Network Packet Sniffer

Overview

Develop a simplified version of a network traffic analyzer like Wireshark to monitor, capture, and analyze real-time network packets on a LAN. The tool provides insights into protocol usage and basic packet metadata.

Tools Required

Programming Language:

- Python

Libraries:

- `scapy` (preferred) or `socket` (for raw packet capture)
- `matplotlib` (for visualizations)
- `csv` (for storing logs)

Development Steps

1. Capture Packets

- Use `scapy.sniff()` or raw sockets to listen to live network traffic
- Define a packet handler to process each captured packet

2. Parse Packet Details

- Extract relevant metadata from each packet:
    - Source IP address

    - Destination IP address
    - Protocol type (TCP, UDP, ICMP, etc.)
    - Packet size

3. Store and Analyze

- Save parsed data into a CSV file for later inspection
- Count the number of packets per protocol
- Use `matplotlib` to generate charts:
    - Pie chart showing protocol distribution
    - Line or bar graphs for time-based analysis (optional)

Expected Output

- Real-time packet capturing script
- CSV log containing essential packet header information
- Visual summary of protocol usage through plots
- Useful for learning network internals and analyzing LAN activity