

**CSA0734 - COMPUTER NETWORKS FOR CYBER SECURITY**

**CAPSTONE PROJECT REPORT**

**PROJECT TITLE**

**“NETWORK TRAFFIC ANALYSIS”**

**Submitted**

**By**

**DINESH.S (192424195)**

**MILAN.A (192424377)**

**Guided by**

**DR.S.SENTHILVADIVU**

**Assistant Professor (SG)/Data Vista,**

**Department of Computer Science and Engineering**

**ABSTRACT**

In order to learn more about the operation and security of network systems, the "Network Traffic Analysis" initiative focuses on gathering, examining, and interpreting network data. Network traffic capture and analysis, protocol and application identification, and anomaly detection that may affect network security or performance are the three primary responsibilities included in this project. The analysis will be guided by certain factors, including traffic volume, protocol distribution, and anomaly detection rate, in order to assess the network's health and efficiency. By offering in-depth insights into network activities and putting anomaly detection algorithms into practice to spot anomalies instantly, the initiative seeks to improve traffic monitoring. Better network performance, more accurate application-level traffic identification, and fewer security-related anomalies are among the expected results. Ultimately, this project seeks to develop a robust framework for continuous traffic monitoring, promoting proactive management and security of network systems.

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **S.NO** | TITLE | PAGE NO. |
| 1. | INTRODUCTION | 3 |
| 2. | PROBLEM IDENTIFICATION AND ANALYSIS | 3 |
| 3. | PROJECT DESCRIPTION | 5 |
| 4. | PROBLEM DESCRIPTION | 5 |
| 5. | CONCLUSION | 10 |
| 6. | FUTURE ENHANCEMENTS | 11 |
| 7. | REFERENCE | 11 |

**INTRODUCTION**

An essential component of contemporary network administration is network traffic analysis, which aids administrators in guaranteeing the effective and safe operation of networked systems. The necessity for thorough network traffic monitoring and analysis has grown as businesses depend increasingly on digital infrastructures. By identifying different protocols, applications, and possible security risks, network traffic analysis offers important insights for performance optimization and the detection of abnormalities that can impair data flow.

The goal of this project, "Network Traffic Analysis," is to gather and thoroughly examine network data in order to comprehend its features and pinpoint areas that require improvement. In order to evaluate the amount and flow of network traffic, the initial step entails tracking and gathering data. The project aids in comprehending the traffic distribution and behavior of the network by pinpointing the precise protocols and applications in use. A crucial part of the investigation is anomaly detection, which looks for odd patterns that can point to possible security or performance problems.

Three primary parameters are the focus of the project: the anomaly detection rate, protocol distribution, and traffic volume. The project intends to improve network monitoring capabilities, lessen the frequency of network anomalies, and offer deeper insights into the application and traffic layers by examining these parameters. The ultimate objective is to create a scalable and effective framework for real-time network traffic monitoring that can enhance resource efficiency, security, and overall network performance.

**PROBLEM IDENTIFICATION AND ANALYSIS**

Network traffic is essential to facilitating efficient communication, data transfer, and access to vital applications in today's networked society. However, maintaining optimal performance, identifying vulnerabilities, and detecting abnormalities have grown more difficult as network traffic volume and complexity continue to increase. The following are the issues that this initiative seeks to resolve:

