Strengthening Network Security: Evaluation of Intrusion Detection and Prevention Systems Tools in Networking Systems

Wahyu Adi Prabowo[1], Khusnul Fauziah[2], Aufa Salsabila Nahrowi[3], Muhammad Nur Faiz[4],
Arif Wirawan Muhammad[5]
Department of Informatics Engineering-Faculty of Informatics, Institut Teknologi Telkom Purwokerto, Indonesia[1, 2, 3]
Department of Computer and Business, Cybersecurity, Cilacap State Polytechnic, Cilacap, Indonesia[4]
Department of Information Security and Web Technology, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia[5]

***Abstract*—This study aims to enhance network security by comprehensively evaluating various Intrusion Detection and Prevention Systems tools in networking systems. The objectives of this research were to assess the performance of different IDPS tools in terms of computer resources utilization, Quality of Service metrics namely delay, jitter, throughput, and packet loss, and their effectiveness in countering Distributed Denial of Service attacks, specifically ICMP Flood and SYN Flood. The evaluation used popular IDPS tools, including Snort, Suricata, Zeek, OSSEC, and Honeypot Cowrie. Real attack scenarios were simulated to measure the tools performance. The results indicated CPU and RAM usage variations among the tools, with Snort and Suricata showing efficient resource utilization. Regarding QoS metrics, Snort demonstrated superior performance in delay, jitter, throughput, and packet loss mitigation for both attack types. The implication for further research lies in exploring the optimal configurations and fine-tuning of IDPS tools to achieve the best possible network security against DDoS attacks. This research provides valuable insights into selecting appropriate IDPS tools for network administrators, cybersecurity professionals, and organizations to fortify their infrastructure against evolving cyber threats.**

Keywords*—IDPS; network security; computer performance; Quality of Service; DDoS attacks

I. Introduction

In today's digital landscape, cybersecurity measures are paramount to protect and protect networks and sensitive data. Among the various cyber threats organizations and individuals face, Distributed Denial of Service (DDoS) attacks pose significant challenges. These attacks involve overwhelming a target system with excessive traffic, rendering it unavailable to legitimate users [1]. Developing effective defense mechanisms against DDoS flooding attacks requires a comprehensive understanding of the problem and the techniques used to prevent, detect, and respond to such attacks [2]. In a DDoS attack, the attacker orchestrates the assault using a network of remotely controlled and widely dispersed nodes. These nodes work collaboratively to flood the victim's network with overwhelming traffic. The primary objective of this attack is not to directly exploit the victim's data but to disrupt the normal functioning of the victim's resources, making it challenging for legitimate users to access the services.

The agent-handler model is a significant structure utilized in DDoS attacks, involving four key participants: the attacker or botmaster, handlers, agents, and the victim [3]. The attacker, also known as the botmaster, communicates indirectly with the agents through the handlers, which act as intermediaries facilitating coordination and communication [4]. The agents compromised devices or systems attack by flooding the victim's network with massive malicious traffic [5], [6].

The agent-handler model provides several advantages for attackers, enabling them to maintain anonymity and distance themselves from the attack [4]. The owners of compromised agent systems often remain unaware that their devices are being exploited to launch DDoS attacks [5]. Moreover, handlers allow the attacker to control multiple agents simultaneously, significantly amplifying the scale and impact of the attack [7]. This model proves particularly effective when targeting web servers during DDoS attacks [5]. By overwhelming the target server with a flood of HTTP requests, such as in an HTTP flood attack, the attacker can exhaust the server's resources, disrupting its availability [7]. Another technique within this model is the Slowloris attack, where partial HTTP requests are sent to the target server, causing it to open additional connections and eventually leading to resource exhaustion [7].

To counter these attacks, Intrusion Detection and Prevention Systems (IDPS) have emerged as a crucial tool in safeguarding networks [8]–[10]. These systems effectively detect and mitigate DDoS attacks to prevent service disruption and data compromise [11]. However, one of the challenges IDPS faces is the ability to effectively detect and defend against evolving and unprecedented attacks [12], [13]. IDPS have traditionally employed two approaches for attack detection: signature-based and anomaly-based [14].

Signature-based detection relies on predefined attack patterns or signatures to identify threats. Although this method can accurately identify known attacks, it becomes ineffective against new or unprecedented attacks that do not match existing signatures [14]. On the other hand, anomaly-based detection analyzes network traffic and identifies abnormal patterns or behavior that deviates from regular network activity [12], [15]. This approach is more effective in detecting unknown attacks, as it does not rely on specific attack signatures but instead focuses on identifying anomalous behavior [16]. Intrusion detection and prevention system (IDPS) is vital to cybersecurity measures. It is crucial to safeguard computer networks and systems from unauthorized access and malicious activities. IDPS monitors network traffic and analyzes it for any signs of suspicious or malicious behavior [17]–[19]. IDPS includes detecting and preventing unauthorized access attempts, malware attacks, and other security breaches [20], [21].

An IDPS can be either network-based or host-based [22]. Network-based IDPS monitors network traffic at various points in the network infrastructure, such as routers and switches, to identify any abnormal patterns or activities. Host-based IDPS, on the other hand, focuses on monitoring an individual host or endpoint device to detect signs of intrusion or malicious activity [23]. The primary function of an IDPS is to detect and prevent unauthorized access to a network or system. It achieves this by analyzing network traffic and comparing it against predefined patterns or signatures of known attacks [24]. Suppose an IDPS identifies any suspicious activity or a match with a known attack signature. In that case, it generates an alert or takes immediate action to prevent further damage and secure the system [25]. An IDPS can also detect and prevent anomalous behavior not covered by known attack signatures [26].

In the comparative analysis of Intrusion Detection and Prevention Systems (IDPS), several popular systems are evaluated, including Snort, Suricata, Zeek, OSSEC, and the honeypot Cowrie. The existing literature on Intrusion Detection and Prevention Systems (IDPS) is extensive and diverse, with each study providing valuable insights. Based on the comprehensive test results in this study [27], the utilization of pfSense and Suricata emerges as the proposed solution to thwart attacks initiated by internal users and curtail assaults stemming from internal networks, as evidenced by the conducted attack test scenarios. With supplementary devices, the next-generation firewall pfSense and Suricata can significantly bolster network security compared to relying solely on traditional firewalls.

Previous studies have examined the use of IDPS in ensuring Quality of Service in various network environments. These studies have highlighted the importance of IDPS in maintaining network performance and protecting against potential cyber threats. One study was conducted by [28]. Focused on using IDPS in cloud environments to achieve desired security in next-generation networks. The study analyzed different intrusions that could affect cloud resources and services' availability, confidentiality, and integrity. Based on their findings, they recommended positioning IDPS in cloud environments as a crucial step towards ensuring network security. Previous studies have also emphasized the need for IDPS to protect against various attacks, such as distributed denial-of-service attacks, malware infections, and unauthorized access attempts. Another QoS study by [29], [30] also emphasized the role of IDPS in maintaining QoS. Specifically, their study focused on using IDPS in wireless sensor networks. By deploying IDPS in wireless sensor networks, they observed improved QoS metrics such as network reliability, latency, and packet delivery ratio.

