Progress Report #2

Network Traffic Analysis Tool

Group 4: Nicholas Yang, Alexander Gottschalk, Youssef Mouzoune, Tristan Kras

**1. Introduction**

The **Packet Sniffer Program** is a Python-based network analysis tool designed to capture, analyze, and log network traffic in real time. It leverages the scapy library to inspect packets, detect anomalies, and generate reports for further analysis.

**Key Objectives:**

* Capture live network traffic with customizable filters.
* Analyze packet metadata (source/destination IP, protocol, size).
* Detect unusual traffic patterns (e.g., DDoS attacks).
* Log captured data for forensic analysis.
* Visualize traffic trends in real time.

**2. Functionalities**

**2.1 Packet Capture & Filtering**

* **Sniffing Engine**: Uses scapy.sniff() to intercept packets.
* **Custom Filters**: Users can filter by:
  + **Protocol** (TCP/UDP/ICMP).
  + **IP Address** (specific hosts).
  + **Port Numbers** (e.g., port 80 for HTTP).

**2.2 Live Traffic Analysis**

* **Metrics Tracked**:
  + **Packets Per Second (PPS)** and **Data Rate (bps)**.
  + **Total Data Transferred** (in KB).
* **Threshold-Based Alerts**: Flags anomalies (e.g., PPS > 20 or bps > 100,000).

**2.3 Data Logging & Export**

* **PCAP Export**: Saves raw packets to .pcap files for Wireshark analysis.
* **CSV Reports**: Logs structured data (timestamp, IPs, protocol, size).

**2.4 Interactive Inspection**

* **Command-Line Interface**:
  + Type stop to halt sniffing and inspect packets.
  + View summaries or full packet details (e.g., packet.show()).
  + A screenshot of a computer

    AI-generated content may be incorrect.

**2.5 Real-Time Visualization**

* **Matplotlib Graph**: Displays live **PPS trends** (updated every second).

A screen shot of a computer

AI-generated content may be incorrect.

**3. Technical Implementation**

**3.1 Core Libraries**

| **Library** | **Purpose** |
| --- | --- |
| scapy | Packet capture/analysis. |
| threading | Concurrent traffic analysis. |
| matplotlib | Live graph rendering. |
| csv | Export logs to CSV. |

**3.2 Key Functions**

* packet\_callback(): Processes each packet (extracts IPs, protocol, size).
* detect\_anomalies(): Checks for traffic spikes.
* save\_log(): Writes metadata to CSV.
* start\_graph(): Updates live PPS graph.

**3.3 Threading Model**

* **Main Thread**: Handles packet capture.
* **Worker Threads**:
  + **Analyze traffic stats** (every 5 seconds).
  + **Process user input** (e.g., stop command).

**4. Sample Output**

**4.1 Terminal Log**

A screen shot of a computer

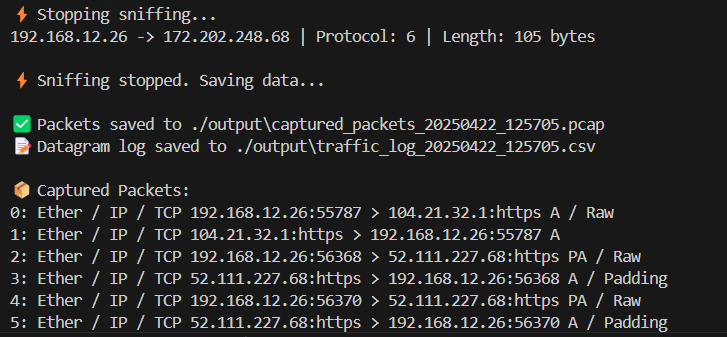
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**4.2 Exported Files**

* captured\_packets\_20240501\_1420.pcap
* traffic\_log\_20240501\_1420.csv



**5. Conclusion**

The Packet Sniffer Program is a **versatile tool** for network monitoring, security analysis, and troubleshooting. Its modular design allows for easy extensions, such as:

* Adding **deep packet inspection** (DPI).
* Integrating **machine learning** for advanced anomaly detection.
* Supporting **remote monitoring** via APIs.

By combining real-time analysis with detailed logging, it serves as a foundation for building more advanced network security applications.