Network Sniffing & Forensic Analysis with Wireshark & NetworkMiner

Cybersecurity Lab Portfolio | Sanitized for Public Sharing

Overview

This lab focused on network traffic analysis and forensic techniques using Wireshark and NetworkMiner. I analyzed multiple PCAP files to identify TCP, UDP, HTTP, HTTPS, and DNS traffic patterns, applied filters, and used visualization tools like I/O Graphs and Flow Graphs to uncover anomalies. With NetworkMiner, I reconstructed files, reviewed DNS queries, and investigated host behavior. These exercises strengthened my skills in packet-level analysis, identifying secure and insecure communication protocols and real-world incident investigation.

Tools Used

- Wireshark – Packet capture and analysis
- NetworkMiner- Network forensics and file reconstruction
- PCAP Files – port80.pcap, port443.pcap, ShortPCAP.pcap
- Npcap – required driver for live capture
- Windows Environment – Alienware M17 R5 (analysis machine)

Methods

- Installed Wireshark and NetworkMiner, confirgured Npcap for packet capture.
- Converted .pcapng files to .pcap format for compatibility with NetworkMiner.
- Applied display filters in Wireshark for TCP, UDP, HTTP, HTTPS, DNS, and ICMP traffic.
- Analyzed TCP streams, Flow Graphs, and I/O Graphs to identify traffic patterns and anomalies.
- Used Endpoint Statistics to identify top talkers and performed reverse DNS lookups.
- Leveraged NetworkMiner to review anomalies, examine host behavior, and reconstruct files.

Key Results

1. TCP Stream Analysis: Encrypted vs Unencrypted Traffic
   Using Wireshark, I followed TCP streams across multiple PCAPs to analyze protocol behavior. HTTP streams revealed full plaintext communication (GET requests and responses), while HTTPS streams showed encrypted data, confirming TLS was in use. The first screenshot shows cleartext HTTP requests and responses, including headers like User-Agent and Server. The second screenshot shows encrypted HTTPS traffic over port 443, demonstrating the presence of TLS, where payloads are unreadable.
   Why this matters: Analysts must confirm encryption to prevent sensitive data exposure and detect insecure protocols on the network.

Figure 1 Wireshark TCP Stream showing unencrypted HTTP communication with sensitive details sanitized

