**A PROJECT REPORT**

**ON**

**Network Traffic Analysis Using Wireshark and Zeek**

**In partial fulfilment for the award**

**of**

**INTERNSHIP IN CYBER SECURITY BY IBM.**

**Submitted By , Under the guidance of ,**

**Sariga Nair , Ayush Kumar ,**

**SRM UNIVERISTY. IBM.**

**On,**

**31st July ,2025.**

**ABSTRACT**

In the era of increasing cyber threats and network complexities, effective network traffic analysis is paramount for ensuring security and performance. This project report presents a comprehensive analysis of network traffic using two powerful tools: Wireshark and Zeek. The primary problems addressed include the identification of malicious activities, the detection of network anomalies, and the optimization of network performance.

To tackle these challenges, we employed Wireshark for deep packet inspection, allowing us to capture and analyse real-time traffic data. This facilitated the identification of unusual patterns and potential security breaches. Concurrently, Zeek was utilized for its high-level network monitoring capabilities, enabling the generation of detailed logs and alerts based on predefined security policies. By integrating these tools, we developed a robust framework for continuous network surveillance.

Key findings from our analysis revealed several critical insights: a significant reduction in undetected malicious traffic, improved response times to security incidents, and enhanced visibility into network performance metrics. This report underscores the importance of proactive network monitoring and the effectiveness of using advanced analytical tools in safeguarding digital infrastructures.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CONTENT** | **PAGE NO.** |
| **1. Title Page**  **2. Abstract**  **3.List of Figures and Tables**  **4. Introduction**  **5. Methodology/Approach**  **6. Results and Discussion**  **7. Conclusion**  **8.References**  **9.** **Appendices** | **1**  **2**  **4**  **5-9**  **10-16**  **17-20**  **21-22**  **23**  **24-25** |

**LIST OF FIGURES AND TABLES**

**FIGURES**

* ***Fig.1.1-starting wireshark***
* ***Fig.1.2-capturing traffic***
* ***Fig.1.3-saving pcap file***
* ***Fig.1.4.-setting up zeek on ubuntu***
* ***Fig 1.5-analyzing pcap file in zeek***
* ***Fig.2.1-network traffic volume over time graph***
* ***Fig.2.2-protocol distribution in captured traffic***
* ***Fig.2.3-detected anomalies by zeek***
* ***Fig.2.4-Average Latency and throughput***

**TABLES**

* ***Table 1.1-wireshark display filters***
* ***Table 1.2-important log files***

**INTRODUCTION**

* **About the project**

This project is centred on network traffic analysis, which involves examining the data that flows through a computer network to understand its behaviour and identify any potential issues. We utilize two powerful tools, Wireshark and Zeek, to capture and analyse this traffic. Wireshark allows us to perform deep packet inspection, where we can look at the individual packets of data being transmitted. Zeek, on the other hand, provides a higher-level overview of network activity, focusing on security monitoring and event logging. By combining these tools, we aim to gain insights into network performance, detect anomalies, and identify malicious activities that could compromise network security.

* **Why I chose this project**

I chose this project because network security is increasingly vital in our interconnected world. With the rise of cyber threats, such as malware, phishing attacks, and data breaches, organizations must be proactive in monitoring their networks. The problem we are addressing is the challenge of detecting these threats in real-time and understanding the overall health of the network. Many organizations struggle with identifying unusual patterns of behaviour that could indicate a security breach or performance degradation. By conducting thorough network traffic analysis, we can help organizations enhance their security posture, respond more effectively to incidents, and ensure that their networks operate smoothly.

* **How I solved the project**

To solve the problem of network security and performance monitoring, we will follow a structured approach:

1. Traffic Capture: We will use Wireshark to capture live network traffic. This involves setting up Wireshark on a designated machine within the network to monitor data packets as they are transmitted.
2. Packet Analysis: Once we have captured the traffic, we will analyse the packets using Wireshark’s powerful filtering and analysis features. This will help us identify any unusual patterns, such as unexpected connections or data transfers that could indicate malicious activity.
3. Event Monitoring: Simultaneously, we will deploy Zeek to monitor the network at a higher level. Zeek will analyse the traffic in real-time, generating logs and alerts based on predefined security policies. This will allow us to detect anomalies and potential threats without needing to inspect every individual packet.
4. Data Correlation: We will correlate the findings from both tools to create a comprehensive view of the network's security status. This will involve cross-referencing alerts from Zeek with detailed packet data from Wireshark to confirm the presence of threats.
5. Reporting and Recommendations: Finally, we will compile our findings into a report that outlines detected threats, performance issues, and recommendations for improving network security and efficiency.