Another relevant study, conducted by [31], explores the approach of integrating a Network Intrusion Detection System (NIDS) and a Host-based Intrusion Detection System (HIDS), which can yield more optimal results in addressing security threats. In this approach, Snort is employed as NIDS to detect network-based intrusions by implementing rules capable of recognizing attack patterns. On the other hand, OSSEC functions as HIDS and effectively detects threats at the host level through log analysis, integrity monitoring, and rootkit detection. Both systems complement each other, with NIDS focusing more on network traffic analysis while HIDS concentrates on device and system protection at the host level.

The study by [32] proposes an analytical queuing model for assessing the impact of IDPS performance on network QoS. It explores the trade-off between security and QoS, demonstrating how enhancing security can lead to improved performance, albeit with some trade-offs. The study by [33] employs a multi-objective Bat algorithm to optimize security and QoS in a real-time operating system. It efficiently selects optimal security policies, ensuring minimal disruptions to Quality of Service. These studies offer valuable insights into enhancing network security and QoS through innovative IDPS approaches, highlighting the importance of balancing security measures with network performance considerations.

Contributions from other research, as presented in the studies by [33]–[38], also provide valuable insights within the domain of IDPS. Researchers examine diversity analysis for open-source IDS, aiding security architects in optimizing system performance. The study in [34] proposes a comprehensive multi-cloud integration security framework incorporating honeypots, significantly enhancing attack detection accuracy. The research in [35] introduces SYNGuard, a dynamic threshold-based SYN flood attack detection and mitigation system in Software-Defined Networks (SDNs), and compares the performance of Snort and Zeek IDS. Researchers [36] and [37] present policy-based security configuration management for IDPS, demonstrating its effectiveness using real-world intrusion detection datasets. Meanwhile, [38] analyzes password attacks via honeypots using machine learning techniques to unveil valuable password attack patterns.

Despite the significant insights provided by previous studies regarding the effectiveness and performance of IDPS systems, a comprehensive analysis of Distributed Denial of Service (DDoS) attacks, particularly ICMP Flood and SYN Flood attacks, on networking systems still needs to be improved. This research aims to fill this gap by evaluating the capabilities of IDPS systems such as Snort, Suricata, Zeek, OSSEC, and Honeypot Cowrie within network traffic. Through meticulous experiments, including real attack scenarios and calculations of Quality of Service (QoS) parameters such as throughput, jitter, delay, and packet loss during ICMP Flood and SYN Flood attacks, this study aims to provide valuable insights for network administrators, cybersecurity professionals, and organizations. The ultimate goal is to assist decision-makers in selecting and implementing the most suitable IDPS tools to safeguard their infrastructure against DDoS attacks, particularly in the context of ICMP Flood and SYN Flood attacks.

In the subsequent sections of this paper, the comprehensive analysis of Intrusion Detection and Prevention Systems (IDPS) in the context of DDoS attacks is explored. Following this introduction, the research methodology is described in Section II. Section III presents the results and findings of the experiments, including an evaluation of Snort, Suricata, Zeek, OSSEC, and the honeypot Cowrie. In Section IV, conclusions are provided based on the results and discussions on potential future works to enhance network security further.

II. Method

This research employs an experimental methodology to evaluate the performance of various intrusion detection tools (Snort, Suricata, Zeek, Ossec, and Honeypot Cowrie) in handling specific cyber-attacks, including ICMP Flood and SYN Flood. The research objective is to analyze how each tool responds to the attacks regarding key performance metrics. The independent variables consist of the intrusion detection tools, while the dependent variables include Delay, Jitter, Throughput, and Packet Loss measured before and after the attacks. In Fig. 1, the experimental design encompasses controlled experiments, where each tool is subjected to the same attack scenarios under consistent network conditions.

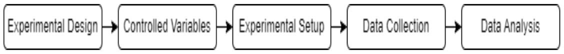


Fig. 1. Research method.

The experimental setup includes deploying the selected tools in a test network environment, and the attacks are initiated to evaluate the detection and response capabilities. The experimental setup involves deploying the selected intrusion detection tools within a controlled test network environment. Subsequently, targeted cyber-attacks are initiated to rigorously evaluate and assess each tool's detection and response capabilities. This evaluation allows for a comprehensive analysis of their performance under realistic attack scenarios, providing valuable insights into their effectiveness in safeguarding computing systems against potential threats.

Data collection involves meticulously recording each tool's performance metrics during the attack simulations. Throughout the simulations, relevant performance data, including Delay, Jitter, Throughput, and Packet Loss, is carefully documented for each detection tool. Quality of Service (QoS) is a method used to measure the quality of a network and determine the level of service it provides. QoS measures specific performance characteristics such as Delay, Jitter, Throughput, and Packet Loss, which are associated with a service [39], [40].

1) Throughput: Throughput refers to the actual bandwidth measured at a specific time when sending a file. Unlike bandwidth, which is measured in bits per second (bps), throughput better represents the actual bandwidth at a specific moment and under certain network conditions, particularly when downloading a particular file. It is calculated as the total number of successfully transmitted data (in bits) divided by the total time taken to transmit that data (in seconds):

$$Throughput = \frac{Total\ amount\ of\ \ transmitted\ data}{\text{Total time to transmit the data}} \qquad (1)$$

2) Packet loss: Packet Loss is the percentage of packets lost during data transmission. Various factors, such as weak signals in the network, network hardware errors, or environmental interference, can cause this. Packet Loss is a critical parameter that illustrates the number of lost packets due to collisions and congestion in the network. It is calculated as follows:

$$Packet\ Loss = \frac{Number\ of\ lost\ packets}{Total\ Number\ of\ packets\ sent} x100\% \qquad (2)$$

3) Jitter: Jitter is the variation in delay (time difference) between packets in the network, which is influenced by the queue length when processing data. It is affected by the traffic load and the number of packets (congestion) in the network, particularly during periods of high traffic. Jitter is calculated using the following equation:

$$Jitter = \frac{Total\ delay\ variation}{Total\ amount\ of\ transmitted\ data} \qquad (3)$$

4) Delay: Delay or Latency is the time it takes for data to travel from the source to the destination. The delay is influenced by distance, physical media, congestion, and processing times. It is calculated as follow:

$$Delay = \frac{Total\ delay}{Total\ amount\ of\ transmitted\ data} \qquad (4)$$

In cybersecurity, particularly in defending against Distributed Denial of Service (DDoS) attacks, IDPS plays a pivotal role. To bolster the effectiveness of IDPS in countering the ever-evolving DDoS threats, it becomes imperative to incorporate more analytical metrics. One such metric that merits heightened attention is the Detection Rate (DR) [41], calculated as follows:

$$Detection\ Rate = \frac{True\ Positive}{\text{True Positive+False Negative}} \qquad (5)$$

The Detection Rate (DR) is a critical metric that gauges the system's ability to identify genuine DDoS attacks accurately among all positive instances. True Positives (TP) represent instances where the IDPS correctly identifies and labels a legitimate DDoS attack. At the same time, False Negatives (FN) indicates instances where the system fails to detect a real DDoS threat, potentially leading to a security breach. In the DDoS mitigation landscape, the significance of DR cannot be overstated. It is a cornerstone for evaluating the IDPS aptitude to identify and thwart DDoS attacks precisely. Achieving a high DR is paramount as it minimizes the risk of false negatives, ensuring that legitimate DDoS threats do not go undetected.

