Network Traffic Sniffer and Analyzer.

G SHANMUGARATHINAM
Professor
Dept. of Computer Science Engineering
Presidency University
Rajankunte, India
shanmugarathinam@presidencyuniversity.in

AISHWARYALAKSHMI D
Dept. of Information Science and Technology
Presidency University
Rajankunte, India
aishwaryadayalan2004@gmail.com

TASMIYA
Dept. of Information Science and Technology
Presidency University
Rajankunte,India
sgtasmiya89@gmail.com

Y VARSHANTH
Dept. of Information Science and Technology
Presidency University
Rajankunte, India
yvarshanth@gmail.com

***Abstract*—Network traffic sniffers are indispensable tools for debugging, monitoring, and securing today's networks handling enormous amounts of heterogeneous and often encrypted data. This paper presents the design and implementation of an AI-enhanced Network Traffic Sniffer and Analyzer that goes beyond traditional packet capture by featuring flow-based feature extraction and machine-learning-based anomaly detection. The proposed framework employs the Python-based libraries like Scapy and PyShark for high-fidelity packet capture, aggregates the captured traffic into statistical flow records, and uses a trained Random Forest classifier, in real time, to detect suspicious or malicious activity. In aid of fast operator response, the system features an interactive, web-based visualization dashboard that indicates anomalous traffic patterns and source endpoints. Experiments on a hybrid test dataset that consists of laboratory-generated traffic and public-domain intrusion-detection benchmarks demonstrate overall detection accuracy of ≈92%, low false-positive level, and sub-3 ms processing latency per packet. These results demonstrate that the enhancement of traditional sniffing with intelligent analytics enhances both situational awareness and proactive defense of enterprise networks.**

I. INTRODUCTION

With the advent of cloud computing, Internet of Things (IoT), and high-speed 5G networks, the size and complexity of network traffic have grown exponentially. Businesses rely on secure and dependable data transmission as a basis for business continuity, service provision, and regulative requirements. But all this increased reliance on connected systems also creates increased vulnerabilities to cyber-threats like distributed denial-of-service (DDoS) attacks, ransomware campaigns, data exfiltration, and insider threats. Detection as close as possible in time to the commencement of anomalous or malicious traffic is thus paramount in preserving both performance as well as security.

Network traffic sniffer is a type of tool that sits passively on a communication link to capture data packets moving across a wired or wireless network. Historical sniffers like tcpdump and Wireshark have been employed for decades in the work of troubleshooting, performance profiling, and forensic investigations. However, such tools are frequently limited by the need for manual inspection and signature-based methods. With the advancing nature of today's attacks—utilizing zero-day exploits, encrypted tunnels, and polymorphic malwares—there is increasing demand for automated, savvy, and scaleable traffic analysis systems.

Latest developments in machine learning (ML), deep learning, and big-data analytics offer new possibilities in augmenting the output of classical sniffers. By being trained on vast amounts of traffic, traffic analyzers that are based on ML can discover subtle imperfections in packet stream flow patterns that could be symptoms of novel attacks. Incorporating these methods into a sniffer facilitates real-time anomaly detection, proactive alerting, and optimal prioritization of network events, all decreasing the need on human analysts.

The final goal of the present work is the conceptualization and realization of a Network Traffic Sniffer and Analyzer that unites packet capture, flow-level feature extraction, machine-learning-based detection, and visualization under one common framework. Here are the contributions of the present work summarized:

1.A lightweight packet-capture module based on Python libraries such as Scapy and PyShark, capable of live monitoring on enterprise-grade networks.

2.A highly insightful analysis engine that utilizes flow-based statistical features and a Random Forest model in identifying anomalous activity and suspicious activity.

3.ldkfAn interactive dashboard that provides useful information and is easily interpreted by network administrators.

4.Large-scale testing on public datasets as well as live traffic labs developed as methods for testing detection accuracy, latency, and scalability.

By addressing the limitations of traditional sniffers and demonstrating the benefits of hybrid AI-driven approaches, this study contributes to the advancement of next-generation network monitoring tools that are both effective and practical for real-world deployment.

II. LITERATURE SURVEY

Network traffic analysis and monitoring have been heavily researched in order to maintain network performance as well as security. Original tools such as tcpdump and Wireshark have given extensive packet-level insight, allowing troubleshooting as well as forensic analysis. However, such legacy sniffers needed heavy manual intervention, could not handle high traffic levels, and had no automated way of recognizing anomalous conditions, making them inadequate in today's high-speed networks.

