Network Systems and Security
Assignment: Network Traffic Analysis

Course: Network Systems and Security

March 28, 2025

1 Introduction

You have been appointed as a **Network and System Security Analyst**. Your organization has received network traffic logs from a firewall or an Intrusion Detection System (IDS). These logs contain network flow details, including traffic between different IPs, packet statistics, protocol information, and timestamps.

For ease of analysis and to avoid unnecessary preprocessing overhead, we are providing the data in **CSV format instead of raw XML files**. This decision is made as a favor to simplify logistics, allowing you to focus on the core analysis rather than the format conversion.

Your task is to extract meaningful insights from this raw data, detect unusual patterns, and evaluate security threats. Given the nature of network monitoring systems, storing exact statistics for all traffic flows is infeasible due to memory constraints. Hence, in certain tasks, you are required to use **sublinear space** instead of storing full data.

Given the nature of network monitoring systems, storing exact statistics for all traffic flows is infeasible due to memory constraints. Hence, in certain tasks, you are required to use **sublinear space** instead of storing full data. You are free to make your own design choices, such as thresholds, duration, etc.

2 Tasks and Evaluation Criteria

2.1 1. Basic Network Traffic Statistics (25 Marks)

To begin, you need to parse the CSV files and extract key traffic details. Implement the following:

- Compute the total number of network flows recorded.
- Identify the **top 5 most used protocols**.
- Identify the **top 10 most active source and destination IPs**.
- Compute the **average packet size**.
- Find the **most common source-destination pair**.
- Identify which IPs are **consistently communicating** over multiple time windows.
- Detect **irregular spikes** in traffic volume over time.
- Compute the **variance of packet sizes** and discuss potential reasons for high variance.

2.2 2. Traffic Estimation Using Sublinear Space (30 Marks)

Since storing every IP address and every packet count is memory-intensive, you must develop space-efficient methods.

(a) Estimating the number of unique IP addresses (10 Marks)

- Instead of storing all unique IPs, design an approach to estimate the total number of distinct IPs using sublinear space.
- Compute the same metric using **both sublinear space and full linear space**, and compare the results.
- Ensure that the error rate (difference between sublinear and linear results) is at most **10%**.
- Analyze the trade-offs in terms of space, accuracy, and computation time.

(b) Identifying frequently contacted destination IPs (10 Marks)

- Identify heavy hitters (destination IPs that receive a significant fraction of traffic).
- Instead of storing per-IP counts, design an efficient method to approximate the most frequently contacted IPs using sublinear space.

- Compare the approximations with exact counts using full memory and analyze the accuracy.
- Ensure the error rate remains below **10%**.

(c) Efficient Membership Testing for Blocklists (10 Marks)

- You need to determine whether an IP has been seen before **without storing all IPs**.
- Implement a **Bloom filter**-based solution to check if an IP was previously observed.
- Compare the false positive rate and trade-offs with an exact membership test using full space.

2.3 3. Advanced Anomaly Detection (25 Marks)

Network anomaly detection is critical for identifying potential security threats. The anomalies to detect include:

(a) Statistical Traffic Analysis (10 Marks)

- Identify and define **statistical thresholds** for anomalies in packet sizes, flow counts, and protocol distributions.
- Compare traffic patterns over different time windows (e.g., per hour vs. per day) to detect significant deviations.
- Flag **outlier IPs** that send a much higher/lower volume of traffic than expected.

(b) Behavioral Analysis (10 Marks)

- Identify IPs that **drastically change behavior** over time (e.g., sudden spikes in traffic after long inactivity).
- Detect **network-wide correlation patterns**, such as whether multiple IPs start communicating with a common target in a short period.
- Flag **any destination IP** that is being contacted by a large number of different sources in a short time.

(c) Suspicious Communication Patterns (5 Marks)

- Identify **long-duration connections** that stay open longer than normal.
- Detect **IPs communicating using multiple protocols** in a short time.

2.4 4. Deep Security Threat Analysis (25 Marks)

Beyond anomalies, you must now conduct a deeper **threat-hunting investigation** using the data.

(a) Detecting Complex Attack Patterns (10 Marks)

- Find **stealthy port scans** where an attacker scans over a long period to avoid detection.
- Identify **slow DDoS attacks** that may not show obvious traffic spikes but consume resources gradually.
- Detect **IP hopping behavior** where a single attacker uses multiple IPs to evade detection.

(b) Malicious Payload Identification (10 Marks)

- Extract payload data from flows and identify unusual payload patterns.
- Flag **encrypted traffic** that does not match normal encrypted traffic behavior.
- Identify network flows that **resemble known malware command-and-control patterns**.

(c) Threat Attribution and Risk Analysis (5 Marks)

- Categorize each detected security event as:
  - Low-risk
  - Medium-risk
  - High-risk
- Provide reasoning based on observed behavior.
- Generate a risk report summarizing findings.

3 Submission Guidelines

- Submit a **Jupyter Notebook or Python script** with your analysis.
- Include **detailed comments and explanations** in your code.
- Provide a short **PDF report** summarizing your findings.
- Deadline: **4th April**

4 Hints

- For **Part 2**, explore techniques such as Bloom filter or any other sublinear technique.
- Use statistical techniques to analyze errors and compare approximate vs. exact counts.
- For anomaly detection, compute baselines for normal behavior and visualize deviations.