By adopting this approach, this research acquires comprehensive and detailed data on the performance of each detection tool under various attack scenarios. Subsequently, data analysis entails statistical comparisons to determine significant differences in performance metrics between the tools. The data analysis process encompasses conducting thorough statistical comparisons to discern notable variations in performance metrics among the different detection tools. Through the application of advanced statistical techniques, the aim is to identify any statistically significant differences in the performance of each tool. This rigorous analysis enables us to gain valuable insights into the relative strengths and weaknesses of the detection tools, facilitating a comprehensive assessment of their capabilities in handling diverse cyber-attacks.

The rigorous experimental methodology aims to provide reliable insights into the efficiency and effectiveness of intrusion detection tools in diverse computing environments, particularly under varying attack conditions. Through comprehensive evaluations and controlled experiments, valuable data is sought to assess the capabilities and performance of these tools in safeguarding computing systems against a wide range of potential cyber threats. Doing so aims to establish a robust understanding of IDPS Tools, enhance cybersecurity practices, and ensures a more secure computing landscape.

III. Result and Finding

A. Experimental Design

The experimental design employed in this study involved conducting controlled experiments to evaluate the performance of each intrusion detection tool under consistent network conditions. All selected tools were subjected to the same attack scenarios in a controlled test network environment to ensure a fair and unbiased assessment. For the attack scenario in Fig. 2, the researchers utilized a computer laboratory comprising ten computers infiltrated with DDoS bots controlled by an attacker operating the handler. This simulation was used to launch attacks on a server, from which the necessary data was obtained during the testing of IDPS (Intrusion Detection and Prevention System) tools, including Snort, Suricata, Zeek, Ossec, and Honeypot Cowrie. This simulation aimed to assess the IDPS tools' performance in detecting and responding to the DDoS attacks orchestrated by the attacker through the compromised bots. The data collected from these simulated attacks served as crucial input for evaluating and analyzing the effectiveness of each IDPS tool in defending against such cyber threats.

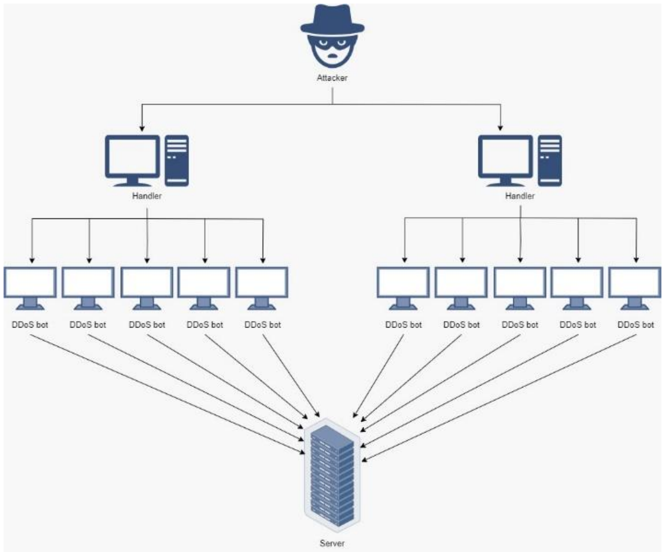


Fig. 2. Attack scenario.

Through controlled experiments, the aim was to eliminate any potential confounding variables and ensure that the observed differences in performance metrics were solely attributed to the capabilities of the intrusion detection tools. Each tool underwent testing under identical conditions, including network traffic, attack intensity, and duration. This standardized approach allowed for objectively comparing the tools' performance and drawing meaningful conclusions about their efficacy in detecting and mitigating various cyber-attacks. The performance metrics, such as RAM usage, CPU utilization, network throughput, delay, jitter, and packet loss, were carefully monitored and recorded during the attack simulations for each tool. Furthermore, to enhance the reliability of the findings, the experiments were repeated multiple times to account for any random variations and ensure the consistency of the results. The aggregated data from the repeated experiments provided a more robust basis for analysis and interpretation.

B. Controlled Variables

A carefully selected set of hardware specifications was strategically employed to ensure the successful acquisition of pertinent data for the research. These specifications were pivotal in establishing a robust experimental environment, enabling controlled experiments and the meticulous recording of performance metrics for the intrusion detection tools under investigation. With utmost attention to detail, specific hardware components were carefully chosen and implemented, tailored precisely to align with the research objectives. The following hardware specifications in Table I were utilized to facilitate data collection.

TABLE I. Experimental Hardware Tools

No	Hardware	Version	Number of Tools	Ip Number
1	switch	cisco sf95d-16 16-port 10/100	2 unit	192.168.100.150 & 192.168.100.151
2	computer server	server dell t150 xeon e-2324g	1 unit	192.168.100.154
3	computer server idps	server dell t40 xeon e-2224g	1 unit	192.168.100.153
4	computer idps console	all in one (aio) pc dell optiplex 7440	1 unit	192.168.100.152
5	computer agent	asus pc all in one v222gak wa141t - dualcore	10 unit	192.168.100.1-10
6	computer handler	asus pc all in one v222gak wa141t - dualcore	2 unit	192.168.100.11 & 192.168.100.12
7	computer attacker	hp pavilion aero 13 be2001au ryzen 5 7535u	1 unit	192.168.100.13

Meticulously designed and implemented a network topology for this research, as illustrated in Fig. 3, which comprised a carefully selected set of computers, each assigned specific roles. At the heart of the topology, the computer server served as a centralized repository for data. At the same time, the deployment of IDPS tools spanned across multiple computers, including servers, effectively safeguarding the network traffic from potential DDoS attacks. The assignment of IP addresses was skillfully managed through the switch, distributing the network across 13 computers. Among these designated systems, ten were dedicated to functioning as agent botnets for DDoS, two served as handlers with control over the agents, and one acted as the attacker. This research opted for the Kali Linux 2023.1 operating system, facilitating the smooth integration of essential intrusion detection tools, namely Snort, Suricata, Zeek, Ossec, and Honeypot Cowrie. This research employed Wireshark 4.0 as the chosen monitoring tool to ensure efficient network traffic monitoring.

