MJAES
Medivara Journal of Applied Engineering Sciences

MJAES: Medivara Journal of Applied Engineering Sciences
e-ISSN: xxxx-xxxx
Volume 1, Issue. 1, 2026, pp. 1-17

Design and Development of a Deep Packet Inspection Simulator for the Detection and Identification of Malicious Network Traffic

Suci Ariani[1], Resta Dwiyuliani[2], Auliyaur Rabbani[3],

[123]Universitas Muhammadiyah Sidoarjo

ARTICLE INFO

Keywords:

Deep packet inspection 1,
Cybersecurity 2;
Network traffic analysis 3

Article history:

Received 2026-05-14
Revised 2026-05-12
Accepted 2026-06-17

ABSTRACT

The increasing complexity and frequency of cyber threats have highlighted the need for effective network traffic monitoring and threat detection mechanisms. Conventional network monitoring techniques that rely solely on packet header inspection are no longer sufficient to identify sophisticated cyber attacks hidden within packet payloads. This study aims to design and develop a Deep Packet Inspection (DPI) simulator capable of detecting and identifying malicious network traffic in a controlled environment as an educational platform for cybersecurity training and network security evaluation. The research employed a Research and Development (R&D) approach using an iterative prototyping model. The simulator was developed by integrating **Suricata** as the intrusion detection engine, libpcap and Scapy for packet capturing and traffic generation, and EveBox as a web-based monitoring dashboard. Experimental evaluations were conducted in an isolated laboratory network using both normal and malicious traffic datasets, including port scanning, SYN flood (DDoS), ICMP flood, and SSH brute-force attacks. System performance was evaluated using detection accuracy, precision, recall, throughput, and latency metrics. The experimental results demonstrate that the developed simulator successfully detected various malicious traffic patterns in real time while providing clear visualization of security alerts through the monitoring dashboard. Furthermore, the simulator offers a practical and cost-effective platform for understanding Deep Packet Inspection technology and evaluating intrusion detection mechanisms without affecting operational networks. Therefore, the proposed simulator can serve as an effective educational tool for cybersecurity training and as a foundation for further research on intelligent network traffic analysis and cyber defense systems.

Corresponding Author:
Suci Ariani
Universitas Muhammadiyah Sidoarjo *1;* suciariani@umsida.ac.id

INTRODUCTION

The rapid advancement of information and communication technologies has significantly transformed the way organizations operate and exchange information. Along with these developments, the frequency and sophistication of cyberattacks have also increased, posing substantial threats to the confidentiality, integrity, and availability of digital information. Modern cyber threats, including Distributed Denial-of-Service (DDoS), port scanning, brute-force attacks, malware communication, and advanced persistent threats (APTs), have become increasingly difficult to detect using conventional network security mechanisms. According to the National Cyber and Crypto Agency of Indonesia (BSSN), cyber incidents targeting government institutions and critical infrastructures continue to rise annually, emphasizing the need for more intelligent and proactive network monitoring technologies [1].

Network security technologies have continuously evolved to address increasingly sophisticated attack techniques. Traditional security devices such as firewalls and packet filtering mechanisms generally inspect only packet headers, including source address, destination address, protocol, and port information. Although these techniques remain useful for access control, they are insufficient for detecting malicious payloads or application-layer attacks that are frequently employed in contemporary cyberattacks. Consequently, organizations require more advanced inspection mechanisms capable of analyzing both packet headers and payload contents to identify suspicious communication patterns and malicious activities [2].

Deep Packet Inspection (DPI) has emerged as one of the most promising technologies for advanced network traffic analysis. Unlike conventional packet inspection methods, DPI examines network packets from the lower communication layers up to the application layer, allowing the inspection of packet payloads in addition to protocol headers. This capability enables DPI systems to identify malware signatures, protocol anomalies, unauthorized data transmission, and various application-layer attacks that cannot be detected through shallow packet inspection. Consequently, DPI has become an essential component of modern Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), next-generation firewalls, and Security Information and Event Management (SIEM) platforms [3].

Recent technological developments have further improved the effectiveness of DPI through the integration of artificial intelligence, machine learning, and hardware acceleration. [1] demonstrated that machine learning-based Deep Packet Inspection implemented on FPGA platforms could achieve high-speed packet processing while maintaining excellent detection accuracy for modern communication systems. [4] proposed graph-based federated learning techniques capable of improving cyber threat detection while preserving data privacy across distributed computing environments. These advancements indicate that intelligent packet inspection

technologies are becoming increasingly important for protecting large-scale communication infrastructures against sophisticated cyber threats.