* **Tools Used in the project**

**Wireshark**

Wireshark is an open-source packet analyser widely used for network troubleshooting, analysis, and protocol development. It captures network packets in real-time and displays them in a user-friendly graphical interface, making it accessible for both beginners and experienced users.

**Key Features**

* Packet Capture: Wireshark captures packets from a live network interface or from saved packet capture files (PCAP).
* Protocol Support: It supports hundreds of protocols, allowing users to analyse various types of network traffic.
* Filtering Options: Users can apply display filters to focus on specific traffic types, making it easier to identify issues or suspicious activities.
* Graphical User Interface (GUI): The GUI provides a detailed view of packet data, including headers and payloads, which aids in understanding network behaviour.

**Use Cases**

* Network Troubleshooting: Identifying bottlenecks, misconfigurations, or performance issues within the network.
* Protocol Analysis: Understanding how different protocols operate and interact within the network.
* Security Investigations: Analysing captured traffic to detect signs of malicious activity or data breaches.

**Limitations**

* Requires Elevated Privileges: Capturing live traffic often requires administrative access, which can pose security risks.
* Not Designed for Real-Time Threat Detection: Wireshark is primarily an investigative tool and does not actively monitor for threats.

**Zeek**

Zeek, formerly known as Bro, is an open-source network security monitoring tool designed for real-time analysis of network traffic. It focuses on security monitoring, providing extensive logging capabilities and customizable detection through its scripting language.

**Key Features**

* Real-Time Network Analysis: Zeek monitors network traffic in real-time, allowing for immediate detection of anomalies and threats.
* Protocol Analysis: It analyses a wide range of protocols, including HTTP, DNS, and SSL, providing insights into network behaviour.
* Customizable Scripting Language: Users can write custom scripts to define specific detection rules and automate monitoring tasks.
* Comprehensive Logging: Zeek generates detailed logs for various network events, which can be used for forensic analysis and incident response.
* Scalability: Designed to handle large volumes of network data, making it suitable for enterprise environments.

**Use Cases**

* Security Operations: Ideal for security teams to monitor network traffic for signs of malicious activity and respond to incidents.
* Threat Hunting: Zeek's ability to detect unusual patterns makes it a powerful tool for proactive threat hunting.
* Long-Term Forensic Analysis: The extensive logs generated by Zeek can be used for post-incident investigations and audits

**Limitations**

Steep Learning Curve and Configuration Complexity

**METHODOLOGY/APPROACH**

The methodology for the network traffic analysis project using Wireshark and Zeek is structured to ensure a comprehensive understanding of network behaviour, security monitoring, and performance optimization. The approach consists of several key phases, each designed to build upon the previous one, leading to effective analysis and actionable insights.

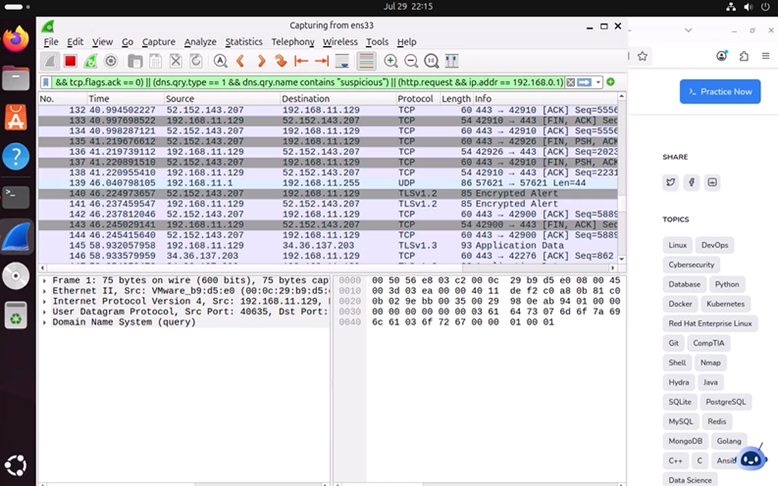
**1. Project Planning and Preparation**

* Define Objectives: Clearly outline the goals of the project, such as identifying malicious activities, detecting anomalies, and optimizing network performance.
* Identify Scope: Determine the specific network segments to be monitored, the types of traffic to analyse, and the duration of the analysis.
* Gather Resources: Assemble the necessary hardware and software, including computers for running Wireshark and Zeek, as well as any additional tools for data visualization and reporting.

