Task 5: Capture and Analyze Network Traffic Using Wireshark — Detailed Step-by-Step

Overview & Objective

This document provides a detailed, step-by-step guide to complete Task 5: capturing live network packets with Wireshark, identifying at least three protocols, exporting a .pcap capture file, and producing a short analysis report. Follow each numbered step and use the example filters and commands.

Tools & Deliverables

Tools: Wireshark (GUI), optional CLI tools: tshark, tcpdump, ping, curl, nslookup/dig.
Deliverables: 1) A .pcap capture file. 2) A written report summarizing protocols identified and packet-level evidence.

Prerequisites / Permissions

- Administrator / root privileges may be required to capture on some interfaces.
- You must have permission to capture on the network you're monitoring — do not capture traffic on networks you don't own or have explicit authorization for.
- Install Wireshark (stable release) from wireshark.org. Optionally install tshark (bundled with Wireshark) for command-line captures.

Installation (short)

Windows: Download installer → Run as Administrator → Allow WinPcap/Npcap installation for packet capture.
macOS: Use the official dmg or Homebrew: brew install --cask wireshark and allow permissions.
Linux (Ubuntu/Debian): sudo apt update && sudo apt install wireshark. Add your user to the 'wireshark' group or run as root.

Step-by-step Capture Process (Wireshark GUI)

1. Open Wireshark.
2. Select the active network interface (e.g., Wi■Fi, Ethernet) from the list. The interface with increasing packet counters is typically the active one.
3. Optional: Click Capture → Options to open capture options.
4. In Capture Options: - Check 'Promiscuous Mode' if you wish to capture all traffic seen by the NIC (use only if appropriate). - (Optional) Set a capture filter (BPF) to limit what Wireshark records (e.g., 'port 53' to capture DNS only).
5. Click the blue shark-fin 'Start' button (or 'Start' in the capture dialog) to begin the capture.
6. Generate traffic: open a browser and visit a few websites, ping a public server (ping 8.8.8.8), perform a DNS lookup (nslookup example.com), or run curl http://example.com. Doing multiple actions helps capture different protocol types.
7. After about 30–90 seconds (or enough events), click the red square 'Stop' button to end the capture.
8. Save the capture: File → Save As... choose **pcap** or **pcapng** format. Use a descriptive name, e.g., task5_capture_2025-09-29.pcap.

Capture filters vs Display filters — quick clarification

Capture filters use BPF (Berkeley Packet Filter) and are applied before saving to disk — they reduce what gets recorded. Examples:

Use case	BPF capture filter (example)
Capture filter — only DNS	port 53
Capture filter — only traffic to/from a host	host 192.168.1.100
Capture filter — only HTTP and HTTPS	tcp port 80 or tcp port 443

Display filters apply after the capture and let you focus on specific protocols/fields. Examples:

Use case	Display filter (Wireshark syntax)

Show only HTTP traffic	http
Show only DNS queries/responses	dns
Show only TCP handshake packets (SYN, SYN-ACK, ACK)	tcp.flags.syn == 1 || tcp.flags.ack == 1
Show ICMP echo requests	icmp.type == 8
Show TLS Client Hello (to view SNI)	tls.handshake.type == 1

Detailed Analysis — what to look for (step-by-step)

- **Open the capture file:** File → Open and select your saved .pcap or .pcapng file.
- **Get a quick protocol summary:** Statistics → Protocol Hierarchy. This shows all protocols seen and their percentages — use it to identify at least three different protocols (e.g., HTTP, DNS, TCP, ICMP, TLS).
- **View top endpoints and conversations:** Statistics → Endpoints and Statistics → Conversations. These show which hosts exchanged the most bytes/packets and the pairwise conversations.
- **Filter for a protocol:** In the top display filter bar enter a filter such as 'http' or 'dns' and press Enter. The packet list updates to show matching frames.
- **Inspect packet details:** Click a packet in the packet list. In the middle pane, expand protocol layers (Ethernet → IP → TCP/UDP → Application layer) to read fields like source/destination ports, hostnames, request methods, DNS query names, etc.
- **Follow a conversation:** Right-click a TCP or HTTP packet → Follow → TCP Stream (or Follow → HTTP Stream). This combines related packets and shows the conversation in order (useful for reconstructing web requests/responses).
- **Check for anomalies:** Use 'Analyze → Expert Information' to find warnings, retransmissions, checksum errors, or malformed packets.
- **Use statistics graphs:** Statistics → I/O Graphs to visualize packet rate/byte rate over time; helpful to show bursts or unusual behavior.

