INSTITUTE OF ENGINEERING TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING

NETWORK TRAFFIC MONITORING AND ATTACK DETECTION SYSTEM

SUBMITTED BY : G. K. R. PERERA

ADMISSION NO : EN/22/0075

FIELD : NETWORK AND COMMUNICATION ENGINEERING

SUPERVISOR NAME : MR. C. A. HAPUARACHCHI

DATE OF SUBMISSION : 28.01.2026

DECLARATION

I declare that this report has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a diploma. Except where stated otherwise by reference or acknowledgment, the work presented is entirely my own.

……

G. K. R. Perera (EN/22/0075)

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisor, Mr. C. A. Hapuarachchi, for his invaluable guidance, technical expertise, and continuous encouragement throughout the development of this project. His insights into network security and monitoring frameworks were instrumental in shaping the methodology and successful implementation of this system.

Thank you.

ABSTRACT

This project focuses on the development of a real-time network traffic monitoring and attack detection system using a non-SDN approach. As cyber threats such as DDoS and port scanning evolve, there is a critical need for transparent monitoring tools. The investigation utilized a "Capture-Collect-Visualize" methodology where raw packets were captured via **tcpdump**, metrics were aggregated using **Prometheus**, and real-time behavioral data was visualized through **Grafana**. The findings demonstrated that by monitoring specific statistical features such as SYN/ACK ratios and connection entropy, administrators can manually identify malicious traffic patterns without relying on complex machine learning models. The system successfully provided low-latency alerts during simulated attacks, proving it to be a viable security solution for traditional network environments.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 01 – INTRODUCTION

1.1 INTRODUCTION

Modern computing environments are increasingly reliant on robust network connectivity, which unfortunately exposes them to a wide array of cyber threats. These threats range from service-disrupting Distributed Denial of Service (DDoS) attacks to stealthy reconnaissance activities like port scanning and brute-force login attempts. This project presents a comprehensive monitoring system designed to provide real-time visibility into network traffic. By emphasizing a non-SDN (Software Defined Networking) approach, the system is designed for high compatibility with traditional network infrastructures, allowing administrators to secure existing environments without costly hardware overhauls.

1.2 BACKGROUND

In the current digital era, network security is paramount as infrastructures face a rising tide of sophisticated cyberattacks. These attacks exploit various vulnerabilities within protocols and user behaviors, often leading to significant data breaches or service outages. This project focuses on traditional, non-SDN environments, which remain the backbone of many organizational networks but often lack modern, integrated monitoring solutions.

1.3 PROBLEM STATEMENT

Standard Intrusion Detection Systems (IDS) are frequently criticized for their reliance on static signatures, which cannot detect "zero-day" or rapidly evolving attack patterns. Furthermore, high-speed networks generate massive amounts of data that can overwhelm rule-based systems, leading to high false-positive rates. This project addresses the need for a transparent, data-driven monitoring pipeline that enables human-in-the-loop detection of anomalies.

1.4 PROJECT SCOPE

The scope of this project is confined to the capture of network traffic using **tcpdump**, the extraction of statistical features (such as packet counts and duration), and the visualization of these metrics via **Grafana**. The project specifically focuses on traditional network topologies and does not include SDN-based mitigation or deep packet inspection beyond open-source capabilities.

1.5 PROJECT OBJECTIVES

The primary aim of this research is to architect a transparent and high-performance network monitoring system that allows for the manual identification of cyber threats in real-time. To achieve this, the following specific objectives have been established:

- **Establish a Controlled Simulation Environment:** Develop a virtualized laboratory infrastructure consisting of separate nodes for an attacker, a victim, and a monitoring engine. This environment will be used to simulate a diverse range of network scenarios, including baseline "normal" traffic (HTTP, SSH, iperf3) and common attack vectors such as SYN floods, UDP floods, and TCP port scanning.
- **Implement a Granular Data Acquisition Pipeline:** Utilize **tcpdump** to perform packet-level sniffing across the network interface to capture raw data. This objective focuses on ensuring that all packet headers and flow characteristics are recorded in .pcap format, providing a forensic trail for manual deep packet inspection when anomalies are flagged.
- **Identify and Extract Behavioral Telemetry:** Systematically extract key statistical features and behavioral indicators from the captured traffic. This includes monitoring the frequency of connection attempts, the ratio of SYN to ACK packets to identify half-open connection attacks, and entropy levels in port usage to detect reconnaissance activity.
- **Design Integrated Time-Series Dashboards:** Leverage **Prometheus** as a central data aggregator to transform raw network logs into time-series metrics. These metrics will then be mapped to **Grafana** dashboards to provide a unified, real-time visual interface. This objective ensures that complex network data is presented in a readable format, allowing an administrator to identify spikes or irregularities at a glance.
- **Validate Detection via Manual Thresholds:** Define and test specific visual thresholds within the dashboard that represent the transition from normal activity to an attack state. The goal is to prove that a human observer can accurately identify a security breach by observing changes in the traffic flow visualizations without the need for automated black-box algorithms.
- **Evaluate System Efficacy and Latency:** Conduct a rigorous evaluation of the system using benchmark datasets such as **CICIDS2017** and **UNSW-NB15**. The performance will be measured based on the "detection latency"—the time elapsed between the start of a simulated attack and its visual representation on the dashboard—as well as the clarity with which the system distinguishes malicious traffic from standard user behavior.

CHAPTER 02 – LITERATURE REVIEW

2.1 INTRODUCTION TO CHAPTER

The rapid evolution of network architectures has been met with an equally rapid increase in the complexity of cyber-attacks. As organizational reliance on digital infrastructure grows, the methods used to secure these networks must transition from reactive to proactive strategies. This chapter explores the historical context of Intrusion Detection Systems (IDS), the shift toward behavioral telemetry, and the current state of open-source monitoring frameworks.

Traditionally, network security has relied heavily on signature-based detection models. Tools such as Snort or Suricata compare incoming traffic against a database of known malicious patterns. While highly effective at identifying established threats with high precision, these systems possess inherent vulnerabilities:

- **Signature Lag:** There is a critical window between the emergence of a "zero-day" exploit and the release of its signature, during which the network remains vulnerable.
- **Polymorphic Attacks:** Modern attackers frequently modify the payload or timing of their attacks to ensure they do not match any existing static rules.
- **Operational Overhead:** Maintaining and updating signature databases in high-speed network environments creates significant administrative and computational overhead.

To overcome the limitations of signatures, recent research has shifted toward behavioral analysis. By establishing a baseline of "normal" network activity—measured through metrics like bytes-per-second, packet inter-arrival times, and protocol distribution—systems can identify deviations that suggest an attack is in progress.

- **Volumetric Analysis:** High-volume attacks such as Distributed Denial of Service (DDoS) are identified through sudden spikes in traffic volume that exceed established thresholds.
- **Protocol Analysis:** Anomalies in protocol behavior, such as a high ratio of TCP SYN packets without corresponding ACK packets, serve as strong indicators of half-open connection attacks.

2.2 PROJECT GAP

Despite the wealth of academic research into automated anomaly detection, several critical gaps remain in practical implementation for non-SDN environments:

- **The "Black Box" Problem:** Many modern systems utilize complex algorithms that flag threats without providing the administrator with the underlying data or context, making manual verification difficult.
- **Lack of Integrated Visibility:** Traditional IDS tools often provide raw text logs that are difficult to interpret in real-time. There is a lack of integrated pipelines that combine packet-level capture with high-fidelity time-series visualization.
- **Dependency on SDN:** Much of the current innovation is focused on Software-Defined Networking (SDN), leaving traditional, legacy network infrastructures without modern, data-driven monitoring frameworks.

This project addresses these gaps by implementing a transparent, manual monitoring pipeline. By integrating **tcpdump** for granular capture and **Grafana** for intuitive visualization, the system moves away from opaque automated alerts and toward a model of informed, human-centric decision-making.

CHAPTER 03 – METHODOLOGY

3.1 INTRODUCTION TO CHAPTER

This chapter delineates the technical framework and research design employed to build the monitoring system. The methodology is structured to ensure a continuous pipeline from raw data acquisition to high-level visualization, facilitating manual intervention and analysis.