The hardware setup and carefully crafted network topology laid the foundation for the controlled experiments, enabling us to systematically assess the performance of each intrusion detection tool under varying attack scenarios. By employing a standardized approach, reliability and accuracy in research results were ensured, providing the means to make informed evaluations regarding the capabilities and effectiveness of these tools in countering diverse cyber threats.

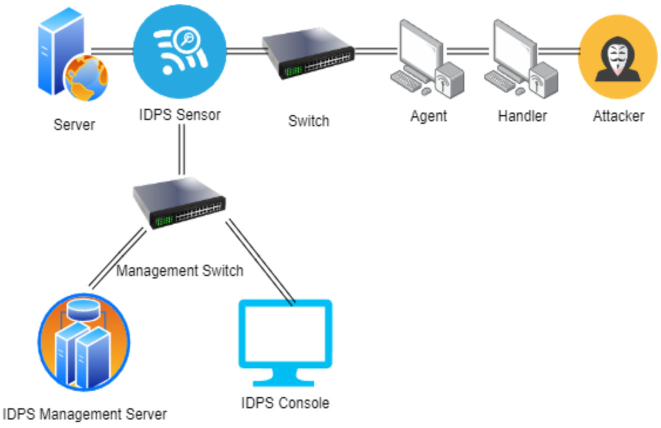
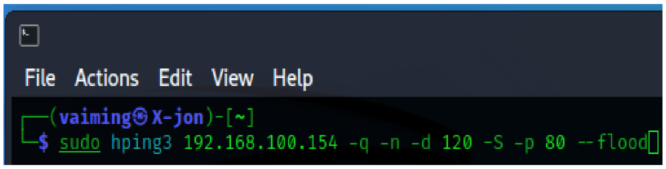


Fig. 3. Network topology.

C. Experimental Setup

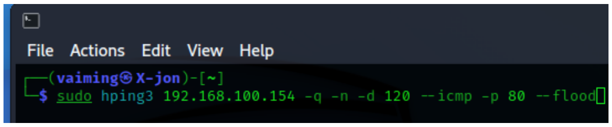
The experimental Setup section of this research focuses on the systematic deployment and evaluation of several intrusion detection tools, namely Snort, Suricata, Zeek, Ossec, and Honeypot Cowrie. Each tool is selected individually and installed with its respective configurations. Subsequently, comprehensive testing assesses their performance in handling DDoS attacks, specifically through ICMP Flood and TCP SYN Flood.

Initiate the evaluation process, the server is configured with rules specific to each IDPS tool, and simulated attacks are launched from an attacker's PC to the server using the DDoS tool Hping3. The commands for the SYN Ddos attack and Icmp Dodos attack simulations are provided as follows in Fig. 4 and Fig. 5:



```
┌──(vaiming㉿X-jon)-[~]
└─$ sudo hping3 192.168.100.154 -q -n -d 120 -S -p 80 --flood
```

Fig. 4. SYN DDoS attack.



```
┌──(vaiming㉿X-jon)-[~]
└─$ sudo hping3 192.168.100.154 -q -n -d 120 --icmp -p 80 --flood
```

Fig. 5. ICMP DDoS attack.

The server's response to the attack is initially observed when protected by the Snort tool with the IPS command. The objective is to determine whether Snort can successfully detect and generate warnings for the simulated attacks. Once the results are obtained from the Snort testing, the same evaluation process is repeated using the other selected tools, Suricata, Zeek, Ossec, and Honeypot Cowrie, on the server.

D. Data Collection

IDPS like Suricata, Zeek, Ossec, and Honeypot Cowrie play pivotal roles in safeguarding digital environments from malicious activities. They operate as the first line of defense, tirelessly monitoring network traffic and system logs. To assess the efficacy of these systems, metrics like the Detection Rate (DR) are paramount. In the context of this research, the Detection Rate (DR) emerges as a pivotal metric in assessing the performance of the IDPS. With 128,027 True Positives, signifying accurate identifications of actual intrusion attempts, and a relatively low 4,241 False Negatives (FN), which represent instances where genuine threats were not detected, the IDPS demonstrates a robust capability in effectively distinguishing malicious activities from benign network traffic. The DR, calculated as the ratio of True Positives to the sum of True Positives and False Negatives (TP / (TP + FN)), reflects the system's ability to capture a high proportion of genuine intrusions. This value is a crucial requirement in this research, where achieving a DR of 128,027, or 97% of all intrusion attempts, is integral to minimizing the risk of false negatives and ensuring the thorough protection of digital assets.

In Table II and Table III, the performance of each IDPS tool was carefully observed during the ICMP flood attack. Snort exhibited a slight increase in RAM usage by 0.07%, followed by a slightly larger increase in CPU usage by 5.00%. Conversely, Suricata experienced a more pronounced rise in RAM usage by 0.19% and a substantial increase in CPU usage by 16.67%. Zeek demonstrated minimal fluctuations in RAM and CPU usage, with only 0.09% and 0.00% changes, respectively. OSsec recorded a moderate uptick in RAM and CPU usage, showing increases of 0.08% and 2.70%, respectively, highlighting its ability to manage ICMP flood attacks without significant overhead.

In contrast, Honeypot Cowrie displayed a noticeable increase in RAM usage by 0.13%, followed by a slightly more substantial rise in CPU usage of 3.61%. Network performance during the ICMP flood attack revealed diverse trends. Snort indicated a moderate upswing in network throughput, measuring 578.79 kb/s. Conversely, Suricata, Zeek, and Ossec experienced slight decreases in network throughput by 1.13 kb/s, 0.66 kb/s, and 0.05 kb/s, respectively. Remarkably, honeypot cowrie showcased a significant spike in network throughput, reaching 701.07 kb/s, underscoring its efficiency in addressing ICMP flood attacks.

TABLE II. COMPUTER PERFORMANCE BEFORE ATTACK

IDPS Tools	DDoS Attack	Before Attack		
		Ram (%)	*Cpu (%)*	*Network (kb/s)*
snort	icmp flood	27.30	0.17	32.38
	syn flood	27.56	0.22	36.4
suricata	icmp flood	32.19	0.66	34.39
	syn flood	36.26	0.82	35.40
zeek	icmp flood	32.69	0.22	32.78
	syn flood	32.57	0.52	32.61
ossec	icmp flood	32.57	0.37	32.70
	syn flood	33	0.22	32.65
honeypot cowrie	icmp flood	29.75	0.415	32.65
	syn flood	31.91	0.52	35.90

TABLE III. COMPUTER PERFORMANCE AFTER ATTACK

IDPS Tools	DDoS Attack	After Attack		
		Ram (%)	*Cpu (%)*	*Network (kb/s)*
snort	icmp flood	27.37	5.17	611.17
	syn flood	27.74	8.11	3238.76
suricata	icmp flood	32.38	7.67	856.26
	syn flood	36.4	9.02	3666.62
zeek	icmp flood	32.78	3.12	983.34
	syn flood	32.61	6.07	3569.07
ossec	icmp flood	32.65	8.08	566.89
	syn flood	32.39	7.73	3,596.54
honeypot cowrie	icmp flood	29.88	6.42	733.72
	syn flood	32.07	8.57	3452.69

Turning to the SYN Flood attack, the IDPS tools once again exhibited diverse patterns of performance adjustment. Snort displayed a slight increase in RAM usage by 0.18%, followed by a more substantial rise in CPU usage by 27.27%. Suricata showcased a more significant uptick in RAM usage by 0.74% and a noteworthy increase in CPU usage of 17.07%. Zeek demonstrated minimal RAM and CPU usage variations, with only 0.09% and 0.00% changes, respectively. Conversely, OSSEC recorded slightly decreased RAM usage by 0.22%, while CPU usage increased by 0.00%. Honeypot Cowrie experienced a noticeable increase in RAM usage by 0.13% and a relatively significant rise in CPU usage of 6.59%.