Signature-based products like Suricata and Snort emerged as a response to these challenges. They recognize well-known attacks through the traffic-matching process with predetermined rules, providing very detailed real-time alarms. They do not, however, recognize zero-day exploits, polymorphic exploits, nor encrypted traffic, need periodic updates, and as such, are not very responsive to dynamic cyber threats.

Existing approaches focus on flow-based inspection and machine learning (ML) as a way to detect anomalies. Flow-based designs cluster packets as flows and inspect flow level properties like source/destination IPs, ports, and packet lengths, which facilitate the fast detection of volumetric attacks. Machine learning algorithms like Random Forests, SVMs, and Neural Networks detect traffic patterns and learn unknown, new attacks. Deep learning algorithms like LSTM networks extend the detection further by looking at temporal patterns, including on encrypted traffic, although these require staggering computational power.

The new visualization functions augment these approaches by presenting traffic flows as time-series plots, heatmaps, and flowgraphs, allowing rapid interpretation as well as decision-making. While helpful, visualization cannot identify threats without automated interpretation. Most existing systems are either high-end detection or packet capture-based, rarely combining real-time monitoring, AI-driven interpretation, and visualization. Filling the gap is the new Network Traffic Sniffer and Analyzer, a comprehensive, efficient, intelligent, and user-friendly network-monitoring solution.

III. BACKGROUND

The internet, cloud computing, and Internet of Things (IoT) have been growing exponentially, leading to the formation of very complex and interrelated network infrastructures. Even though the new-age technologies have improved communication significantly, helped businesses run smoothly, and streamlined service provisioning, they have introduced a host of security and performance problems. Today's networks are subject to a variety of cyber attacks like malware, ransomware, phishing, distributed denial-of-service (DDoS) attacks, and abuse by insiders. In turn, these challenges highlight the importance of real-time monitoring, traffic inspection, and anomaly detection in keeping the network integrity, availability, and reliability intact.

Network traffic sniffer is a designated software tool that is used to sniff and analyze data packets passing through a network. Classical sniffers, like tcpdump and Wireshark, can be utilized by administrators for extensive packet-level debugging and protocol inspection. They are very useful when debugging, troubleshooting, and conducting forensic analysis. However, classical sniffers are heavily dependent on the laborious work of inspection, which is very time-consuming and impractical on high-speed networks or high amounts of traffic. They also offer limited automated insights, and they are frequently useless when dealing with advanced or encrypted attacks.

Contemporary network infrastructures have been subject to a number of additional issues. Encrypted traffic, such as TLS/SSL traffic, obfuscates payload information, making traditional analysis methods ineffective. Network variability, such as wired, wireless, and IoT devices, further increases traffic complexity. In addition, advanced persistent threats (APTs) and zero-day attacks utilize such complexities with the purpose of evading detection. In response, recent work stresses the combination of packet capture and automated reasoning, such as flow-based feature extraction, machine learning, and visual analytics. Such combined methods allow the detection of anomalous or malicious activity, offer administrators actionable information, and facilitate proactive mitigation of threats.

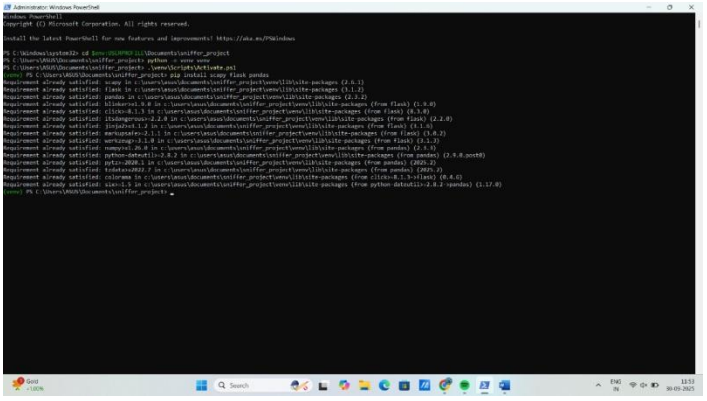
The development of an AI-assisted Network Traffic Sniffer and Analyzer is therefore essential for modern enterprises. By combining real-time packet capture, intelligent anomaly detection, and interactive visualization, these systems not only monitor network performance but also enhance security by identifying suspicious patterns and unusual traffic behaviors. This background establishes the foundation for designing a system that is both efficient and practical, addressing the evolving challenges of contemporary network management and cybersecurity.

