**Title: Protocol Capture Using Wireshark**  
**Name:** Jingar Vishal  
**Date:** 4/6/25

**1. Objective**

The objective of this task is to capture and analyze the traffic of different network protocols using **Wireshark**. The protocols targeted for capture include:

* TCP (Transmission Control Protocol)
* HTTP (HyperText Transfer Protocol)
* DNS (Domain Name System)

Screenshots and .pcap files of each protocol capture are included to validate the process.

**2. Tools Used**

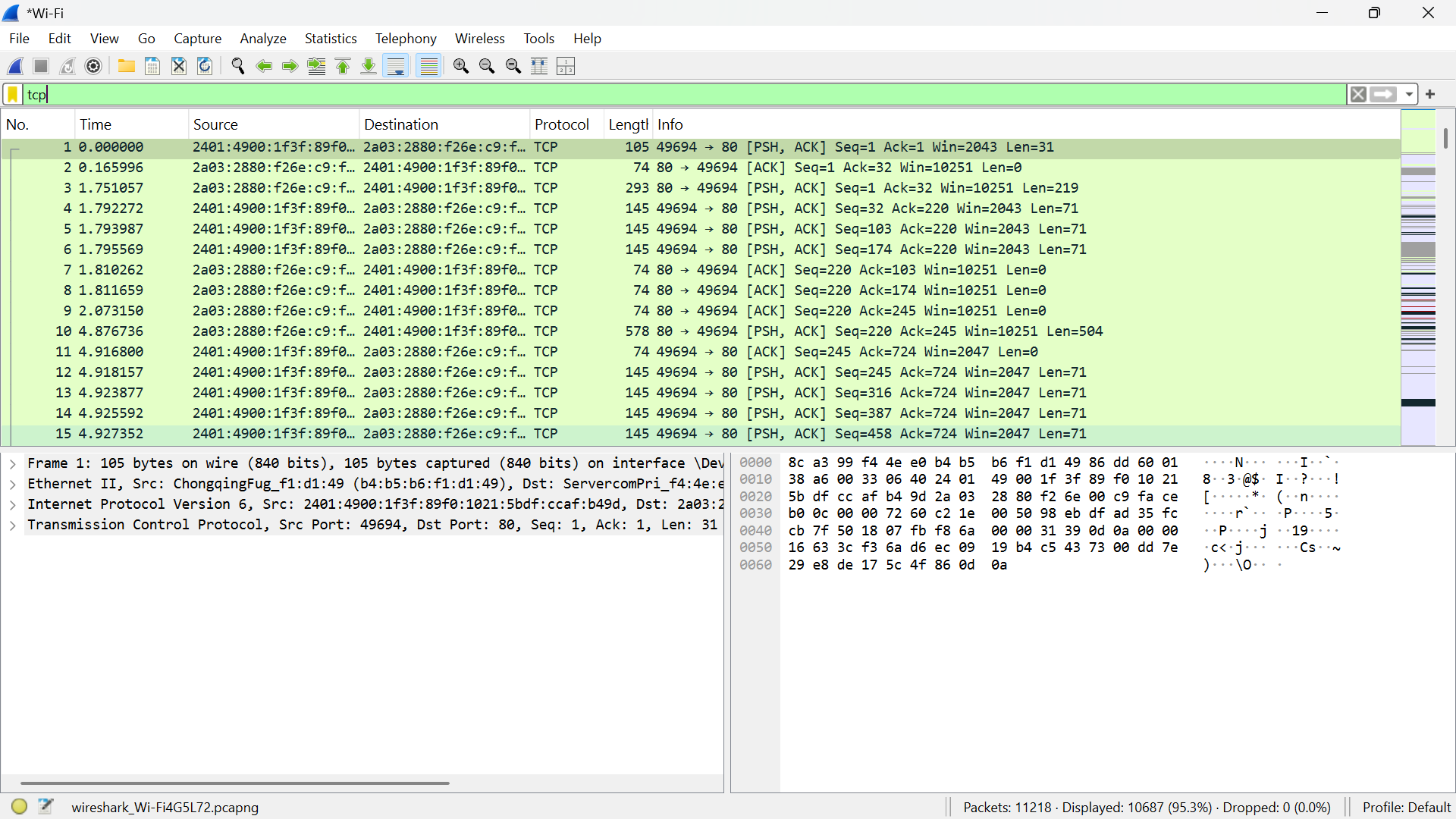
* **Wireshark** – A network protocol analyzer tool
* **Web Browser / Terminal** – To generate network traffic
* **Operating System:** Windows 11

**3. Procedure**

**Step 1: Launch Wireshark**

* Open the Wireshark application.
* Select the appropriate network interface ( Wi-Fi).
* Click the **Start Capture** button (blue shark fin icon) to begin monitoring packets.

