Analysis of Snort Rules to Prevent Synflood Attacks on Network Security

Karmadenur
Technical Information Universitas Mercu Buana
Jalan Meruya Selatan No. 1, Kembangan Kebon Jeruk, Jakarta, Indonesia

Raka Yusuf
Universitas Mercu Buana
Jalan Meruya Selatan No. 1, Kembangan, Kebon Jeruk, Jakarta, Indonesia

ABSTRACT
Snort rules are a form of the database whose attack pattern is applied to a Snort server to filter out the types of attacks, so that the type of attack detected can be isolated, the Snort rule database must be updated so that if there are new types of attack patterns it can be found by Snort rules. This analysis will provide input to update regularly so that if a new type of attack is detected and can be detected. Snort rules to prevent SYN Flood attacks, the type of denial of service that has been formulated using loopholes when connecting to TCP / IP is done. In network security analysis it is very important to formulate an attack pattern that will attack the network so that it can be overcome by Snort rules. Maintaining a secure network from interference can be overcome by Snort rules. Analysis of Snort's rules is to prevent SYN Flood attacks on network security and makes it easier for administrators to report the types of attacks that enter Snort rules and make it easier to make policies improve based on the logs in Snort rules.

General Terms
Snort Rules to Prevent Synflood Attacks

Keywords
Snort Rules, SYN Flood attack, Option Rules, DDoS

1. INTRODUCTION
Network security is an important aspect that must be considered to maintain system stability and smoothness. Snort is a security application tool that serves to detect network intrusions including infiltration, attack, and various forms of threats that are used to simplify analyzing incoming data traffic to minimize the presence of packages that are awkward and unwanted, which can result in the network not functioning normally (hangs) so that data traffic on the network is interrupted. The snort that has been running in command line mode works well to perform its functions and rules.

Some attacks on servers often occur which sometimes can not be separated from reality even though not all goals are carried out based solely on politics or business. For this reason network security requires Snort rules, which are databases that contain signature attack patterns. Snort rules must always be updated regularly so that after a new attack technique appears the attack can be detected, as well as preventing SYN Flood attacks. Also, the types of attacks that are currently developing are like DoS attacks which are currently developing into Distributed Denial of Service (DDoS). Some types of attacks that are currently carried out by attackers with various types of attacks usually attack occur very much distributed so that when the attack occurs, it can immobilize the target very quickly. Making Snort rules can prevent flood SYN attacks such as a type of DoS attack, where Traffic Flooding attacks will make full network traffic come quickly so that official users don't make it into the network system. Analysis of Snort rules to prevent SYN Flood attacks on network security adds updates that then provide Snort rules so that it can make it easier for administrators to see reports of the types of attacks that enter Snort rules.

2. RELATED RESEARCH
In the study discussed the rules of Snort to prevent Syn Flood attack attacks, there are several main things to detect and prevent an attack on network interference. In developing network security it is very important to produce and counteract the types of attacks that will cause the system to die, to overcome this can be overcome with the latest rules. Anomaly detection cannot be considered a perfect solution for new threats. Transmission Control Protocol (TCP) Syn Flood attacks carry out a type of attack that is continuous, to maintain competent and efficient types of attacks to reduce Syn Flood attacks Snort rules are needed [1][2]. Syn flood attacks network security that happens a lot. So to prove the proposed security efficiency so that it can be integrated into Snort's rules by using Code Refactoring, which is an external behavior that deliberately reflects the code without changing the behavior from the outside to make a strong intrusion detection machine [3].

Signature-based intrusion detection system using machine learning algorithms. Another problem that has the potential to generate additional insight is the analysis and rules. This algorithm was developed to extract protocol information from Snort rules and determine distribution in the ruleset [4].

Whereas in the explanation the rules are made to filter the packages sent by network users. In the same pattern the Snort system can be automated or with initial rules, to integrate honeypot and IDS can activate Snort rules by the server, rules that have been automatically created will filter packets sent by the network [5] [6].