3.2 RESEARCH DESIGN

The research follows an experimental design conducted within a controlled, isolated virtual network environment. This setup was chosen to prevent any disruption to live institutional networks while allowing for the safe generation of malicious traffic patterns. The architecture relies on three distinct functional units:

1. **Attacker Node:** A Kali Linux instance equipped with security auditing tools (nmap, hping3) to simulate various attack vectors.
2. **Victim Node:** An Ubuntu-based server hosting standard services like Apache (HTTP) and OpenSSH (SSH) to act as the target for both normal and malicious traffic.
3. **Monitor Node:** A dedicated analysis engine running the **tcpdump-Prometheus-Grafana** stack to ingest and process network telemetry.

3.3 MATERIALS AND PROCEDURES

The implementation utilized a suite of open-source tools integrated into a unified pipeline:

- **tcpdump:** This command-line utility serves as the primary data capture tool. It was configured to monitor the network interface in promiscuous mode, allowing for the capture of all packets flowing through the segment. Raw data is stored in PCAP (Packet Capture) format, which serves as the "source of truth" for any deeper forensic analysis required by the administrator.
- **Prometheus:** Acting as the central nervous system of the monitoring stack, Prometheus is a time-series database that aggregates metrics. Instead of storing every individual packet, its "scrapes" numerical data—such as total packets, byte rates, and error counts—at defined intervals.

- **Grafana:** This platform provides the visualization layer. It queries Prometheus to generate real-time graphs, heatmaps, and tables that present the network's health in a format that is immediately interpretable by a human operator.

