PROJECT REPORT

NETWORK TRAFFIC ANALYSIS USING WIRESHARK AND ZEEK

Submitted By

Dheerendra Patel

Submitted To

Nikhil Pandey

Department of Computer Science and Engineering

United College of Engineering and Research, Prayagraj

July 30, 2025

ABSTRACT

In today's digitally connected world, network security is a critical concern for organizations and individuals alike. Analyzing network traffic is a vital technique used to monitor, investigate, and secure networks against potential cyber threats. This project, titled "Network Traffic Analysis Using Wireshark and Zeek," focuses on using two open-source tools — Wireshark and Zeek — to capture and analyze real-time network data in order to detect suspicious patterns, unauthorized access, and potential attacks.

Wireshark is used to capture live packet data from network interfaces, providing a deep inspection of network protocols. Zeek, on the other hand, is employed for high-level traffic analysis by parsing .pcap files and generating structured log files for HTTP, DNS, and connection events. The combination of these tools allows for both low-level and high-level insight into the behavior of network traffic.

The project was implemented within a virtual machine environment running Ubuntu to ensure a secure testing platform. Network traffic was captured and analyzed for various parameters, including port scans, excessive DNS queries, and abnormal HTTP requests. Zeek's logs were further visualized using graphs and charts to interpret traffic patterns and anomalies.

This project not only enhances understanding of network protocols and threat detection but also builds practical cybersecurity skills. The findings demonstrate how real-world tools can help detect and analyze malicious activities, laying the foundation for automated network monitoring solutions.

TABLE OF CONTENTS

INTRODUCTION

In the modern digital era, networks serve as the backbone for communication, data transfer, and operations for individuals, businesses, and governments. With the growing reliance on internet-connected devices and online platforms, ensuring the security and reliability of computer networks has become a top priority. One of the fundamental approaches in cybersecurity is **Network Traffic Analysis**, which involves monitoring, recording, and analyzing data packets transmitted across a network to identify potential threats, intrusions, and abnormal behavior.

Network Traffic Analysis plays a vital role in identifying unauthorized access, detecting malware activities, and uncovering vulnerabilities within the network infrastructure. It is widely used by cybersecurity professionals to gain visibility into how data flows through systems and to determine whether malicious actors are attempting to exploit or disrupt services. Unlike traditional security measures such as firewalls and antivirus software, traffic analysis focuses on the real-time behavior of the network, offering deeper insights into activities that may otherwise go unnoticed.

This project, titled **"Network Traffic Analysis Using Wireshark and Zeek,"** aims to demonstrate how open-source tools can be effectively used to monitor and analyze network traffic in a controlled environment. The main objective is to detect anomalies such as port scanning, DNS tunneling, or suspicious HTTP requests that might indicate the presence of a threat. For this purpose, two powerful tools have been used — **Wireshark**, a packet sniffer that captures live traffic, and **Zeek** (formerly Bro), a network security monitor that analyzes traffic logs and generates high-level insights.

The project was carried out in a **virtual machine (VM)** setup using Ubuntu Linux to provide a safe, isolated, and reproducible environment for testing. Wireshark was used to capture network packets over live interfaces such as `eth0` or `wlan0`, allowing for inspection of protocols like HTTP, DNS, TCP, and UDP. These packets were saved in the `.pcap` format and later analyzed using Zeek, which generated a wide variety of log files including `conn.log` (connections), `dns.log` (DNS requests), `http.log` (web traffic), and `notice.log` (alerts or anomalies).

The reason for choosing this project is its **practical relevance and importance** in the field of cybersecurity. Organizations today face frequent cyber threats including reconnaissance, scanning, and data exfiltration attempts. Understanding how to capture and analyze network traffic is a core skill for any aspiring cybersecurity analyst, system administrator, or network engineer. By performing this project, students not only gain theoretical knowledge but also hands-on experience in using industry-standard tools that are widely adopted by security professionals around the world.

Moreover, tools like Wireshark and Zeek are **open-source and platform-independent**, making them ideal for academic and professional use. Wireshark provides a graphical interface to view and analyze packets in detail, allowing users to apply filters, examine

headers, and follow communication streams. Zeek, on the other hand, is a command-line tool that provides a different perspective by transforming captured traffic into structured logs that are easier to interpret, automate, and correlate over time.

In summary, this project bridges the gap between classroom knowledge and real-world application by teaching how to **detect and analyze network threats through traffic inspection**. It lays the foundation for advanced network security tasks like intrusion detection, incident response, and threat hunting. The combination of Wireshark and Zeek enables both low-level (packet) and high-level (log-based) traffic analysis, offering a comprehensive overview of what happens inside a network.

