Assignment - 2

of

Cyber Security Laboratory

(CSE612)

Bachelor of Technology (CSE)

By

Ramoliya Kaushal (22000409)

Third Year, Semester 6

Course In-charge: Prof. Ninad Bhavsar

Department of Computer Science and Engineering

School Engineering and Technology

Navrachana University, Vadodara

Spring Semester

(2025)

Q1. Install Wireshark in Windows 10/11.

Step-by-Step Installation of Wireshark on Windows 11:

Step 1: Download Wireshark

- Open your web browser (like Chrome or Edge).
- Go to the official Wireshark website: https://www.wireshark.org/download.html
- Click on “Windows Installer (64-bit)” or “Windows Installer (32-bit)” (most likely you need 64-bit).

Step 2: Run the Installer

- Once downloaded, open the installer (Wireshark-win64-x.x.x.exe).
- If prompted by Windows, click Yes to allow changes.

Step 3: Start the Installation

- Click Next on the welcome screen.
- Accept the License Agreement → click Next.

Step 4: Select Components

- You can keep the default selections.
- Ensure “TShark” and “NPcap” are selected (these are needed for packet capturing).
- Click Next.

Step 5: Install NPcap

- The installer will now install NPcap, a packet capturing driver.
- Click Next → Accept the license → Install NPcap with default settings.

Step 6: Choose Installation Location

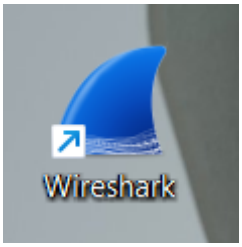
- Choose where to install Wireshark (default location is fine).
- Click Next → then click Install.

Step 7: Complete Installation

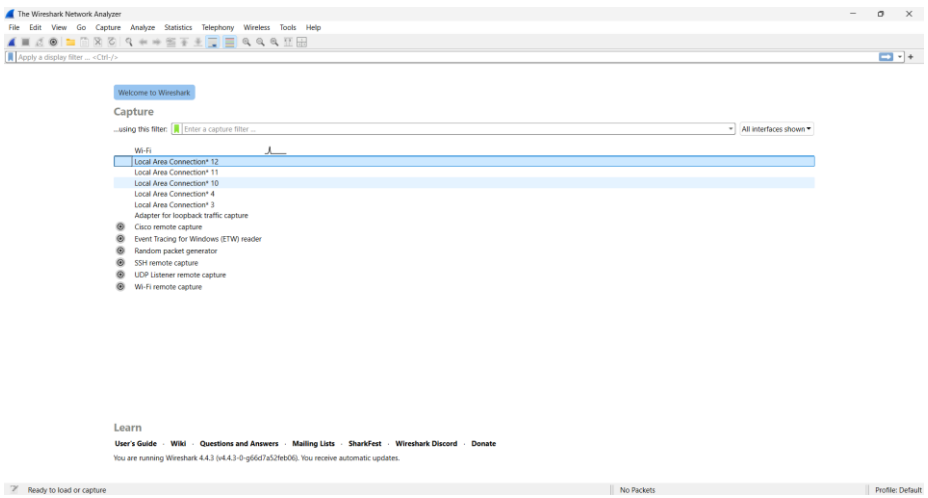
- Wait for the installation to complete.
- Click Finish when done.

How to Verify Wireshark is Installed

- Go to Start Menu → Search “Wireshark”.
- Click on the Wireshark app.



1. **The Wireshark interface will open, showing a list of network interfaces (Wi-Fi, Ethernet, etc.).**



Conclusion:

Wireshark successfully installed on Windows 11. It will now allow you to capture and analyse live network packets in real time.

Q2. Demonstrate how live traffic can be captured (either through wired or wireless interface, as per your device/laptop configuration)

Step-by-Step: Capture Live Network Traffic

Step 1: Open Wireshark

- Click Start Menu → Type Wireshark → Open it.