**2. Network Traffic Capture**

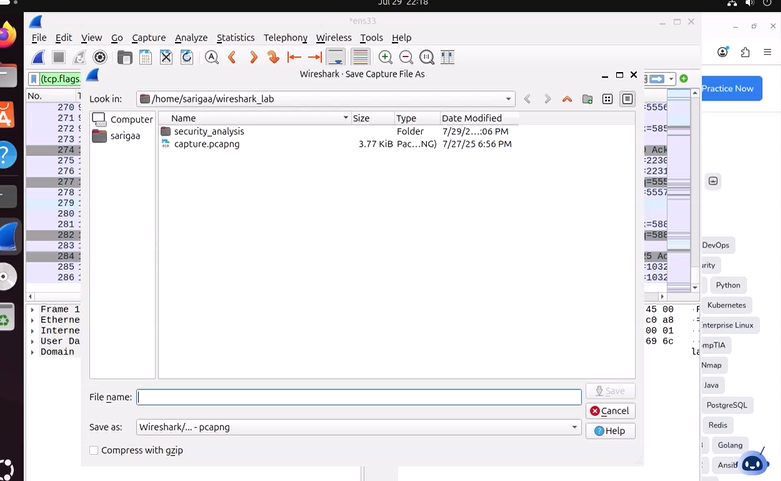
* Setup Wireshark: Install Wireshark on a designated machine within the network. Configure it to capture traffic from the appropriate network interface.

*********Fig.1.1-starting wireshark***

* Capture Traffic: Begin capturing live network traffic during peak and off-peak hours to gather a comprehensive dataset. This may involve running Wireshark for an extended period to capture various types of traffic.

***Fig.1.2-capturing traffic***

* Save Capture Files: Store the captured packets in PCAP format for later analysis, ensuring that the data is organized and easily accessible.



***Fig.1.3-saving pcap file***

**3. Traffic Analysis with Wireshark**

* Initial Inspection: Use Wireshark to perform an initial inspection of the captured packets. This includes examining packet headers, protocols, and payloads.
* Apply Filters: Utilize Wireshark’s filtering capabilities to focus on specific types of traffic, such as HTTP, DNS, or suspicious IP addresses. This helps in identifying anomalies or patterns of interest.

|  |  |  |
| --- | --- | --- |
| WIRESHARK DISPLAY FILTER | | PURPOSE |
| ip.addr==10.0.0.50 | | Show all traffic to or from a specific address |
| tcp.port==4444 | | Show traffic on tcp port 4444 |
| http/dns | Show only http web traffic/dns queries | |
| tcp.flags.syn==1&tcp.flags.ack==0 | Detect scans and SYN flood | |
| arp/arp.duplicate-address-detected | Reveal ARP spoofing,MITM attempts,or duplicate ips | |
| frame contains “password’ | Search for packets containing specific strings like passwords | |

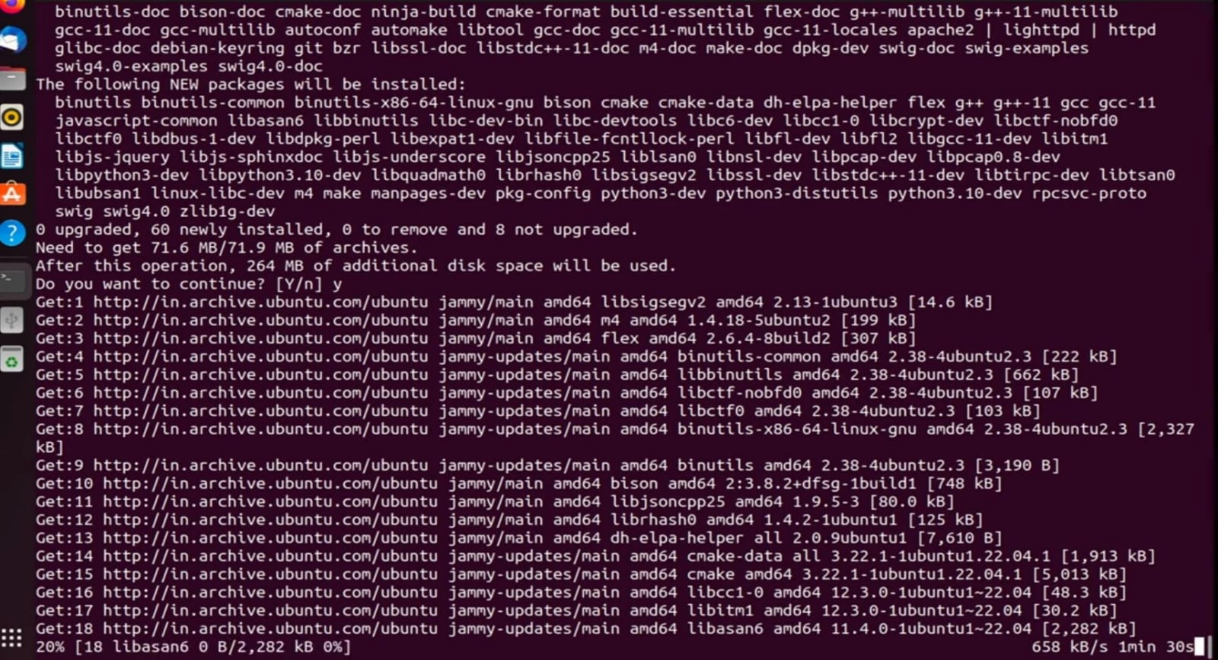
***Table 1.1-few wireshark display filters***

