

# 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.

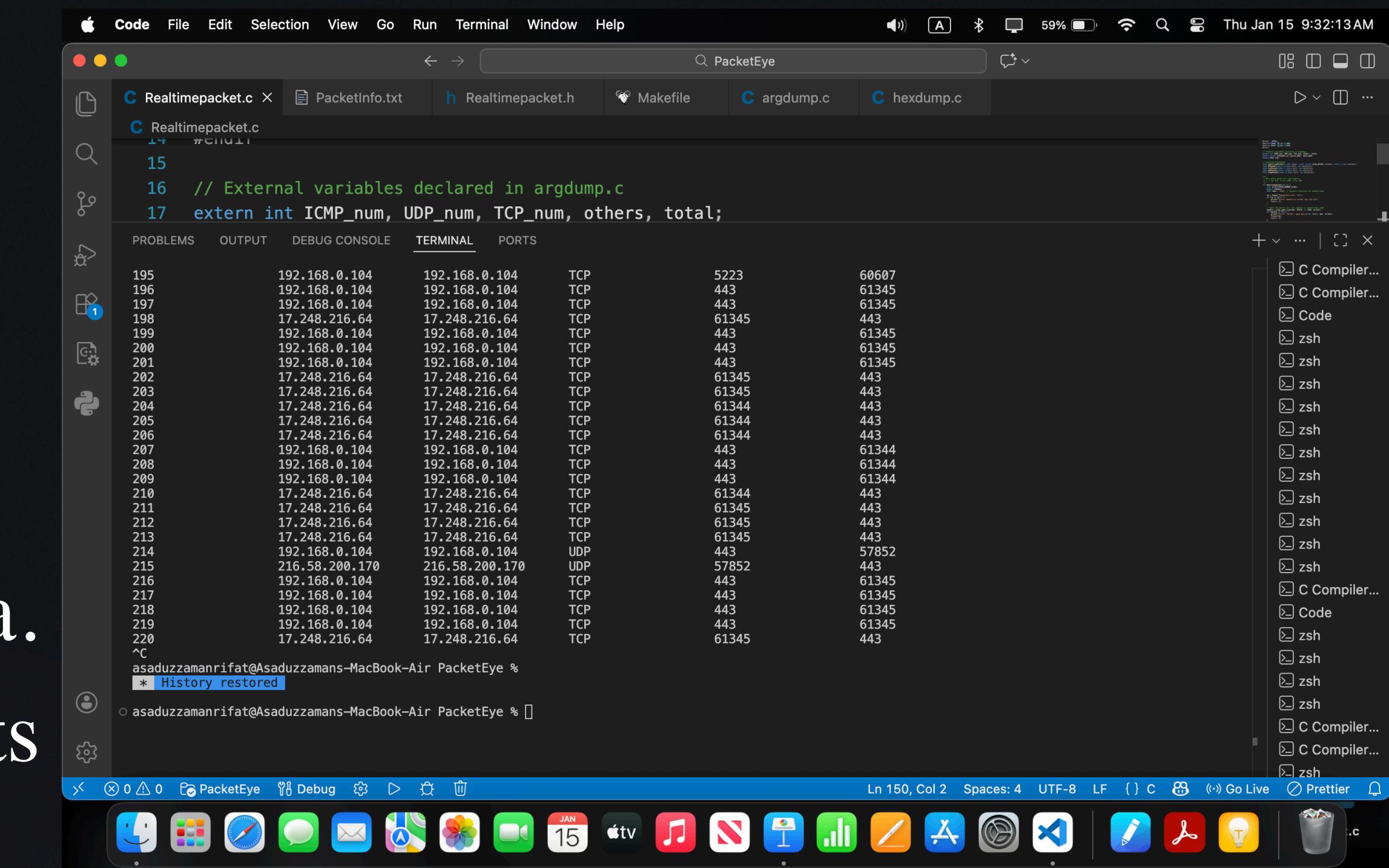
## Main Objectives:

- **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.



```
Code File Edit Selection View Go Run Terminal Window Help
Realtimepacket.c X PacketInfo.txt Realtimepacket.h Makefile argdump.c hexdump.c
15
16 // External variables declared in argdump.c
17 extern int ICMP_num, UDP_num, TCP_num, others, total;

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
195 192.168.0.104 192.168.0.104 TCP 5223 60607
196 192.168.0.104 192.168.0.104 TCP 443 61345
197 192.168.0.104 192.168.0.104 TCP 443 61345
198 17.248.216.64 17.248.216.64 TCP 61345 443
199 192.168.0.104 192.168.0.104 TCP 443 61345
200 192.168.0.104 192.168.0.104 TCP 443 61345
201 192.168.0.104 192.168.0.104 TCP 443 61345
202 17.248.216.64 17.248.216.64 TCP 61345 443
203 17.248.216.64 17.248.216.64 TCP 61345 443
204 17.248.216.64 17.248.216.64 TCP 61344 443
205 17.248.216.64 17.248.216.64 TCP 61344 443
206 17.248.216.64 17.248.216.64 TCP 61344 443
207 192.168.0.104 192.168.0.104 TCP 443 61344
208 192.168.0.104 192.168.0.104 TCP 443 61344
209 192.168.0.104 192.168.0.104 TCP 443 61344
210 17.248.216.64 17.248.216.64 TCP 61344 443
211 17.248.216.64 17.248.216.64 TCP 61345 443
212 17.248.216.64 17.248.216.64 TCP 61345 443
213 17.248.216.64 17.248.216.64 TCP 61345 443
214 192.168.0.104 192.168.0.104 UDP 443 57852
215 216.58.200.170 216.58.200.170 UDP 57852 443
216 192.168.0.104 192.168.0.104 TCP 443 61345
217 192.168.0.104 192.168.0.104 TCP 443 61345
218 192.168.0.104 192.168.0.104 TCP 443 61345
219 192.168.0.104 192.168.0.104 TCP 443 61345
220 17.248.216.64 17.248.216.64 TCP 61345 443
^C
asaduzzamanrifat@Asaduzzamans-MacBook-Air ~
* History restored
asaduzzamanrifat@Asaduzzamans-MacBook-Air ~
Ln 150, Col 2 Spaces: 4 UTF-8 LF () C ⌘ Go Live ⌘ Prettier ⌘
```

## 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