Method or analysis in extracted features using the back-propagation algorithm. On the other hand, the intrusion has been classified as a Denial of Service (DoS) category and an attack probe that is warned in storage through log data. Using the Snort backpropagation algorithm application, the detection of unknown attacks is only one way the result is reduced attack detection [7] [8].

Distributed Denial of Service (DDoS) combines mitigation schemes in attacks currently with new models to IT devices, most of which are used to produce DDoS attacks, namely collaboration that distributes using Distributed Denial of Service (DDoS) attack mitigation schemes [1] [5]. Testing rules to be able to reveal various benefits of information regarding the type of network traffic that is considered dangerous. The first phase of the algorithm was developed to extract protocol information from Snort rules and to determine

the distribution in the set of rules that are more detailed and updated. Snort is the most popular software used for network detection, Snort Flexibility is easily configured in detecting intrusion [9] [10].

The method proposes how to calculate the network topology between Snort rules based on set theory [11] [4]. Lately, the paradigm has gotten much attention for implementing a data center network that provides efficient solutions for security [8] [12]. The development of this very high network system, the huge amount of data traffic needs to be analyzed by fiber processed using high-speed infrastructure. To advance the performance of network intrusion detection systems and reduce the time of network traffic processing, intrusion detection systems on high-speed networks are needed Parallel techniques as an alternative [13] [14].

Syn flooding Attack is one of the most popular types of DDoS attacks known as TCP Syn Flood Attack. TCP Syn flood attacks are generally done to reduce server resources [2]. The normal data sender from the client is done using Syn to synchronize to the destination server, after that the destination server will answer the client with an Ack form with evidence that the data is received. Present the Internet of Things (IoT) that will get a greater threat of attacks from DoS or DDoS attacks, so it is necessary to increase security against network attacks [15].

This type of attack the server will be filled with many Syn packages, if the client sends a Syn packet, the server also immediately sends Syn-Ack data to the client. Then the server continuously records and creates a backlog queue to interfere with the Ack response from the client sending the Syn packet, the backlog provided is very small. In the backlog queue when it is full, the system will not accommodate other TCP Syn packets that enter, resulting in the server hangs [16]. The TCP Syn-Ack packet queue in the backlog can only be wasted when the specified time is up, TCP and shows that there is no response from the sender. The TCP client connection to the server can be explained as follows:

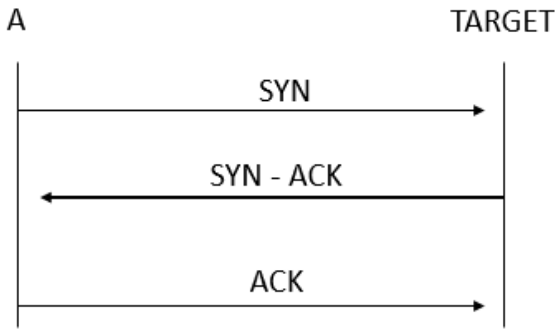

Figure 1. Process of Syn Flooding Attack.

Explanation of the image above addressing side A will send a packet with TCP Syn, sending by manipulating a fake address such as the Target side can be called IP Spoofing. The purpose of receiving the Syn shipment is then sent to the intended destination Syn-Ack while the storage request from side A in the queue will be processed if the target Ack is the side A. Because the process of sending Syn to Ack is repeated repeatedly, the server service becomes die. Like this attack on a network attack, it is called the Daniel of Service attack.

So that the server, which should work without stopping every day, will stop and cannot receive data packets sent from any side, even though the server must serve clients during day and night without stopping. With the occurrence of DoS attacks, the service is stopped, this is how attacks often occur on network security.