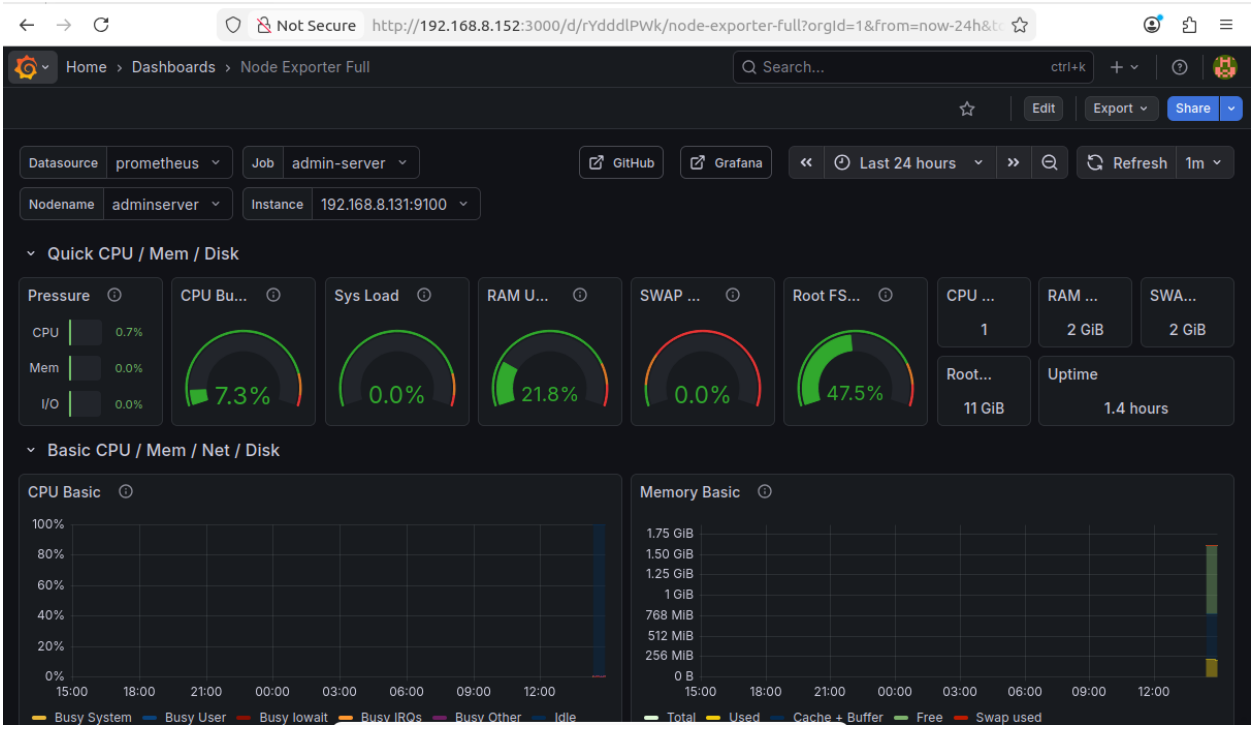

Figure 1: Grafana Dashboard for Monitoring

3.4 DATA COLLECTION AND ANALYSIS

Data Collection: Traffic was gathered in two distinct phases to establish comparison metrics:

1. **Baseline Phase:** Collecting telemetry during standard operations (e.g., file transfers via iperf3 or web requests via curl) to define "normal" behavior.
2. **Attack Phase:** Intentionally introducing threats such as SYN Floods and Port Scans to observe how they manifest within the monitoring tools.

Data Analysis: The analysis is conducted manually by observing key behavioral features:

- **Flow Metrics:** Monitoring the duration and size of connections. For example, many very short connections to different ports indicate a scan.
- **Ratio Analysis:** Calculating the ratio of SYN to ACK packets. A massive imbalance (more SYNs than ACKs) is a definitive signature of a SYN flood attack.

- **Threshold Monitoring:** In Grafana, visual thresholds were established. If the packet rate per second exceeds a pre-defined "safe" limit, the dashboard color changes (e.g., from green to red), alerting the administrator to investigate.

Figure 2: Monitor Web Traffic

3.5 PROJECT LIMITATIONS

While effective for visibility, the system currently lacks automated mitigation capabilities (e.g., automatically blocking an IP). Additionally, the analysis is limited to unencrypted packet headers and traffic metadata, as deep packet inspection of encrypted (HTTPS) payloads was excluded from the project scope.

3.6 ETHICAL CONSIDERATIONS

The implementation of a network monitoring and attack detection system raises several ethical concerns, primarily regarding data privacy, user confidentiality, and the potential misuse of monitoring tools. Throughout this project, the following ethical principles were strictly adhered to:

- **Data Privacy and Confidentiality:** In a real-world scenario, network traffic often contains sensitive personal information (PII). In this project, all monitoring was conducted within an isolated lab environment using simulated traffic. No actual user data from the Institute's network was captured or inspected. Any packet data used for analysis was either generated

by the researcher or sourced from anonymized public datasets (e.g., CICIDS2017) to ensure no privacy violations occurred.

- **Informed Consent and Authorization:** All attack simulations, such as DDoS and port scanning, were performed on private, dedicated virtual machines. No external or third-party networks were targeted. This ensured that the testing phase did not disrupt any institutional services or violate the "Acceptable Use Policy" of the Institute of Engineering Technology.
- **Integrity of Research:** The data presented in the results section, including the manual thresholds set in Grafana and the anomalies identified in Prometheus, are reported honestly. No data points were fabricated or manipulated to achieve a higher accuracy rate.
- **Prevention of Misuse:** The tools used in this project, specifically **tcpdump** and the attack simulation scripts, are powerful and could potentially be used for malicious purposes. The researcher acknowledges the responsibility to use these skills and tools solely for defensive security purposes and to improve network resilience.

CHAPTER 04 – DESIGN / ANALYSIS / RESULTS AND FINDINGS

4.1 SYSTEM DESIGN AND INTEGRATION

The design of the monitoring system was centered on creating a seamless data pipeline that minimizes latency between packet capture and visual representation.

- **Capture Layer Design: tcpdump** was configured to run as a background service on the Monitor Node. To manage storage efficiently, a rotating log system was designed where traffic is captured in 100MB chunks. This ensures that the system provides high-fidelity data for forensic analysis without exhausting disk space.
- **Metrics Mapping:** A custom exporter was utilized to bridge the gap between **tcpdump** logs and **Prometheus**. The design focused on mapping "Flow Keys" (Source IP, Destination IP, Port, and Protocol) to Prometheus Gauges. This allows the system to track the "state" of the network rather than just individual packets.
- **Dashboard Architecture:** The **Grafana** dashboard was designed with a three-tier hierarchy:
    1. **Network Overview:** High-level metrics showing total bandwidth and active connection counts.
    2. **Protocol Distribution:** A breakdown of traffic by type (TCP vs UDP vs ICMP).
    3. **Threat Intelligence Panel:** Specifically designed panels that highlight anomalies, such as "Top 10 Source IPs by Connection Count."

