Wireshark Network Traffic Analysis Report

Title:

Network Traffic Capture & Analysis Using Wireshark

1. Objective

The objective of this project is to capture real-time network traffic using Wireshark and analyze it using different filters to identify normal and suspicious communication patterns.

2. Tools Used

- Wireshark
- Windows 10 Network Adapter
- Chrome Browser

3. Steps Performed

Step 1 — Packet Capture

- Launched Wireshark and selected the Wi-Fi interface.
- Started live capture and visited websites to generate traffic.
- Stopped capture after sufficient packets were collected.
  (Screenshot 1, 2, 3)

Step 2 — HTTP Traffic Analysis

- Applied filter: http
- Observed HTTP GET/POST packets.
  (Screenshot 4)

Step 3 — DNS Traffic Analysis

- Applied filter: dns
- Observed DNS queries and responses (e.g., google.com, windowsupdate.com).
  (Screenshot 5)

Step 4 — TCP Error Packets

- Applied filter: tcp.flags.reset == 1

- Observed TCP Reset packets indicating failed/blocked connections.
  (Screenshot 6)

Step 5 — Encrypted TLS Traffic

- Applied filter: tls
- Observed HTTPS encrypted packets (Client Hello / Server Hello).
  (Screenshot 7)

Step 6 — Identifying Suspicious Traffic

- Found repeated RST packets and unknown IP communication which may indicate blocked/failed or unusual connections.
  (Screenshot 8)

---

4. Findings

- Continuous DNS activity observed — normal for browsing.
- HTTP and TLS packets confirm both secure and insecure web traffic.
- TCP RST packets indicate unsuccessful connection attempts.
- Some IPv6 unknown addresses were detected — likely CDN or system background services.

---

5. Conclusion

Wireshark helped in analyzing different packet types such as DNS, HTTP, TCP, and TLS.
This project demonstrates basic network traffic analysis and identification of unusual patterns.