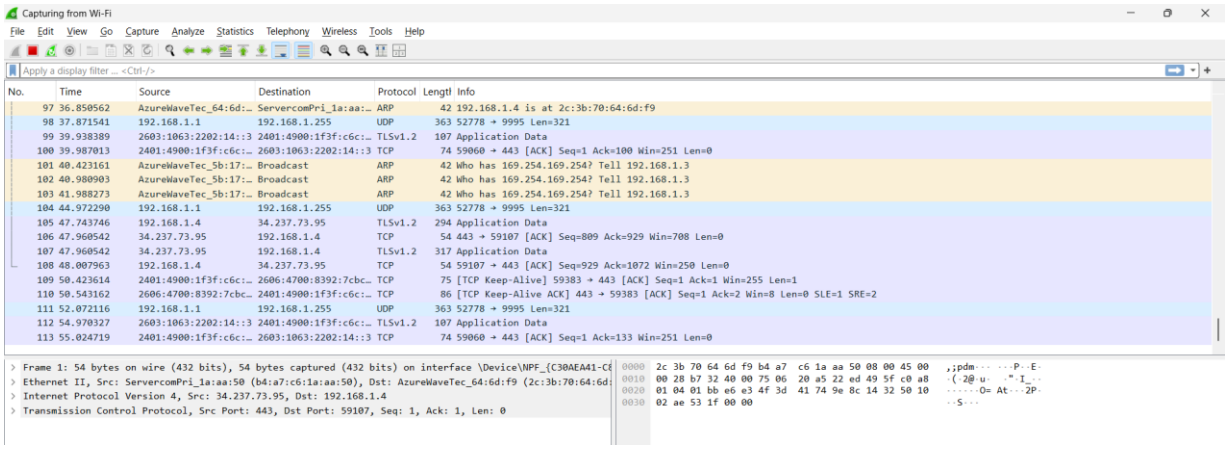
Step 2: Select Network Interface

- When Wireshark opens, you'll see a list of interfaces like:
  - Wi-Fi – if you're connected wirelessly
  - Ethernet – if you're connected via cable (LAN)

Choose the one that is active (has traffic shown as moving bars or numbers).

Step 3: Start Capturing

- Click on the interface name (e.g., Wi-Fi) to start capturing.
- You'll see packets appearing live with details like:
  - Time
  - Source
  - Destination
  - Protocol
  - Info