Network performance during the SYN Flood attack also revealed distinct behavior. Snort and Suricata exhibited moderate increases in network throughput, measuring 2205.59 KB/s and 1587.86 KB/s, respectively, demonstrating their efficient responses to SYN Flood attacks. Zeek demonstrated a slight decrease in network throughput by 1.34 KB/s, while OSSEC experienced a significant surge in network throughput, reaching 1929.89 KB/s. Notably, honeypot cowrie significantly increased network throughput, measuring 2413.52 KB/s, further highlighting its robustness in handling SYN Flood attacks.

These observations suggest differences in the tools' ability to detect and counter such attacks. In the case of ICMP Flood attacks, it was observed that certain IDPS tools experienced notable increases in resource utilization, such as RAM, CPU, and network throughput, after the attacks. These observations imply varying sensitivity and adaptability of these tools to the attack type. Similarly, during SYN Flood attacks, the IDPS tools exhibited diverse patterns of resource usage alterations, suggesting differences in their ability to detect and counter such attacks. The observed changes in performance metrics underscore the need for a nuanced evaluation of IDPS tools under distinct attack scenarios.

This study in Fig. 6 conducted QoS measurements for throughput during ICMP Flood and SYN Flood DDoS attacks using different Intrusion Detection and Prevention Systems (IDPS) tools, namely Snort, Suricata, Zeek, Ossec, and Honeypot. The results indicated variations in throughput values across these tools for both attack types. Among the tested tools, Snort demonstrated the highest throughput during ICMP Flood attacks, reaching 26,485 bits per second. At the same time, Suricata and Zeek showed similar throughput values at 32,400 and 32,438 bits per second, respectively. Ossec and Honeypot yielded slightly lower throughputs at 26,052 and 39,897 bits per second, respectively.

For SYN Flood attacks, Snort exhibited the highest throughput of 29,701 bits per second, followed closely by Honeypot at 34,701 bits per second. Suricata and Zeek yielded lower throughput values at 25,029 and 21,970 bits per second, respectively. Ossec demonstrated the lowest throughput among the tested tools for SYN Flood attacks, registering 21,970 bits per second.

Snort exhibited strong throughput values for both ICMP Flood and SYN Flood attacks, making it a viable choice for mitigating these attack types. Suricata and Zeek also demonstrated competitive throughput, indicating their potential effectiveness in handling DDoS attacks.

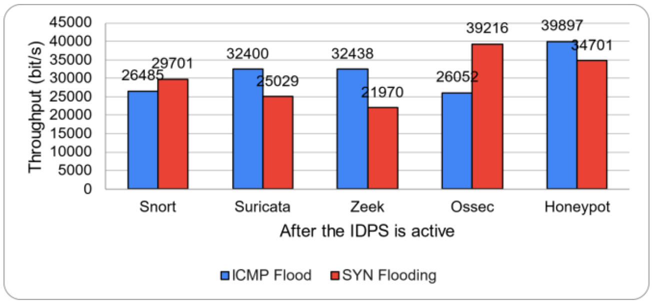


Fig. 6. Throughput DDoS attack.

The Delay values for different types of DDoS attacks were evaluated using various Intrusion Detection and Prevention Systems (IDPS), including Snort, Suricata, Zeek, Ossec, and Honeypot Cowrie in Fig. 7. For ICMP Flood attacks, Snort exhibited a delay of 223.53 ms, Suricata had a delay of 183.85 ms, Zeek showed a delay of 45.59 ms, Ossec had a delay of 130.9 ms, and Honeypot Cowrie displayed the lowest delay of 22.88 ms. Similarly, for SYN Flood attacks, Snort demonstrated a delay of 187.17 ms, Suricata had a delay of 104.59 ms, Zeek exhibited a delay of 17.9 ms, Ossec showed a delay of 60.8 ms, and Honeypot Cowrie had a delay of 187.2 ms. These Delay values provide insights into the responsiveness of each IDPS in detecting and mitigating ICMP and SYN Flood attacks. It is worth noting that Honeypot Cowrie consistently displayed lower Delay values, indicating its potential effectiveness in handling such attacks with minimal delay.

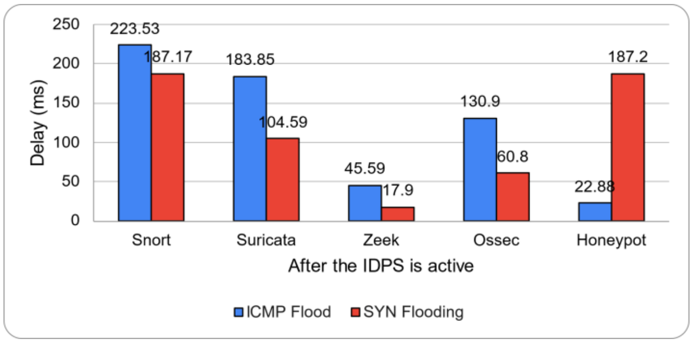


Fig. 7. Delay DDoS attack.

Analyzing jitter values across various DDoS attack scenarios and corresponding Intrusion Detection and Prevention Systems (IDPS) tools reveals distinct patterns in Fig. 8. In the case of ICMP Flood attacks, Zeek stands out with remarkably low jitter (0.88 ms), indicating stable and consistent packet delay. Conversely, Snort (7.37 ms), Suricata (1.8 ms), Ossec (1.01 ms), and Honeypot (11.5 ms) exhibit comparatively higher jitter values, suggesting potential fluctuations in delay times. A similar trend emerges during SYN Flooding attacks, where Zeek maintains its superior performance in jitter control (2.02 ms). Suricata (5.63 ms) and Ossec (6.08 ms) demonstrate increased jitter, while Snort (1.81 ms) and Honeypot (1.82 ms) exhibit relatively better control. These findings underscore Zeek's consistent jitter management capabilities across both attack types.