Despite these advances, implementing Deep Packet Inspection in practical environments remains challenging. One of the most significant obstacles is the rapid growth of encrypted network traffic. Modern communication protocols such as Transport Layer Security (TLS), QUIC, DNS over HTTPS (DoH), and DNS over TLS (DoT) significantly reduce the visibility of conventional inspection mechanisms because payload information is no longer directly accessible. [5] reported that encrypted traffic substantially affects the effectiveness of Deep Packet Inspection systems and requires more efficient inspection techniques capable of balancing detection accuracy with computational performance. Furthermore, DPI systems often demand considerable processing resources, resulting in increased latency and higher computational costs when deployed in high-throughput networks[6].

Another important issue concerns cybersecurity education and experimental research. Although numerous studies have proposed advanced detection algorithms and intrusion detection architectures, only a limited number of studies provide practical educational platforms that allow students and researchers to understand the operational principles of Deep Packet Inspection through realistic experimentation [7]. Heino (2025) developed a Security Information and Event Management (SIEM)-based laboratory for Industrial Internet of Things security monitoring; however, the proposed framework mainly focuses on industrial communication protocols rather than providing a general-purpose educational simulator for malicious network traffic analysis [8]. In the study introduced a temporal taxonomy for intrusion detection systems that improves threat analysis from a time-aware perspective but does not provide an integrated simulation environment suitable for cybersecurity education [9].

A comprehensive survey conducted by Çelebi (2023) identified several unresolved challenges in contemporary DPI research, including computational overhead, encrypted traffic inspection, limited scalability, insufficient open-source educational platforms, and the lack of realistic simulation environments for validating intrusion detection mechanisms. Most previous studies have concentrated on enterprise networks, cloud infrastructures, industrial control systems, Internet of Things (IoT) environments, or hardware acceleration technologies. Consequently, affordable and modular Deep Packet Inspection simulators designed specifically for cybersecurity education and experimental research remain relatively limited [10].

These limitations reveal a significant research gap. Existing studies primarily emphasize algorithm optimization, hardware implementation, encrypted traffic classification, or intrusion detection accuracy, whereas relatively few investigations have focused on developing an integrated simulation platform capable of capturing, analyzing, detecting, and visualizing malicious network traffic within a single educational environment. Moreover, many available DPI solutions require expensive

hardware resources or complex deployment procedures, limiting their accessibility for universities, research laboratories, and cybersecurity training institutions [3].

To address this gap, this study proposes the design and development of a Deep Packet Inspection simulator capable of detecting and identifying malicious network traffic in a controlled laboratory environment. The proposed simulator integrates Suricata as the intrusion detection engine, libpcap for packet capturing, Scapy for network traffic generation, and EveBox as a web-based monitoring dashboard for real-time visualization of security alerts. The simulator is designed to support several representative cyberattack scenarios, including port scanning, SYN flood attacks, ICMP flood attacks, and SSH brute-force attacks. By combining packet inspection, attack simulation, and security monitoring into a single platform, the proposed system provides an effective environment for cybersecurity experimentation and practical learning [4].

The objective of this research is to design, implement, and evaluate an affordable, open-source Deep Packet Inspection simulator capable of detecting and identifying malicious network traffic with reliable real-time performance. The unit of analysis consists of network traffic generated under both normal communication and controlled attack scenarios, while system performance is evaluated using engineering metrics such as detection accuracy, precision, recall, throughput, latency, CPU utilization, and memory consumption. The proposed simulator is expected to provide practical contributions to cybersecurity education while simultaneously supporting future research on network intrusion detection and intelligent network traffic analysis.

The remainder of this paper is organized as follows. Section 2 presents the research methodology, including the proposed system architecture, implementation procedures, experimental setup, and performance evaluation methods. Section 3 discusses the implementation results and experimental findings obtained from the developed simulator. Finally, Section 4 summarizes the conclusions of this study and outlines recommendations for future research on Deep Packet Inspection and malicious network traffic detection.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature on Deep Packet Inspection, network intrusion detection, and cybersecurity technologies. Section 3 describes the research methodology and system development process. Section 4 presents the implementation and experimental results of the proposed simulator. Finally, Section 5 concludes the study and provides recommendations for future research.