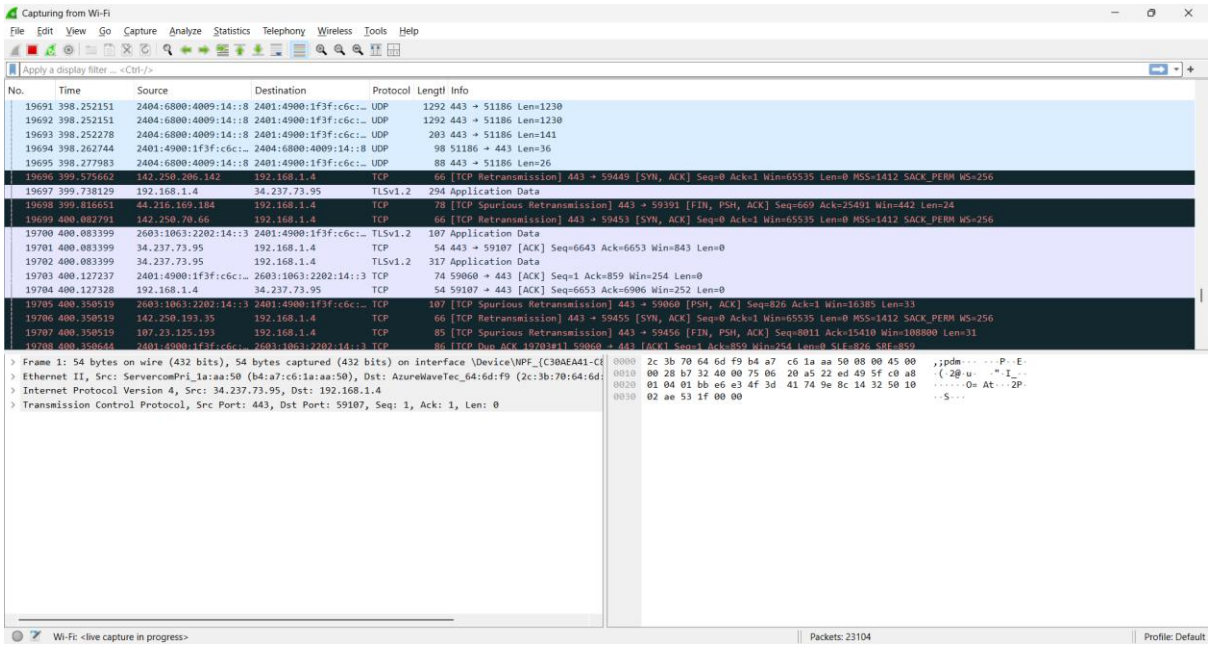


Step 4: Generate Some Traffic

To see some activity, you can:

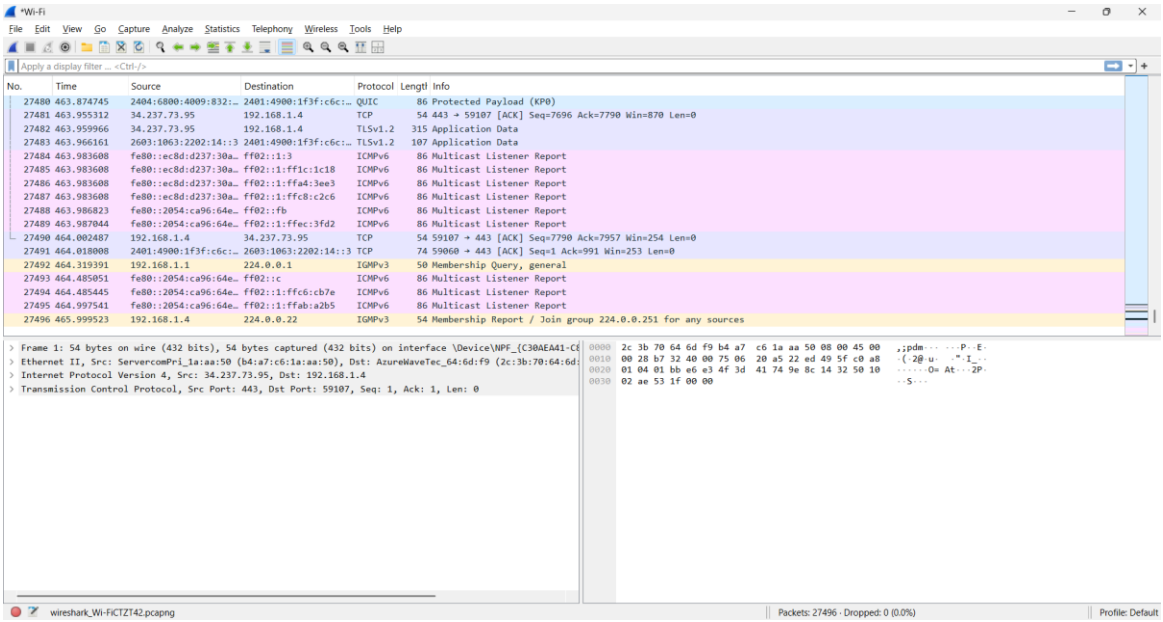
- Open a browser and visit any website (e.g., www.google.com)
- Stream a video or refresh a page

Wireshark will start capturing packets related to those activities.



