



**Date: 06/08/2025**

### **Lab Practical #09:**

Study Packet capture and header analysis by Wireshark (HTTP, TCP, UDP, IP, etc.)

### **Practical Assignment #09:**

#### **1. Explain usage of Wireshark tool.**

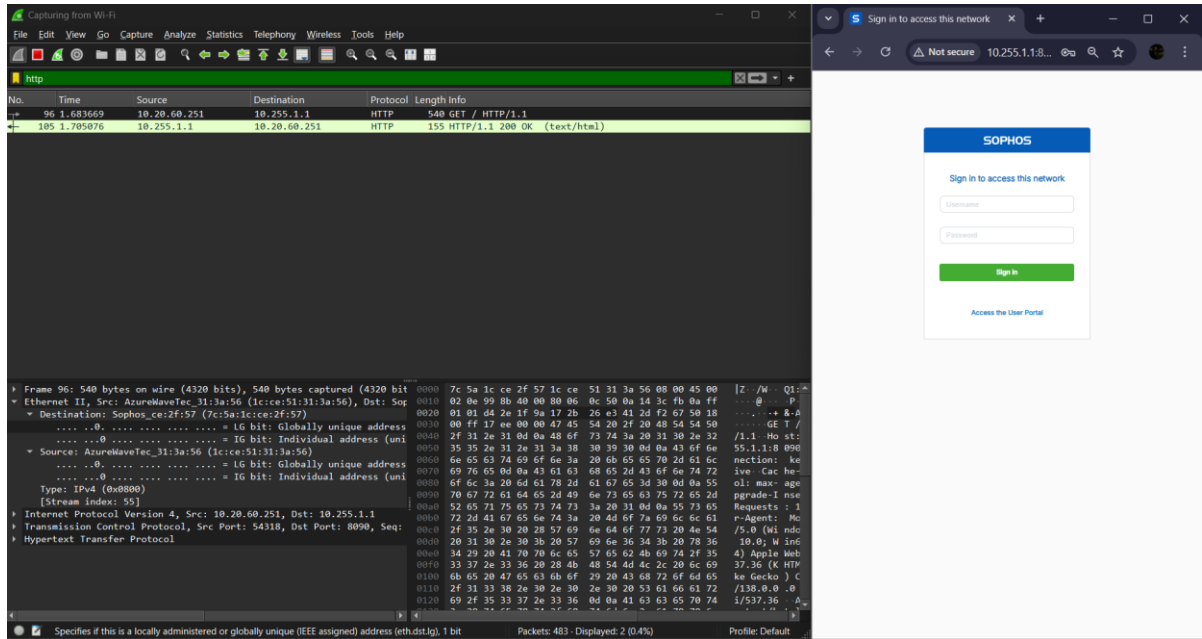
Wireshark is a tool that helps you see what's really happening on a network. Think of it like a microscope for network traffic—it lets you capture and study the data that travels between computers, servers, and devices.

##### **Usage:**

1. To Watch Network Traffic in Real-Time
  - You can see every packet (tiny piece of data) going in and out of your network.
  - It shows details like where it came from, where it's going, and what protocol it uses.
2. To Understand Network Protocols
  - Wireshark can “translate” thousands of different protocols (like HTTP, DNS, or TCP) into a readable form, so you don't have to decode them yourself.
3. To Fix Network Problems
  - If your internet feels slow or connections keep dropping, Wireshark can help find the cause—like packet loss, delays, or misconfigured devices.
4. To Keep an Eye on Performance
  - You can check how much bandwidth is being used and whether the network is overloaded.
5. To Filter and Focus
  - Wireshark lets you filter out unnecessary information and look only at the data that matters to you.
6. To Save and Share Results
  - You can save what you capture and share it with your team for further investigation.