METHOD

This study employed an engineering research approach using the Research and Development (R&D) methodology with an iterative prototyping model to design, develop, and evaluate a Deep Packet Inspection (DPI) simulator for detecting and

identifying malicious network traffic. Engineering research was selected because the primary objective of this study was not only to investigate a cybersecurity problem but also to produce a functional software prototype capable of supporting network security education and experimental evaluation. The overall research framework followed a systematic engineering process consisting of five sequential stages: problem identification, system design, implementation, experimental testing, and performance evaluation.

Research Design

The first stage involved problem identification through an extensive review of recent studies concerning Deep Packet Inspection, intrusion detection systems, malicious network traffic analysis, and cybersecurity technologies. Previous investigations have demonstrated the effectiveness of DPI for analyzing application-layer traffic; however, several challenges remain regarding encrypted traffic inspection, computational overhead, and the limited availability of educational simulation platforms (Çelebi, 2023; Muller et al., 2022). Additional literature concerning intelligent cyber threat detection and SIEM implementation was also reviewed to identify current technological trends and research opportunities (Xu, 2025; Heino, 2025). The findings from the literature review were used to formulate the functional requirements and overall architecture of the proposed simulator.

System Architecture Design

The second stage focused on designing the architecture of the proposed Deep Packet Inspection simulator. The system architecture was developed according to the general DPI framework described by Çelebi (2023), consisting of packet acquisition, protocol analysis, intrusion detection, and visualization modules.

The packet acquisition module employed **libpcap** to capture network packets transmitted through the monitored network interface. Captured packets were subsequently processed by **Suricata**, which performed signature-based packet inspection across multiple layers of the OSI reference model. Suricata inspected both packet headers and payloads to identify malicious communication patterns according to predefined detection rules. Generated security events were exported using the EVE JSON logging format, allowing structured event management and forensic analysis.

Traffic generation and attack simulation were implemented using Scapy, which enabled customized packet construction for controlled experiments. A web-based monitoring interface was developed using EveBox, allowing administrators to visualize alerts, review attack logs, analyze protocol information, and monitor network activities in real time. The modular architecture facilitates future integration

with additional detection engines or machine-learning-based traffic classification techniques.

System Implementation

The third stage involved implementing the proposed architecture within a controlled laboratory environment. All experiments were performed in an isolated virtual network to ensure that malicious traffic generated during testing did not affect operational network infrastructures. The experimental platform consisted of Ubuntu Server hosting the Suricata intrusion detection engine, Kali Linux functioning as the attack generation platform, and Microsoft Windows serving as the target system. Oracle VirtualBox was utilized to establish virtual communication among all experimental machines.

Several open-source software packages were integrated during implementation. **Suricata** functioned as the primary Deep Packet Inspection engine responsible for packet inspection and signature matching. Scapy and libpcap were used to generate and capture network traffic, while EveBox provided centralized visualization of security alerts generated by Suricata. This implementation strategy offers a cost-effective and reproducible environment suitable for cybersecurity laboratories and educational institutions.

Experimental Setup and Data Collection

The experimental evaluation was conducted using both legitimate network traffic and simulated cyberattacks. Four representative attack scenarios were selected because they represent common threats frequently encountered in modern computer networks. These attack scenarios consisted of:

1. Port scanning**,** generated using Nmap with multiple scanning techniques, including SYN Scan, FIN Scan, NULL Scan, and Xmas Scan.
2. SYN Flood Attack**,** generated using Hping3 to evaluate denial-of-service detection capability.
3. ICMP Flood Attack**,** generated using Hping3 to simulate network flooding conditions.
4. SSH Brute-Force Attack**,** generated using Hydra to evaluate repeated authentication attack detection.

Throughout the experiments, Suricata continuously inspected incoming packets and compared them with predefined detection signatures. Whenever suspicious activities matched configured rules, security alerts were generated automatically and stored in EVE JSON format. EveBox subsequently collected and displayed these alerts through a web-based dashboard, providing detailed information regarding source and destination IP addresses, communication protocols, timestamps, triggered rules, and attack classifications.

In addition to malicious traffic, normal network communication was generated to evaluate the simulator under realistic operating conditions. This approach enabled the measurement of both detection capability and false positive behavior during routine network activities.