Syn Flooding attack is an attack on the server by sending a data packet to the server, after the server responds to the data packet and gives an answer signal to the client via the sender's IP address, it turns out the answer received by the server to reply to the client's reply is sent using a fake IP address, and this happens continuously, which causes the server to send an answer to the packet with a fake IP. This attack does not escape the reconnaissance server that will be used as an object. Technically Syn Flood occurs when there is a computer device connected to the server so that it is called TCP to the server, then the client sends it to the server then there is an exchange of information that usually happens like this. The client requests a connection to the server by sending the Syn (Synchronize) code to the server, the server recognizes the acknowledges of this request by sending the Syn code to Ack, then returning the request to the client. The client sends back the received response and produces a connection between the client and the server.

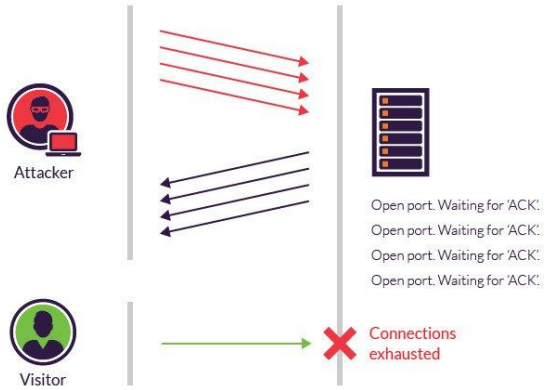

Figure 2. Illustration of a Denial of Service attack using the SYN Flooding Attack method.

Syn Flooding usually attacks by sending a Syn packet to ports in the "Listening" condition or registered on the target port to be attacked. So that it will make a shipment by modifying the authenticity of the source until the target will respond to the Syn-Ack packet to the address registered to the Syn packet, then wait for the Acknowledgment data package. So that the attacked ports will hold until the buffer he receives holds the final expiration time, if the port accepts it will respond by sending the Syn / Ack package by the package sent then it will be processed by the operating system.

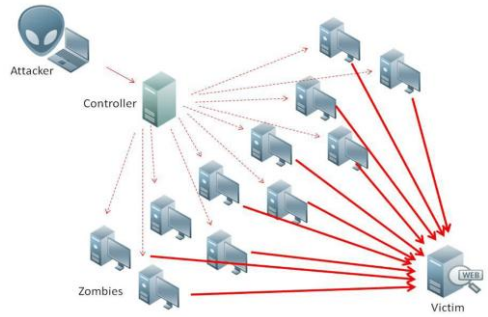

Figure 3. DDoS Attack attack efforts.

3. METHODOLOGY

3.1 Types of Research

This research is an experimental method that is carried out using Snort Applications which are added rules to explain the types of attacks that enter the network and detected by snort rules. Simulated using the Low Orbit Ion Cannon application which is used to attack the server with SYNflood, by determining the port to be attacked and the IP address that will be attacked so that it can make the system down. This research is a type of quantitative research, because research is carried out in the form of analysis, and verification of methods and the results obtained from testing emphasize the use of numbers and graphs. This research is an application that analyzes the rules of Snort and the synflood system [12].

3.2 Data Collection Methods

Data collection in this study was carried out for analytical material which would be an important part of the Snort rule analysis process. To have valid data, you must update Snort rules by using updates from www.snort.org

4. MODEL

4.1 Snort

Snort is a database that is used to create rules derived from other types of suspicious attacks [4]. Snort distribution applies a set of rules that can include known attacks such as buffer overflow or exploit distribution. If there are shortcomings or weaknesses in Snort, they will be added to the specified rules. In designing a network detection monitoring system that uses Snort. Snort distribution uses a set of rules that can close attacks. The Snort distribution is obtained from the extraction from the header attack and then the Snort distribution by collecting a rule that can close attacks or spread exploits. Snort rules will add to Snort if suspicious things are found. The detection is collected in the rules and stored in the database specified by Snort.

Snort is a suspicious activity detection system in a network system, which can analyze network traffic in real-time and provide alerts. Snort uses a system of rules that is relatively easier to learn when detecting and logging various attacks on computer networks. Snort can detect intrusions and can log packages against attacks. Snort is an open-source application based on the GNU (General Public License) so that it can be used and developed unpaid to get the source code. Snort is software that functions as a firewall and still runs command-line based.