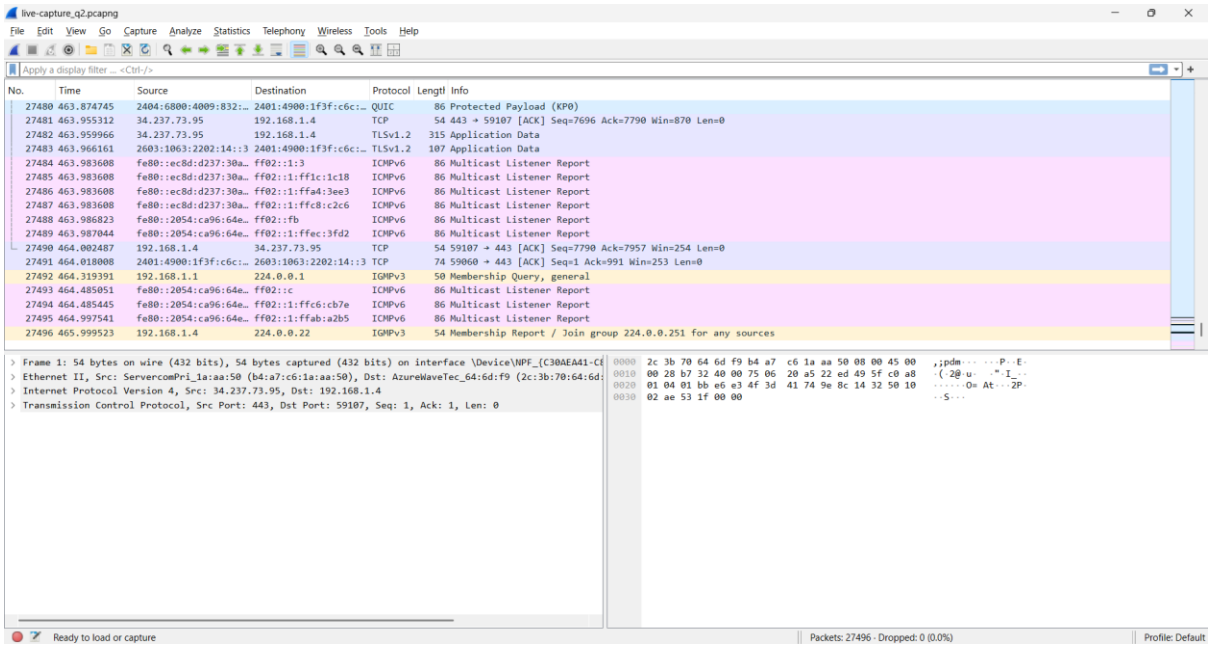
Step 5: Stop the Capture

- Click the Red Square "Stop" button at the top-left to end the capture.



Step 6: Save the Capture

- Go to File > Save As
- Choose a location and give your file a name like live-capture_q2.pcapng
- Click Save



Conclusion:

Live network traffic was successfully captured using Wireshark on a wireless (or wired) interface. Real-time packets related to browsing and system processes were displayed and recorded.

Q3. Demonstrate how live ICMP traffic can be captured. (Hint: Use Ping)

Step 1: Open Wireshark

- Start Wireshark
- Select your active network interface (usually Wi-Fi or Ethernet)
- Click "Start Capturing"

Step 2: Start a Ping in Command Prompt

- Open Command Prompt
- Type the following command:

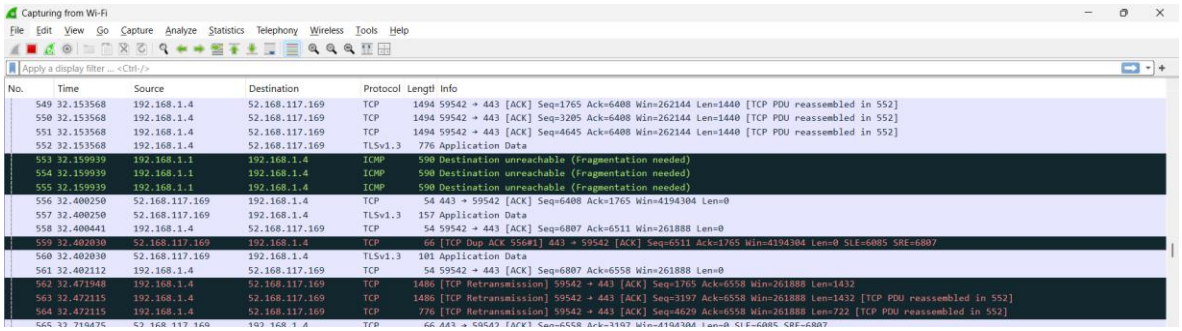
  ping www.google.com

```
C:\Users\kaush>ping www.google.com

Pinging www.google.com [2404:6800:4002:81f::2004] with 32 bytes of data:
Reply from 2404:6800:4002:81f::2004: time=96ms
Reply from 2404:6800:4002:81f::2004: time=96ms
Reply from 2404:6800:4002:81f::2004: time=94ms
Reply from 2404:6800:4002:81f::2004: time=95ms

Ping statistics for 2404:6800:4002:81f::2004:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 94ms, Maximum = 96ms, Average = 95ms

C:\Users\kaush>
```

