PackAnalyzer
A GUI based Network Traffic Analysis & Security Testing Tool

Student's Name: Asaduzzaman Rifat

Course: SPL 1

Date: January 15, 2026

Project Overview & Goals

PackAnalyzer is a powerful tool built in **C** to capture, read, and understand network traffic as it happens.

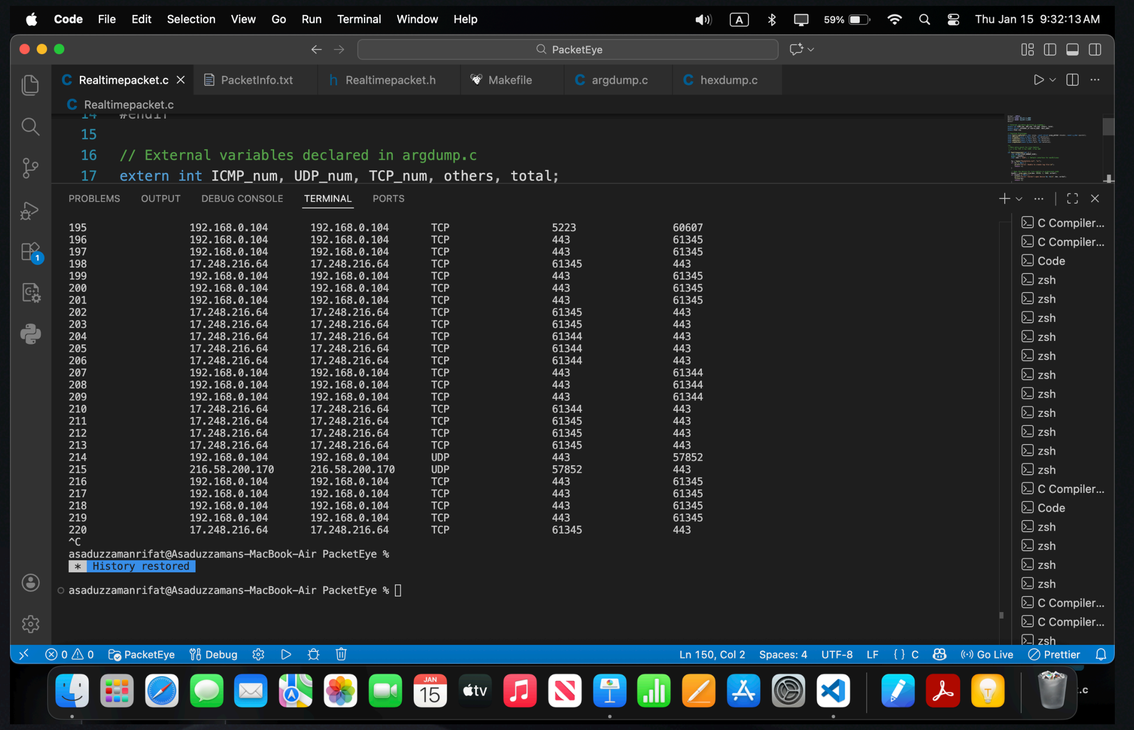
Main Objectives:

- **Live Capture:** Use the `libpcap` library to listen to network traffic.
- **Deep Analysis:** Break down packets (IP, TCP, UDP, ICMP) to see what is inside.
- **Modular Design:** Easy to add new features like security testing later.

System Architecture

The project is structured according these files,

- **`argdump.c` (The Menu):** Handles user commands and filters.
- **`Realtimepacket.c` (The Engine):** Connects to the hardware to "sniff" the data.
- **`analyzer.c` (The Translator):** Converts raw 0s and 1s (binary) into readable info.
- **`Makefile` (The Builder):** Compiles the whole project with one simple command.



Technical Working Process:

1. **Removing the Shell:** The tool skips the first 14 bytes (Ethernet header) to get to the actual data.
2. **Checking the Protocol:** It looks at the **IP header** to see if the packet is a Website request (TCP), a Video stream (UDP), or a Ping (ICMP).
3. **Saving the Data:** Everything is shown on the screen and saved in "`PacketInfo.txt`" so anyone can check it later.

Future Roadmap:

Upcoming Security & Diagnostic Tools

- **Network Stress Testing:** Simulating "Flood Attacks" to check if a network or server is strong enough.
- **Local Device Mapping:** Identifying every device connected to the network using their physical (MAC) addresses.
- **Deep Inspection:** A "Hex-View" mode to see the raw, hidden details inside any packet.
- **Offline Analysis:** The ability to open and read saved network files (`.pcap`) from other tools like Wireshark.

Live Demonstration

Step 1: Run `make` to build the program.

Step 2: Start the sniffer using `sudo ./PackAnalyzer -i 1`.

Step 3: Watch the live traffic on the screen and check the **`PacketInfo.txt`** file for the log.

Conclusion

Current Progress

Completed: The core engine for capturing and reading traffic is 100% ready.

Next Goal: Adding the security testing and network mapping features.

Thank You