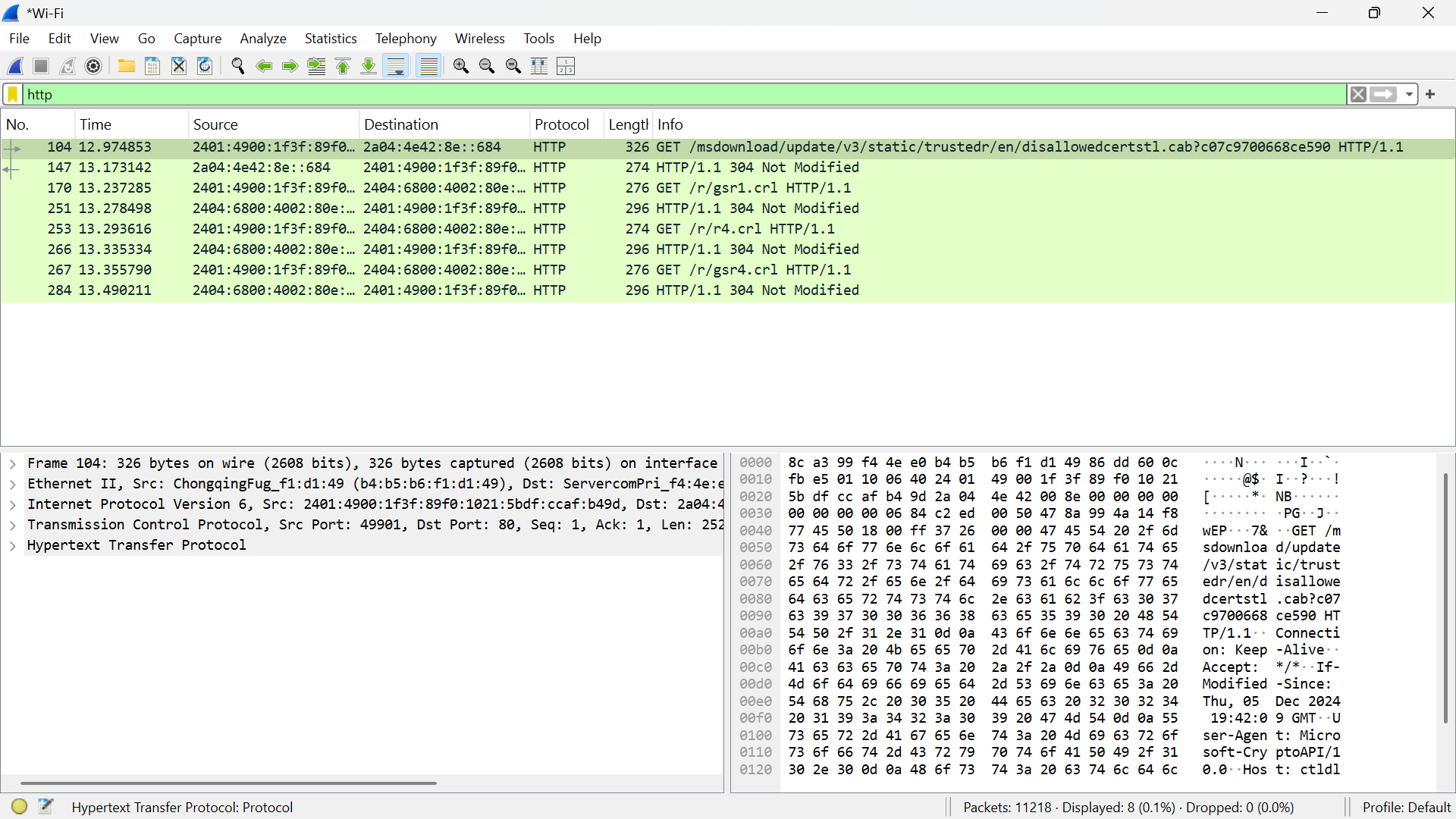
**Step 2: Capture TCP Traffic**

* Generate TCP traffic by opening a website or application that uses TCP.
* In Wireshark, apply the filter: tcp
* You will see packets with the TCP protocol listed.
* **Screenshot:**  
  
* **Saved File:** [tcp.pcapng](https://1drv.ms/u/c/b5f0684d8a46f95d/EQhXByws4BtDgdig-Tzu-lYBONf0myLoqTV8bm8fp_LVkg?e=T2WhOY)