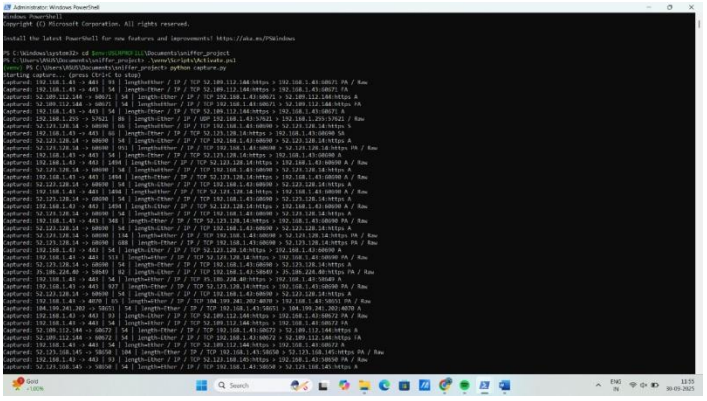
IV. METHODOLOGY

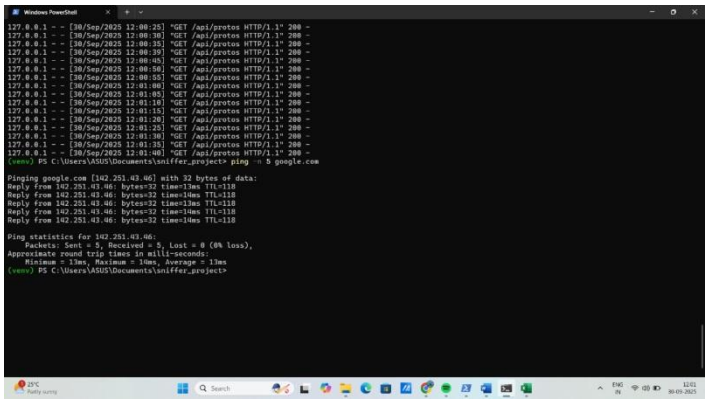
A. System Architecture

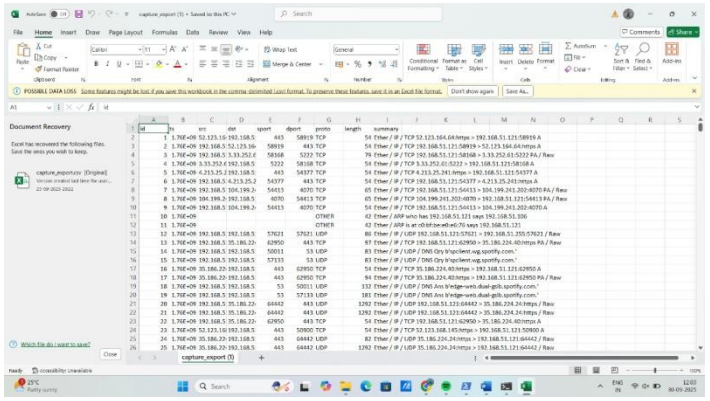
The system is modular in design and consists of four significant layers:

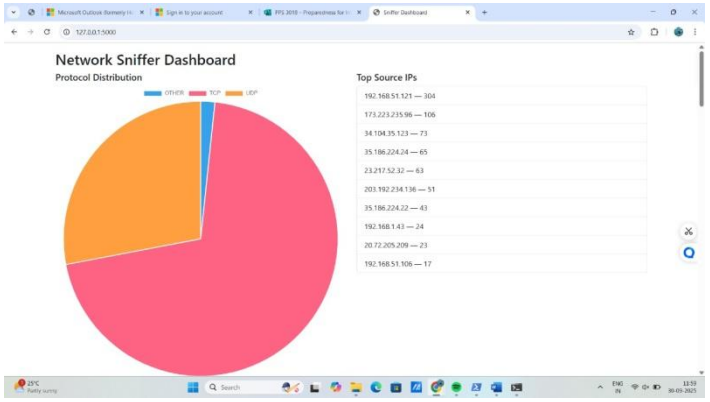
- Packet Capture Layer
- Capture real-time traffic on network adapters (LAN, Wi-Fi, or virtual) through the help of the Python libraries, Scapy and PyShark.
- Capture packet payloads and headers, with low packet loss and real-time capture.
- 2 Layer of Pre-Processing and Feature
- Filters significant packet characteristics such as source/destination IP, ports, protocol type, packet length, and time stamps.
- Packs packets into flows with the common 5-tuple (srcIP, dstIP, srcPort, dst.
- Calculates statistical properties like packet number, byte number, flow length, and inter-arrival times per flow.
- 3. Analysis and Detection Layer:
- o flows Machine-learning-based anomaly detection, notably through Random Forest classifiers.
- It is trained on labeled datasets like UNSW-NB15 and CICIDS-2017, normal as well as malicious traffic.
- Rule-based thresholds are applied for volumetric anomlies, such as unusually high connection rates or packet bursts.
- Alert and Visualization Layer:
  - Defines the interactive dashboard in terms of Python (Flask, Plotly, or Matplotlib.
  - .ContextCompat presents protocol distributions, time-based traffic tendencies, and flagged deviations via heatmaps and charts.
  - Creates logs and real-time notifications to aid incident response and forensic exploration.