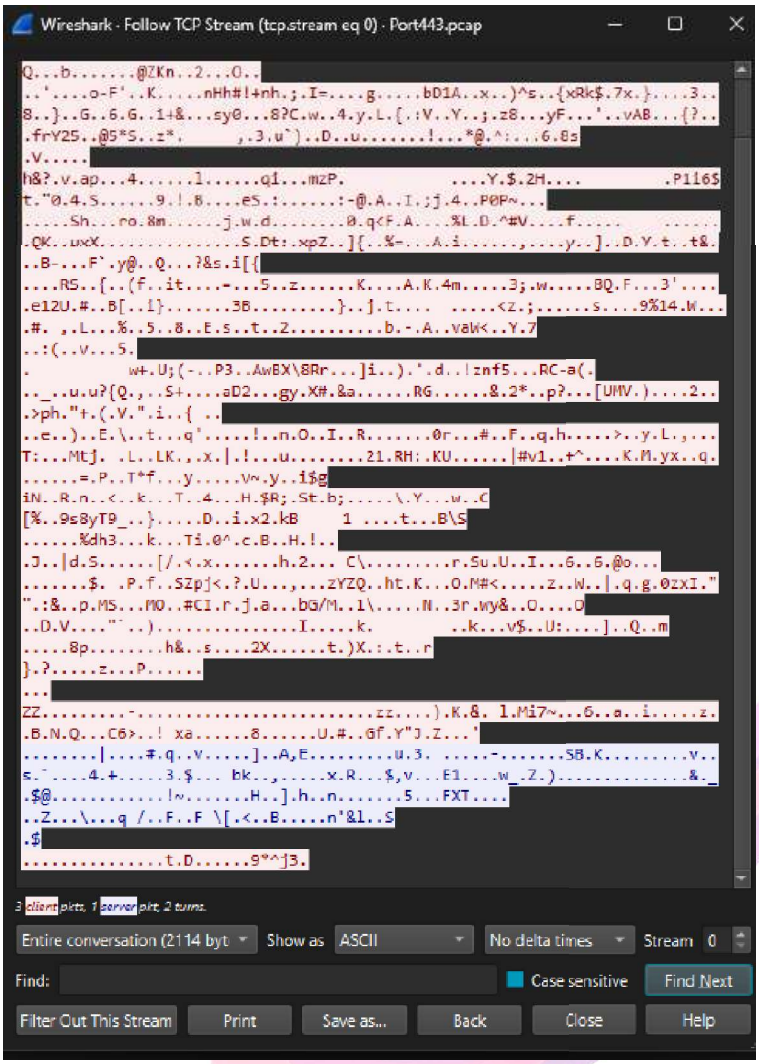

Figure 2 Wireshark TCP Stream showing encrypted HTTPS communication over port 443.

2. Traffic Anomaly Detection with I/O Graphs
   I created I/O graphs to visualize packet activity over time. One ShortPCAP file displayed a noticeable spike, which correlated to QUIC traffic (secure, UDP-based). This helped identify the protocol and activity type efficiently.
   Why this matters: Visualization is critical for spotting unusual patterns, such as DDoS attempts or data exfiltration.