LITERATURE REVIEW

Network traffic analysis has been an essential area of research and practical application in the field of cybersecurity and digital forensics. It involves the systematic monitoring and examination of network data to detect anomalies, performance issues, or malicious activities. Several tools and techniques have been developed over the years to facilitate network traffic inspection, each with its unique approach to capturing and analyzing traffic.

Wireshark is one of the most widely used open-source tools for packet analysis. Originally known as Ethereal, Wireshark allows users to capture live network packets and analyze them in real-time or from previously saved capture files. It supports deep inspection of hundreds of protocols and provides a user-friendly graphical interface. Numerous academic and industry studies have emphasized Wireshark's effectiveness in understanding network behaviors, troubleshooting problems, and identifying protocol-level anomalies.

Zeek(formerly known as Bro) is another powerful open-source platform for network monitoring and analysis. Unlike Wireshark, Zeek does not focus on individual packet inspection; instead, it converts network traffic into high-level, structured logs that are easier to interpret and automate. Research studies and practical implementations have shown that Zeek is especially effective in detecting suspicious patterns over time, such as port scans, brute-force login attempts, and DNS tunneling. Its scripting capabilities allow customization of detection logic based on an organization's specific security policies.

Several researchers have compared Wireshark and Zeek, indicating that while Wireshark excels in low-level inspection and education, Zeek provides scalability and automation for larger networks. Both tools have been incorporated into Security Information and Event Management (SIEM) systems and are often used as part of incident response strategies in enterprises.

In recent years, there has been a growing trend in integrating traffic analysis tools with machine learning algorithms for anomaly detection. However, the foundational layer of all such advanced methods remains rooted in the accurate and thorough capture and parsing of network data—tasks where Wireshark and Zeek continue to serve as gold standards.

This literature review underlines the relevance and credibility of using Wireshark and Zeek in academic and practical cybersecurity projects, forming the technological base for the hands-on analysis performed in this study.

METHODOLOGY

This section explains the step-by-step approach used to execute the project “Network Traffic Analysis Using Wireshark and Zeek.” The goal was to capture real network traffic in a safe environment and analyze it for unusual activity such as port scans, DNS abuse, or suspicious HTTP requests.

1. Planning and Environment Setup

To perform network analysis safely, a virtual lab was created using Oracle VirtualBox. The guest operating system was Ubuntu 22.04 LTS. This Linux distribution supports both Wireshark and Zeek, and it allowed complete control over system behavior.

The virtual machine (VM) was configured with the following:

- RAM: 4 GB

- Processor: 2 cores

- Disk: 30 GB virtual disk space

- Network: Bridged or NAT adapter for internet access

This setup provided a safe and reproducible environment for capturing and analyzing traffic.

2. Tool Installation

Two tools were used:

• Wireshark – for packet capturing

• Zeek – for behavioral analysis of the captured packets

Installation was done using the following commands in the Ubuntu terminal:

- sudo apt update

- sudo apt install wireshark

- sudo apt install zeek

Non-root users were granted permission to capture packets during Wireshark installation. After installation, both tools were tested and confirmed working.

3. Capturing Network Traffic with Wireshark

Wireshark was launched with appropriate permissions. The process for capturing traffic included:

- Selecting the active network interface (eth0/wlan0)

- Starting live capture

- Simulating network traffic (e.g., browsing, pinging, downloading files)

- Applying display filters such as:

  - http

  - ip.addr == 192.168.x.x

  - tcp.port == 80

- Saving the captured packets in `.pcap` format

Wireshark's graphical interface allowed packet-level inspection, including IP addresses, protocols, headers, and payloads.

4. Analyzing Traffic with Zeek

The `.pcap` file was analyzed using the following command:

```
zeek -r capture.pcap
```

Zeek generated structured log files:

- conn.log → All connections

- dns.log → DNS queries

- http.log → Web requests

- notice.log → Warnings & detections

- weird.log → Unexpected or abnormal behavior

These logs provided high-level summaries of sessions, hosts, protocols, and anomalies.

5. Detection of Suspicious Activity

The logs were reviewed manually for anomalies. Key indicators included:

- Multiple short-lived connections from a single source IP → Port scanning

- Excessive DNS queries to random domains → DNS tunneling

- HTTP GET requests to suspicious domains → Possible C2 activity

Each behavior was cross-verified using timestamps, IP addresses, and connection metadata.