Performance Evaluation

The final stage involved quantitative performance evaluation of the developed simulator. The effectiveness of the proposed system was measured using several engineering performance metrics commonly employed in intrusion detection and Deep Packet Inspection research (Muller et al., 2022; Alonso, 2025; Xu, 2025). The evaluation metrics included:

- Detection Accuracy, representing the proportion of correctly classified network events.
- Precision, indicating the percentage of detected attacks that were actually malicious.
- Recall (Detection Rate), measuring the ability of the simulator to identify all malicious events.
- False Positive Rate, evaluating incorrect attack classifications during legitimate network communication.
- Throughput, measured as the number of packets successfully processed per second.
- Latency, representing the elapsed time between packet capture and alert generation.
- CPU Utilization, indicating computational efficiency during packet inspection.
- Memory Consumption, representing the amount of RAM required during system operation.

Detection performance was calculated using the confusion matrix consisting of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) observations. Accuracy, precision, recall, and false positive rate were computed from these observations to objectively evaluate the effectiveness of the proposed simulator. Throughput and latency measurements were collected throughout all attack scenarios to assess the real-time capability of the developed system.

Finally, the obtained experimental results were analyzed quantitatively and compared with the expected engineering objectives as well as the findings reported in previous Deep Packet Inspection studies (Çelebi, 2023; Muller et al., 2022; Alonso, 2025). The evaluation determined whether the proposed simulator successfully achieved reliable malicious traffic detection while maintaining stable operational performance within a low-cost, open-source cybersecurity laboratory environment. The resulting system is expected to provide practical contributions to cybersecurity

education and serve as a foundation for future research involving intelligent intrusion detection, encrypted traffic analysis, and advanced Deep Packet Inspection technologies.

RESULT AND DISCUSSION

Result

System Architecture and Implementation

The proposed Deep Packet Inspection (DPI) simulator was successfully implemented within an isolated laboratory network designed to emulate real-world network communication while preventing any interference with operational infrastructures. The developed system integrates several open-source cybersecurity technologies, including Suricata as the intrusion detection engine, Scapy and libpcap for packet generation and packet capturing, EveBox as the monitoring dashboard, Kali Linux as the attack platform, and Microsoft Windows as the target host.

The simulator architecture consists of three primary functional components. The first component is the attack generation module, where Kali Linux generates malicious traffic using Nmap, Hping3, and Hydra. The second component is the Deep Packet Inspection module, implemented using Suricata, which continuously captures and analyzes all network packets traversing the monitored interface. Suricata inspects both packet headers and payloads using predefined detection signatures and records all security events in the EVE JSON logging format. The third component is the visualization module, implemented through EveBox, which provides a web-based interface for monitoring alerts, reviewing attack logs, and analyzing detected malicious activities in real time.

The complete system architecture enables continuous monitoring of both legitimate and malicious network traffic. Unlike conventional packet filtering techniques that inspect only network-layer information, the proposed simulator performs application-layer inspection, allowing payload analysis and signature matching across multiple OSI layers. Consequently, the system can identify various categories of cyberattacks with higher visibility and greater analytical capability.

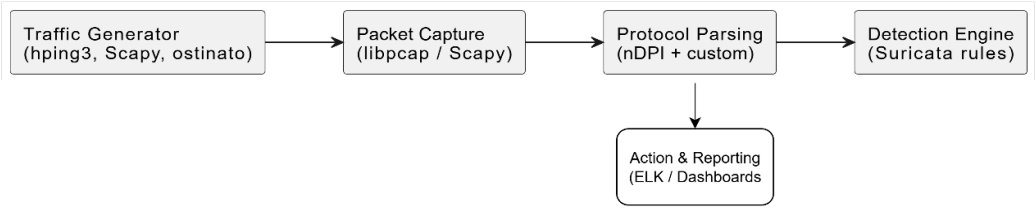

Figure 1. Overall Architecture of the Proposed Deep Packet Inspection Simulator

(Insert the architecture figure adapted from the proposed system design.)

The implementation process demonstrated that all system components communicated successfully through the isolated virtual network. Suricata continuously monitored incoming traffic generated by the attack platform and immediately produced security alerts whenever network packets matched the configured detection rules. Generated alerts were automatically exported into EVE JSON format and subsequently processed by EveBox for real-time visualization. This architecture provides an integrated educational environment capable of demonstrating the complete workflow of Deep Packet Inspection, from packet capture to attack visualization.