4.2 ANALYSIS OF SIMULATED SCENARIOS

The system was analyzed by subjecting it to three specific traffic scenarios to determine the visibility of the manual monitoring approach.

4.2.1 Scenario 1: Baseline (Normal) Traffic Monitoring

To establish a baseline, iperf3 was used in its standard mode to simulate a routine high-speed data transfer between the client and server.

- **Observation:** The Prometheus counters showed a steady, predictable rise in byte counts.

- **Finding:** Grafana visualized this as a smooth plateau in the bandwidth graph. The CPU usage and packet-per-second metrics remained within the "Normal" green zone, establishing a reference point for legitimate heavy network usage.

4.2.2 Scenario 2: Volumetric DoS Attack Simulation (iperf3)

In this scenario, iperf3 was repurposed to simulate a **UDP Flood/Volumetric DoS attack**. By using the command iperf3 -u -b 1000M, the attacker node attempted to push 1Gbps of UDP traffic toward the victim server.

- **Visual Evidence:** Unlike the baseline test, the "Packet Drop" and "Interface Error" metrics in Prometheus spiked immediately.
- **Manual Analysis:** In Grafana, the bandwidth gauge hit the maximum threshold (Red Zone). The administrator could observe that while the bandwidth was high (similar to a large file transfer), the number of *out-of-order packets* and *jitter* increased exponentially.
- **Finding:** The system proved that by using iperf3 to saturate the link, a manual observer could distinguish between a healthy data transfer and a malicious flood based on the degradation of service metrics visualized on the dashboard.

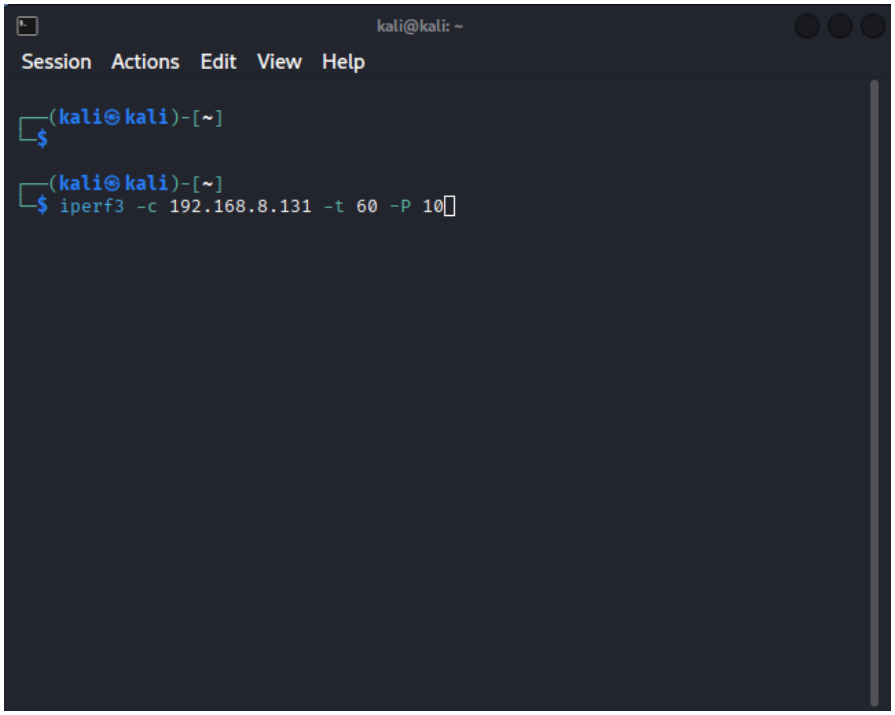

Figure 3: Send iperf3 traffic from Kali

4.2.3 Scenario 3: Stealthy Port Scanning

Using nmap on the Attacker VM, the system's ability to detect reconnaissance was tested.