  (or ping any IP address like ping 8.8.8.8)

Step 3: Watch Wireshark

- In Wireshark, packets will start appearing in real time
- You'll see packets with Protocol = ICMP



What You Are Seeing:

- Echo Request = Your ping going out
- Echo Reply = Google's server responding

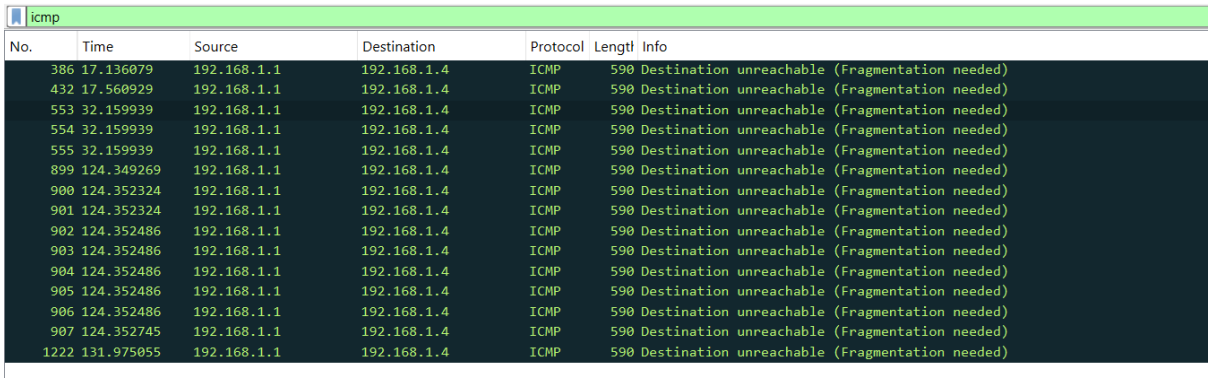
Conclusion:

You have successfully captured live ICMP traffic using Wireshark. In your case, the ICMP messages are showing "Destination Unreachable (Fragmentation needed)", which is a type of ICMP error message. This proves ICMP is actively used by devices on your network to communicate issues, not just for ping.

Q4. Demonstrate how to view only ICMP traffic, assuming all traffic types are being captured.

Use a Filter in Wireshark

- In the top filter bar, type: icmp
- This shows only ICMP packets (makes it cleaner to view).



icmp

No.	Time	Source	Destination	Protocol	Length	Info
386	17.136079	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
432	17.560929	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
553	32.159939	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
554	32.159939	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
555	32.159939	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
899	124.349269	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
900	124.352324	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
901	124.352324	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
902	124.352486	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
903	124.352486	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
904	124.352486	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
905	124.352486	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
906	124.352486	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
907	124.352745	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)
1222	131.975055	192.168.1.1	192.168.1.4	ICMP	590	Destination unreachable (Fragmentation needed)

In Screenshot Shows:

Field	Explanation
Protocol	ICMP – You're capturing Internet Control Message Protocol traffic, just like we wanted.
Source	192.168.1.1 – This is your router or another device sending ICMP messages.
Destination	192.168.1.4 – This is your PC receiving those messages.
Info	Destination Unreachable (Fragmentation needed)

Conclusion:

To view only ICMP traffic in Wireshark, use the filter icmp in the top filter bar. This removes all non-ICMP packets and shows only ICMP communication like ping requests, replies, and ICMP error messages.

