Detailed Report: Capture Network Traffic with Wireshark

1. Objective
The objective of this task is to capture and analyze network traffic using Wireshark. The focus is on filtering HTTP traffic and studying request/response packets.

2. Tools Used
- Operating System: Ubuntu Linux (or equivalent)
- Tool: Wireshark
- Protocols Analyzed: HTTP (requests and responses)

3. Steps Performed

Step	Action	Description
1	Install Wireshark	Installed using `sudo apt install wireshark -y`.
2	Start Capture	Opened Wireshark, selected active network interface, began capturing packets.
3	Generate Traffic	Visited websites to generate HTTP/HTTPS requests.
4	Apply Filter	Used filter `http` to isolate HTTP traffic.
5	Analyze Packets	Inspected HTTP GET/POST requests and server responses.

4. Findings

Packet Type	Details
HTTP Request	Example: GET /index.html HTTP/1.1, Host: example.com, User-Agent: Mozilla/5.0
HTTP Response	Example: HTTP/1.1 200 OK, Content-Type: text/html
Headers	Observed Host headers, content type, and server info.

5. Significance of Capture
- Demonstrates how HTTP requests and responses flow between client and server.
- Provides insights into headers, methods, and server responses.
- Useful for troubleshooting, monitoring, and learning protocol behavior.

6. Conclusion
Wireshark successfully captured live network traffic. Filtering HTTP packets highlighted how clients request resources and servers respond. This enhances understanding of web communication and supports troubleshooting and security analysis.

7. Screenshot Evidence

<< Insert Screenshot from Wireshark video here >>

8. Recommendations
- Use HTTPS whenever possible to protect sensitive data in transit.
- Regularly monitor network traffic for anomalies or suspicious packets.
- Apply filters effectively to focus on specific protocols.

- Save captures in `.pcap` format for future reference and analysis.