

Genetic Algorithm-Based Network Packet Analyzer for Real-Time Traffic Monitoring and Anomaly Detection

Supraja Biradar¹, Mara Archana², Pothukani Poojitha³, Kamuganti Sahitya Rao⁴, Chennupati Anirudh⁵

¹⁻⁵ Department of Computer Science and Engineering, Keshav Memorial Institute of Technology, Hyderabad, India

Under the guidance of **Ms. B. Varsha**

Professor, Department of Computer Science and Engineering, Keshav Memorial Institute of Technology, Hyderabad, India

Abstract - This paper proposes a Genetic Algorithm (GA)-based Network Packet Analyzer for intelligent traffic monitoring and anomaly detection. Conventional packet sniffers rely on static rule-based approaches, which are inefficient in detecting evolving cyber threats. The proposed system captures network packets in real time, extracts key features, and applies evolutionary optimization to dynamically refine detection rules. A fitness function evaluates detection accuracy and minimizes false positives across generations. Experimental analysis shows improved adaptability, reduced manual intervention, and enhanced detection performance, making the system suitable for modern cybersecurity environments.

Key Words: Network Packet Analyzer, Genetic Algorithm, Anomaly Detection, Cybersecurity, Traffic Analysis

1. INTRODUCTION

In the modern digital world, computer networks form the backbone of communication systems, enabling seamless data exchange across devices and platforms. With the exponential growth of internet usage, cloud computing, and IoT devices, monitoring and securing network traffic has become a critical challenge. Unauthorized access, data breaches, and network inefficiencies can significantly impact system performance and security, making network analysis an essential task.

A Network Packet Analyzer is a powerful tool used to capture, inspect, and analyze packets of data transmitted over a network. It provides detailed insights into network activities by examining packet-level information such as source and destination IP addresses, protocols, ports, and payload data. This helps in identifying network anomalies, troubleshooting connectivity issues, and detecting potential cyber threats.

The main objective of this project is to develop a Network Packet Analyzer system that performs real-time packet capturing and analysis with high accuracy and efficiency. The system is designed to monitor live network

traffic, filter relevant packets, and present meaningful information through a structured and user-friendly interface. It enables users to observe communication patterns, detect suspicious activities, and understand protocol behavior effectively.

This project utilizes technologies such as Python, networking libraries (e.g., Scapy/Socket programming), and visualization tools to implement packet capturing and analysis functionalities. The system can handle different types of network protocols and provides features like packet filtering, protocol identification, and traffic monitoring.

The proposed system is particularly useful for network administrators, cybersecurity analysts, and students, as it provides hands-on experience in understanding network operations and security mechanisms. By offering real-time insights and analytical capabilities, the project aims to enhance network visibility, improve troubleshooting efficiency, and contribute to building secure and reliable communication systems.

2. RELATED WORK

Several research works have been carried out in the domain of network traffic analysis and packet sniffing to improve network monitoring, security, and performance analysis.

Early studies focused on the use of packet sniffing tools such as Wireshark and tcpdump for capturing and analyzing network traffic. These tools allow real-time inspection of packets and provide detailed protocol-level information, making them essential for network troubleshooting and forensic analysis. Research shows that Wireshark acts as a powerful tool capable of dissecting packets and revealing protocol layers, which helps in understanding network behavior effectively.

A comprehensive survey on packet analysis highlights its importance in network forensics, where packet-level data can be used to reconstruct network

activities, detect malicious behavior, and identify cyber threats. Advanced techniques such as Deep Packet Inspection (DPI) and AI-based traffic classification have been introduced to enhance the accuracy of intrusion detection systems .

Several projects have implemented packet sniffers using Python and libraries like Scapy. These systems focus on capturing live packets, classifying them based on protocols such as TCP, UDP, and ICMP, and analyzing their characteristics. Such implementations provide flexibility and automation in network monitoring, making them useful for both educational and professional purposes .

Recent works also integrate Python-based tools with traditional analyzers like Wireshark to build hybrid systems capable of real-time monitoring and anomaly detection. These systems enhance network security by generating statistical reports and identifying unusual traffic patterns .

