Introduction

This project is a real-time Network Packet Sniffer with a graphical user interface (GUI) built using Python, Scapy, and Flet. It allows users to monitor live network traffic, apply protocol/IP-based filters, and export logs to CSV format. The application is designed to be beginner-friendly while offering foundational insights into network-level packet analysis.

Abstract

The purpose of this project is to build a simple yet effective tool that captures packets from the local network interface, displays important metadata (source/destination IP, protocol, size), and provides filtering and export features. It aids learning about TCP/IP layers, network protocols, and provides a hands-on experience with raw socket-based network monitoring in Python.

Tools Used

1. Python 3.10+
2. Scapy - For capturing and parsing network packets
3. Flet - For creating a cross-platform desktop GUI
4. CSV - For exporting logs
5. Batch file (.bat) - For launching with admin rights on Windows

Steps Involved in Building the Project

1. Installed required libraries using pip (Scapy, Flet).
2. Created the packet capture logic using Scapy's sniff() method.
3. Built a modern GUI using Flet with input fields, dropdowns, and buttons.
4. Implemented filters for IP address and protocols (TCP, UDP, ICMP, ARP, HTTP, DNS).
5. Added live logging to the interface and stored packets in a list.
6. Enabled CSV export of packet logs.
7. Designed a batch file to launch the GUI with administrator privileges for raw socket access.
8. Prepared documentation and created GitHub-ready project structure.

Conclusion

This packet sniffer project provides real-time traffic visibility and helps understand fundamental concepts in networking and cybersecurity. It combines practical packet analysis using Scapy with a user-friendly UI in Flet, making it a useful learning tool and a strong addition to a cybersecurity or software development portfolio.