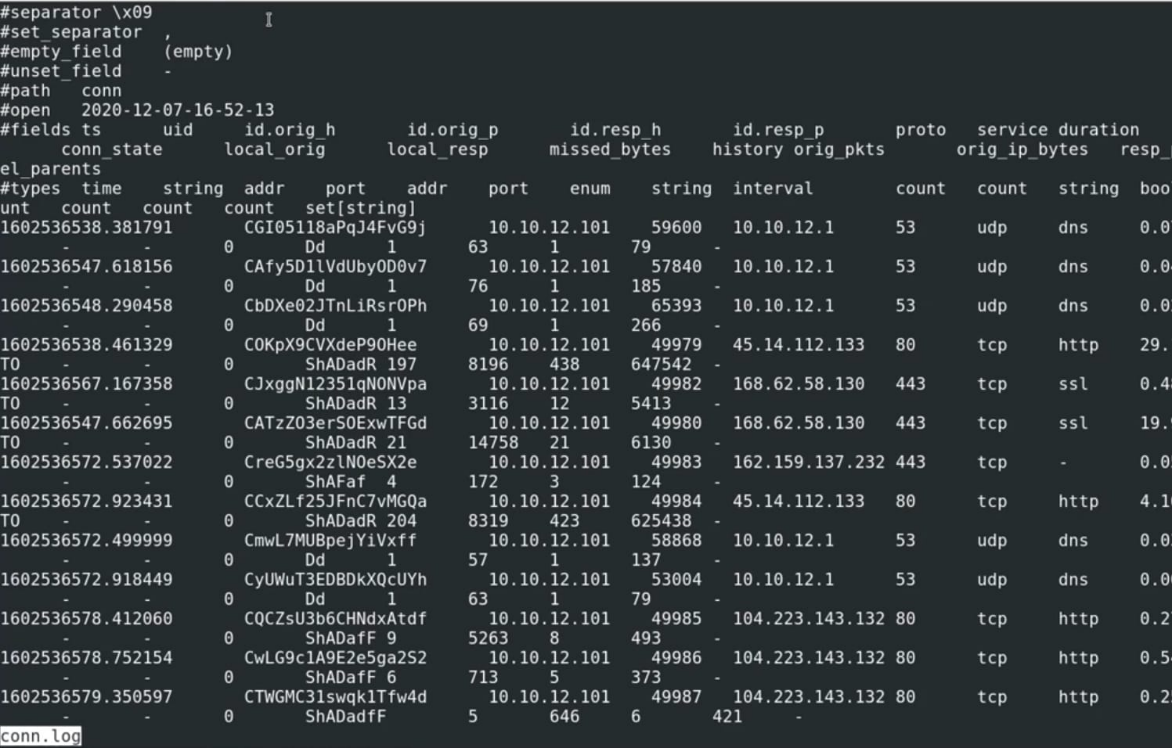
* Identify Anomalies: Look for unusual patterns, such as unexpected connections, high traffic volumes to specific ports, or repeated failed login attempts, which may indicate potential security threats.