Q5. Demonstrate how to capture only ICMP traffic

This question is different from Q4. In Q4, we captured all traffic and used a filter to view only ICMP. But in Q5, we are setting up Wireshark to capture only ICMP packets right from the start, so we avoid capturing other unnecessary data like TCP, UDP, etc.

Capture Only ICMP Traffic

Step 1: Open Wireshark

Launch Wireshark on your system.

Step 2: Select Your Interface

Choose the correct interface for capturing (Wi-Fi, Ethernet, etc.).

Step 3: Set a Capture Filter

This is different from the display filter!

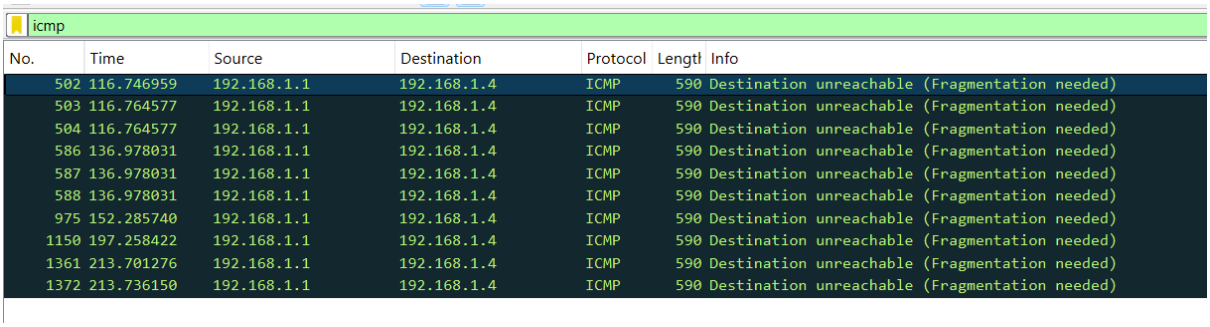
- Click in the field labeled **"Capture Filter"** (next to the interface you selected).

- Type the following: Icmp

Step 4: Start Capturing

Click on Start to begin capturing packets.

Now, Wireshark will capture only ICMP traffic (e.g., ping requests and replies).

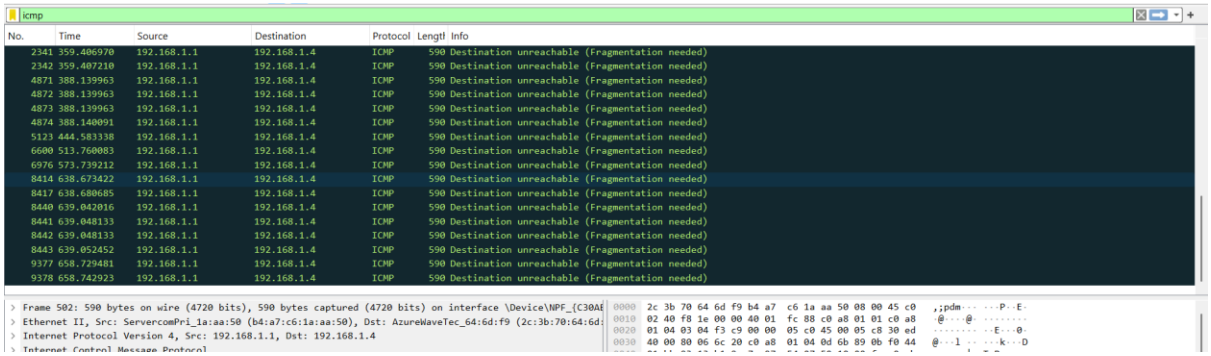


Bonus Tip:

To generate some ICMP traffic during capture, open a terminal or command prompt and type:

ping google.com

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

Pinging google.com [2404:6800:4002:81b::200e] with 32 bytes of data:
Reply from 2404:6800:4002:81b::200e: time=48ms
Reply from 2404:6800:4002:81b::200e: time=44ms
Reply from 2404:6800:4002:81b::200e: time=42ms
Reply from 2404:6800:4002:81b::200e: time=43ms