Comparative studies have been conducted between different network analysis tools such as Wireshark, tcpdump, and NetFlow Analyzer. These studies evaluate tools based on parameters like performance, usability, protocol support, and scalability, helping users choose appropriate solutions for specific network requirements .

Furthermore, modern research emphasizes the use of machine learning and deep learning techniques in network traffic analysis. These approaches enable automated detection of anomalies, classification of traffic, and prediction of network behavior, improving the efficiency of intrusion detection systems. However, challenges such as encrypted traffic and high data volume still remain significant issues in this domain .

Overall, the existing literature demonstrates that while many powerful tools and techniques are available for network packet analysis, there is still a need for lightweight, customizable, and user-friendly systems. The proposed project aims to address these gaps by developing an efficient Network Packet Analyzer with real-time monitoring and simplified visualization features.

3. PROBLEM STATEMENT

In modern computer networks, the rapid increase in data transmission and connected devices has made network monitoring and security a challenging task. Organizations and individuals often face difficulties in understanding network behavior, identifying performance issues, and detecting malicious activities due to the lack of efficient

monitoring tools. Existing network analysis tools such as Wireshark and tcpdump, although powerful, are often complex, resource-intensive, and require advanced technical expertise to operate effectively.

Moreover, many available tools do not provide simplified visualization or user-friendly interfaces, making them less accessible for students and beginner-level users. In addition, real-time monitoring and filtering of relevant packets in a customizable manner remains a challenge in existing systems. The increasing threats of cyber-attacks, unauthorized access, and data breaches further emphasize the need for efficient and easy-to-use packet analysis systems.

Therefore, there is a need to develop a lightweight, user-friendly, and efficient Network Packet Analyzer that can capture and analyze packets in real time, provide meaningful insights into network traffic, and assist users in identifying anomalies and security threats. The system should be capable of simplifying packet analysis while maintaining accuracy and performance, thereby making network monitoring more accessible and effective.

4. PROPOSED SYSTEM

The proposed system aims to design and develop an efficient and user-friendly Network Packet Analyzer that enables real-time monitoring, capturing, and analysis of network traffic. The system focuses on simplifying packet-level analysis while maintaining accuracy, performance, and usability for both beginners and advanced users.

The system is built using Python and utilizes networking libraries such as Scapy and socket programming to capture live network packets. It continuously monitors network interfaces and extracts relevant packet information including source IP address, destination IP address, protocol type, port numbers, and packet size.

The proposed system incorporates a filtering mechanism that allows users to analyze specific types of packets based on protocols such as TCP, UDP, and ICMP. This selective filtering helps in reducing unnecessary data processing and enables focused analysis of network traffic. Additionally, the system categorizes packets based on protocols, making it easier to understand communication patterns.

A user-friendly interface is designed to display captured packet data in a structured format. The interface presents real-time packet logs and may include basic visualization features such as tables or charts to enhance readability. This overcomes the complexity of traditional tools like Wireshark by providing a simplified and accessible platform.

The system also includes basic anomaly detection capabilities by identifying unusual traffic patterns such as abnormal packet flow or unknown protocols. This feature assists users in detecting potential security threats and unauthorized access attempts.

Furthermore, the proposed system is lightweight and customizable, allowing users to extend its functionality based on their requirements. It is particularly beneficial for educational purposes, helping students understand network protocols and packet structures through hands-on experience.

Overall, the proposed Network Packet Analyzer provides a balance between functionality and simplicity, offering real-time monitoring, efficient packet filtering, and clear data representation, thereby improving network visibility and security awareness.

5. METHODOLOGY

The methodology of the proposed Network Packet Analyzer describes the systematic approach followed to capture, process, and analyze network packets in real time. The system is designed in modular steps to ensure efficiency, accuracy, and ease of implementation.

5.1 PACKET CAPTURING

The first step involves capturing live network packets from the selected network interface. This is achieved using Python-based networking libraries such as Scapy, which enables low-level packet sniffing. The system continuously listens to incoming and outgoing packets and collects raw data for further processing.