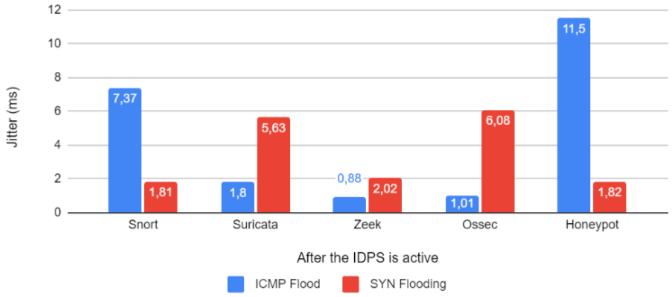


Fig. 8. Jitter DDoS attack.

The investigation into packet loss rates during ICMP Flood and SYN Flooding attacks, evaluated across a range of Intrusion Detection and Prevention Systems (IDPS) tools, yielded distinct outcomes. In Figure 9, Snort and Suricata exhibited minimal packet loss, recording percentages of 0.32% and 0.44% for ICMP Flood and 0.56% and 0.29% for SYN Flooding, respectively. Zeek displayed effective packet loss mitigation, with rates of 0.25% for ICMP Flood and 0.14% for SYN Flooding. Ossec and Honeypot Cowrie demonstrated slightly higher packet loss percentages, at 0.33% and 0.19% for ICMP Flood and 0.56% for SYN Flooding. These findings illuminate the diverse packet loss responses of IDPS tools to specific attack scenarios, empowering network administrators and cybersecurity practitioners with valuable insights for optimizing DDoS protection strategies.

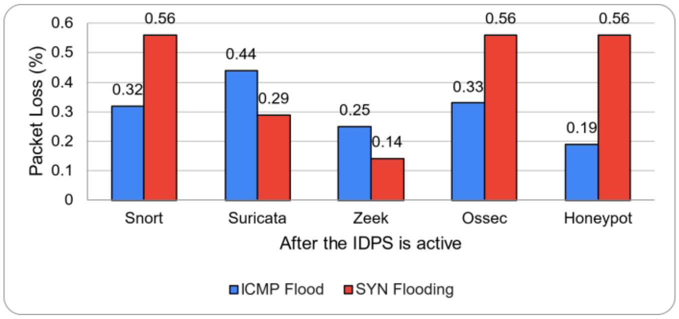


Fig. 9. Packet loss DDoS attack.

The Quality of Service (QoS) analysis of the network performance before and after different attack scenarios, as measured by various metrics, offers valuable insights into the effectiveness of the Intrusion Detection and Prevention Systems (IDPS) tools. Among the tested tools, Snort and Zeek consistently demonstrate a relatively robust ability to mitigate the impact of attacks on network Delay and Packet Loss. Suricata and OSSEC, on the other hand, exhibit more susceptibility to disruptions caused by the attacks, with increased Delay, Jitter, and Packet Loss, especially evident in SYN Flood attacks. Notably, Honeypot Cowrie proves adept at maintaining network stability during ICMP Flood attacks, showcasing lower Jitter and relatively stable Throughput. These observations underline the varying QoS responses of different IDPS tools to distinct attack types, providing crucial insights for making informed decisions regarding network defense strategies. The ICMP Flood attacks generally result in increased Delay and Packet Loss, while the SYN Flood attacks tend to affect Delay, Jitter, and sometimes throughput. Among the IDPS tools, Snort and Zeek exhibit relatively better network performance maintenance, while Suricata and OSSEC show more impact from the attacks. Honeypot Cowrie maintains network performance relatively well, particularly for ICMP Flood attacks. It is important to note that these observations provide insights into how each IDPS tool responds to specific attack types regarding QoS metrics.

E. Data Analysis

The data analysis phase serves as the foundation of the investigation, shedding light on the performance dynamics of distinct intrusion detection tools when confronted with diverse cyber threats. This research systematically compared key performance metrics through meticulous experimentation and keen observation before and after simulated attacks. The analytical focus encompassed critical parameters, including delay, jitter, throughput, and packet loss, offering a comprehensive view of each tool's response. Notably, Snort exhibited commendable efficiency in managing ICMP Flood attacks, showcasing minimal network latency and jitter disruption. Suricata demonstrated adeptness in mitigating SYN Flood attacks with modest fluctuations. Zeek's proficiency shone through its stable network throughput during ICMP Flood scenarios.

Meanwhile, OSSEC displayed a robust defense mechanism against ICMP Flood attacks, containing packet loss within acceptable bounds. Honeypot Cowrie effectively mitigated packet loss while experiencing elevated jitter during ICMP Flood incidents. The analysis, bolstered by robust statistical techniques, revealed nuanced performance differences allowing us to conclude each tool's strengths and limitations. These insights offer essential guidance for practitioners configuring intrusion detection tools advancing cybersecurity defense strategies with precision.

This research analysis of countering ICMP DDoS and SYN Flood attacks highlighted varying degrees of efficacy among the IDPS tools. Snort stood out in addressing ICMP Flood attacks, effectively minimizing network disruption, latency, and jitter. Similarly, Suricata exhibited proficiency in mitigating SYN Flood attacks, maintaining stable network throughput, and responding to anomalous traffic patterns. On the other hand, Zeek displayed commendable network throughput during ICMP Flood scenarios but showed moderate fluctuation against SYN Flood attacks. While OSSEC contained packet loss during ICMP Flood incidents, it faced challenges maintaining network stability under SYN Flood onslaughts. Honeypot Cowrie, resilient against packet loss, experienced elevated jitter during ICMP Flood attacks. These findings collectively suggest that Snort and Suricata are potent contenders for countering ICMP DDoS and SYN Flood attacks, offering consistent and robust responses. The nuanced strengths and limitations underscore the importance of tailored tool selection based on the specific threat landscape and operational requirements.

A comprehensive analysis of QoS data and computer/networking performance metrics reveals that Snort is a standout performer in countering ICMP Flood attacks. This conclusion is drawn from consistent and commendable results across various parameters. Snort effectively mitigated delays and jitter, ensuring optimal network responsiveness and maintaining impressive throughput levels, all while demonstrating minimal packet loss. Moreover, Snort efficiently utilized CPU and RAM resources, indicating its ability to handle ICMP Flood attacks without overstraining the system. These findings position Snort as the most robust IDPS tool for effectively countering ICMP Flood attacks, making it a compelling choice for defending against such threats and ensuring network stability and performance.

Similarly, the analysis indicates that Zeek is the most effective IDPS tool for countering SYN Flood attacks. Zeek consistently demonstrated remarkable performance in minimizing delays and jitter during SYN Flood attacks, maintaining stable network responsiveness. Additionally, Zeek maintained competitive throughput levels and remarkably low packet loss, showcasing its proficiency in managing SYN requests. From a computer and networking performance standpoint, Zeek efficiently allocated CPU and RAM resources, indicating its capability to handle SYN Flood attacks without burdening the system. Overall, Zeek's strong performance across QoS metrics and resource management makes it the optimal choice for countering SYN Flood attacks and safeguarding network stability.