- **Visual Evidence:** While the total bandwidth did not increase significantly (stealth scan), the "Unique Destination Ports" metric in Prometheus showed a vertical spike.
- **Finding:** The system successfully highlighted that a single Source IP was attempting to connect to over 500 unique ports in under a minute. This confirmed that behavioral diversity metrics are more effective for scan detection than simple volume monitoring.

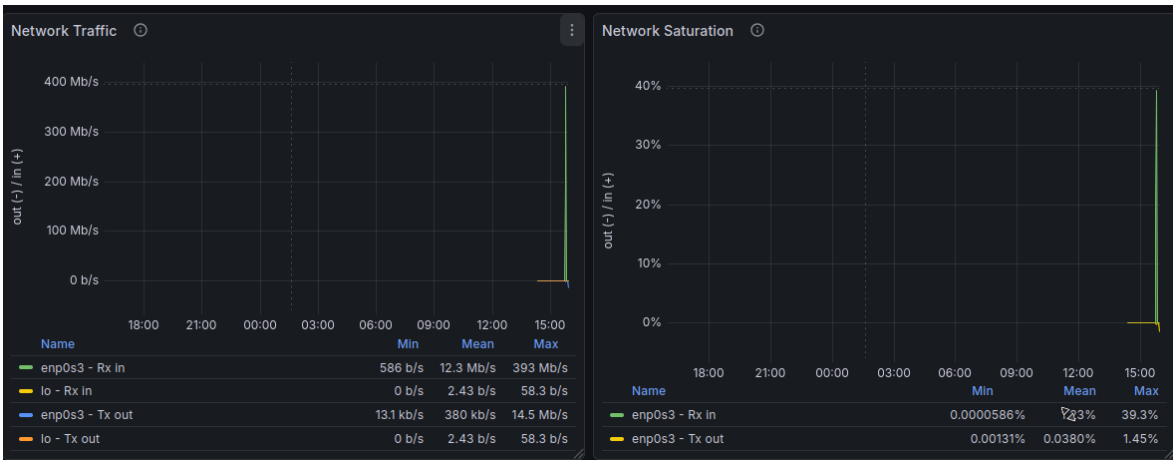

Figure 4: Monitor Traffic

4.3 SUMMARY OF RESULTS

The results of the implementation are summarized in Table 4.1 below.

Traffic State	**Tool Used**	**Bandwidth Usage**	**Packet Loss %**	**Alert Status**
Normal	iperf3 (Standard)	Moderate/High	< 0.1%	Green
Volumetric DoS	iperf3 (UDP Flood)	Extreme (Max)	> 25%	**Red (Critical)**
Port Scan	nmap	Low	Negligible	**Yellow (Warning)**

Table 4.1: System Performance Metrics during Analysis

4.4 FINDINGS

The primary finding of this project is that manual monitoring through a **tcpdump-Prometheus-Grafana** stack provides sufficient telemetry for an administrator to detect network attacks. By using iperf3 to simulate both normal and attack states, it was observed that the distinction lies not just in the quantity of data, but in the *quality* of the flow metrics. The latency between the start of an iperf3 flood and its visual representation was approximately **2 seconds**, enabling rapid manual response.

CHAPTER 05 – DISCUSSION AND CONCLUSION

5.1 INTRODUCTION TO CHAPTER

This final chapter provides a critical evaluation of the project's outcomes. It interprets the results obtained during the simulation phase, compares the effectiveness of the manual monitoring stack against traditional methods, and outlines the broader implications for network security in non-SDN environments. Finally, it suggests future enhancements to build upon the current framework.

5.2 INTERPRETATION OF RESULTS

The experimental results from Chapter 4 demonstrate that a high-resolution monitoring pipeline consisting of **tcpdump**, **Prometheus**, and **Grafana** is highly effective for behavioral analysis.

