JURNAL INFOTEL
Informatics - Telecommunication - Electronics
Website: http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN: 2085-3688; e-ISSN: 2460-0997

Implementation of intrusion prevention system (IPS) to analyze triad CIA on network security attacks

Amilia Anggraeni[1,*], Jafaruddin Gusti Amri Ginting[2], Syariful Ikhwan[3]
[1]Institut Teknologi Telkom Purwokerto
[1]Jl. D.I. Panjaitan, No. 128, Purwokerto 53147, Indonesia
[*]Corresponding email: jafaruddin@ittelkom-pwt.ac.id

Received 27 July 2022, Revised 19 September 2022, Accepted 23 September 2022

Abstract — Information security serves to avoid damage or loss caused by attack activities during the communication process. Information security aspects are Confidentiality, Integrity, and Availability (CIA). A web server is a service provider that is used to receive HTTP requests that work on port 80 which is more vulnerable to attack threats. Threats of attacks that can occur on the web server are port scanning, Brute Force and DDoS. Intrusion prevention system is a solution that can maintain network security from various attacks. Intrusion prevention system acts as a protector on the network by detecting and preventing suspicious traffic on a network. In this study, the intrusion prevention system uses the snort and IP-Tables tools as well as the signature based detection method. The test is carried out using two scenarios, namely before IPS is activated and after IPS is activated. The results of the study are that the three attacks tested have different characteristics of cause and effect. Port scanning and Brute Force attacks can be prevented by IPS because the characteristics of both attacks are easily recognized by the rules in the snort database. In DDoS attacks, snort only speeds up the attack time to be accessible again because the characteristics of slow HTTP attacks are sending incomplete packets in large numbers gradually and maintaining connection session times so that it is difficult for Snort to recognize. In a DDoS attack with an action rule drop, the web server can be accessed again at 160 seconds while the reject rule action can be accessed again at 145 seconds where the normal attack time can be accessed again at 165 seconds. In CPU usage, IPS can reduce usage by 9.2 % on DDoS attacks.

Keywords – CIA Triad, intrusion prevention system, snort

I. Introduction

The advancing times have led to an increase in internet use and utilization across all industries and activities. Based on data from the news website surat.com, there were 202.6 million internet users in Indonesia as of January 2021 [1]. The information security system serves to prevent damage or loss of data caused by attack activities during the communication process. Various threats of attacks on the security of telecommunications networks greatly affect the aspects of confidentiality, integrity, and availability of service providers. Based on data from the National Cyber Security Operations Center (POKSN) and the National Cyber and Crypto Agency (BSSN), there were 88,414,296 cases of cyber attacks in Indonesia, from January 1, 2020, to April 12, 2020 [2].

As a service provider and user interface used to receive HTTP requests, the web server is the most vulnerable to attack threats. Several attack threats that can occur on the web server are port scanning, Brute Force and Distributed Denial of Service (DDoS). One solution that can be done by network administrators to minimize the possibility of attacks that occur is by implementing an Intrusion Prevention System (IPS). IPS itself is software for monitoring, detecting and preventing intrusion of suspicious activity on network traffic and is a combination of the Intrusion Detection System (IDS) and firewall function. IDS is a method that can be used to identify, provide reports on network. Activities and firewall are a technique used to protect computer network security by filtering incoming and outgoing data packets on the network [3], [4].

The network security attacker will study, analyse, modify, and even steal data on the target system where changes or system damage can hinder the course of user activities. Hackers typically target servers with the intention of uncovering security weaknesses, obstructing service providers from providing their services, seeking profit, or simply showcasing their expertise [5]. Aspects of Confidentiality, Integrity, and Availability (CIA) are common problems [6]. Various kinds of attacks in this study such as DDoS and Denial of Service (DoS) were carried out with the aim of exhausting the connectivity and processing of the target server resources which could allow obstacles to access services by legitimate users where the type of slow HTTP is a new class of DoS attack that exploits vulnerabilities in the application layer [7]. Brute force, where the attacks use the matchmaking list method or in other words match words to find the password that is being searched for [8]. Port scanning is used to find active or open ports on a computer network and so on [9].