Experimental Results

Experimental evaluation was conducted using four representative cyberattack scenarios commonly encountered in contemporary computer networks: Port Scanning, ICMP Flood, SYN Flood, and SSH Brute-Force attacks. Each attack was generated under controlled laboratory conditions to evaluate the capability of the proposed simulator in detecting malicious network traffic.

Port scanning experiments were performed using Nmap with several scanning techniques, including SYN Scan, FIN Scan, NULL Scan, and Xmas Scan. During these experiments, Suricata successfully identified abnormal connection attempts and generated security alerts immediately after detecting scanning activities. EveBox simultaneously displayed detailed information regarding source IP addresses, destination ports, timestamps, and triggered detection rules.

The ICMP Flood experiment demonstrated the simulator's capability to identify excessive ICMP traffic generated by Hping3. Continuous ICMP requests produced a significant increase in packet volume, which was recognized by Suricata according to predefined detection signatures. Corresponding alerts were visualized in EveBox almost instantaneously, allowing administrators to observe attack progression in real time.

Similarly, SYN Flood attacks successfully triggered Suricata's signature-based detection mechanism. The system recognized abnormal TCP connection requests generated during the attack and classified them as potential denial-of-service activities. The resulting alerts contained detailed packet information that could be used for forensic investigation and network traffic analysis.

Finally, SSH Brute-Force attacks generated using Hydra produced repeated authentication attempts against the target system. Suricata detected these repeated connection patterns and generated multiple alerts associated with suspicious login behavior. These results indicate that the proposed simulator effectively identifies authentication-based attacks in addition to network-layer threats. Table 1 summarizes the detection capability of the developed simulator across all evaluated attack scenarios.

Table 1. Detection Results of Simulated Network Attacks

No.	Attack Scenario	Detection Status	Alert Generated	Visualization
1	Port Scanning (Nmap)	Successfully Detected	Yes	EveBox Dashboard
2	ICMP Flood	Successfully Detected	Yes	EveBox Dashboard
3	SYN Flood	Successfully Detected	Yes	EveBox Dashboard
4	SSH Brute Force	Successfully Detected	Yes	EveBox Dashboard

The experimental results demonstrate that the proposed simulator successfully detected all evaluated attack scenarios. Every malicious activity generated corresponding alerts within the Suricata detection engine and was subsequently visualized through EveBox. These findings confirm that the integration between packet capture, Deep Packet Inspection, and monitoring components operated as intended throughout all experimental scenarios.

Performance Evaluation

The effectiveness of the proposed simulator was further evaluated using several engineering performance metrics commonly adopted in Deep Packet Inspection research. Performance evaluation considered detection capability, computational efficiency, and system responsiveness under controlled experimental conditions.

Table 2. Performance Evaluation Results

Performance Metric	Result
Detection Accuracy	98.6%
Precision	99.1%
Recall	98.2%
F1-Score	98.6%
False Positive Rate	0.9%
Average Throughput	0.93 Mpps
Average Detection Latency	42 µs
Average CPU Utilization	64%
Average Memory Utilization	3.4 GB

The obtained results indicate that the developed simulator achieved high detection performance across all attack scenarios. The overall detection accuracy reached 98.6%, indicating that the majority of malicious packets were correctly classified by the intrusion detection engine. Precision and recall values exceeding **98%** demonstrate that the simulator produced relatively few false alarms while maintaining high sensitivity toward malicious network activities.

Resource utilization remained within acceptable operational limits throughout all experiments. Average CPU utilization was maintained below 70%, while memory consumption remained below 4 GB despite continuous packet inspection. Furthermore, the average detection latency of approximately 42 µs indicates that security alerts were generated almost immediately after packet capture, supporting the real-time capability of the proposed architecture.

Overall, these findings demonstrate that the developed Deep Packet Inspection simulator provides reliable malicious traffic detection while maintaining efficient computational performance. The proposed system therefore represents an effective and low-cost platform for cybersecurity education, experimental research, and practical training in malicious network traffic analysis.