In this comparative analysis of the results, we have examined this research alongside relevant previous studies. The study by [28], [35] introduces an analytical model for assessing IDPS configurations, emphasizing theoretical modeling. In contrast, the results of this research delve into practical IDPS implementation within a networking system environment to defend against specific threats, emphasizing real-world application. The studies cited as [11], [12], [29], [30], [33], [36]–[38], on the other hand, differ significantly from this research outcome. Given these variations in goals and approaches, direct result comparisons can be challenging. This study's results highlight practical implementation and threat defense, distinguishing it from theoretical modeling and the differing contexts in previous studies.

IV. Conclusion and Future Works

This study undertook a comprehensive analysis of diverse Intrusion Detection and Prevention Systems (IDPS) tools, namely Snort, Suricata, Zeek, OSSEC, and Honeypot Cowrie, with a primary focus on their effectiveness in countering Distributed Denial of Service (DDoS) attacks. Through a meticulous evaluation encompassing aspects of network traffic analysis, Quality of Service (QoS) metrics, computer performance, and attack mitigation, this research gained insights into the capabilities of these tools. In this assessment, research revealed distinct performance characteristics for each IDPS tool. Snort excelled in network-based intrusion detection, efficiently identifying and countering threats at the network level. Suricata demonstrated prowess in packet processing and rule matching, making it a strong contender for network security. With its emphasis on comprehensive traffic analysis, Zeek offered valuable insights into network activity. OSSEC showcased robust host-based intrusion detection capabilities, providing effective log analysis and threat identification. Honeypot Cowrie displayed potential while highlighting areas for improvement in QoS metrics and computer performance. Regarding Quality of Service (QoS), the analysis unveiled Snort as the most effective IDPS tool in countering ICMP Flood and SYN Flood attacks, consistently exhibiting superior throughput, lower delay, minimal jitter, and commendable packet loss rates. These QoS metrics reflect Snort's adeptness in preserving network integrity and minimizing disruption during DDoS incidents.

Future research avenues include integrating advanced machine learning techniques into IDPS tools to optimize detection accuracy while minimizing false positives. Additionally, exploring the deployment of IDPS in dynamic cloud and hybrid environments, understanding their scalability, and adapting them to varying network conditions would provide valuable insights. In conclusion, this study provides valuable insights into the performance of diverse IDPS tools against DDoS attacks. By addressing identified limitations and pursuing avenues for future research, this research can advance the field of network security and contribute to developing resilient defense mechanisms against evolving cyber threats.

References

[1] W. A. Al-Khater, S. Al-Maadeed, A. A. Ahmed, A. S. Sadiq, and M. K. Khan, “Comprehensive review of cybercrime detection techniques,” IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3011259.

[2] S. T. Zargar, J. Joshi, and D. Tipper, “A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks,” IEEE

Communications Surveys and Tutorials, vol. 15, no. 4, 2013, doi: 10.1109/SURV.2013.031413.00127.

[3] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, “Botnet in DDoS Attacks: Trends and Challenges,” IEEE Communications Surveys and Tutorials, vol. 17, no. 4, 2015, doi: 10.1109/COMST.2015.2457491.

[4] M. Masdari and M. Jalali, “A survey and taxonomy of DoS attacks in cloud computing,” Security and Communication Networks, vol. 9, no. 16. 2016. doi: 10.1002/sec.1539.

[5] E. Alomari, S. Manickam, B. B. Gupta, S. Karuppayah, and R. Alfaris, “Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art,” Int J Comput Appl, vol. 49, no. 7, 2012, doi: 10.5120/7640-0724.

[6] S. Specht and R. Lee, “Taxonomies of Distributed Denial of Service Networks, Attacks, Tools, and Countermeasures,” 2003. [Online]. Available: www.princeton.edu

[7] M. K. Kareem, O. D. Aborisade, S. A. Onashoga, T. Sutikno, and O. M. Olayiwola, “Efficient model for detecting application layer distributed denial of service attacks,” Bulletin of Electrical Engineering and Informatics, vol. 12, no. 1, 2023, doi: 10.11591/eei.v12i1.3871.

[8] M. Shaohui, G. Tuerhong, M. Wushouer, and T. Yibulayin, “PCA mix-based Hotelling’s T2 multivariate control charts for intrusion detection system,” IET Inf Secur, vol. 16, no. 3, 2022, doi: 10.1049/ise2.12051.

[9] P. Sai Chowdary and D. Vinod, “Host Intrusion Detection System Using Novel Predefined Signature Patterns by Comparing Random Forest over Decision Tree Algorithm,” in Advances in Parallel Computing, 2022. doi: 10.3233/APC220092.

[10] J. Gabirondo-Lopez, J. Egana, J. Miguel-Alonso, and R. Orduna Urrutia, “Towards Autonomous Defense of SDN Networks Using MuZero Based Intelligent Agents,” IEEE Access, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3100706.

[11] K. Alsubhi and H. M. AlJahdali, “Intrusion detection and prevention systems as a service in could-based environment,” International Journal of Advanced Computer Science and Applications, vol. 9, no. 7, 2018, doi: 10.14569/IJACSA.2018.090738.

[12] A. H. B. Aighuraibawi, R. Abdullah, S. Manickam, and Z. A. A. Alyasseri, “Detection of ICMPv6-based DDoS attacks using anomaly based intrusion detection system: A comprehensive review,” International Journal of Electrical and Computer Engineering, vol. 11, no. 6. 2021. doi: 10.11591/ijece.v11i6.pp5216-5228.

[13] S. Laqtib, K. El Yassini, and M. L. Hasnaoui, “A technical review and comparative analysis of machine learning techniques for intrusion detection systems in MANET,” International Journal of Electrical and Computer Engineering, vol. 10, no. 3. 2020. doi: 10.11591/ijece.v10i3.pp2701-2709.

[14] P. Araujo et al., “Impact of Feature Selection Methods on the Classification of DDoS Attacks using XGBoost,” Journal of Communication and Information Systems, vol. 36, no. 1, 2021, doi: 10.14209/jcis.2021.22.

[15] N. Z. M. Safar, N. Abdullah, H. Kamaludin, S. A. Ishak, and M. R. M. Isa, “Characterising and detection of botnet in P2P network for UDP protocol,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 18, no. 3, 2020, doi: 10.11591/ijeecs.v18.i3.pp1584-1595.

[16] C. Oh, J. Ha, and H. Roh, “A survey on tls-encrypted malware network traffic analysis applicable to security operations centers,” Applied Sciences (Switzerland), vol. 12, no. 1, 2022, doi: 10.3390/app12010155.

[17] I. Mukhopadhyay, M. Chakraborty, and S. Chakrabarti, “A Comparative Study of Related Technologies of Intrusion Detection &amp; Prevention Systems,” Journal of Information Security, vol. 02, no. 01, 2011, doi: 10.4236/jis.2011.21003.