Ping statistics for 2404:6800:4002:81b::200e:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 42ms, Maximum = 48ms, Average = 44ms

C:\Users\kaush>
```



This will send ICMP Echo Requests and generate ICMP replies—great for testing.

Conclusion:

To capture only ICMP packets in Wireshark, set the capture filter to icmp before starting the capture. This helps you collect only the traffic you need (like pings), reducing noise from other protocols.

Q6. Identify the source and destination IP address, and the ICMP Message Code number of the request and reply messages.

Step 1: Open Wireshark

Make sure your ICMP traffic is already captured (you can generate it using ping).

Step 2: Use the Display Filter

To view only ICMP packets: icmp

In the ICMP traffic captured:

- Source IP: 192.168.1.1
- Destination IP: 192.168.1.12
- ICMP Type: 3 (Destination Unreachable)
- ICMP Code: 4 (Fragmentation Needed and DF set)

This message indicates that the packet could not be delivered because it requires fragmentation, but the “Don’t Fragment” flag is enabled.

Conclusion:

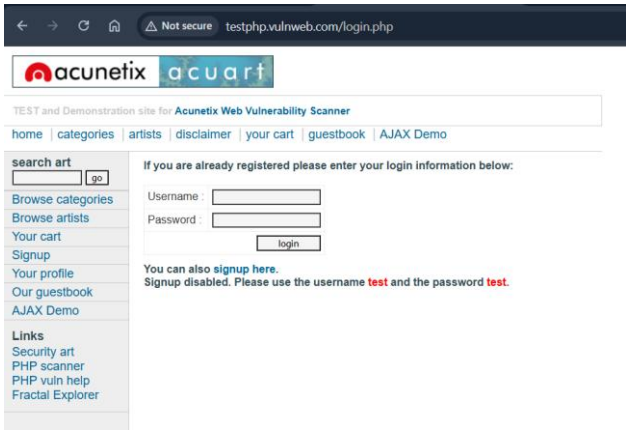
In this activity, ICMP packets were successfully captured and analyzed using Wireshark. The source IP was 192.168.1.1 and the destination IP was 192.168.1.12. The ICMP message type was 3 (Destination Unreachable), and the code was 4 (Fragmentation Needed).
This confirms that the device attempted to send a large packet that could not be delivered because fragmentation was required, but the “Don't Fragment” flag was set.