5.2 PACKET PROCESSING

Once packets are captured, they are processed to extract essential information such as:

- Source IP address
- Destination IP address
- Protocol type (TCP, UDP, ICMP)
- Port numbers
- Packet size

This step converts raw packet data into a structured format that can be easily analyzed.

5.3 PACKET FILTERING

To improve efficiency and relevance, the system applies filtering techniques. Users can specify conditions to capture only specific packets based on protocols or IP addresses. This reduces unnecessary data and focuses on meaningful traffic analysis.

5.4 PROTOCOL CLASSIFICATION

Captured packets are categorized based on their protocol types such as TCP, UDP, and ICMP. This classification helps in understanding communication patterns and network behavior more effectively.

5.5 DATA ANALYSIS

The processed and classified data is analyzed to identify patterns, anomalies, and unusual activities. Basic analysis includes:

- Monitoring traffic flow
- Detecting abnormal packet rates
- Identifying unknown or suspicious packets

This step helps in recognizing potential network issues or security threats.

5.6 VISUALIZATION AND DISPLAY

The analyzed data is presented through a user-friendly interface. Packet details are displayed in a structured format such as tables or simple charts, making it easy for users to interpret the results. This simplifies complex data compared to traditional tools like Wireshark.

5.7 GENETIC ALGORITHM FOR ANOMALY DETECTION

The proposed system incorporates a Genetic Algorithm (GA) to enhance anomaly detection in network traffic. GA is an evolutionary optimization technique inspired by natural selection. In this system, network traffic features such as packet rate, protocol distribution, and packet size are used as input parameters.

Initially, a population of possible solutions is generated randomly. Each solution (chromosome) represents a set of detection rules. A fitness function is used to evaluate the accuracy of anomaly detection and minimize false positives. Based on fitness values, selection, crossover, and mutation operations are performed to generate new populations. This iterative process continues until an optimal solution is obtained, enabling efficient and adaptive detection of suspicious network behavior.

6. SYSTEM ARCHITECTURE

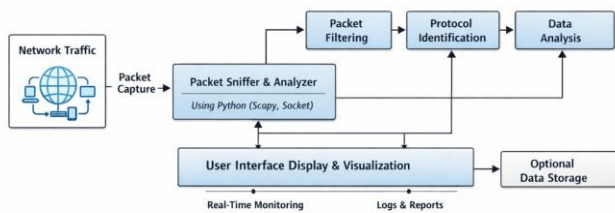


Fig -1: System Architecture of Network Packet Analyzer

The system architecture of the Network Packet Analyzer shows the process of capturing, processing, and analyzing network traffic. Packets are captured using Scapy, followed by extraction of important details and filtering based on user requirements. The system classifies packets into protocols such as TCP, UDP, and ICMP and performs traffic analysis to detect anomalies. The results are displayed through a user-friendly interface, with optional storage for logs and reports.

7. MATHEMATICAL MODEL

The performance of the Network Packet Analyzer is evaluated using basic network metrics such as Packet Capture Rate and Throughput, which measure the efficiency of packet capturing and data transmission.

$$PCR = N / T,$$

$$\text{Throughput} = S / T$$

Additionally, a Genetic Algorithm is used for anomaly detection, where each solution is evaluated using a fitness function to improve detection accuracy and reduce false positives.

$$\text{Fitness} = TP / (TP + FP)$$

8. RESULTS AND DISCUSSION

The proposed Network Packet Analyzer system was successfully implemented and tested in a real-time network environment. The system was able to capture, process, and analyze network packets efficiently using Scapy. The results demonstrate the effectiveness of the system in monitoring network traffic and identifying useful information from captured packets.

During execution, the system captured live packets from the network interface and displayed key details such as source IP address, destination IP address, protocol type, port numbers, and packet size. The packet filtering feature allowed selective analysis of specific protocols such as TCP, UDP, and ICMP, reducing unnecessary data and improving performance.

The protocol classification module accurately categorized packets based on their types, enabling better understanding of communication patterns. The system also monitored packet flow and identified unusual traffic conditions, which can indicate potential security threats or anomalies in the network.