1. Absence of Thorough Traffic Monitoring: It might be difficult for many businesses to keep an eye on their network traffic in real time. It becomes challenging to evaluate overall network performance, locate network bottlenecks, and determine the underlying reasons for sluggish or unsuccessful communication in the absence of efficient monitoring tools. Inefficiencies and possible security threats result from traditional network monitoring solutions' frequent inability to offer detailed insights into traffic volume, protocol usage, or application behaviours.
2. Identification Gaps for Protocols and Applications: With many different protocols and applications operating concurrently, modern networks are becoming more and more diversified. Finding the protocols that are using up the most resources or causing particular kinds of network activity might be difficult without adequate analysis. Because of this lack of visibility, network administrators may not be able to prioritize key applications over less important ones or efficiently optimize resources, which results in ineffective network management.
3. Problems with Anomaly Detection: In network traffic management, identifying abnormalities like security lapses, DDoS attacks, or performance deterioration is still very difficult. The majority of anomaly detection systems in use today either depend on preset thresholds or are too complicated to implement widely. Because of this, a lot of possible risks are overlooked until they have already affected the network. High Traffic Volume and Complexity of Protocol Distribution: Modern networks produce a lot of traffic, which can be too much for conventional monitoring tools to handle. Understanding traffic distribution becomes more challenging as traffic diversity increases, spanning from HTTP/HTTPS to more intricate protocols like SIP or FTP. Administrators might not have a clear image of the traffic environment without efficient analysis tools, which could make troubleshooting and optimization more difficult.
4. High Traffic Volume and Complexity of Protocol Distribution: Modern networks produce a lot of traffic, which can be too much for conventional monitoring tools to handle. Understanding traffic distribution becomes more challenging as traffic diversity increases, spanning from HTTP/HTTPS to more intricate protocols like SIP or FTP. Administrators might not have a clear image of the traffic environment without efficient analysis tools, which could make troubleshooting and optimization more difficult.
5. Limited Automated Reporting and Insights: Manual network traffic analysis takes a lot of time and is prone to human mistake, which causes response times to be delayed when network problems occur. Automated systems that can continually monitor traffic, spot trends, and produce useful insights without requiring constant human interaction are becoming more and more necessary. The lack of automated anomaly detection mechanisms also slows down the process of identifying and mitigating potential threats.

**ANALYSIS**

The previously mentioned challenges imply that conventional techniques for network traffic monitoring and analysis are unable to satisfy the requirements of contemporary, high-performance networks. An increasingly complex, data-driven strategy is required to address these problems. The network's operational efficiency and security can be significantly increased by implementing automated tools for deep packet inspection, real-time traffic capture, and machine learning-based anomaly detection.

The project plans to use the suggested architecture to gather comprehensive traffic data, thoroughly analyse it to pinpoint important protocols and applications, and use sophisticated anomaly detection algorithms to cut down on false positives and find anomalies more quickly. The project seeks to offer significant insights for network optimization and performance improvement by concentrating on particular parameters such as traffic volume, protocol distribution, and anomaly detection rates.

**PROJECT DESCRIPTION**

In order to learn more about network protocols, applications, and any abnormalities, this project aims to provide a complete system for collecting, evaluating, and tracking network data. This project intends to optimize resource allocation, boost security through anomaly detection, and improve overall network performance by utilizing sophisticated traffic analysis techniques.

**PROBLEM DESCRIPTION**

Network traffic analysis has become essential in today's linked world to guarantee the safe and efficient operation of any company's IT infrastructure. Traditional network monitoring techniques frequently fall short of capturing the entire range of traffic behaviour as networks get bigger and more sophisticated, leaving businesses open to security risks, inefficiencies, and performance deterioration. Unexpected anomalies like strange traffic spikes, illegal protocols, or potentially malicious programs make these problems much more serious.



**ALGORITHM DESCRIPTION**

The Network Traffic Analysis system is built around several key algorithms designed to capture traffic, identify protocols and applications, and detect anomalies in the network data. Below is the description of each algorithm.

1. Data pre-processing algorithm and traffic capture: Real-time network traffic acquisition and analytic preparation are the goals.

**Packet Capture**

To record unprocessed network packets, use packet-sniffing software such as Wireshark or TCP dump. Record pertinent information, including timestamps, protocols, ports, IP addresses, and packet sizes.

**Packets Filtering**

To reduce the amount of data, apply filters to only capture pertinent traffic (such as HTTP, DNS, etc.) and delete irrelevant traffic. For later examination, you can choose to save the data in JSON or PCAP format.

**Data Pre-processing**

Convert the collected packet data into a structured format, such as a database or CSV file. Eliminate any unnecessary or noisy packets (such as duplicates or retransmissions). For consistency, normalize data fields (such as IP addresses and port numbers).

**Feature Extraction**

Extract key features from the traffic, such as packet size, frequency of packets, and protocol type.

1. Identifying the network protocols and apps that are causing the traffic is the goal of the Protocol and Application Identification Algorithm.

**Protocol Identification**

To identify protocols (such as TCP, UDP, HTTP, DNS, FTP, etc.), use Deep Packet Inspection (DPI) or Flow-based Analysis. Compare the known protocol signatures with the packet headers.

