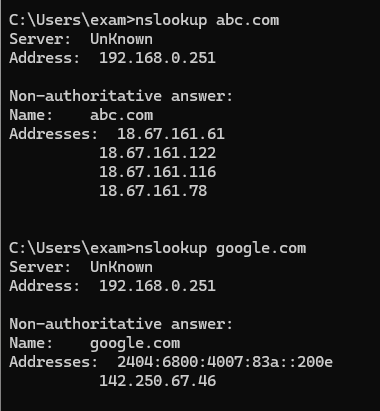
We as humans usually use the website address known as Uniform Resource Locator (URL) to identify the destination host. But IP address is used by the devices to identify the server in the network. Domain Name System (DNS) service by the DNS server helps the client to translate an unknown URL to an IP address and also cache it in the device itself for subsequent access. The local cache maintains DNS record having a map between URL and IP address.

In it is most basic operation, *nslookup* tool allows the host running the tool to query any specified DNS server for a DNS record. To accomplish this task, *nslookup* sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result.



The IP address of abc.com is 18.67.161.78

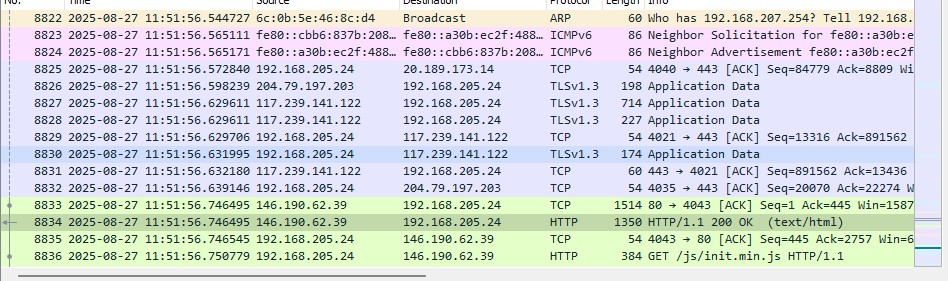
The IP address of google.com is 142.250.67.46



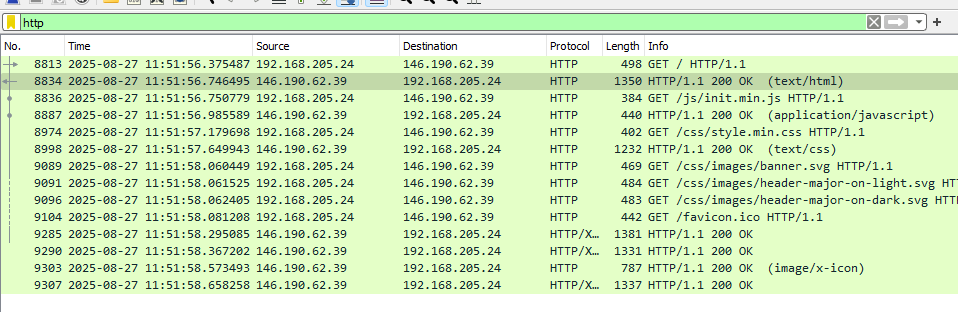
Find the IP address of all websites that you are using for Part2.

**Part2: Understanding the TCP/IP protocol stack using Wireshark**

1. Open Packet sniffer [Wireshark] Application and Capture the Wi-Fi/ Ethernet Interface



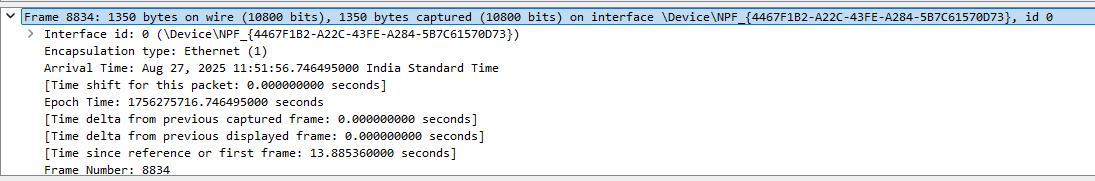
1. Do this activity and capture frames.
   1. Request for a web page by typing the least used URL in the webserver

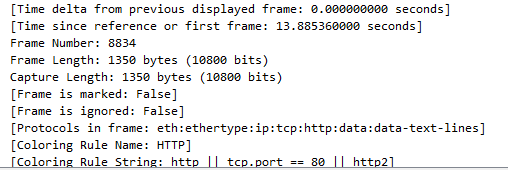


1. Briefly explain the **Encapsulation** process in at least one http request frame of the protocol analyzed. Also complete the table below by PDU contents and details requested in each layer. Also try to provide proof in the form of relevant screenshot.