Snort generally functions as follows:

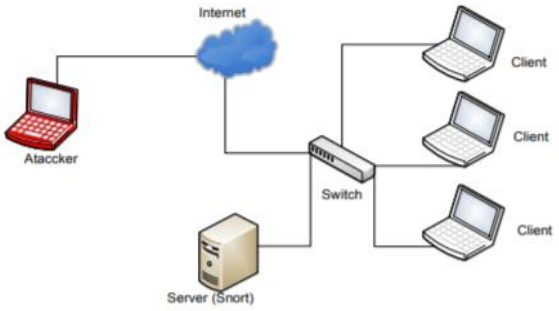

Figure 4. Network scheme for Snort detectors.

4.2 Snort Rules

Snort Rules are rules that contain databases with attack patterns, Snort rules database must be updated frequently so that if there are new types of attack patterns can be found by Snort rules. In this discussion of Snort rules to prevent Syn Flood attacks, writing rules must be written in one line and Snort rules are divided into two: header rules and option rules containing actions, protocols, sources, and destination IP addresses, netmask, source, and destination ports. The regulatory option consists of a warning message and various information that forms the basis of the package, containing alert messages and various information where the package should be placed [17].

Rules Snort is a rule in the form of files whose names are adapted to the category of attacks, for example, viruses. Rules are files containing snort rules that detect packages containing viruses. The Snort rule functions as a standard rule as a detector of system data vulnerabilities analyzed.

All existing rules follow the rule format:

action protocol address port direction address port (rules option)

Examples of rules

alert tcp any any -> 10.20.100.253 23|(msg:" Ada Telnet!';sid:1000001;)
Rules Header | *Rules Options*

Explanation of the snort Rules is divided into two parts, namely the rule header and rule options. The header rule contains details of the actions that occur if there is an attack signature that matches the rule while the rule option is detailed about the value of the data package. This will generate an alert if there is a TCP activity that enters IP 10.20.100.253 with port number 23. Rule Header is a rule that must exist in the snort rule format, while the option rule is an additional rule that functions as an attack rule identity.

An action is an action for an action that takes place in the form of "alert." The choice of action is "log" or "alert". Action "alert" will write all alert data to the same notification or "alert" file. While the log will store traffic for each problematic IP address in a folder for further analysis. In the development of snort mode, the alert action will be replaced with an action drop to filter incoming traffic.

The protocol section can be filled in "TCP", "UDP", or "ICMP". All packages (Any) that go to telnet with a port (port 23) are not allowed. Port can use range and operator "!" To machine 10.20.100.253 Address can be CIDR notation and

additional readable words, such as "There is Telnet!", And sid - rule ID starts from 1000000 as snort id number.

The rules for making rules are used to prevent SYNflood attacks so that they can be applied to snort rules.

