**Network Traffic Analysis using Wireshark And Zeek.**

**Name :** Krishnendra Kumar

**Institution Name :** United college of engineering and research Allahabad

**Course :** B.Tech

**Date :** 31/07/2025

**Supervisor :-** Mr. Ayush Kumar

**Abstract**

The goal of this project is to use two popular tools—Wireshark and Zeek—to monitor and analyze network traffic. By recording live traffic and examining trends, the main objective is to spot odd behavior or possible security risks in network activities. While Zeek makes high-level protocol analysis and event correlation possible, Wireshark aids in deep packet examination. As part of the project, live network traffic had to be recorded, unnecessary data had to be filtered out, communication patterns had to be examined, and possible abnormalities had to be found. A deeper comprehension of network behavior, the identification of particular traffic patterns, and the detection of potential scanning or intrusion efforts are the results. The significance of proactive network monitoring in cybersecurity is emphasized by this effort.

**Table of content**

1. Title Page
2. Abstract
3. Table of Contents
4. Introduction
5. Literature Review
6. Methodology/Approach
7. Results and Discussion
8. Conclusion
9. Recommendations
10. References
11. Appendices

**Introduction**

An essential component of cybersecurity is the examination of network traffic. Given the rise in cyberthreats, network behavior monitoring has become crucial for early attack detection. The goal of this project is to use Wireshark and Zeek to examine real-time network data in order to find odd or perhaps hostile activity. Zeek facilitates protocol-based analysis and event correlation, whereas Wireshark permits packet-level inspection. Gaining practical expertise in network monitoring and threat detection—two essential competencies for every cybersecurity professional—is the driving force behind this initiative.

**Literature Review**

A considerable body of work supports the use of network traffic analysis tools in cybersecurity. Tools like Wireshark and Zeek are widely recommended in academic and industry contexts.

Wireshark, due to its user-friendly GUI, is extensively used for protocol learning and packet-level inspection. Research papers emphasize its role in teaching and forensics. For example, "Packet-Level Network Analysis with Wireshark" (IEEE, 2017) discusses how Wireshark helps visualize network flows and protocol structures.

Zeek, formerly Bro, is praised for its flexibility and high-level insights. Studies like "Anomaly Detection Using Zeek in Enterprise Networks" show its capability to generate logs and detect suspicious activity at scale. Zeek's scripting language makes it suitable for writing custom policies and detectors.

Together, these tools cover a wide spectrum—from raw data collection to contextual behavior analysis—making them ideal for security analysts and students alike.

**Methodology**

**Approach:**

We aimed to create a controlled environment to monitor, capture, and analyze network traffic using Wireshark and Zeek. The focus was on capturing HTTP, DNS, and TCP streams and detecting any unusual behavior.

**Tools and Technologies Used:**

* **Wireshark:**

GUI-based network protocol analyzer.

* **Zeek:**

CLI-based network analysis framework that produces structured logs.

* **VirtualBox + Kali Linux + Windows VM:**

Simulated internal traffic.

* **PCAP files:**

Used for replay and analysis.

**Step-by-Step Process:**

Environment Setup:

Installed Kali Linux and Zeek on a virtual machine.

Installed Wireshark on the host system.

Created a simulated internal network using NAT settings.

Traffic Capture with Wireshark:

Launched Wireshark and captured real-time network traffic.

Applied filters like http, dns, tcp.port==80 for protocol-specific capture.

Exported traffic to .pcap files.

Analysis in Wireshark:

Followed TCP streams to analyze conversations.

Identified unusual packet sizes, retransmissions, and malformed packets.

Used statistics → Protocol Hierarchy to summarize traffic types.

Zeek Log Generation:

Ran Zeek against captured .pcap files using command:

zeek -r capture.pcap

Analyzed conn.log, http.log, dns.log, and weird.log.

Parsed results using tools like zeek-cut.

Anomaly Detection:

Detected DNS tunneling patterns from long TXT queries.

Logged potential scanning behavior from multiple TCP SYN requests.

Observed failed login attempts in HTTP logs (possible brute-force).

**Results and Discussion**

**Results:**

* **Wireshark:**
  + Detected multiple TCP retransmissions to a specific IP.
  + Identified malformed HTTP headers indicating poorly constructed requests.
  + Visualized packet timing using IO graphs.
* **Zeek:**
  + conn.log: Showed frequent short-lived connections (scan-like).
  + dns.log: Logged an unusually high number of TXT-type DNS queries.
  + weird.log: Detected non-standard HTTP methods like “BREW”.
  + http.log: Tracked unusual user agents and host headers.

**Discussion:**

The use of both tools showed how low-level and high-level analysis can reveal different facets of the same traffic. Wireshark helped inspect individual packets, while Zeek provided a broader picture through logs. Challenges included handling large capture files and filtering out background noise (broadcast/multicast traffic).

**Conclusion**

The project successfully demonstrated that both Wireshark and Zeek serve critical yet distinct roles in network traffic analysis. Wireshark excels in real-time packet inspection, which is valuable for learning and quick troubleshooting. Zeek, with its structured log output, provides deeper behavioral insights.

Through the combined use of these tools, we were able to detect potentially malicious activities such as scanning attempts, DNS tunneling, and anomalous HTTP behavior. This validates the need for layered analysis when securing networks.

**Key Learnings:**

* Interpreting live and saved network traffic.
* Generating and analyzing Zeek logs.
* Applying filters to extract relevant data.

**Future Work:**

* Integration with SIEM platforms like ELK or Splunk.
* Using Zeek scripts for automated alerting.
* Real-time deployment in enterprise networks for intrusion detection.

**Recommendations**

* Small organizations can deploy Zeek on their gateway to monitor internal traffic.
* Use Wireshark for incident response and packet-level investigation.
* Regularly monitor DNS traffic to detect tunneling behavior.
* Combine both tools with a SIEM for better visibility and alerting.

**References**

1. Wireshark Official Docs - https://www.wireshark.org/docs/
2. Zeek Network Security Monitor - https://docs.zeek.org/
3. Scarfone, K., & Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (NIST).
4. Paxson, V. (1999). Bro: A System for Detecting Network Intruders in Real-Time.
5. YouTube and GitHub tutorials for Zeek + Wireshark.

**Appendices**

**Appendix A: Sample conn.log from Zeek:**

1521911720.622501 CgJHgz2mGy 192.168.1.100 49162

93.184.216.34 80 tcp http 0.002685 281 1099 SF ...

**Appendix B: Wireshark Filter Commands**

* http.request.method == "GET"
* tcp.analysis.retransmission
* dns.qry.type == 16 (for TXT)