Q7. Demonstrate how passwords can be cracked for http /non https sites.

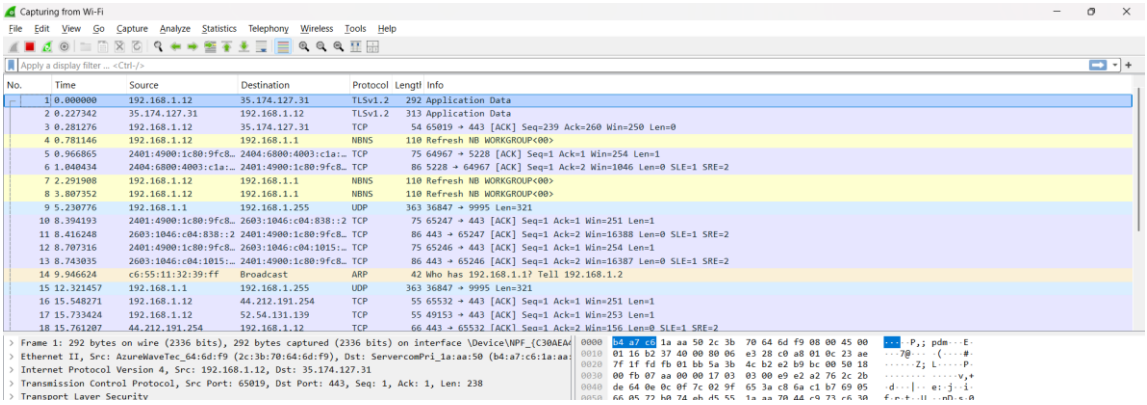
1. Set up a test HTTP login page.

You can use a local server with a simple HTML form, or use known test sites like:

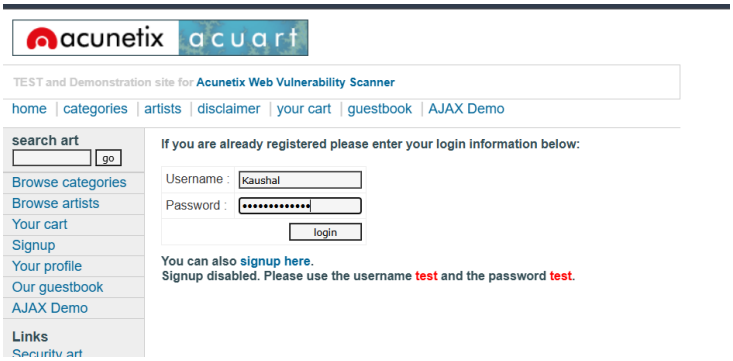
http://testphp.vulnweb.com/login.php



2. Open Wireshark and start capturing on the active network interface (Wi-Fi or Ethernet).



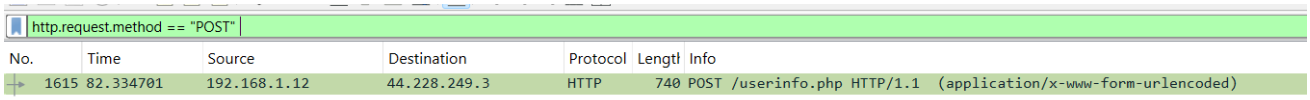
3. In your browser, visit the test HTTP site and submit the login form with any username and password (e.g., username: Kaushal, password: Kaushal@98765).



4. Stop capturing in Wireshark after submission.

5. Use the following filter in Wireshark to narrow down the results:

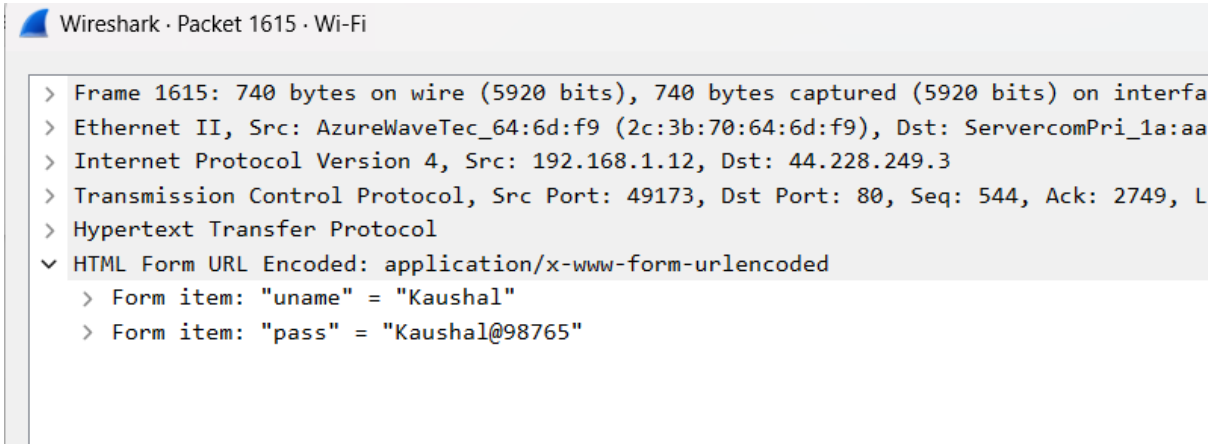
http.request.method == "POST"



6. Inspect the packets:

- Click on the HTTP POST packet in the capture list.
- In the packet details section, look under:

Hypertext Transfer Protocol → Form item



This shows that the password is clearly visible in plain text.

Conclusion:

This activity demonstrates that when login credentials are submitted through an HTTP (non-secure) website, they can be captured and viewed in plain text using tools like Wireshark. This proves that HTTP is not secure and sensitive data like usernames and passwords should always be transmitted using HTTPS, which encrypts the data in transit.