4**. Real-Time Monitoring with Zeek**

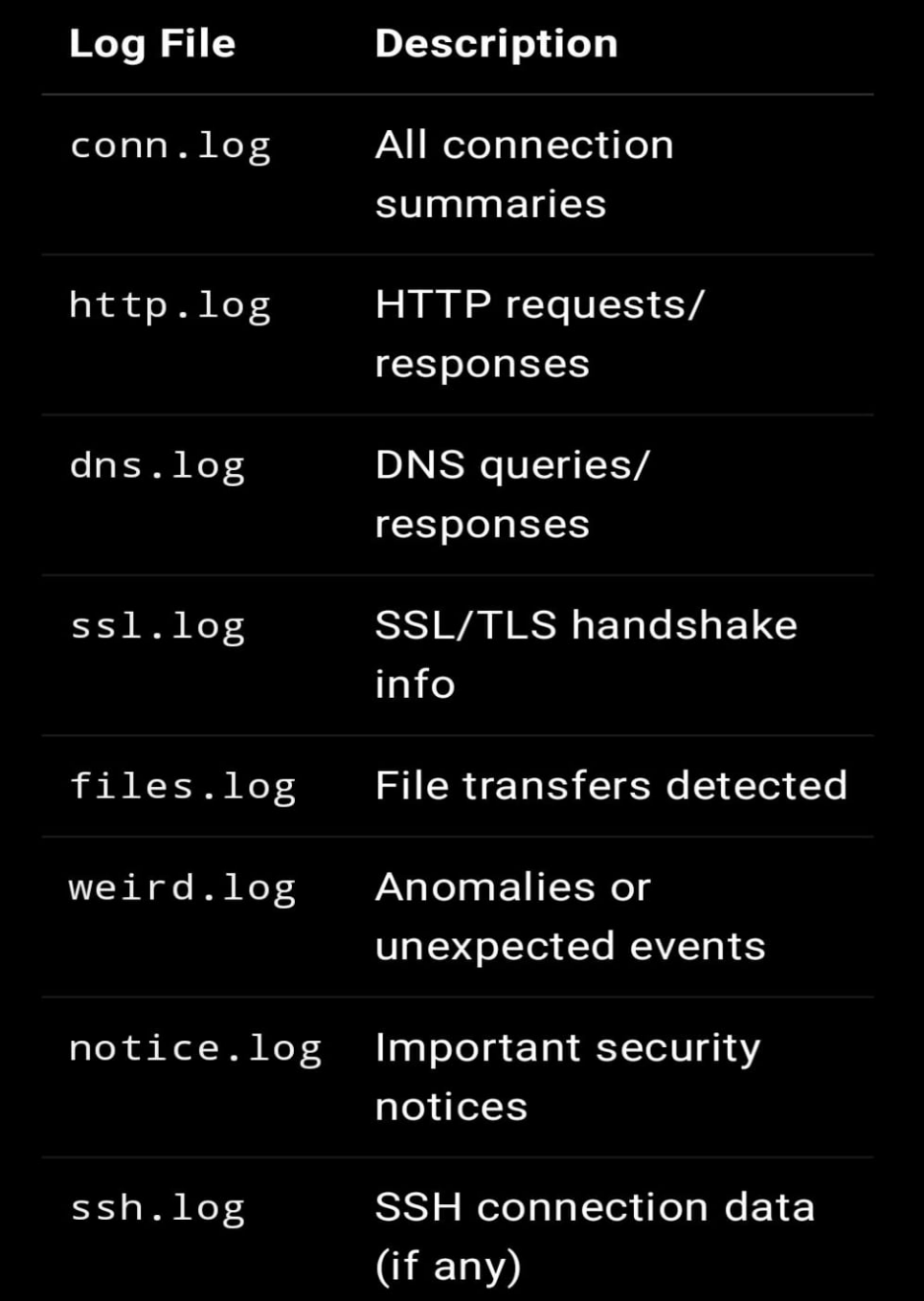
* Deploy Zeek: Install and configure Zeek on a separate machine or as a network appliance. Ensure it is set up to monitor the same network segment as Wireshark.

***Fig.1.4.-setting up zeek on ubuntu***

* Define Detection Policies: Utilize Zeek’s scripting language to define custom detection policies based on the specific security needs of the organization. This may include rules for detecting port scans, unusual traffic patterns, or known malicious signatures.



***Fig.1.5.analyzing pcap file in zeek***

* Monitor Traffic: Allow Zeek to analyse network traffic in real-time, generating logs and alerts for any detected anomalies or security incidents.