6. Visualization of Results

To better interpret the data, the logs were visualized using Excel and Python:

- Line graph → DNS query volume over time

- Bar chart → IP addresses with the most connections

- Pie chart → Traffic protocol distribution (TCP/UDP/DNS)

These visuals helped clearly represent the network behavior and supported the findings in a more intuitive format.

7. Tools and Technologies Used

Tool	Technology and Purpose
Wireshark	Capture and inspect raw network packets
Zeek	Generate structured logs from packet data
Ubuntu 22.04	Linux OS for the virtual lab
VirtualBox	Isolated and safe testing environment
Excel / Python	Graph plotting and visual analysis of logs

This structured approach ensured effective, safe, and insightful traffic analysis. The combination of Wireshark and Zeek allowed both packet-level and log-level understanding of network activities.

The following section will discuss the actual results obtained, including screenshots, charts, and observations from the log files.

RESULTS AND DISCUSSION

This section presents the outcomes of the network traffic analysis using Wireshark and Zeek. After capturing live network traffic in a controlled virtual environment and analyzing it with both tools, several observations were made. These include the detection of normal versus suspicious traffic, analysis of logs, and visualizations of specific behaviors like DNS queries and IP-based connections.

1. Packet Capture Using Wireshark

The network interface was monitored using Wireshark while common online activities were performed, such as visiting websites, downloading files, and DNS lookups. Wireshark successfully captured thousands of packets in real time. Filters like `http`, `dns`, and `tcp` were applied to isolate relevant traffic for analysis.

Key observations:

- Clear display of packet details (source/destination IPs, ports, protocols)

- HTTP GET requests to known and unknown URLs

- DNS queries to multiple domains in short time spans

- TCP 3-way handshakes and session terminations

The `.pcap` file containing the captured traffic was saved and later used for Zeek analysis.

2. Zeek Log Analysis

The `.pcap` file was processed using Zeek, which generated detailed logs including `conn.log`, `dns.log`, and `http.log`. These files offered a structured overview of connections and protocol-level activities.

conn.log observations:

- Multiple short-lived TCP connections from a single internal IP to various ports on a public IP address.

- Connections had low duration and byte count – common signs of port scanning.

dns.log observations:

- A high volume of DNS queries to random, non-standard domain names from one internal IP.

- Suggested potential DNS tunneling activity or a misconfigured client.

http.log observations:

- Detected multiple HTTP GET requests to uncommon or untrusted domains.

- URLs contained suspicious query strings – possible indicators of malware C2 communication.

notice.log:

- Zeek flagged a few sessions as "Potentially Malicious" based on its built-in detection rules.

These logs proved effective in understanding both regular and suspicious network behaviors.

3. Visualization of Findings

To support the analysis, visual charts were created based on Zeek logs:

- **Figure 1: DNS Query Volume Over Time**

  A line graph showing spikes in DNS requests, highlighting moments of unusual activity.

- **Figure 2: Top Source IPs by Connection Count**

  A bar chart representing which IP addresses initiated the most connections – helping identify scanning sources.

- **Figure 3: Protocol Usage Distribution**

  A pie chart showing the share of TCP, UDP, HTTP, and DNS traffic in the capture.

These graphs made it easier to interpret the raw log data and visually pinpoint abnormalities.

4. Interpretation of Suspicious Behavior

The combined analysis from Wireshark and Zeek revealed indicators of potential threats:

- **Port Scanning**: A large number of short connections to sequential ports, likely an attacker probing for open services.

- **DNS Abuse**: A surge in DNS requests from one client, often associated with tunneling or botnet check-ins.

- **HTTP Anomalies**: Requests to strange or suspicious URLs may represent malware communication.

All findings were manually verified to ensure accuracy and relevance. While these behaviors did not originate from actual malicious attacks (as this was a test environment), they mimicked real-world attack patterns.

5. Advantages of Dual Tool Use

Using both Wireshark and Zeek provided significant advantages:

- Wireshark offered low-level visibility into individual packets.

- Zeek provided high-level summaries and pattern detection capabilities.

- Together, they enabled a holistic view of the network, from detailed headers to behavioral trends.

6. Summary of Results

Observation Type	Tool Used	Key Insight
Port Scan Detection	Zeek	Found multiple short TCP sessions from single IP
DNS Abnormality	Zeek	Repeated queries to random domains
Suspicious HTTP URLs	Both	Accessed unknown external domains
Protocol Distribution	Zeek	HTTP and DNS were dominant

These results demonstrated how effective open-source tools can be in monitoring, detecting, and analyzing real-time network activity in a secure environment.