In 2020, Alamsyah, *et al.* [10] do a study ”Analisa Keamanan Jaringan Menggunakan Network Intrusion Detection and Prevention System” conducted research by building an IPS system that uses Suricata and IP-Tables as a firewall which works with signature-based mode and with NIDPS mode. In its implementation, the authors identified that the research did not build a network using NIDPS but with HIDPS, because the protected side is a web server where, when a server or host is attacked by attackers on the Suricata side, it cannot detect and stop the attack. In 2018, Putra, *et al.* [11] doing study ”Implementasi dan Analisis Keamanan Jaringan Virtual HIPS Snort Pada Layanan Web Server Dengan Penyerangan DoS dan DDoS” found that HIPS can recognize and withstand attacks both TCP and Syn Flood. The HIPS presentation can block attacks on 1 attacker, which is 97.98% and 97.8% on 4 attackers. Test results on bandwidth found no significant difference between the attacks of 1 attacker with 4 attackers. The data transfer rate used in this study is about 20 Mbps.

In 2019, Suwanto, *et al.* [12] conducted a study ”Implementation of Intrusion Prevention System (IPS) Using Snort and IPTables on Website-Based Local Network Monitoring”. In this study, it can detect and prevent attacks with a success percentage of 90% on ping of death attacks and 85% on port scanning attacks. However, because the prevention system used is IP blocking on IP-Tables and rules that are still very common, it allows legal users to be detected as attackers. To overcome the weaknesses that exist in research [12], in this study, an option rule that is more specific and closer to the characteristics of the attack is used so that attack prevention can be carried out appropriately so that legal users are no longer detected as attacks. IPS, which the server can use to detect and block an external action that is considered suspicious by a network [13]. The tools used are Snort and IP-Tables which are implemented by the NIPS method, where IPS not only protects one host but protects all hosts on the network [12], [14]. In addition to maintaining security, this study analyzes the characteristics of each attack that occurred based on the CIA Triad aspect. In this study, it was carried out at the application layer where this layer will provide an interface to the application that we are using to exchange information so that it is the most vulnerable part to be attacked [15].

The tools used for the attack include NMAP which is used for the application layer to provide an interface to the application that we are using to exchange information, Hydra is used to break passwords and usernames according to the wordlist that has been created, and Slowhttptest is used to exploit existing vulnerabilities in the application layer [14], [17], [18]. In this study, the device used to analyze packets is Wireshark [19].

Based on this background, the purpose of this research is to implement IPS as a security server. Then, this research will analyze the characteristics of each attack that tried to enter the server based on the CIA Triad.

II. Research Method

In this section, we discuss the flowchart, network topology, IPS server configuration, tools of attacker configuration, and system test.

A. Research Flowchart

This stage is the flow of the scheme during the research include of the literature study, setting up the device until the result analysis of testing and measurement. The flowchart of this research consist of several steps as shown in Fig. 1.

B. Network Topology

The topology used in this study consists of a server in which there is an IPS server and a web server, two switches, an attacker PC, and a client PC. In this study there are two experimental scenarios. Fig. 2 is the first scenario, namely the network topology before configuring the IPS rule and Fig. 3 is the second scenario, namely the network topology after configuring the IPS rule.

C. IPS Server Configuration

At this stage, the author configures Snort which acts as an IPS. The configuration stages are divided into several parts, including Snort.lua configuration, rule file configuration, IP-Tables configuration and Snort activation.

In the snort.lua configuration file there are several commands that must be adjusted, including the declaration of HOME_NET and EXTERNAL _NET, as well as the declaration of the directory that will be used to store logs. Configuring inline.lua is done to specify the method used as the inline mode. In this study, the inline method used is nfq with queue = 0.