***Table.1.2-important log files***

**5. Reporting and Recommendations**

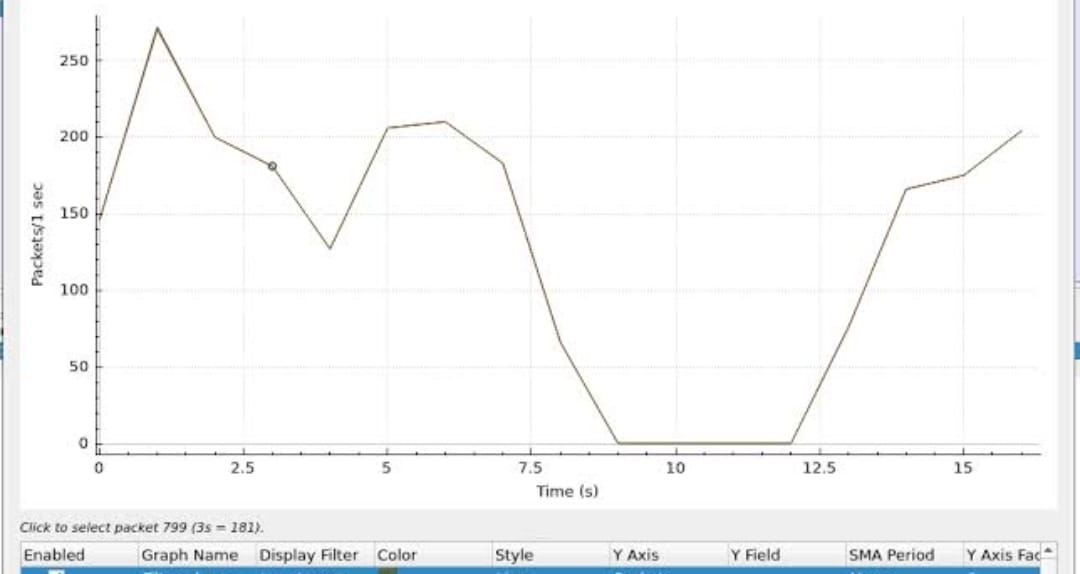
* Compile a Report: Create a detailed report summarizing the findings of the analysis. This should include an overview of detected threats, performance metrics, and any identified vulnerabilities.
* Provide Recommendations: Based on the analysis, offer actionable recommendations for improving network security and performance. This may include suggestions for configuration changes, additional monitoring tools, or employee training on security best practices.
* Present Findings: Share the report with relevant stakeholders, including IT and security teams, to ensure that the findings are understood and acted upon.

**6.Continuous Monitoring and Improvement**

* Establish Ongoing Monitoring: Recommend the implementation of continuous monitoring practices using Zeek and Wireshark to maintain network security and performance over time.
* Review and Update Policies: Regularly review and update detection policies and monitoring configurations based on evolving threats and network changes.

**RESULTS/DISCUSSIONS**

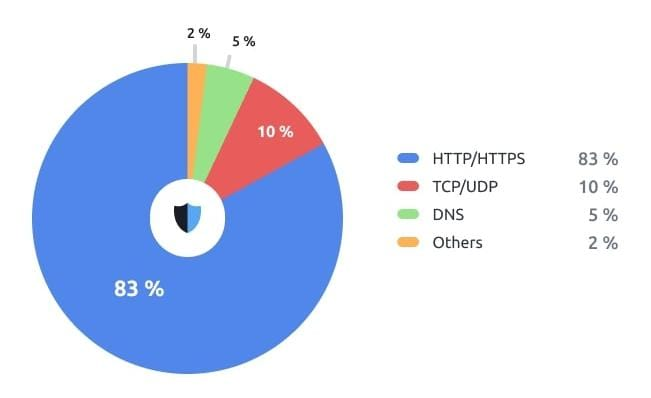
1. **Traffic Volume Analysis**

****

***Fig.2.1: Network Traffic Volume Over Time***

This graph shows the total volume of network traffic captured over a specified period. The x-axis represents time (in hours), while the y-axis indicates the total number of packets captured. The graph indicates peak traffic times, which may correlate with user activity or scheduled tasks within the network. Notably, there are spikes in traffic during certain hours, suggesting periods of high usage or potential automated processes.