Date: 06/08/2025

## 2. Packet capture and header analysis by Wireshark (HTTP, TCP, UDP, IP, etc.)

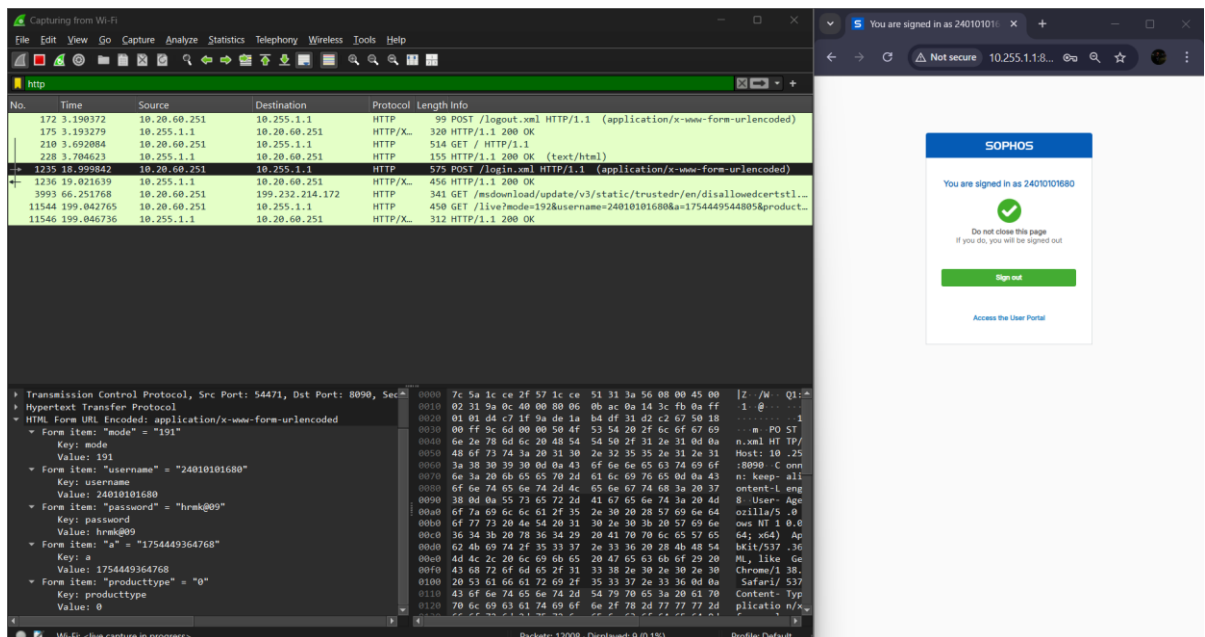


The screenshot shows a Wireshark packet capture of an HTTP GET request. The packet list pane shows a single packet (No. 105) at time 1.683669, source 10.20.60.251, destination 10.255.1.1, protocol HTTP, length 155. The packet details pane shows the following layers:

- Ethernet II, Src: AzureWaveTec\_31:3a:56 (Icice51:31:3a:56), Dst: Sophos\_ces2f:57 (7c:5a:1c:ce:2f:57)
- Internet Protocol Version 4, Src: 10.20.60.251, Dst: 10.255.1.1
- Transmission Control Protocol, Src Port: 54318, Dst Port: 8090, Seq: 155
- Hypertext Transfer Protocol

The packet bytes pane shows the raw data of the packet, including the Ethernet II header, IP header, TCP header, and the HTTP GET request body.