In this study, three rules are used to be used in different attacks and then use the ’reject’ action where

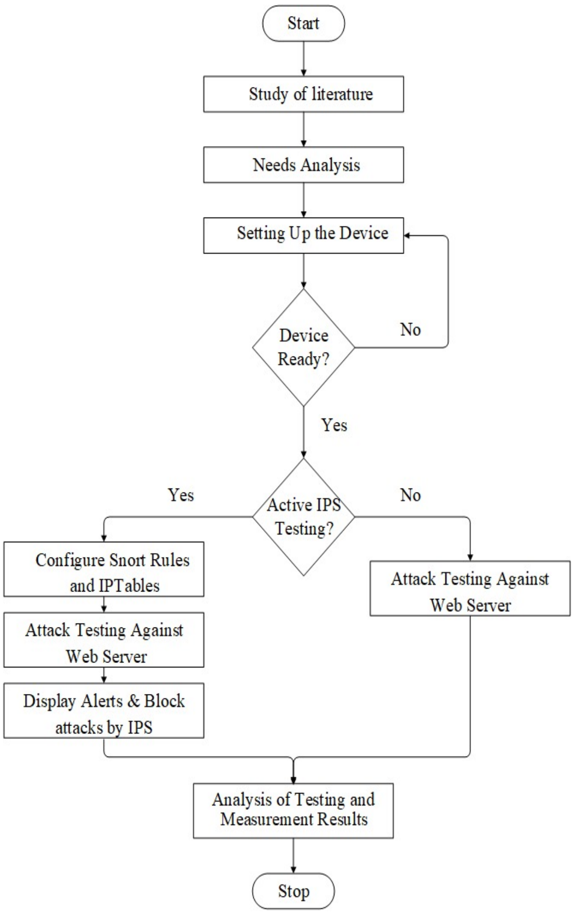

Start
Study of literature
Needs Analysis
Setting Up the Device
Device Ready?
No
Yes
Active IPS Testing?
Yes
No
Configure Snort Rules and IPTables
Attack Testing Against Web Server
Display Alerts & Block attacks by IPS
Attack Testing Against Web Server
Analysis of Testing and Measurement Results
Stop

Fig. 1. Research flowchart.

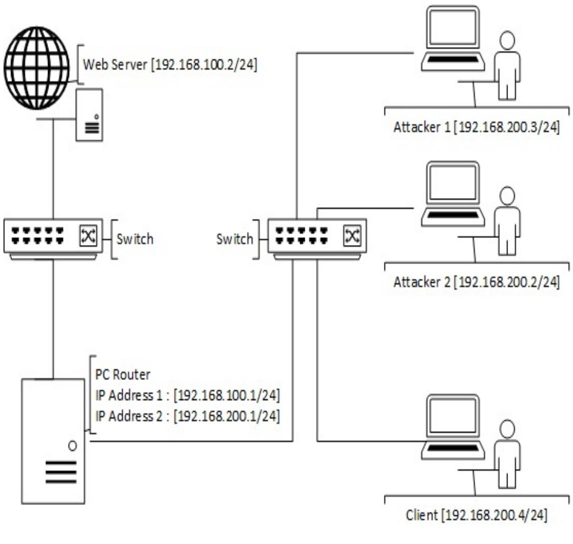

Web Server [192.168.100.2/24]
Switch
Switch
Attacker 1 [192.168.200.3/24]
Attacker 2 [192.168.200.2/24]
PC Router
IP Address 1 : [192.168.100.1/24]
IP Address 2 : [192.168.200.1/24]
Client [192.168.200.4/24]

Fig. 2. Topology before IPS rule configuration.

each packet will be dropped and recorded in the log then will send a TCP reset [20]. The rules used for Port Scanning attacks can be seen in Fig. 4, the rules used for brute force attacks can be seen in Fig. 5 and the rules used for DDoS attacks are shown in Fig. 6.

This study uses three rules and configuration IPTables to be used in different attacks and then uses the 'reject' action where each package will be dropped and logged then sends a TCP reset. The rules used for Port Scanning attacks can be seen in Fig. 4, the rules used for brute force attacks can be seen in Fig. 5, the rules used for DDoS attacks are shown in Fig. 6 and the configuration used for IP-Tables is shown in Fig. 7.