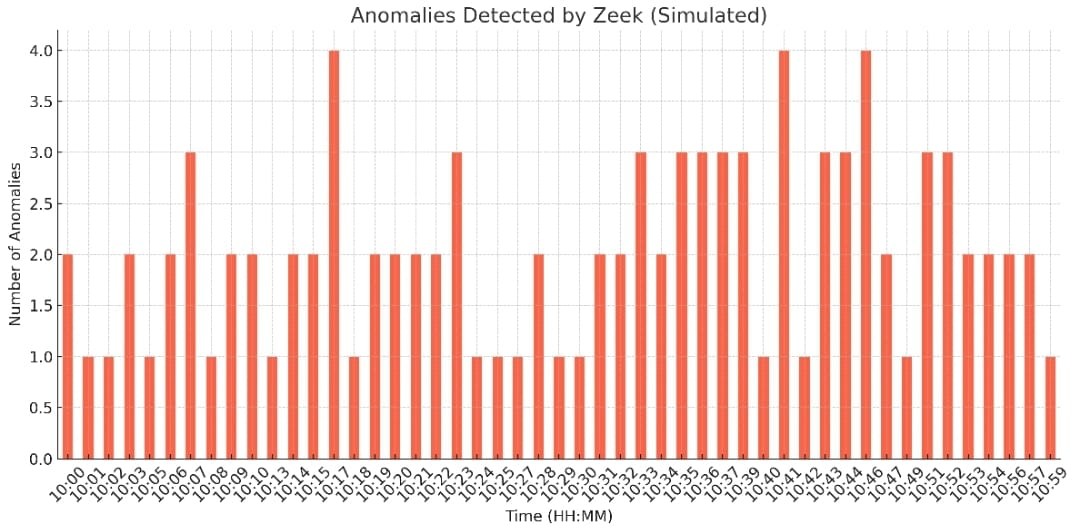
1. **Protocol Distribution**



***Fig2.2: Protocol Distribution in Captured Traffic***

This pie chart illustrates the distribution of different protocols observed in the captured traffic. The segments represent various protocols such as HTTP, DNS, TCP, and others. The analysis reveals that HTTP traffic constitutes the largest portion of the captured data, which is expected in a typical organizational environment. However, the presence of a significant amount of DNS traffic indicates active domain name resolution processes. Any unusual spikes in less common protocols could warrant further investigation.

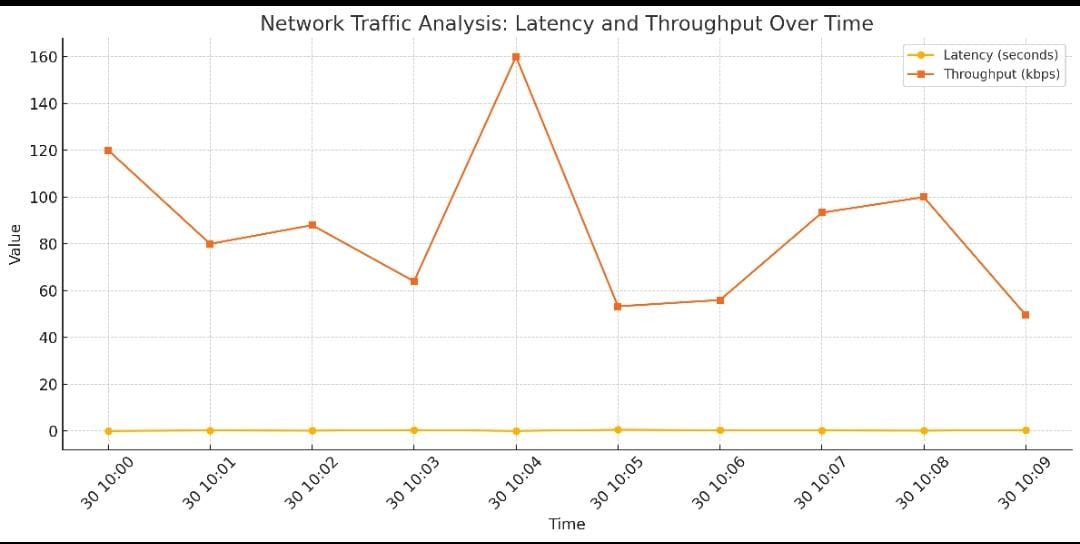
**3. Anomaly Detection Results**



**Fig 2.3: Detected Anomalies by Zeek**

This bar chart displays the number of anomalies detected by Zeek over the analysis period. Each bar represents a different type of anomaly, such as port scans, unusual connection attempts, and failed login attempts. The results show a notable number of port scan detections, which may indicate reconnaissance activities by potential attackers. The presence of failed login attempts suggests possible brute-force attacks. These findings highlight the importance of continuous monitoring and prompt response to mitigate potential threats.