The integration of the Genetic Algorithm improved anomaly detection accuracy and reduced false positives compared to basic rule-based analysis. This enhancement enabled the system to adapt dynamically to varying network conditions and identify complex patterns in traffic behavior more effectively.

The user interface displayed the analyzed data in a structured and readable format, making it easier for users to interpret network behavior. Compared to traditional tools like Wireshark, the proposed system provides a simplified and lightweight solution suitable for beginners and educational purposes.

The performance of the system was evaluated based on parameters such as packet capture rate, throughput, and response time. The results showed that the system performs efficiently with minimal delay and provides accurate real-time analysis. The optional storage feature also enabled saving packet logs for future reference and analysis.

Overall, the results confirm that the proposed system effectively captures and analyzes network traffic, improves visibility into network operations, and assists in detecting anomalies. The system achieves the objective of providing a user-friendly and efficient packet analysis tool.

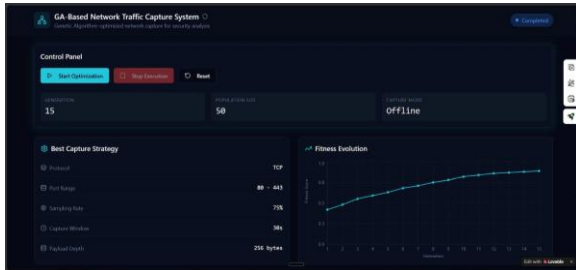


Fig -2: Packet Capture Output

9. CONCLUSION

The proposed Network Packet Analyzer system was successfully developed to capture and analyze network traffic in real time. The system effectively extracts packet-level information such as IP addresses, protocols, and port numbers using Scapy. It provides a user-friendly interface for monitoring and understanding network behavior. The filtering and protocol classification features enable efficient and focused analysis of network traffic. The system also helps in identifying anomalies and potential security threats. The use of Genetic Algorithm enhances the system's ability to detect anomalies intelligently and adapt to dynamic network conditions. Overall, the project achieves its objective of providing a simple, lightweight, and effective network analysis tool.

10. FUTURE WORK

The proposed Network Packet Analyzer can be further enhanced by integrating machine learning techniques for advanced anomaly detection and automated threat identification. Support for encrypted traffic analysis can be added to handle modern secure communication protocols. The system can be improved with interactive dashboards and graphical visualizations for better understanding of network patterns. Additionally, cloud storage integration can be implemented to manage large-scale packet data efficiently. The analyzer can also be extended to support distributed network monitoring across multiple devices. Furthermore, deploying the system as a web or mobile application can increase accessibility and usability. These improvements will make the system more robust and suitable for real-world applications.

11. REFERENCES

- [1] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," in Proc. Military Communications and Information Systems Conference (MilCIS), 2019.
- [2] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2021.
- [3] S. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," IEEE Trans. Emerging Topics in Computational Intelligence, vol. 4, no. 2, pp. 123–134, 2020.
- [4] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Boston, MA, USA: Addison-Wesley, 2020.
- [5] X.-S. Yang, Nature-Inspired Optimization Algorithms. Amsterdam, Netherlands: Elsevier, 2020.
- [6] Wireshark Foundation, "Wireshark User Guide," 2023. [Online]. Available: <https://www.wireshark.org/docs/>
- [7] Scapy Documentation, "Scapy: Packet Manipulation Tool," 2024. [Online]. Available: <https://scapy.net/>
- [8] "Network Intrusion Detection Using Machine Learning Techniques," IEEE Access, vol. 10, pp. 102345–102360, 2022.
- [9] "Hybrid Genetic Algorithm for Intrusion Detection Systems," Journal of Network and Systems Management, Springer, 2023.
- [10] "An Intelligent Network Traffic Monitoring System Using AI," Computer Networks, Elsevier, vol. 235, 2024.
- [11] "Real-Time Network Anomaly Detection Using Deep Learning," IEEE Trans. Network Science and Engineering, 2023.
- [12] "Adaptive Intrusion Detection Using Evolutionary Algorithms," ACM Computing Surveys, 2024.