Wireshark
Cheat Sheet

Raj Bhatia
https://www.linkedin.com/in/raj-bhatia-1790901a8/

Wireshark

A popular tool for recording and analyzing network traffic in real time is Wireshark, a network protocol analyzer. It is a free, open-source program that works with Linux, macOS, and Windows, among other operating systems. An effective tool for cybersecurity investigation, performance optimization, and network troubleshooting is Wireshark.

Wireshark is a potent network analysis tool that supports multiple protocols, including HTTP, TCP, UDP, and DNS, and records data packets moving over wired or wireless network interfaces in real time. For in-depth protocol study, it decodes packets into a structured, readable format. To concentrate on particular traffic, users can filter and search packets according to IP addresses, protocol types, or port numbers. It makes it possible to effectively monitor network performance and traffic flow with visualization tools including flow graphs, I/O graphs, and comprehensive data. Wireshark is an essential tool for network troubleshooting and security investigation since it is extensible through scripts and plugins and allows offline analysis by exporting packets in common formats like PCAP and PCAPNG.

Real-World Uses:

- IT Support: Resolving client or staff connectivity problems.
- Cybersecurity: Keeping an eye out for weaknesses or efforts at incursion.
- DevOps: Improving service-to-service connectivity in a microservices framework.
- Education: imparting knowledge on cybersecurity and networking concepts.

Cheat Sheet for Analyzing PCAP Files in Wireshark

Basic Steps

Action	Description	Shortcut/Command
Open a PCAP File	Load a .pcap or .pcapng file for analysis.	File > Open or Ctrl + O
Save Filtered Packets	Export filtered packets to a new file.	File > Export Specified Packets
View Detailed Packet Information	Select a packet to see details in the middle and bottom panes.	N/A

Display Filters for Analysis

Filter	Purpose
ip.addr == 192.168.1.1	Display all traffic involving a specific IP address.
tcp.port == 80	Show traffic on a specific TCP port (e.g., HTTP).

udp.port == 53	Show DNS traffic (UDP port 53).
http	Display HTTP traffic.
ssl or tls	Show SSL/TLS encrypted traffic.
tcp.flags.syn == 1	Show TCP SYN packets (connection attempts).
tcp.flags.reset == 1	Display TCP RST packets (connection resets).
dns.flags.response == 1	Show only DNS responses.
frame contains "example.com"	Filter packets containing a specific string.
tcp.analysis.retransmission	Display TCP retransmissions (possible issues).
tcp.analysis.flags	Show TCP packets with issues (e.g., retransmissions, resets).
icmp	Display ICMP traffic (e.g., ping requests).
arp	Show ARP requests and responses.

Tools for Analysis

Tool	**Location**	**Description**
Protocol Hierarchy	Statistics > Protocol Hierarchy	View breakdown of all protocols in the capture.
Conversations	Statistics > Conversations	List all conversations (by IP, MAC, or transport layer).
Endpoints	Statistics > Endpoints	Show endpoints (e.g., IPs, MACs) involved in the traffic.
Flow Graph	Statistics > Flow Graph	Visualize the flow of packets between endpoints.
I/O Graphs	Statistics > I/O Graph	Plot network traffic over time.
Resolved Addresses	Statistics > Resolved Addresses	View hostnames for IPs (if resolved).
Expert Information	Analyze > Expert Information	Highlight errors, warnings, and noteworthy packets.

Follow Stream

Stream Type	Action	Purpose
TCP Stream	Right-click a packet > Follow > TCP Stream	Reconstruct a TCP conversation between endpoints.
UDP Stream	Right-click a packet > Follow > UDP Stream	Reconstruct a UDP conversation.
HTTP Stream	Right-click a packet > Follow > HTTP Stream	View HTTP headers and body content in readable format.

Statistics Filters

Analysis Task	Command/Shortcut	Purpose
Protocol Usage Breakdown	Statistics > Protocol Hierarchy	Identify which protocols dominate the traffic.
Communication Details	Statistics > Conversations	See detailed info about communication between hosts.
Packet Flow	Statistics > Flow Graph	Analyze the sequence of packets and detect anomalies.
Response Times (DNS, HTTP)	Statistics > Service Response Time	Measure response times for DNS and HTTP requests.

Graphing and Visualization

Tool	Usage	Purpose
I/O Graphs	Statistics > I/O Graph	Visualize packet rates, bandwidth usage, or custom filters.
TCP Stream Graphs	Right-click a TCP packet > TCP Stream Graphs	Analyze round-trip time, throughput, and window scaling.

Tips for Specific Analysis Scenarios

Scenario	Filter or Approach
Detect Port Scanning	Filter for packets with many destination ports: ip.src == 192.168.1.1 && tcp.flags.syn == 1

Analyze Slow Connections	Look for retransmissions or high latency: tcp.analysis.retransmission or I/O Graphs.
Identify Suspicious Traffic	Filter for unusual IPs or unknown protocols: !(ip.addr == <trusted network>).
Examine Downloads	Follow the HTTP Stream or use http.file_data to view file contents.
Inspect ARP Spoofing	Look for duplicate ARP responses: arp.duplicate-address-frame.

Wireshark Keyboard Shortcuts

File Operations

- **Open Capture File**: `Ctrl + O`
- **Save Capture File**: `Ctrl + S`
- **Close Wireshark**: `Ctrl + Q`

Capture Control

- **Start/Stop Capture**: `Ctrl + E`
- **Restart Capture**: `Ctrl + R`

Filters

- **Apply Display Filter**: Press `Enter` in the Filter Bar
- **Clear Display Filter**: `Ctrl + Shift + X`
- **Toggle Between Capture and Display Filters**: `Shift + Ctrl + T`

Navigating Packets

- **Move to Next Packet**: ↓
- **Move to Previous Packet**: ↑
- **Go to First Packet**: `Ctrl + Home`
- **Go to Last Packet**: `Ctrl + End`
- **Jump to Packet**: `Ctrl + G`

Expand/Collapse Packet Details

- **Expand Packet Details**: →
- **Collapse Packet Details**: ←

- **Expand All Details**: `Shift + →`
- **Collapse All Details**: `Shift + ←`

View and Layout

- **Show/Hide Packet Details Pane**: `Ctrl + Shift + D`
- **Show/Hide Packet Bytes Pane**: `Ctrl + Shift + B`
- **Zoom In**: `Ctrl + =`
- **Zoom Out**: `Ctrl + -`
- **Reset Zoom**: `Ctrl + 0`

Find and Search

- **Find Packet**: `Ctrl + F`
- **Repeat Find (Next)**: `Ctrl + N`
- **Repeat Find (Previous)**: `Ctrl + Shift + N`