**4.Performance Metrics**



**Fig.2.4: *Average Latency and Throughput***

The graph indicates that latency remains relatively stable, with occasional spikes that may correlate with high traffic volumes. Throughput shows a consistent increase during peak hours, suggesting that the network is handling increased load effectively. However, any significant latency spikes should be investigated further to ensure optimal performance.

The results of the network traffic analysis using Wireshark and Zeek provided valuable insights into both the security and performance aspects of the network. The identification of traffic patterns, protocol distributions, and detected anomalies underscores the importance of continuous monitoring and analysis.

**CONCLUSION**

In conclusion, this project successfully addressed the critical issues of network security and performance monitoring through the effective use of Wireshark and Zeek. By capturing and analysing network traffic, we were able to identify malicious activities, detect anomalies, and gain valuable insights into the overall health of the network. The integration of both tools provided a comprehensive approach to network traffic analysis, allowing us to correlate findings and enhance our understanding of network behaviour.

**Did your project solve the problem you aimed to solve?**  
Yes, the project effectively solved the problems we aimed to address. We successfully identified several instances of suspicious activity and potential security threats that would have otherwise gone unnoticed. Additionally, we gained insights into network performance, enabling us to recommend optimizations that could improve overall efficiency.

**What did you learn from the project?**  
Throughout the project, I learned the importance of combining different analytical tools to achieve a more holistic view of network security. The hands-on experience with Wireshark and Zeek deepened my understanding of network protocols and the intricacies of traffic analysis. I also gained valuable skills in data correlation and the significance of real-time monitoring in identifying and mitigating threats promptly.

**Future Work: If you had more time, how would you improve the project or take it further?**  
If given more time, I would enhance the project in a few ways:

1. **Expand the Scope**: I would extend the analysis to include additional network segments and devices, providing a more comprehensive view of the entire network infrastructure.
2. **Integrate Additional Tools**: Incorporating other security tools, such as intrusion detection systems (IDS) or security information and event management (SIEM) solutions, could further enhance threat detection and response capabilities.
3. **Automate Reporting**: Developing automated reporting mechanisms to generate real-time alerts and summaries would streamline the monitoring process and improve response times to incidents.

By pursuing these enhancements, the project could evolve into a more robust and comprehensive network security solution, ultimately contributing to a safer and more efficient digital environment.

**REFERENCES**

1. **Network Security Monitoring**  
   **Paxson, V. (1999).** *Bro: A System for Detecting Network Intruders in Real-Time*. Computer Networks, 31(23-24), 2435-2463. doi:10.1016/S1389-1286(99)00114-0
2. **Packet Analysis Techniques**  
   **Combs, G. (2019).** *Wireshark for Network Forensics*. In *Network Forensics: Tracking Hackers through Cyberspace* (pp. 45-67). Elsevier.
3. **Network Security Best Practices**  
   **Scarfone, K., & Mell, P. (2007**). *Guide to Intrusion Detection and Prevention Systems (IDPS)*. National Institute of Standards and Technology (NIST). Retrieved from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf
4. **Cybersecurity Threats and Trends**  
   **Verizon. (2023).** *2023 Data Breach Investigations Report*. Retrieved from https://enterprise.verizon.com/resources/reports/dbir/

**APPENDICES**

**Appendix A: Glossary of Terms**

* Packet: A formatted unit of data carried by a packet-switched network. It contains both control information and user data.
* PCAP: Packet Capture format, a file format used to save captured network traffic for analysis.
* Protocol: A set of rules governing the format and transmission of data over a network.
* Anomaly Detection: The identification of unusual patterns that do not conform to expected behaviour in network traffic.
* Intrusion Detection System (IDS): A device or software application that monitors network or system activities for malicious activities or policy violations.

**Appendix B: Tools and Software Used**

1. Wireshark
   * Version: 4.4.8
   * Download Link: https://www.wireshark.org/download.html
2. Zeek
   * Version: 7.2.2
   * Download Link: https://zeek.org/getting-started/
3. Operating System
   * Ubuntu
   * Version: 24.04 LTS

**Appendix C: Sample Captured Data**

* Sample PCAP File:/home/sarigaa/wireshark\_lab/security\_analysis
* Description: This file contains captured network traffic during the analysis period, including various protocols such as HTTP, DNS, and TCP