A logo of a graduation cap

Description automatically generatedLab program Number: 14 Date: 2082-05-14

Title: Introduction to Network Traffic Analysis using Wireshark

**Theory**

**Wireshark**

Wireshark is a free and open-source network protocol analyser used for network troubleshooting, analysis, software and communications protocol development, and education. It allows users to capture and interactively browse the traffic running on a computer network, providing detailed information about each packet, including source and destination addresses, protocol headers, and payload data. Wireshark supports a vast range of protocols and has a friendly graphical user interface (GUI) as well as a command-line interface called T-Shark.

**Key Concepts of Wireshark**

**Packets**: Units of data transmitted over a network that Wireshark captures and analyses.

**Protocols:** Rules and standards for network communication that Wireshark decodes to interpret traffic.

**Capture Filters:** Criteria set before capturing to limit which packets are recorded by Wireshark.

**Display Filters:** Used to narrow down captured traffic, allowing focus on specific protocols, addresses, or packet types for easier analysis.

**Packet Details:** Provides in-depth information about each packet, including headers, payload, and protocol fields.

**Real-time Monitoring:** Enables live observation of network activity, helping detect issues, anomalies, or suspicious traffic instantly.

**Interface of Wireshark**

**Main Toolbar**

Provide quick access to essential function like starting/stopping captures, opening and saving files and setting preferences. It streamlines navigation within Wireshark.

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Main Toolbar of Wireshark

**Packet List Pane**

Displays all captured packets in a list format, showing key information like packet number, timestamp, source/destination IP, protocol, and length.

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Packet List Pane of Wireshark

**Packet Details Pane**

Shows a detailed breakdown of the selected packet, displaying information layer-by-layer (e.g., Ethernet, IP, TCP) to help analyse the packet structure.

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Packet Details Pane of Wireshark

**Packet Bytes Pane**

Displays the raw data of the selected packet in hexadecimal and ASCII formats, allowing for a deeper look into the packet's actual content.

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Packet Bytes Pane of Wireshark

**Basic Network Capture and Analysis**

**Selecting a Network Interface**

Step1: Open Wireshark and click the "Capture" menu.

Step 2: Choose the correct network interface from the list (e.g., Ethernet or Wi-Fi).  
Step 3: Look for the interface with active traffic (represented by a graph with fluctuating data).  
Step 4: Choose the required interface by clicking on it.

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Choosing the Required Interface

**Starting Packet Capture**

Step 1: Click the blue "Shark Fin" icon (Start Capture) on the toolbar to begin capturing.  
Step 2: Open a web browser and navigate to example.com to generate traffic.



Fig: Navigate example.com in Web Browser

Step 3: Wireshark will capture the packets sent and received during the visit to the website.  
Step 4: With the use of filter bar packets related to examplw.com can be captured also type ip.addr = <ip address of src>.

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Capturing Packet

**Stopping and Saving Captures**

Step 1: Click the red "Square" icon (Stop Capture) on the toolbar when you have enough data.  
Step 2: Go to "File" and select "Save As" to save the captured data in your preferred location.  
Step 3: Choose the appropriate format (e.g., pcap or. pcapng) for saving the file.

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Saving a File

**Export the Captured Data**

Step 1: After stopping the capture, click on "File" and choose "Export Specified Packets".  
Step 2: Apply any filters if you only want to export specific packets (e.g., by IP address or protocol).

Step 3: Save the exported file in the desired format for further analysis or sharing with others.

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Exporting the Capture Packet

**Conclusion**

In this lab, we successfully demonstrated the use of Wireshark as a powerful tool for capturing and analysing real-time network traffic. By selecting a network interface and testing with example.com, we observed how Wireshark decodes various protocols, visualizes packet data, and aids in identifying potential issues. This hands-on experience highlighted its importance in troubleshooting, performance monitoring, and security auditing, making Wireshark an essential tool for network administrators.