The next section concludes the project by summarizing key learnings and future scope of improvement.

CONCLUSION

This project successfully demonstrated how open-source tools like Wireshark and Zeek can be used effectively to analyze and monitor network traffic for security insights. Through real-time packet capturing and log-based traffic analysis, we gained a practical understanding of how normal and abnormal behaviors appear within network environments.

Wireshark provided a deep, packet-level view of the data flowing across the network, which helped in identifying the structure and nature of individual communication sessions. Zeek complemented this by offering high-level behavioral insights through structured log files, highlighting potentially malicious activities such as port scanning and DNS abuse.

By working in a controlled virtual machine environment, we simulated network traffic and detected suspicious patterns through both visual and log-based analysis. The integration of visual tools like Excel and Python further enhanced our ability to interpret large volumes of log data efficiently.

Overall, the project reinforced the importance of proactive network monitoring and threat detection. It also built essential skills in using industry-relevant tools, interpreting network logs, and understanding real-world cybersecurity challenges.

In the future, this project could be extended by incorporating machine learning models to automate threat detection based on patterns in network logs. Additionally, real-time alerting systems and integration with SIEM tools can make the solution even more robust for enterprise-level deployment.

RECOMMENDATIONS

Based on the experience and results obtained during the project, several recommendations can be made to further improve the effectiveness and scope of network traffic analysis using Wireshark and Zeek:

1. **Integration with Real-Time Alerting Tools**

While Zeek can log suspicious activity, integrating it with real-time alerting systems like ELK Stack (Elasticsearch, Logstash, Kibana) or SIEM solutions (e.g., Splunk, Wazuh) would provide instant notifications and dashboards for monitoring critical network events.

2. **Automating Threat Detection with Machine Learning**

Anomaly detection models can be trained using patterns extracted from Zeek logs to automate the identification of unknown threats. Machine learning algorithms can classify network behavior and flag outliers more efficiently than manual analysis.

3. **Expanding Log Analysis with Custom Zeek Scripts**

Zeek's scripting engine allows for the creation of custom detection policies. Writing and deploying custom scripts could enable detection of specific behaviors relevant to a given environment, such as brute-force login attempts or unauthorized file transfers.

4. **Scaling for Enterprise Networks**

For larger networks, capturing traffic on multiple interfaces and distributing analysis across several Zeek nodes can help manage high traffic volumes without performance loss.

5. **Continuous Learning and Simulation**

Regular use of simulated attack environments (like Kali Linux or Metasploit inside a VM) could improve learning and testing of detection capabilities, allowing a deeper understanding of both attacker behavior and defensive response.

6. **Policy Implementation Based on Findings**

The insights gained from this analysis can be used to inform firewall rules, DNS filtering, and internal policies to prevent specific types of traffic or access to malicious domains.

In summary, combining the strengths of Wireshark and Zeek with automation, alerting, and AI-powered analysis can lead to a powerful and scalable network security monitoring system. Future students and professionals are encouraged to explore these advanced capabilities for more robust cybersecurity defense.

REFERENCES

1. Wireshark Official Documentation. "Wireshark User Guide."

   Available at: https://www.wireshark.org/docs/

2. Zeek Network Security Monitor. "Zeek Documentation."

   Available at: https://docs.zeek.org/

3. Scarfone, K., & Mell, P. (2007). "Guide to Intrusion Detection and Prevention Systems (IDPS)."

   National Institute of Standards and Technology (NIST), Special Publication 800-94.

4. Bejtlich, R. (2005). "The Tao of Network Security Monitoring: Beyond Intrusion Detection."

   Addison-Wesley Professional.

5. Anderson, J. P. (1980). "Computer Security Threat Monitoring and Surveillance."

   James P. Anderson Company.

6. Open Security Training. "Packet Analysis and Network Monitoring Courses."

   Available at: https://www.opensecuritytraining.info/

7. Orebaugh, A., Ramirez, G., & Beale, J. (2007). "Wireshark & Ethereal Network Protocol Analyzer Toolkit."

   Syngress Publishing.

8. Zeek Scripts and Tutorials – GitHub Repository.

   Available at: https://github.com/zeek/zeek

9. Conti, G. (2007). "Security Data Visualization: Graphical Techniques for Network Analysis."

   No Starch Press.

10.Various Articles and Blogs on Medium and TowardsDataScience related to Zeek and Wireshark.