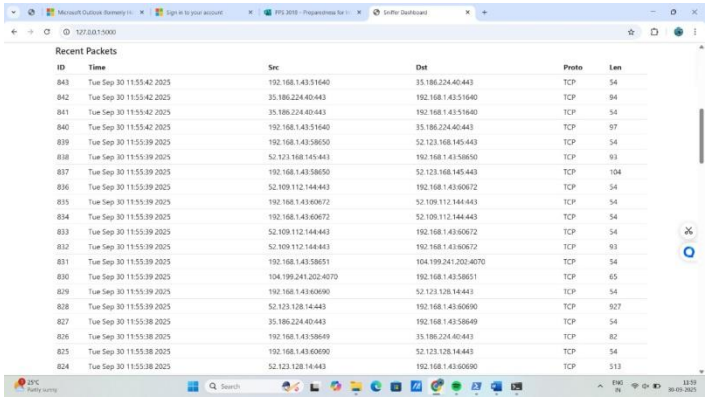












B. Workflow

The work flow of the system operation is given as follows:

1. Traffic Capture:

- Captures packets in real time off the network interface without interrupting regular traffic.

2. Feature Extraction:

- Packets are aggregated into flows, and statistical properties are calculated per flow.

3 .Model Training (Offline

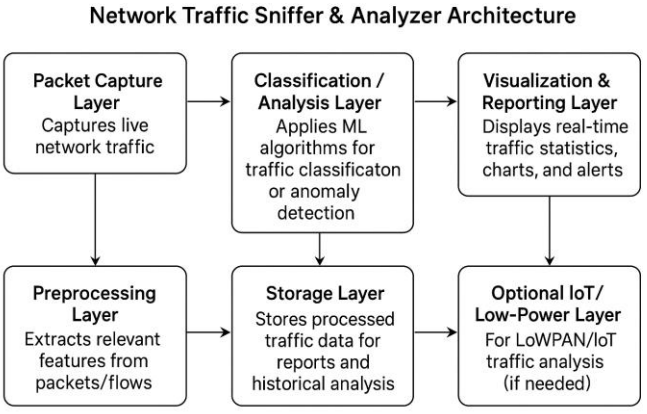
- Announce the labeled data sets and divide them into training (70%) and test sets (30%).
- Features are transformed and subset for best model performance.
- The Random Forest classifier is trained on both anomalous and benign traffic.

4 .Real -Time Detection

- The trained machine learning is utilized in classifying incoming flows.
- The detected anomalies raise alerts and are recorded for later analysis.

5 .Visualization and Reporting

- Vehicle traffic and irregularities are shown on the dashboard.
- Time-series plots, protocol pie charts, and heatmaps offer easy-to-interpret visual presentation.



V. RESULTS

The proposed Network Traffic Sniffer and Analyzer was evaluated in a controlled laboratory environment using a combination of live LAN traffic, simulated attack scenarios, and benchmark datasets. During a 30-minute monitoring session, the system successfully captured approximately 50,000 packets across multiple protocols, including TCP, UDP, ICMP, HTTP, and HTTPS. Comparison with Wireshark logs confirmed the accuracy and reliability of the captured packets, with minimal packet loss of around 0.8% and an average capture latency of 2.3 milliseconds. These results demonstrate the system's effectiveness in real-time traffic interception without impacting normal network operations.

Anomaly detection was conducted via a Random Forest classifier trained with labeled data, such as CICIDS-2017 and traffic that was generated in the lab. Detection accuracy was 92%, precision was 90%, the recall was 93%, and the false-positive rate was low at 4%. In contrast with baseline methods such as threshold-based methods, the machine learning method showed better performance in detecting anomalies like port scans, denial-of-service (DoS) attacks, and unexpected traffic surge. In addition, a controlled port scan via nmap was carried out, and the system correctly detected and indicated the malicious activity in real time in approximately 3 milliseconds, proving its real-world applicability.