### 1.1 Analysis of HTTP without login Sophos



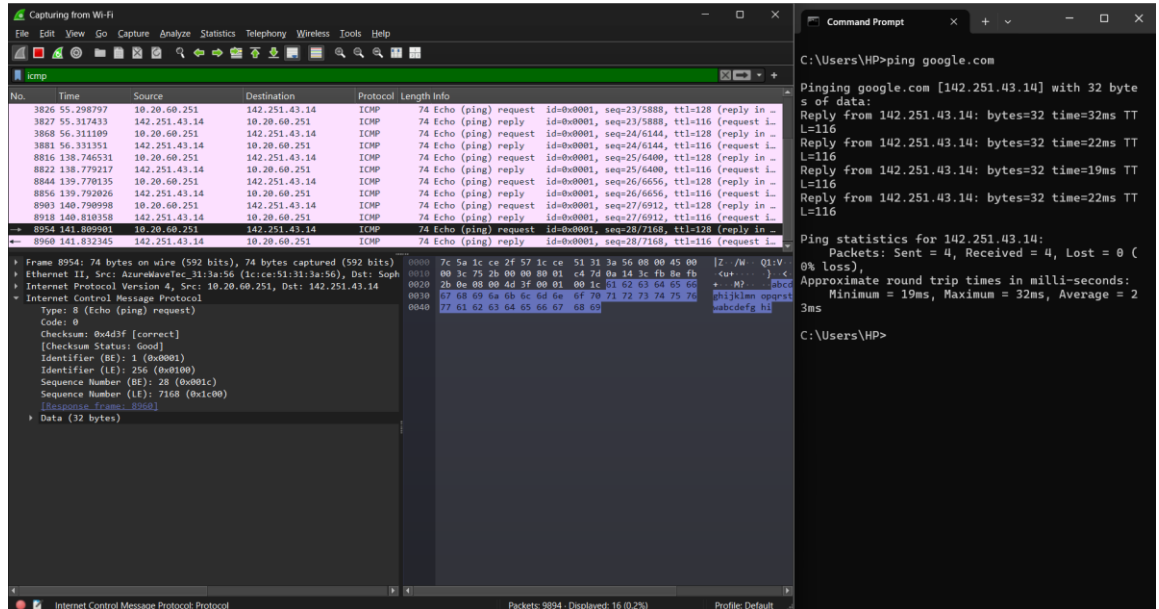
The screenshot shows a Wireshark packet capture of an HTTP POST request. The packet list pane shows a single packet (No. 1235) at time 1.622842, source 10.20.60.251, destination 10.255.1.1, protocol HTTP, length 578. The packet details pane shows the following layers:

- Ethernet II, Src: AzureWaveTec\_31:3a:56 (Icice51:31:3a:56), Dst: Sophos\_ces2f:57 (7c:5a:1c:ce:2f:57)
- Internet Protocol Version 4, Src: 10.20.60.251, Dst: 10.255.1.1
- Transmission Control Protocol, Src Port: 54471, Dst Port: 8090, Seq: 3993
- Hypertext Transfer Protocol

The packet bytes pane shows the raw data of the packet, including the Ethernet II header, IP header, TCP header, and the HTTP POST request body. The HTML Form URL Encoded data is visible in the packet details pane, showing fields for mode, username, password, and producttype.

### 1.2 Analysis of HTTP with login Sophos

Date: 06/08/2025



The image shows a Wireshark packet capture of ICMP Echo (ping) requests and replies between 10.20.60.251 and 142.251.43.14. The packet list shows several successful ping requests and replies. The packet details pane for packet 8954 shows the ICMP Echo (ping) request structure, including the Echo (ping) request type, code, checksum, identifier, and sequence number. The packet bytes pane shows the raw data of the ICMP request.

Command Prompt output:

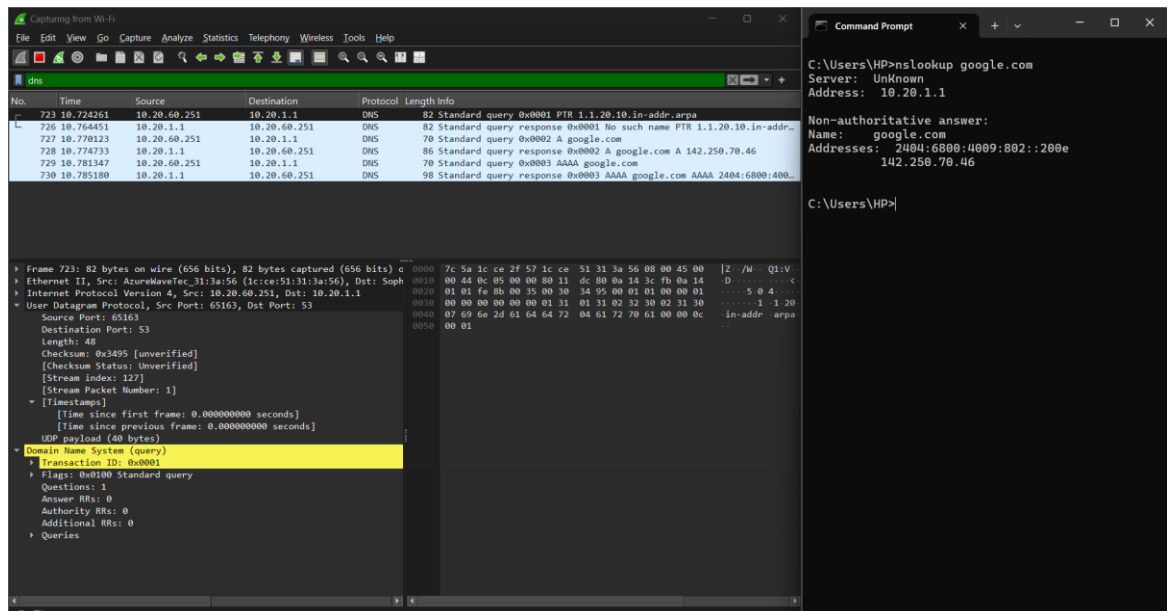
```
C:\Users\HP>ping google.com

Pinging google.com [142.251.43.14] with 32 bytes of data:
Reply from 142.251.43.14: bytes=32 time=32ms TTL=116
Reply from 142.251.43.14: bytes=32 time=22ms TTL=116
Reply from 142.251.43.14: bytes=32 time=19ms TTL=116
Reply from 142.251.43.14: bytes=32 time=22ms TTL=116

Ping statistics for 142.251.43.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 19ms, Maximum = 32ms, Average = 23ms

C:\Users\HP>
```

### 1.3 Analysis of ICMP



The image shows a Wireshark packet capture of DNS traffic. The packet list shows a standard query for PTR 1.1.20.10.in-addr.arpa and a standard query response. The packet details pane for packet 723 shows the DNS query structure, including the transaction ID, flags, questions, and answers. The packet bytes pane shows the raw data of the DNS query.