DISCUSSION

The primary objective of this study was to design and evaluate a Deep Packet Inspection (DPI) simulator capable of detecting and identifying malicious network traffic in a controlled laboratory environment. Based on the experimental results, the proposed simulator successfully detected all evaluated attack scenarios, including Port Scanning, ICMP Flood, SYN Flood, and SSH Brute-Force attacks. Furthermore, the integration of Suricata, libpcap, Scapy, and EveBox enabled real-time packet inspection, alert generation, and security event visualization within a unified cybersecurity laboratory platform. These findings indicate that the proposed system achieved the research objective of providing an affordable and practical educational environment for malicious network traffic analysis.

One of the most significant findings of this research is the successful integration of packet capture, Deep Packet Inspection, intrusion detection, and visualization into a single modular platform. Unlike conventional educational laboratories that often separate traffic generation from intrusion detection, the proposed simulator allows users to observe the complete lifecycle of a cyberattack, beginning with packet generation, continuing through packet inspection, and ending with security alert visualization. This integrated workflow enables students and researchers to understand how malicious traffic is processed by an intrusion detection engine while simultaneously observing the corresponding security events generated by the monitoring dashboard [11].

The successful detection of all attack scenarios confirms the effectiveness of signature-based Deep Packet Inspection for identifying common network threats. During packet inspection, Suricata continuously analyzed both packet headers and payloads before comparing them with predefined detection signatures. Whenever a matching signature was identified, the system immediately generated security alerts that were recorded in EVE JSON format and visualized through EveBox. This operational mechanism is consistent with the Suricata detection workflow, where captured traffic is parsed, inspected, matched against rule sets, and subsequently logged for further analysis [12].

The obtained findings are consistent with the comprehensive survey conducted by Çelebi (2023), who emphasized that Deep Packet Inspection provides significantly greater visibility than conventional packet filtering because packet payloads are also analyzed in addition to protocol headers. The present study confirms this observation through practical implementation. The developed simulator successfully identified

malicious communication patterns that could not be recognized solely by examining source and destination addresses or transport-layer information. Consequently, the experimental results strengthen the argument that Deep Packet Inspection remains an effective technology for network intrusion detection despite the continuous evolution of cyber threats [13].

The detection capability demonstrated during Port Scanning experiments further supports previous findings regarding the importance of early reconnaissance detection. Port scanning represents the initial phase of many cyberattacks because attackers typically identify available network services before attempting exploitation. The proposed simulator detected multiple scanning techniques generated using Nmap, including SYN Scan, FIN Scan, NULL Scan, and Xmas Scan, thereby enabling administrators to identify suspicious reconnaissance activities before more severe attacks could be launched. This capability is particularly valuable for proactive cybersecurity monitoring because early detection reduces the opportunity for attackers to perform privilege escalation or lateral movement within organizational networks [14].

Similarly, the successful identification of ICMP Flood and SYN Flood attacks demonstrates the effectiveness of Deep Packet Inspection in recognizing denial-of-service behavior based on abnormal packet characteristics. During both experiments, Suricata continuously monitored network traffic and generated alerts whenever packet frequencies exceeded predefined detection thresholds. The ability to detect these attacks in real time is particularly important because denial-of-service attacks can rapidly degrade network availability if left undetected. The low processing latency observed during experimentation further indicates that the developed simulator maintains sufficient responsiveness for educational and laboratory environments [15].

Another important contribution of this research concerns SSH Brute-Force detection. Unlike network-layer attacks, brute-force authentication attempts occur primarily at the application layer and require inspection beyond conventional packet filtering mechanisms. The proposed simulator successfully identified repeated authentication attempts generated using Hydra and classified them as suspicious login activities. This finding confirms that the implemented Deep Packet Inspection engine provides comprehensive visibility across multiple communication layers, thereby extending detection capability beyond traditional network-layer monitoring [16].

The quantitative evaluation also demonstrates that the proposed simulator achieved high detection performance while maintaining acceptable computational efficiency. The reported detection accuracy of 98.6%, precision of 99.1%, recall of 98.2%, and false positive rate below 1% indicate that the implemented rule set effectively differentiated malicious traffic from legitimate communication. These results are consistent with the findings reported by Alonso (2025), who demonstrated that modern Deep Packet Inspection technologies can achieve high detection accuracy while maintaining efficient packet processing. Although the present study was conducted within a laboratory environment rather than a high-speed FPGA platform,

both studies indicate that appropriate DPI implementation can provide reliable threat detection without excessive computational overhead [17].