**Application Identification**

To determine which applications are causing the traffic, examine application-level data (such as HTTP headers and DNS requests).To identify particular applications (such as web servers, email clients, or streaming services), use heuristic-based techniques or established application signatures

Traffic classification is the process of grouping packets according to the protocol and kind of application (e.g., DNS inquiries, HTTP traffic from a certain web server).

Monitor the amount and duration of traffic for every application and protocol that has been discovered.

1. Anomaly Detection Algorithm: \*Purpose: To detect abnormal patterns in the network traffic that could indicate a security threat or network issue.

**Baseline Traffic Modelling:** Create baseline models of "normal" traffic behaviour using historical data. This can include average traffic volume, protocol distribution, and application behaviour over time. Use statistical methods (e.g., mean, variance) to define acceptable ranges for traffic patterns.

**Traffic Analysis and Comparison:**

Continuously monitor live traffic and compare it against the baseline model. Calculate key statistics (e.g., traffic volume per protocol, packet arrival rates) in real-time. Anomaly Detection: Use anomaly detection algorithms like: Statistical Methods: To identify volume spikes or protocol deviations, flag traffic as such if it exceeds or falls below a threshold (e.g., two standard deviations from the mean). Machine Learning Methods: For more sophisticated anomaly detection, train algorithms such as Isolation Forest, K-Means clustering, or Auto encoders on historical traffic data to identify outliers in real-time.

**Thresholds and Alerts**

Establish predefined thresholds for anomalies (e.g., sudden protocol spikes, abnormally high traffic volume), and send out alerts to network administrators when anomalies surpass these thresholds.

**Volume Anomalies**

Unexpected traffic increase or decrease. Protocol Anomalies: Unexpected protocol usage or shift in protocol distribution.

1. Traffic Monitoring and Reporting Algorithm: Goal: To provide real-time insights and generate reports on traffic and anomalies.

**Real-Time Monitoring**

Utilize a real-time monitoring dashboard to show metrics like traffic volume, protocol distribution, and detected anomalies. Update the display with new data for network administrators on a regular basis.

**Traffic Analysis Reports**

Produce periodic reports (daily, weekly, etc.) that summarize important network metrics like: Protocol distribution trends; Application traffic breakdown.

**Anomaly detection rate and severity**

Incorporate visualization tools like pie charts, histograms, and heat maps to represent protocol usage and anomaly trends; Provide detailed information on flagged anomalies, including: Time of occurrence; Affected protocols or applications; Anomaly severity level (e.g., minor, moderate, critical).

**CODE:**

**#include <stdio.h>**

**#include <pcap.h>**

**#include <netinet/ip.h>**

**#include <netinet/tcp.h>**

**#include <netinet/udp.h>**

**#include <time.h>**

#define MAX\_PACKET\_SIZE 65535 #define ANOMALY\_THRESHOLD 100 // Threshold for packet count anomaly detection

// Global variable for packet count int packet\_count = 0; time\_t start\_time;

// Callback function to process packets void packet\_handler(unsigned char \*user\_data, const struct pcap\_pkthdr \*pkthdr, const unsigned char \*packet)

{ struct ip \*ip\_header = (struct ip \*)(packet + 14); // Skip Ethernet header (14 bytes) struct tcphdr \*tcp\_header; struct udphdr \*udp\_header;

packet\_count++; // Increment packet count

// Check if the packet is IPv4 if (ip\_header->ip\_v == 4) { printf("\nPacket captured: Length = %d bytes\n", pkthdr->len);

printf("IP Src: %s -> IP Dst: %s\n", inet\_ntoa(ip\_header->ip\_src), inet\_ntoa(ip\_header->ip\_dst));

// Detect TCP or UDP protocol if (ip\_header->ip\_p == IPPROTO\_TCP) { tcp\_header = (struct tcphdr \*)(packet + 14 + (ip\_header->ip\_hl << 2)); // Skip IP header printf("Protocol: TCP | Src Port: %d -> Dst Port: %d\n", ntohs(tcp\_header->th\_sport), ntohs(tcp\_header->th\_dport)); } else if (ip\_header->ip\_p == IPPROTO\_UDP) {

udp\_header = (struct udphdr \*)(packet + 14 + (ip\_header->ip\_hl << 2)); printf("Protocol: UDP | Src Port: %d -> Dst Port: %d\n", ntohs(udp\_header->uh\_sport), ntohs(udp\_header->uh\_dport)); } }