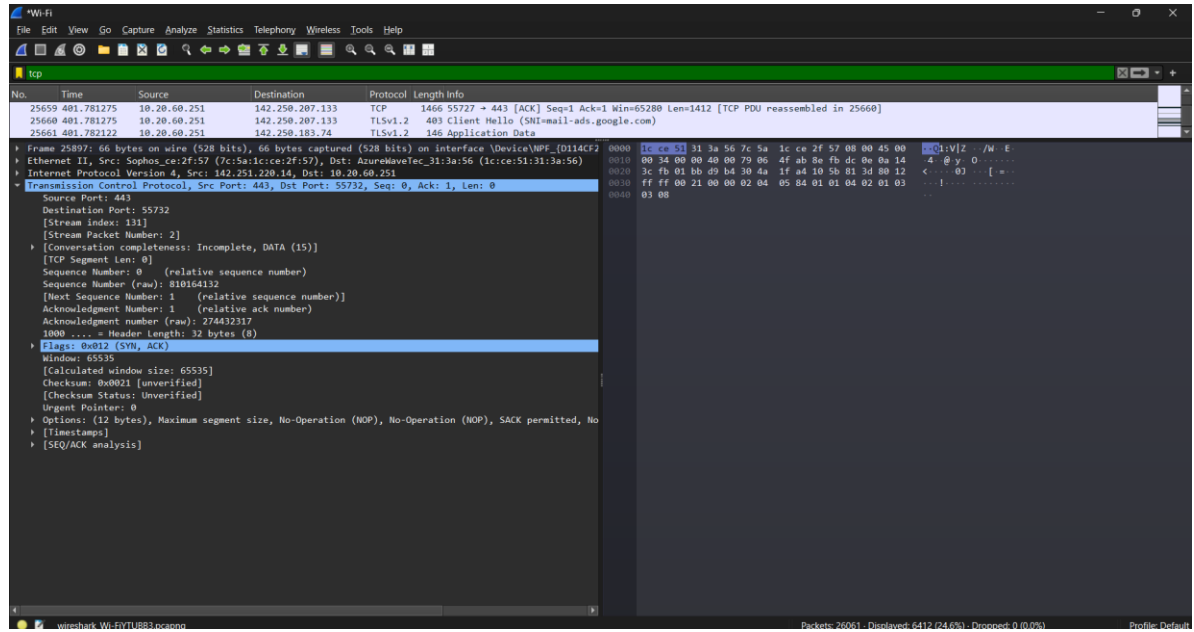
Command Prompt output:

```
C:\Users\HP>nslookup google.com
Server: Unknown
Address: 10.20.1.1

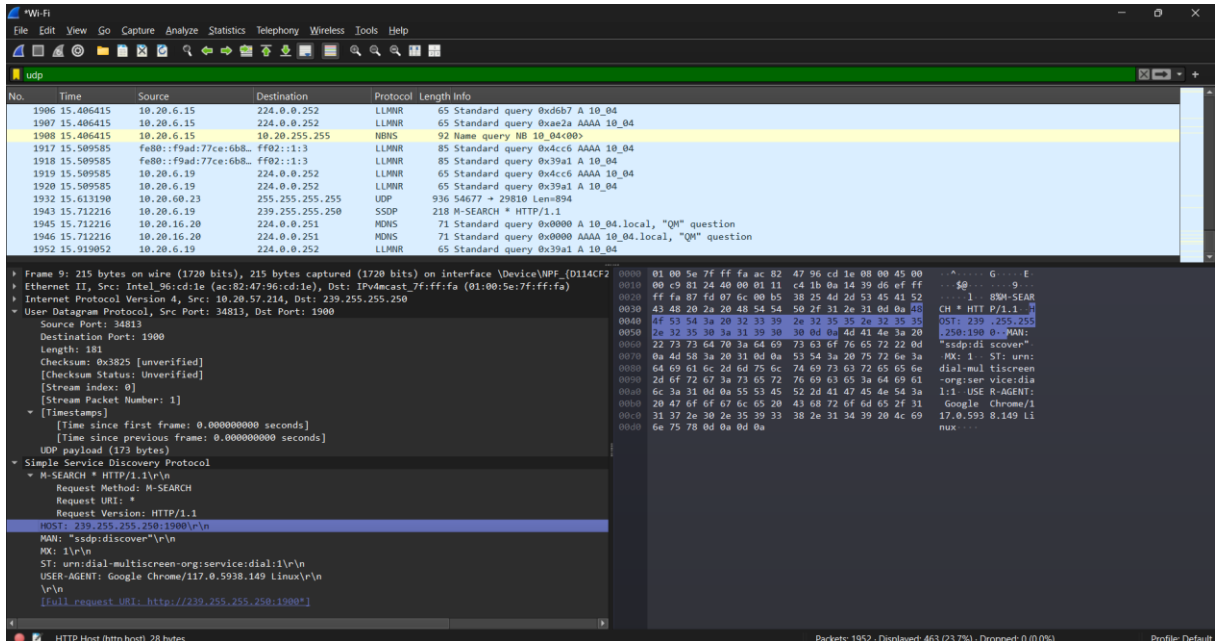
Non-authoritative answer:
Name:   google.com
Address: 2404:6800:4009:802::200e
        142.250.70.46

C:\Users\HP>
```

### 1.4 Analysis of DNS



### 1.5 Analysis of TCP



### 1.6 Analysis of UDP