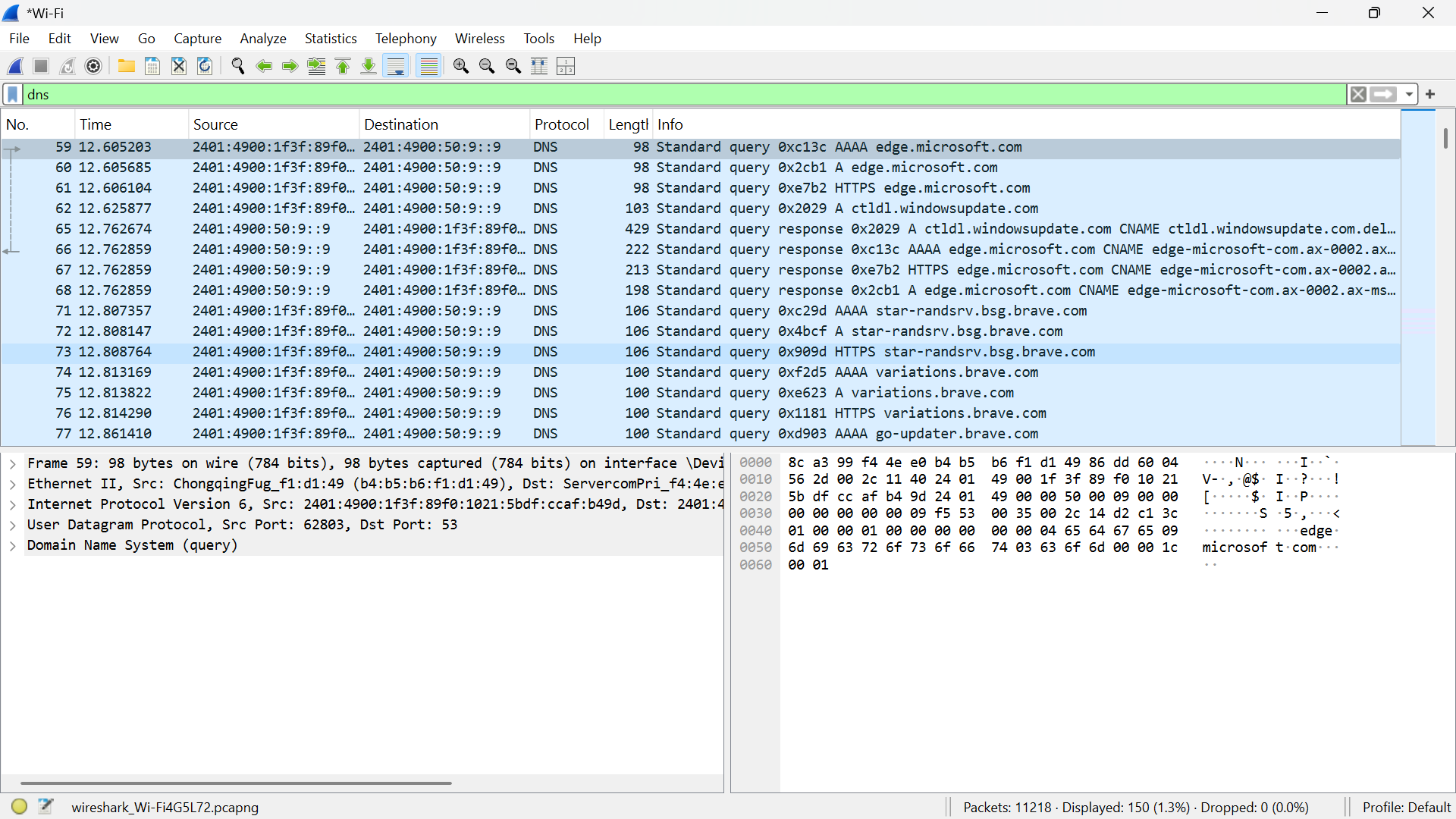
**Step 3: Capture HTTP Traffic**

* Open a browser and visit a website using **HTTP** (not HTTPS), such as:

http://example.com

* In Wireshark, apply the filter:http
* You should see HTTP GET/POST requests.
* **Screenshot:**  
  
* **Saved File:** [http port.pcapng](https://1drv.ms/u/c/b5f0684d8a46f95d/EcCPP-fXqYBAnSFq_9pImVwBXJbCz5w8YCCofahG53wtEQ?e=joCQ1U)

**Step 4: Capture DNS Traffic**

* Open the Command Prompt or Terminal.
* Run a DNS lookup command to generate traffic:
* In Wireshark, apply the filter:
* You will observe DNS queries and responses.
* **Screenshot:**  
  
* **Saved File:** [dns.pcapng](https://1drv.ms/u/c/b5f0684d8a46f95d/EWLDTDdhzJtAqsRKvKMFwXUBa09NfaNIaFnzfQAsJQB7Xw?e=LzzREM)

**4. Observations**

* TCP packets show the handshake and data transfer between endpoints.
* HTTP requests reveal the web content being requested and status codes in return.
* DNS traffic contains queries made to resolve domain names to IP addresses.

**5. Conclusion**

This lab exercise demonstrated the process of capturing and analyzing protocol-specific traffic using Wireshark. By applying protocol filters, we were able to observe and understand how TCP, HTTP, and DNS behave over a network.