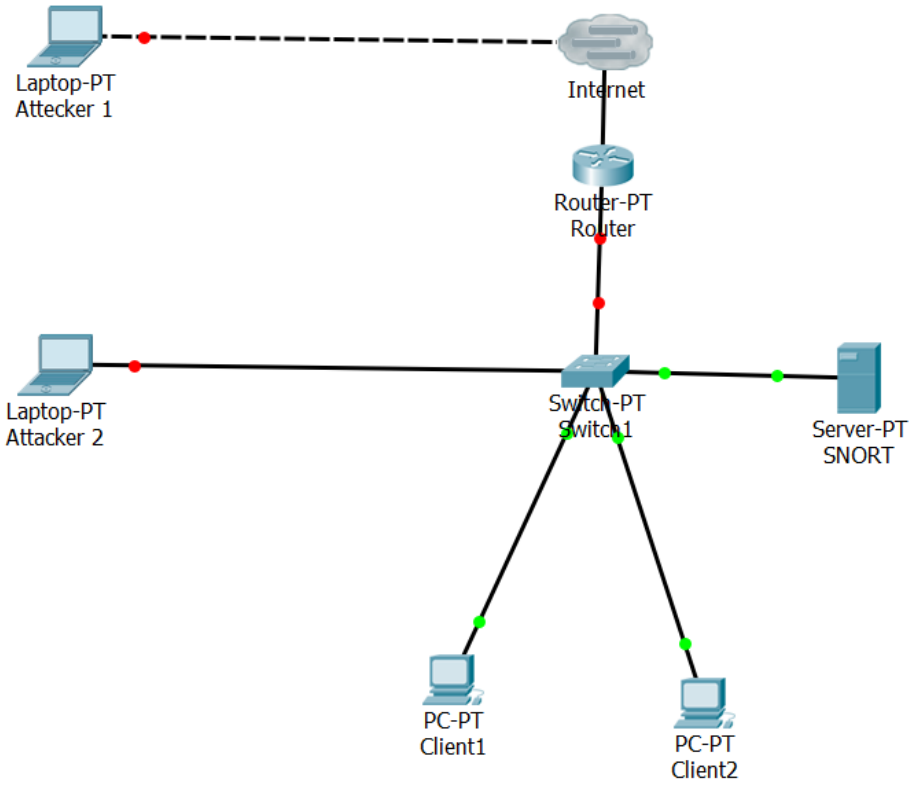

Figure 5. Network Topology

5. RESULTS AND DISCUSSION

Discussion Snort types of attacks often occur as part of a system intrusion that can be in the form of scanning Nmap ports, LOIC, SQL Injection, and database access [18]. Some technical aspects must be elaborated such as: How the system handles a variety of effective security that is faced with a high rate of growth in the volume of data, and how aspects of information security will be addressed [19]. Rule Snort is a Snort rule that contains databases with attack patterns.

Figure 6. Low Orbit Ion Cannon

Low Orbit Ion Cannon is an application that is used to attack a server with SYNflood by selecting the port that will be attacked and which IP Address will be attacked so that it can turn off the system. After the Snort rule is generated, to get results, that the resulting rules can prevent SYNflood attacks, the testing process is carried out. The client acting as the attacker sends an SYN packet to the port that is open on the target server. After testing the SYNflood attack, network traffic that was successfully blocked by the existing system can be known. Of course, the system cannot be penetrated and can be prevented by the rules that have been produced [20].

Test without Snort rules and use rules:

In this test shows the difference between active Snort without rule and active Snort with a rule when synflood attacks.

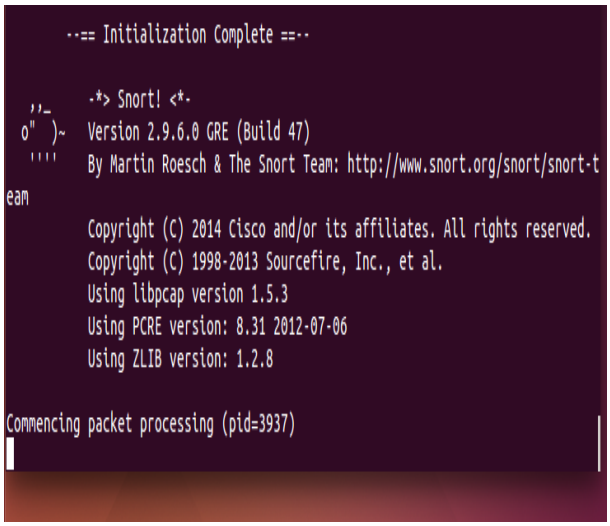

Figure 7. Testing without Snort rules

It can be seen in the picture there is no visible Alert or detection and prevention when sending packets with TCP SYN flood LOIC.