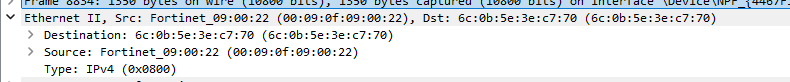
|  |  |  |  |
| --- | --- | --- | --- |
| Layer | Protocol | Important Contents (You can get details on clicking on a packet) | Purpose of the content specified in the Layer |
| Application | http | HTTP/1.1 200 OK\r\n [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n] Response Version: HTTP/1.1 Status Code: 200 [Status Code Description: OK] Response Phrase: OK Date: Wed, 27 Aug 2025 06:08:49 GMT\r\n Server: Apache/2.4.18 (Ubuntu)\r\n Last-Modified: Sat, 08 Jul 2017 17:58:19 GMT\r\n | Contains protocols required for the connection with another device |
| Transport | TCP | Transmission Control Protocol, Src Port: 80, Dst Port: 44667, Seq: 6446, Ack: 1081, Len: 604 [3 Reassembled TCP Segments (3524 bytes):  #2096(1460), #2097(1460), #2098(604)] [Frame: 2096, payload: 0-1459 (1460 bytes)] [Frame: 2097, payload: 1460-2919 (1460 bytes)] [Frame: 2098, payload: 2920-3523 (604 bytes)] [Segment count: 3] [Reassembled TCP length: 3524 | Process to process communication between two devices in a network |
| Network | IP | Internet Protocol Version 4, Src: 107.181.87.5, Dst: 192.168.205.5 0100 .... = Version: 4 .... 0101 = Header Length: 20 bytes (5) Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) Total Length: 644 Identification: 0x6912 (26898) Flags: 0x40, Don't fragment Fragment Offset: 0 Time to Live: 64 | route data packets between different networks by determining the best paths and managing logical addressing |
| Data Link | Ethernet II | Ethernet II, Src: Fortinet\_09:00:22 (00:09:0f:09:00:22), Dst: 24:6a:0e:77:0e:d66a:0e:77:0e:d6) Destination: 24:6a:0e:77:0e:d6 (24:6a:0e:77:0e:d6) Source: Fortinet\_09:00:22 (00:09:0f:09:00:22) Type: IPv4 (0x0800) | Converts the response data into frames and sends it over the network interface |

PHYSICAL LAYER

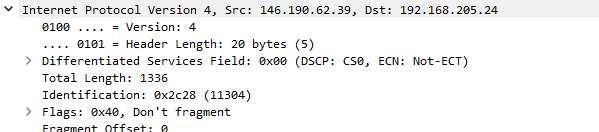


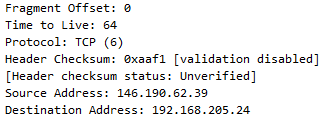


DATA LINK LAYER

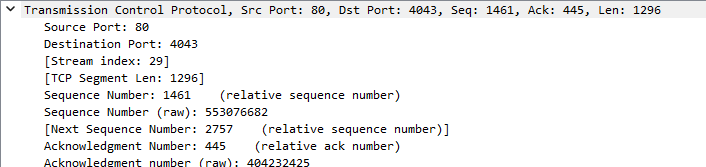


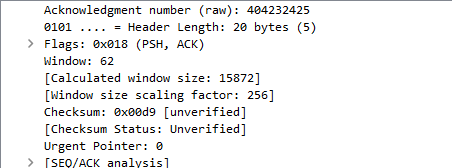
NETWORK LAYER

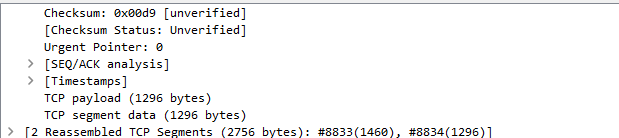




TRANSPORT LAYER







APPLICATION LAYER