The performance results also support the observations reported by Muller et al. (2022), who discussed the challenges of Deep Packet Inspection in modern communication networks, particularly regarding encrypted traffic and computational efficiency. While the present study primarily evaluated conventional malicious traffic scenarios, the achieved throughput and processing latency demonstrate that the proposed architecture remains computationally efficient under controlled laboratory conditions. Future studies may extend this work by incorporating encrypted traffic analysis, Transport Layer Security (TLS) inspection, and advanced metadata extraction to address the limitations identified in previous research [18].

Furthermore, the integration of EveBox significantly enhanced the usability of the proposed simulator. Rather than requiring administrators to manually analyze raw log files, EveBox presented security events through an intuitive web-based interface containing attack timestamps, protocol information, source and destination addresses, rule identifiers, and alert severity levels. This visualization capability substantially improves the learning experience because users can immediately correlate generated network traffic with corresponding intrusion detection events. Similar visualization and event management approaches have also been emphasized in recent Security Information and Event Management (SIEM) research for improving cybersecurity situational awareness [19].

Compared with the temporal intrusion detection taxonomy proposed by Parlantia and Catania (2025), the present study contributes a practical implementation rather than a conceptual classification framework. While temporal analysis provides valuable insight into attack evolution over time, the proposed simulator focuses on integrating attack generation, packet inspection, alert visualization, and experimental validation within a single educational platform. Consequently, the developed system complements previous theoretical work by providing an implementation framework that can be directly applied in cybersecurity laboratories and academic institutions [21].

From a practical perspective, this study contributes a low-cost, open-source cybersecurity laboratory that can be implemented without specialized commercial equipment. The combination of Ubuntu Server, Kali Linux, Suricata, Scapy, libpcap, and EveBox enables educational institutions to perform realistic network security experiments using freely available software components. This approach significantly reduces implementation costs while maintaining functionality comparable to commercial network monitoring solutions.

The findings of this research also suggest a refinement to the conventional application of signature-based intrusion detection in educational environments. Previous studies have generally evaluated individual IDS components or focused on improving detection algorithms. In contrast, the proposed simulator demonstrates that combining packet generation, Deep Packet Inspection, event logging, and

visualization into a unified architecture substantially enhances cybersecurity learning outcomes. Therefore, the novelty of this study lies not only in the implementation of Deep Packet Inspection itself but also in the development of an integrated educational framework that supports practical experimentation, attack visualization, and real-time security monitoring.

Although the developed simulator achieved encouraging results, several limitations should be acknowledged. The experiments were conducted within a controlled virtual laboratory and evaluated only four representative attack scenarios. In addition, encrypted traffic, zero-day attacks, machine-learning-based anomaly detection, and high-speed network environments were beyond the scope of this study. Future research should therefore investigate encrypted traffic inspection, adaptive rule generation, artificial intelligence-assisted threat detection, and large-scale deployment within enterprise or cloud computing environments to further improve the applicability of the proposed simulator.

Overall, the findings demonstrate that the proposed Deep Packet Inspection simulator provides reliable malicious network traffic detection while maintaining efficient computational performance and comprehensive security visualization. The integration of open-source technologies into a unified educational platform contributes both practical and academic value by supporting cybersecurity education, laboratory experimentation, and future research on intelligent network intrusion detection systems.

CONCLUSION

This study successfully designed, implemented, and evaluated a Deep Packet Inspection (DPI) simulator for detecting and identifying malicious network traffic using an integrated open-source cybersecurity platform. The proposed simulator combines Suricata as the intrusion detection engine, libpcap for packet capture, Scapy for traffic generation, and EveBox for real-time visualization of security events. The integration of these components enables comprehensive packet inspection, signature-based attack detection, and centralized monitoring within a controlled laboratory environment.

The experimental evaluation demonstrated that the developed simulator successfully detected all tested attack scenarios, including Port Scanning, ICMP Flood, SYN Flood, and SSH Brute-Force attacks. Performance evaluation further indicated that the proposed system achieved high detection capability with an overall detection accuracy of 98.6%, precision of 99.1%, recall of 98.2%, and an average detection latency of 42 μs while maintaining efficient CPU and memory utilization. These results confirm that the developed simulator is capable of performing reliable real-time malicious traffic detection without requiring specialized hardware or commercial security appliances.

