EXP NO: 14	**Capture, save, and analyse traffic (TCP/UDP/IP/HTTP/ARP/DHCP/ICMP/DNS) using Wireshark**

Aim

Capture live network traffic, apply protocol-specific filters, save PCAPs, and analyze sessions (HTTP requests, DNS queries, DHCP leases, ICMP pings) to investigate events and network behavior.

Procedure / Algorithm

1. Start capture on the correct interface in Wireshark (select physical or virtual interface).
2. Use **capture filter** (optional) to limit volume (e.g., `port 53` or `host 10.10.10.5`).
3. Use **display filters** for focused analysis:
   - `http` — HTTP traffic
   - `dns` — DNS queries/responses
   - `bootp || dhcp` — DHCP messages
   - `arp` — ARP traffic
   - `icmp` — ICMP messages
   - `tcp.port==80` — traffic on port 80
4. Use **Follow → TCP Stream** to view a full session (HTTP request/response).
5. Use **Statistics → Protocol Hierarchy / Conversations / Endpoints** to summarize traffic.
6. Export objects: `File → Export Objects → HTTP` to recover downloaded files.
7. Save capture: `File → Save As` → `analysis_capture.pcap`.

Sample steps & expected findings

- Follow TCP Stream of an HTTP GET to see the full request and response (useful to detect exfiltration or malicious downloads).
- In Statistics → Conversations, identify top talkers (most bytes sent) — these may be exfiltration suspects.
- Use Export Objects → HTTP to extract files for malware inspection.

Results

Wireshark provides the definitive packet-level view; pair with pcapxray for visual summaries.

- When sharing captures, sanitize sensitive IPs or use redact tools.
- Large captures: use editcap to trim or split files before analysis.