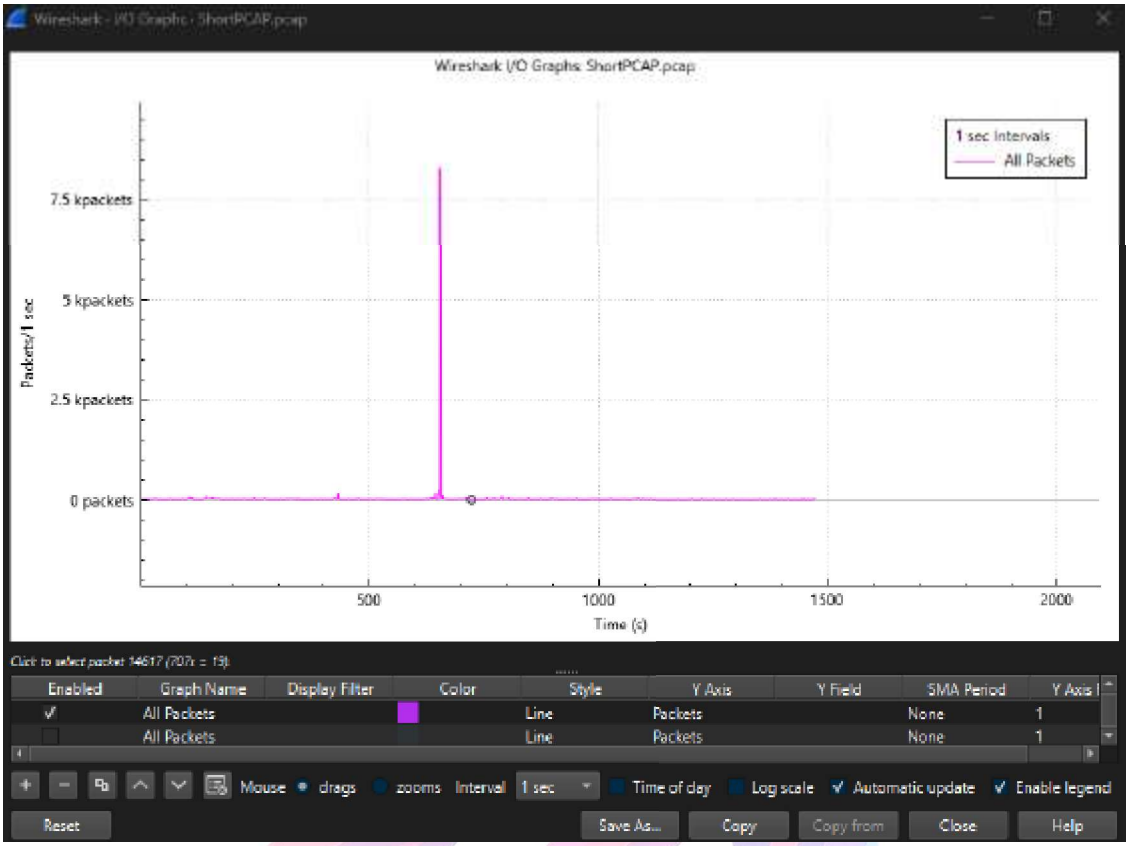

Figure 3 Wireshark I/O Graph displaying packet activity with a significant spike.

3. Host and File Reconstruction in NetworkMiner
   NetworkMiner revealed detailed host metadata and successfully reconstructed a .cer certificate file, proving its forensic capability. This demonstrated how captured packets can expose sensitive files in unencrypted or partially encrypted sessions. Why this matters: File reconstruction is a key step in incident response and forensic investigations.

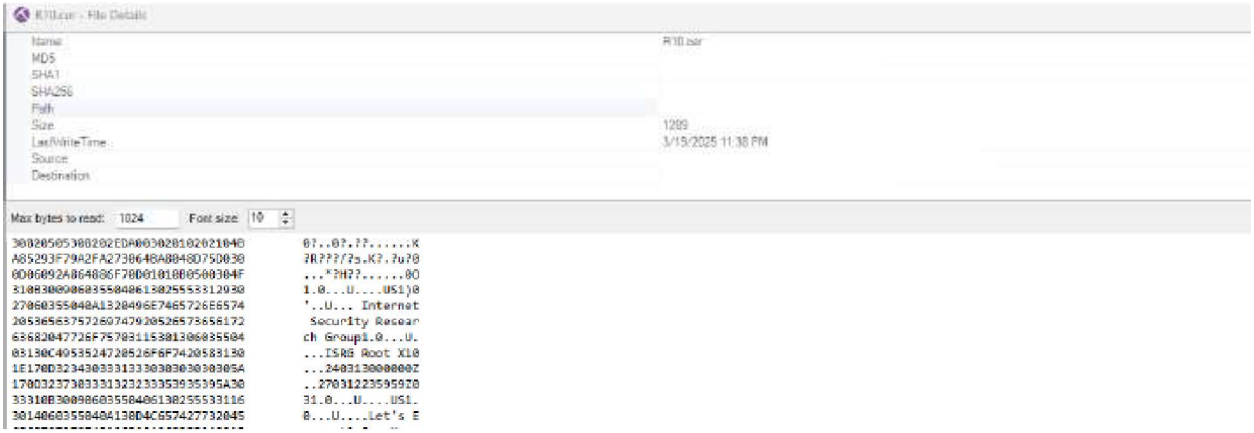

Figure 4 NetworkMiner file reconstruction showing a recovered certificate file from captured traffic (sensitive IPs and file paths redacted).

Discussion

This lab reinforced the importance of packet-level analysis in detecting anomalies and understanding network behavior. Through Wireshark, I explored the differences between plaintext and encrypted protocols, visualized traffic spikes using I/O Graphs, and leveraged NetworkMiner to extract artifacts for forensic review. These skills translate directly to real-world scenarios such as incident response, intrusion detection, and threat hunting, where analysts must identify insecure communication, reconstruct evidence, and interpret traffic patterns under time-sensitive conditions.

Key Takeaways

- Demonstrated proficiency in Wireshark for filtering traffic and analyzing TCP, UDP, HTTP, HTTPS, DNS and ICMP.
- Compared unencrypted vs encrypted traffic, validating TLS protection.
- Used I/O Graphs to detect abnormal packet spikes, a common indicator of anomalies.
- Performed file reconstruction and host analysis in NetworkMiner, simulating real-world forensic tasks.
- Strengthened ability to correlate technical findings with security implications.