[18] M. Ozkan-Okay, R. Samet, O. Aslan, and D. Gupta, “A Comprehensive Systematic Literature Review on Intrusion Detection Systems,” IEEE Access, vol. 9. 2021. doi: 10.1109/ACCESS.2021.3129336.

[19] A. S. Putra and N. Surantha, “Internal threat defense using network access control and Intrusion Prevention System,” International Journal of Advanced Computer Science and Applications, vol. 10, no. 9, 2019, doi: 10.14569/ijacsa.2019.0100948.

[20] M. H. Kamarudin, C. Maple, T. Watson, and N. S. Safa, “A New Unified Intrusion Anomaly Detection in Identifying Unseen Web Attacks,” Security and Communication Networks, vol. 2017, 2017, doi: 10.1155/2017/2539034.

[21] M. Ahsan, K. E. Nygard, R. Gomes, M. M. Chowdhury, N. Rifat, and J. F. Connolly, “Cybersecurity Threats and Their Mitigation Approaches Using Machine Learning—A Review,” Journal of Cybersecurity and Privacy, vol. 2, no. 3, 2022, doi: 10.3390/jcp2030027.

[22] V. Vasilyev and R. Shamsutdinov, “Distributed Intelligent System of Network Traffic Anomaly Detection Based on Artificial Immune System,” 2019. doi: 10.2991/itids-19.2019.7.

[23] B. Hameed, A. AlHabshy, and K. ElDahshan, “Distributed Intrusion Detection Systems in Big Data: A Survey,” Al-Azhar Bulletin of Science, vol. 32, no. 1, 2021, doi: 10.21608/absb.2021.63810.1100.

[24] O. Alkadi, N. Moustafa, and B. Turnbull, “A Review of Intrusion Detection and Blockchain Applications in the Cloud: Approaches, Challenges and Solutions,” IEEE Access, vol. 8. 2020. doi: 10.1109/ACCESS.2020.2999715.

[25] T. Andrysiak, Ł. Saganowski, and W. Mazurczyk, “Network anomaly detection for railway critical infrastructure based on autoregressive fractional integrated moving average,” EURASIP J Wirel Commun Netw, vol. 2016, no. 1, 2016, doi: 10.1186/s13638-016-0744-8.

[26] I. Singh, S. Singhal, and V. Kumar, “Database intrusion detection using role and user level sequential pattern mining and fuzzy clustering,” International Journal of Engineering Research and Technology, vol. 13, no. 6, 2020, doi: 10.37624/ijert/13.6.2020.1173-1178.

[27] A. J. Alhasan and N. Surantha, “Evaluation of Data Center Network Security based on Next-Generation Firewall,” International Journal of Advanced Computer Science and Applications, vol. 12, no. 9, 2021, doi: 10.14569/IJACSA.2021.0120958.

[28] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, “A survey of intrusion detection techniques in Cloud,” Journal of Network and Computer Applications, vol. 36, no. 1. 2013. doi: 10.1016/j.jnca.2012.05.003.

[29] C. Birkinshaw, E. Rouka, and V. G. Vassilakis, “Implementing an intrusion detection and prevention system using software-defined networking: Defending against port-scanning and denial-of-service attacks,” Journal of Network and Computer Applications, vol. 136, 2019, doi: 10.1016/j.jnca.2019.03.005.

[30] H. Hendrawan, P. Sukarno, and M. A. Nugroho, “Quality of service (QoS) comparison analysis of snort IDS and Bro IDS application in software define network (SDN) architecture,” in 2019 7th International Conference on Information and Communication Technology, ICoICT 2019, 2019. doi: 10.1109/ICoICT.2019.8835211.

[31] D. W. Y. O. Waidyarathna, W. V. A. C. Nayantha, W. M. T. C. Wijesinghe, and K. Y. Abeywardena, “Intrusion Detection System with correlation engine and vulnerability assessment,” International Journal of Advanced Computer Science and Applications, vol. 9, no. 9, 2018, doi: 10.14569/ijacsa.2018.090947.

[32] K. Alsubhi, M. F. Zhani, and R. Boutaba, “Embedded Markov process based model for performance analysis of Intrusion Detection and Prevention Systems,” in GLOBECOM - IEEE Global Telecommunications Conference, 2012. doi: 10.1109/GLOCOM.2012.6503227.

[33] H. Asad and I. Gashi, “Dynamical analysis of diversity in rule-based open source network intrusion detection systems,” Empir Softw Eng, vol. 27, no. 1, 2022, doi: 10.1007/s10664-021-10046-w.

[34] T. Alyas et al., “Multi-Cloud Integration Security Framework Using Honeypots,” Mobile Information Systems, vol. 2022, 2022, doi: 10.1155/2022/2600712.

[35] M. Rahouti, K. Xiong, N. Ghani, and F. Shaikh, “SYNGuard: Dynamic threshold-based SYN flood attack detection and mitigation in software-defined networks,” IET Networks, vol. 10, no. 2, 2021, doi: 10.1049/ntw2.12009.

[36] S. R. M. Zeebaree, K. Jacksi, and R. R. Zebari, “Impact analysis of SYN flood DDoS attack on HAProxy and NLB cluster-based web servers,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 19, no. 1, 2020, doi: 10.11591/ijeecs.v19.i1.pp505-512.

[37] S. R. M. Zeebaree, K. H. Sharif, and R. M. Mohammed Amin, “Application Layer Distributed Denial of Service Attacks Defense

Techniques : A review," Academic Journal of Nawroz University, vol. 7, no. 4, 2018, doi: 10.25007/ajnu.v7n4a279.

[38] H. TAŞÇI, S. GÖNEN, M. A. BARIŞKAN, G. KARACAYILMAZ, B. ALHAN, and E. N. YILMAZ, "Password Attack Analysis Over Honeypot Using Machine Learning Password Attack Analysis," Turkish Journal of Mathematics and Computer Science, vol. 13, no. 2, 2021, doi: 10.47000/tjmcs.971141.

[39] 3Gpp-Ts-23.107, "3GPP TS 23.107: Quality of Service (QoS) Concept and Architecture," 3GPP:Technical Specification Group Services and System Aspects., vol. 0, no. Release 1999, 2009.

[40] F. L. Rodríguez, U. S. Dias, D. R. Campelo, R. de O. Albuquerque, S. J. Lim, and L. J. G. Villalba, "QoS management and flexible traffic detection architecture for 5G mobile networks," Sensors (Switzerland), vol. 19, no. 6, 2019, doi: 10.3390/s19061335.

[41] N. A. majeed Alhammadi, "Comparative study between (SVM) and (KNN) classifiers by using (PCA) to improve of intrusion detection system," Iraqi Journal of Intelligent Computing and Informatics (IJICI), vol. 1, no. 1, 2022, doi: 10.52940/ijici.v1i1.4.