Example packet evidence (sample entries you should capture and include in the report)

- **Packet #125 — TCP SYN:** Description: Start of TCP handshake. Fields to note: src IP, dst IP, src port, dst port, tcp.flags.syn == 1, tcp.flags.ack == 0. Display filter: tcp.flags.syn == 1 && tcp.flags.ack == 0.
- **Packet #126 — TCP SYN-ACK:** Description: Response from server. Fields: tcp.flags.syn == 1 && tcp.flags.ack == 1. This confirms a server responded to connection request.
- **Packet #127 — TCP ACK:** Description: Client completes 3-way handshake with ack. Session established.
- **Packet #200 — HTTP GET:** Description: HTTP request to example.com for path '/'. Fields: http.request.method == 'GET', http.host == 'example.com'. Use Follow → HTTP Stream to view full request/response.
- **Packet #205 — DNS Query:** Description: DNS Question: 'example.com' type A. Filter: dns.qry.name == "example.com". Check response IP addresses in the DNS Answer section.
- **Packet #210 — ICMP Echo Request/Reply:** Description: Ping; shows round-trip. Filters: icmp.type == 8 (request), icmp.type == 0 (reply). Note times to compute RTT.

Exporting captures and artifacts

- To save the full capture: File → Save As... choose pcap or pcapng.
- To save only the currently displayed packets (after applying a display filter): File → Export Specified Packets... → Choose 'Displayed' and save.
- To extract transferred files (HTTP): File → Export Objects → HTTP (or SMB/SMB2, FTP depending on traffic) and save files of interest.
- To export packet bytes: Right-click packet → Export Packet Bytes...
- To create a smaller example pcap for submission: apply a display filter to show the relevant conversation(s) and Export Specified Packets → Displayed.

Sample Report Template (copy this into your deliverable report)

Capture metadata:

Date/Time: Interface: Capture duration: Capture filter used: Display filters used during analysis:

Summary of protocols (table):

List protocols seen and short notes (e.g., HTTP — web browsing — GET requests observed).

Detailed packet evidence:

List at least 3 packets with packet number, timestamp, source/destination IPs and ports, protocol, and short description of why it matters.

Findings & conclusion:

Summarize what the capture showed, any anomalies, retransmissions, or suspicious behavior, and final conclusions.

Useful Display Filters & CLI equivalents

Goal	Wireshark Display Filter	tshark / tcpdump example
Show HTTP traffic	http	tshark -r capture.pcap -Y http
Show DNS traffic	dns	tshark -r capture.pcap -Y dns
Show ICMP (ping)	icmp	tcpdump -r capture.pcap icmp
Show TCP retransmissions	tcp.analysis.retransmission	tshark -r capture.pcap -Y "tcp.analysis.retran

Legal & Ethical Notice

Only capture traffic on networks you own or where you have explicit permission. Capturing network traffic can expose sensitive information — handle pcap files securely and redact or avoid including private data in shared deliverables.

Appendix — Example Completed Analysis (example text you can adapt)

Capture metadata: - Date/Time: 2025-09-29 13:40 IST - Interface: Wi■Fi (wlan0) - Duration: 1 minute 12 seconds - Capture filter: none (full capture saved to task5_capture.pcapng) Summary of protocols observed: - DNS: Multiple queries for visited domains (example.com). Observed authoritative answers with IPv4 addresses. - TCP: Numerous TCP connections; handshakes and data transfer observed (port 80 and port 443). - HTTP: Observed HTTP GET requests and responses (when visiting non-HTTPS pages). - ICMP: Echo Request/Reply when pinging 8.8.8.8. Selected packet evidence (packet numbers are illustrative): - Packet #125 – TCP SYN – src: 192.168.1.10:50234 → dst: 93.184.216.34:80 – tcp.flags.syn==1 – start of 3-way handshake. - Packet #126 – TCP SYN-ACK – src: 93.184.216.34:80 → dst: 192.168.1.10:50234 – tcp.flags.syn==1 && tcp.flags.ack==1. - Packet #200 – HTTP GET – Request for http://example.com/ – http.request.method == GET – Host: example.com. - Packet #205 – DNS Query – dns.qry.name == "example.com" – Response contains IPv4 93.184.216.34. - Packet #210 – ICMP Echo Request/Reply – Shows RTT of ~24 ms. Findings: - Normal web browsing behavior observed. No obvious malicious traffic in the short capture window. - No packet-level anomalies other than expected retransmissions due to normal network conditions. Conclusion: The capture shows typical DNS resolution followed by TCP connection establishment and HTTP requests. See saved pcap file for raw frames.

Deliverables you should submit:

- task5_capture_YYYYMMDD.pcap (or .pcapng) — the raw capture file (or a filtered subset containing the relevant conversations).
- task5_report_YYYYMMDD.pdf — a written report using the template provided above and including packet evidence.