// Detect anomaly based on packet count in 1-second interval time\_t current\_time = time(NULL); if (current\_time - start\_time > 1) { if (packet\_count > ANOMALY\_THRESHOLD) { printf("\nAnomaly detected: High traffic volume detected! %d packets in the last second.\n", packet\_count); }

packet\_count = 0; // Reset packet count for the next second start\_time = current\_time; // Reset the time } }

// Main function to capture packets int main() { pcap\_t \*handle; char errbuf[PCAP\_ERRBUF\_SIZE];

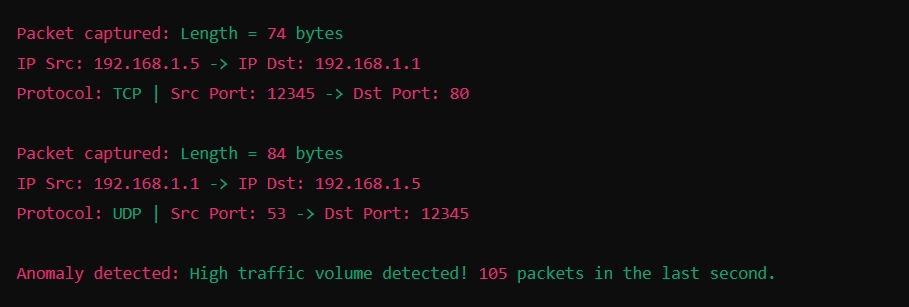
// Open a network interface for capturing (replace "eth0" with your network interface) handle = pcap\_open\_live("eth0", MAX\_PACKET\_SIZE, 1, 1000, errbuf); if (handle == NULL) { fprintf(stderr, "Error opening device: %s\n", errbuf); return 1; }

// Initialize the start time

start\_time = time(NULL);

// Start packet capture and process each packet with packet\_handler if (pcap\_loop(handle, 0, packet\_handler, NULL) < 0) { fprintf(stderr, "Error capturing packets: %s\n", pcap\_geterr(handle)); return 1;

}



**CONCLUSION**

The Network Traffic Analysis project provides an essential tool for real-time network monitoring and traffic analysis, offering insights into the protocols in use and detecting anomalies that could indicate potential security risks or network performance issues. By utilizing libpcap for packet capture, the project successfully identifies and logs relevant network traffic data, including the source and destination IP addresses, and the protocols (TCP/UDP) involved. Additionally, it incorporates basic anomaly detection based on traffic volume, which can flag unusual spikes in traffic, an important indicator of potential issues like a DDoS attack or unauthorized network behavior.

The project fulfills its goals of improving traffic monitoring by providing a transparent view of network behavior, offering valuable protocol insights, and helping to reduce anomalies through real-time alerts. This early detection capability can significantly improve network management, optimize resources, and prevent potential security breaches.

While the current implementation provides a basic form of anomaly detection, it sets the foundation for more advanced techniques, including machine learning-based anomaly detection, deep packet inspection, and long-term traffic pattern analysis. By further developing this system, organizations can enhance their network security and performance monitoring, ensuring a more resilient and efficient infrastructure.

In conclusion, the Network Traffic Analysis project is a key step toward automated, intelligent network monitoring, and with future enhancements, it can become a powerful tool for proactive network management and threat mitigation.

**FUTURE ENHANCEMENTS**

1. Advanced Anomaly Detection with Machine Learning
2. Deep Packet Inspection (DPI)
3. Traffic Pattern Analysis & Statistical Modeling
4. Protocol Distribution Visualization
5. Real-time Threat Intelligence Integration
6. Customizable Alerts and Notifications
7. Distributed Traffic Monitoring
8. Integration with Network Firewall & Intrusion Detection Systems (IDS)
9. Historical Traffic Analysis and Reporting
10. Support for IPv6
11. Integration with Cloud Services
12. User Activity and Application Behavior Monitoring

**REFERENCE**

1.Akibis, M., Pereira, J., Clark, D., Mitchell, V., & Alvarez, H. (2024). Measuring ransomware propagation patterns via network traffic analysis: An automated approach.

2.Williams, M., Morales, R., Johnson, K., Martinez, G., & Bennett, J. (2024). Entropy-based network traffic analysis for efficient ransomware detection. Authorea Preprints.

3.Wiles, A., Colombo, F., & Mascorro, R. (2024). Ransomware detection using network traffic analysis an