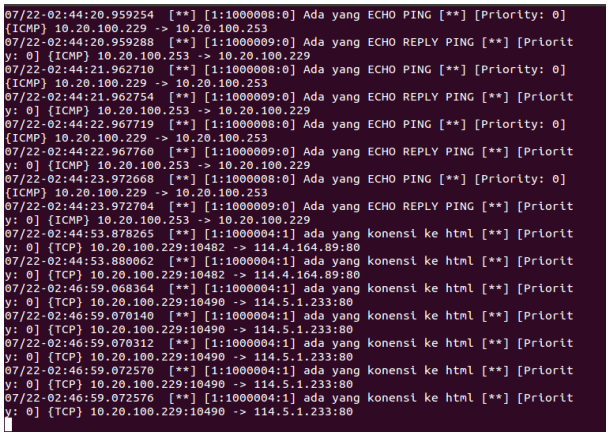

Figure 8. Testing using Snort rules

Alert or detection and prevention are performed successfully when the Attacker attacks the TCP SYN flood, which indicates that the snort rule has successfully detected and prevented the Drop on the console.

Traffic Measurement
DOS attacks can consume memory and bandwidth resources, so you need to analyze Network Traffic to find out how much bandwidth is spent by DOS when testing for 1 minute, testing is done by attackers 1 and attackers 2 compared to active Snort rules and Snort rules are not active.

Usable Value Resources Bandwidth table

Attack1 Per minute	Snort is not active	Snort Active	Attack1 Per minute	Snort is not active	Snort Active
1	13,45	10,32	1	19,13	17,01
2	14,55	11,31	2	13,32	11,34
3	18,85	12,52	3	11,76	10,01
4	18,95	12,47	4	15,65	13,89
5	13,35	12,78	5	18,04	14,75
6	15,67	11,07	6	20,11	16,34
7	18,32	13,82	7	17,84	14,53
8	12,75	10,31	8	14,64	12,34
9	14,54	11,02	9	14,34	12,36
10	13,65	10,43	10	14,32	12,31
Average	15,41	11,61	Rata-rata	15,92	13,49

According to the results of the following table, the average value of the bandwidth used in the event of a Snort rules attack is not active and Snort rules are active.

The scenario of an attack on snort rules 1 inactive Attacker generated an average value of used bandwidth of 15.41 Mbps and when the active snort rules average value of bandwidth used is 11.61 Mbps.

In the scenario of Attacker 2 attack, when the snort rules are inactive, the average value of used bandwidth is 15.92 Mbps when the active snort rules are used for an average bandwidth value of 13.49 Mbps.

The presentation of attacker test 1 saves 25%, while attacker testing to 2 saves 15% Slightly different the amount of bandwidth used during testing with snort inactive and snort active within 10 minutes counted 1 minute on the network.

6. CONCLUSION AND FUTURE WORK

Based on the test results, Snort rules to prevent Synflood attacks on network security were successfully tested and resulted in Snort rule conditions that were better able to handle SYNflood attacks. It was concluded that the creation of the Snort rule developed had functioned well in preventing SYNflood attacks. Rules that have been created with the Snort rule system can recognize and resist TCP SYN flood attacks. So that it makes it easier for Administrators to get data in addition to increasing network security from Sys flooding attacks and makes it easier for administrators to make policies improve based on logs in Snort.

In the future, what needs to be developed is Snort Rule which is formulated integrated with the application so that it makes it easier or easier to display the log database because, in this discussion it has not been made, it is necessary to develop applications that are integrated with the Snort rule database.

7. REFERENCES

[1] T. Ubale and A. K. Jain, "SRL : An TCP SYNFLOOD DDoS Mitigation Approach in Software-Defined Networks," 2018 Second Int. Conf. Electron. Commun. Aerosp. Technol., no. Iceca, pp. 956–962, 2018.

[2] N. Jongsawat and J. Decharoenchitpong, "Creating behavior-based rules for snort based on Bayesian network learning algorithms," Proc. 2015 Int. Conf. Sci. Technol. TICST 2015, pp. 267–270, 2015.