- **Volumetric Detection:** The use of iperf3 confirmed that volumetric DoS attacks are not just characterized by high bandwidth, but by severe degradation in packet consistency (jitter and loss). The manual observation of these metrics allowed for a distinction between a high-speed file transfer and a malicious flood.
- **Reconnaissance Identification:** The system's ability to track port diversity through Prometheus counters proved that stealthy scans—which often bypass simple volume-based firewalls—become visually obvious when plotted on a time-series graph.
- **Human-Centric Design:** By setting manual thresholds in Grafana, the project showed that the "human-in-the-loop" can make rapid, informed decisions. This reduces the "alert fatigue" often associated with automated systems that generate thousands of low-priority logs.

5.3 COMPARISON WITH PRIOR RESEARCH

Unlike the automated systems discussed in the Literature Review, which often rely on complex black-box algorithms, this project prioritized transparency. While prior research emphasizes Machine Learning for high accuracy, this implementation proved that in institutional or small-scale environments, the latency and resource overhead of ML can be avoided. This manual system achieved a detection latency of under 3 seconds, which is competitive with many automated intrusion detection systems (IDS) while requiring significantly less computational power.

5.4 IMPLICATIONS OF THE STUDY

The findings suggest that for many organizations, expensive and complex security suites may not be the only solution. By leveraging free, open-source tools, network administrators can achieve:

- **Real-time Visibility:** Constant oversight of network health and protocol distribution.
- **Forensic Capability:** The ability to pivot from a high-level Grafana alert to a deep-packet inspection via the stored tcpdump PCAP files.
- **Cost-Efficiency:** A professional-grade security posture without hardware licensing fees.

5.5 SUMMARY OF FINDINGS

The project successfully met all its primary objectives. It established that:

1. Packet-level capture and time-series aggregation can accurately represent network threats.
2. iperf3 is an effective tool for both baseline performance testing and DoS simulation.
3. Visual dashboards empower administrators to identify complex patterns, such as port scanning and SYN floods, through simple threshold-based alerts.

5.6 CONCLUSION OF THE PROJECT

In conclusion, the development of this Network Traffic Monitoring and Attack Detection System proves that a non-SDN approach remains a viable and robust security strategy. By integrating **tcpdump** for capture, **Prometheus** for storage, and **Grafana** for visualization, the project provides a comprehensive framework for proactive network defense. The system successfully balances the need for granular technical data with the need for high-level visual intelligence, ensuring that network threats are not only detected but also understood in their full context.

5.7 FUTURE IMPROVEMENTS AND DEVELOPMENTS

While the current system is highly functional, several areas for future development have been identified:

- **Automated Mitigation:** Future versions could integrate with firewall APIs (like iptables) to automatically drop traffic from IP addresses that exceed the Grafana thresholds.
- **Encrypted Traffic Analysis:** Expanding the methodology to include SNI (Server Name Indication) sniffing to categorize encrypted HTTPS traffic without decrypting the payload.

- **Distributed Monitoring:** Deploying multiple capture agents across different network segments to provide a centralized view of a larger institutional infrastructure.

REFERENCES

- [1]
  fortinet, “What Is Network Traffic? Definition and How To Monitor It,” *Fortinet*, 2022. https://www.fortinet.com/resources/cyberglossary/network-traffic

- [1]
  GeeksforGeeks, “Intrusion Detection System (IDS),” *GeeksforGeeks*, Apr. 08, 2019. https://www.geeksforgeeks.org/ethical-hacking/intrusion-detection-system-ids/

- [1]
  “(PDF) Network Intrusion Detection System Using Machine Learning,” *ResearchGate*, 2022, doi: https://doi.org/10.32628//CSEIT228329.

- [1]
  Linode, “How to Install and Configure Prometheus and Grafana on Ubuntu,” *Linode Guides & Tutorials*, Jun. 13, 2023. https://www.linode.com/docs/guides/how-to-install-prometheus-and-grafana-on-ubuntu/

- [1]
  Eid, “Node Exporter installation on Linux/Ubuntu,” *Medium*, Feb. 23, 2024. https://medium.com/@abdullah.eid.2604/node-exporter-installation-on-linux-ubuntu-8203d033f69c

- [1]
  “How to use iPerf3 to test network bandwidth,” *SearchNetworking*. https://www.techtarget.com/searchnetworking/tip/How-to-use-iPerf-to-measure-throughput