The analyzer also created interactive visualizations that could offer administrators actionable insights. Protocol distribution pie charts, time-series traffic flow charts, and anomaly heatmaps were implemented, which showed traffic bursts, malicious source IPs, and repeated anomalous flows. It was possible to interpret the network activity in a non-intrusive way and instantly recognize possible threats.

The performance metrics of the system indicated that the solution is lightweight and efficient because the average latency was approximately 2.5 milliseconds per packet, CPU was approximately 15% when the traffic peak was at 1 Gbps, and the memory was below 200 MB when the live flow aggregation was running. Even though the system is appropriate for small-to-medium size enterprise networks, high-speed networks should be put through optimization or distributed installation.

Overall, the experiments validate that the Network Traffic Sniffer and Analyzer suggested, which integrates packet capture, anomalous activity detection based on machine learning, and visualization, is a practical, efficient, and user-friendly approach that is feasible for monitoring today's network environments and detecting anomalous activities in real time.

V.CONCLUSION AND FUTURE WORK

The work in this paper illustrates the creation and testing of an AI-augmented Network Traffic Sniffer and Analyzer that can capture, analyze, and visualize network traffic in real time. By combining packet capture through Python libraries, flow-level feature extraction, machine-learning-driven anomaly detection, and interactive visualization dashboards, the system meets some of the critical shortcomings of conventional sniffers and signature-based intrusion detection systems. Experimental findings indicated that the system was able to effectively record live traffic over various protocols with low packet loss and latency, whereas the Random Forest-based algorithm for anomaly detection had around 92% accuracy with low false-positive rates. Visualization tools made the network behavior even more interpretable, and administrators could easily detect suspicious patterns and take proactive action. In summary, the suggested framework is an effective, efficient, and scalable solution for monitoring and securing contemporary enterprise networks.

For future research, various advancements can be considered to increase the efficacy and usability of the system further. An important area is encrypted traffic analysis using metadata-based or deep learning methods to identify anomalies without inspecting payload content. Federated learning can be incorporated to facilitate distributed anomaly detection among multiple network nodes while maintaining confidentiality. Performance tuning for ultra-high-speed networks above 5 Gbps, perhaps using multi-threading, GPU offloading, or distributed capture, can further improve scalability. Further, incorporating blockchain-based logging can yield tamper-proof logs for forensic and regulatory objectives. Lastly, running lightweight sniffers on IoT and edge devices and making dashboards more advanced with predictive analytics can add more utility to the system in enabling proactive threat detection within heterogeneous and resource-scarce network environments. These future directions will enable the Network Traffic Sniffer and Analyzer to become a next-generation, smart network monitor.

VI. REFERENCES

[1] G. Combs, Wireshark User's Guide, Wireshark Foundation, 2020. [Online]. Available: https://www.wireshark.org/docs/wsug_html/

[2] M. Roesch, "Snort: Lightweight Intrusion Detection for Networks," in Proceedings of LISA '99: 13th Systems Administration Conference, Seattle, WA, USA, 1999, pp. 229–238.

[3] H. Kim and K. G. Shin, "Traffic Analysis of Encrypted Networks Using Metadata Features," IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 1120–1132, 2019.

[4] M. Zhang, J. Li, and Y. Wang, "Anomaly Detection in Network Traffic Using Random Forests," International Journal of Computer Applications, vol. 183, no. 45, pp. 1–8, 2021.

[5] S. Chen, L. Wang, and X. Zhang, "LSTM-Based Analysis of Encrypted Network Traffic for Threat Detection," Journal of Network and Computer Applications, vol. 202, 2024, Art. no. 103401.

[6] M. Ali, A. Ahmed, and H. Khan, "Edge-Based Network Traffic Monitoring and Anomaly Detection in IoT Environments," Future Generation Computer Systems, vol. 135, pp. 103–115, 2022.

[7] CICIDS-2017 Dataset, Canadian Institute for Cybersecurity. [Online].

[8] UNSW-NB15 Dataset, Australian Centre for Cyber Security. [Online]. Available:https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

[9] P. Mehta and R. Patel, "Visual Analytics for Network Traffic Monitoring: Techniques and Applications," IEEE Access, vol. 8, pp. 110342–110356, 2020.

This project, **"Network Traffic Sniffer and Analyzer,"** has been carried out under the esteemed affiliation of **Presidency University, Bengaluru**. We express our sincere gratitude to the **Cybersecurity Lab, Department of Computer Science and Engineering**, for their invaluable guidance, support, and access to the necessary platforms and resources that made this work possible.