[3] R. T. Gaddam and M. Nandhini, "An analysis of various snort based techniques to detect and prevent intrusions in networks," Proc. Int. Conf. Inven. Commun. Comput. Technol. ICICCT 2017, no. Icicct, pp. 10–15, 2017.

[4] C. Turner and A. Joseph, "A Statistical and Cluster Analysis Exploratory Study of Snort Rules," Procedia Comput. Sci., vol. 114, pp. 106–115, 2017.

[5] S. Hameed and H. A. Khan, "Leveraging SDN for collaborative DDoS mitigation," 2017 Int. Conf. Networked Syst. NetSys 2017, no. March, 2017.

[6] A. Motivaciones, "Security for WI SP through Mikrotik equ uipment krotik)," 2015 Chil. Conf. Electr. Electron. Eng. Inf. Commun. Technol., pp. 229–233, 2015.

[7] R. F. Olanrewaju, B. U. Islam Khan, A. R. Najeeb, K. N. A. Ku Zahir, and S. Hussain, "Snort-based Smart and Swift Intrusion Detection System," Indian J. Sci. Technol., vol. 11, no. 4, pp. 1–9, 2018.

[8] D. Ibdah, M. Kanani, N. Lachtar, N. Allan, and B. Al-Duwairi, "On the security of SDN-enabled smartgrid systems," 2017 Int. Conf. Electr. Comput. Technol. Appl. ICECTA 2017, vol. 2018-Janua, pp. 1–5, 2018.

[9] P. Singh, S. Behal, and K. Kumar, "Performance enhancement of a Malware Detection System using score based prioritization of snort rules," Proc. 2015 Int. Conf. Green Comput. Internet Things, ICGCIoT 2015, pp. 1150–1155, 2016.

[10] C. Networks, S. K. Patel, and A. Sonker, "Internet Protocol Identification Number based Ideal Stealth Port Scan Detection using Snort," 2016.

[11] Y. Yin, Y. Wang, and N. Takahashi, "Set-based calculation of topological relations between snort rules," Proc. - 2014 2nd Int. Symp. Comput. Networking, CANDAR 2014, pp. 617–619, 2014.

[12] M. Sadikin, R. Yusuf, and A. R. Dwiyanto, "Load Balancing Clustering on Moodle LMS to Overcome Performance Issue of e-Learning System," TELKOMNIKA (Telecommunication Comput. Electron. Control., vol. 17, no. 1, pp. 281–289, 2019.

[13] F. I. Shiri, B. Shanmugam, and N. B. Idris, "A parallel technique for improving the performance of signature-based network intrusion detection system," A Parallel Tech. Improv. Perform. Signature-Based Netw. Intrusion Detect. Syst., pp. 692–696, 2011.

[14] I. Nurhaida and Ngadiyono, "Quality of Service for Traffic Monitoring System based on Static Routing using EoIP Tunnel over IPSec," no. 1, pp. 91–99, 2019.

[15] S. S. Bhunia and M. Gurusamy, "Dynamic Attack Detection and Mitigation in IoT using SDN," 2017.

[16] A. Masys, "Networks and network analysis for defence and security," pp. 1479–1480, 2014.

[17] N. Khamphakdee, N. Benjamas, and S. Saiyod, "Improving intrusion detection system based on Snort rules for network probe attack detection," 2014 2nd Int. Conf. Inf. Commun. Technol. ICoICT 2014, pp. 69–74, 2014.

[18] N. Suteva, A. Mileva, and M. Loleski, "Computer Forensic Analisys of Some Web Attacks," pp. 42–47, 2014.

[19] A. Garg, "Performance Analysis of Snort-based Intrusion Detection System," pp. 0–4, 2016.

[20] I. Coonjah and P. C. Catherine, "Performance Evaluation and Analysis of Layer 3 Tunneling between OpenSSH and OpenVPN in a Wide Area Network Environment," pp. 1–4, 2015.