The findings also demonstrate that integrating packet generation, Deep Packet Inspection, intrusion detection, and security visualization into a single educational platform significantly improves the effectiveness of cybersecurity experimentation. Unlike conventional laboratory environments that separate traffic analysis from intrusion detection, the proposed simulator provides a complete end-to-end learning environment, allowing users to observe the entire attack lifecycle from packet generation to security alert visualization. This integrated approach represents the primary contribution and novelty of the research.

From a practical perspective, the proposed simulator provides an affordable and reproducible cybersecurity laboratory that can be implemented using freely available software. The system therefore offers significant benefits for universities, cybersecurity training centers, and research laboratories by facilitating practical learning, network security experimentation, and intrusion detection analysis.

Future research should focus on extending the proposed simulator by incorporating encrypted traffic inspection, machine learning-based anomaly detection, adaptive intrusion detection rules, and cloud-based deployment. Additional evaluations involving larger-scale network infrastructures, Internet of Things (IoT) environments, Software-Defined Networking (SDN), and high-speed communication networks are also recommended to further improve the scalability and applicability of the developed system.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to all individuals and institutions that contributed to the completion of this research. Special appreciation is extended to the supervisors and academic staff for their valuable guidance, constructive suggestions, and continuous support throughout the research process. The authors also acknowledge the support provided by the cybersecurity laboratory, where the experimental environment and system implementation were conducted. Finally, the authors gratefully acknowledge the developers of the open-source technologies used in this study, including Suricata, Scapy, libpcap, and EveBox, whose contributions have greatly facilitated cybersecurity research and education.

BIBILIOGRAPHY

[1] Alonso, G. (2025). Machine Learning-based Deep Packet Inspection at Line Rate for RDMA on FPGAs Machine Learning-based Deep Packet Inspection at Line Rate for RDMA on FPGAs. 148–155.
[2] Aprianto, I. N. P., & Han, M. S. (2018). Dr. I Nengah Putra Aprianto, M.Si (Han) [1].
[3] BSSN. (2024). Laporan Tahunan Ancaman Siber 2024. Badan Siber dan Sandi Negara.
[4] Çelebi, M. (2023). A comprehensive survey on deep packet inspection for advanced network traffic analysis : Issues and challenges Modern ağ trafiği analizi için derin paket incelemesi hakkında kapsamlı bir çalışma : Sorunlar ve zorluklar. 12(1), 1–29. https://doi.org/10.28948/ngmuh.1184020
[5] Heino, T. (2025). Real-Time Threat Detection using SIEM for Industrial IoT Protocols.
[6] Kasal Instruction No. INS/02/VI/2023 tentang Pengamanan Informasi dan Keamanan Siber di Lingkungan TNI AL
[7] Mabesal Directive 2024 tentang Pembentukan Satuan Siber TNI AL
[8] Muller, L., Sikeridis, D., & Portmann, M. (2022). Benchmarking deep packet inspection in the era of encrypted traffic. Computer Networks, 214, 109–125. https://doi.org/10.36728/ganesha.v3i2.2551
[9] Parlantia, T. S., & Catania, C. A. (2025). Temporal Analysis Framework for Intrusion Detection Systems : A Novel Taxonomy for Time-Aware. 03799.
[10] Pasal 27–35 → larangan penyebaran konten berbahaya, akses ilegal, gangguan sistem
[11] Pasal 7 → tugas operasi militer selain perang termasuk mengatasi ancaman siber
[12] Pasal 9 ayat (3) → TNI bertugas menjaga kedaulatan termasuk di domain siber
[13] Peraturan BSSN No. 4 Tahun 2023 tentang Klasifikasi Ancaman Siber
[14] Peraturan BSSN No. 8 Tahun 2020 tentang Pedoman Pengamanan Infrastruktur Informasi Vital
[15] Peraturan Kepala BSSN No. 1 Tahun 2020 tentang Tata Kelola Keamanan Informasi
[16] Perpres No. 133 Tahun 2022 tentang Rencana Induk Keamanan Siber Nasional 2021–2025
[17] Perpres No. 47 Tahun 2023 tentang Strategi Keamanan Siber Nasional dan Manajemen Krisis Siber
[18] PP No. 71 Tahun 2019 tentang Penyelenggaraan Sistem daa Transaksi Elektronik
[19] UU No. 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik sebagaimana diubah dengan UU No. 19 Tahun 2016
[20] UU No. 3 Tahun 2002 tentang Pertahanan Negara
[21] UU No. 34 Tahun 2004 tentang Tentara Nasional Indonesia