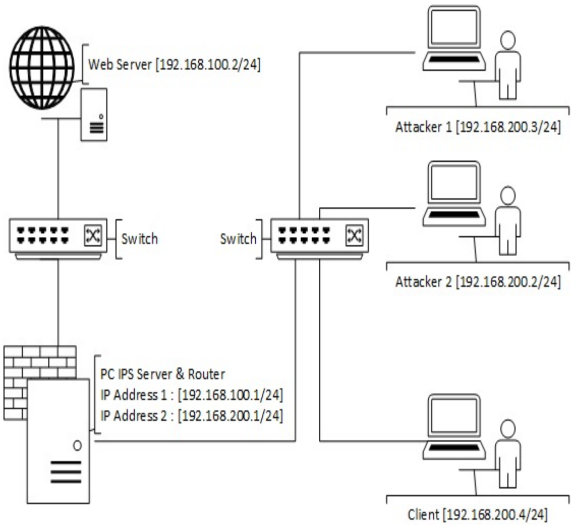

Web Server [192.168.100.2/24]
Attacker 1 [192.168.200.3/24]
Switch
Switch
Attacker 2 [192.168.200.2/24]
PC IPS Server & Router
IP Address 1 : [192.168.100.1/24]
IP Address 2 : [192.168.200.1/24]
Client [192.168.200.4/24]

Fig. 3. Topology after IPS rule configuration.

```
reject tcp any any -> any any (msg: "TCP Port
Scanning!!!"; sid : 10000003; rev : 1; flags
: S; flow : stateless; classtype : attempted-
recon;)
```

Fig. 4. Rule Snort attack port scanning.

```
reject tcp any any -> any any (msg: "Serangan
Brute Force ke Web Server"; content : "Login
Gagal" ; sid : 1000005 ; rev : 1 ; classtype
: attempted-admin ; detection_filter : track
by_src,  count  5,  seconds  10;  flow  :
stateless;)
```

Fig. 5. Rule Snort attack brute force.

```
drop tcp any any -> any 80 (msg: "DDoS HTTP
Flooding!!!";    flow:to_server,established;
flags: PA; detection_filter: track by_dst,
count 200, seconds 30; classtype:attempted-
dos; sid:1000008; rev:1;)
reject tcp any any -> any 80 (msg: "DDoS HTTP
Flooding!!!";    flow:to_server,established;
flags: PA; detection_filter: track by_dst,
count 200, seconds 30; classtype:attempted-
dos; sid:1000008; rev:1;)
```

Fig. 6. Rule Snort attack DDoS.

```
IPTABLES -I FORWARD -j NFQUEUE
IPTABLES -I INPUT -j NFQUEUE
IPTABLES -I OUTPUT -j NFQUEUE
```

Fig. 7. Configuration IP-Tables.

D. Tools of Attacker Configuration

This stage is carried out when preparing the device to carry out an attack. The author configures the tools that will be used to support testing attacks on the server. The tools used include Wireshark, NMAP, slowhttptest and Hydra.

In the configuration of tools for Port Scanning attacks, there are several tools that need to be prepared. The tools used are Wireshark and NMAP. Wireshark tools are used as an initial step carried out by attackers to simply see the target IP address. This attack is used to find out the port gap of the target host with the NMAP -sS 192.168.100.2 command. The type of Port Scanning used in this study is SYN Scan, where the attacker will send a SYN packet, then if the port on the target is open, then the target will send a SYN-ACK packet reply, then the attacker will send an RST packet to close the network before the connection ends.

In the configuration of tools for brute force attacks, there are several things that need to be prepared. The tools used are Hydra and two supporting files in the form of a username list file and a password list file. In the configuration of tools for DDoS attacks, Slowhttptest aims to exploit HTTP GET or POST requests on the server so that it can result in the inability of the server to perform services. In this attack there are 5 times with the attack parameters in the Table 1.

Table 1. DDoS Attack Parameters

No	Parameeter	Value
1	Number of connections	1000
2	Action time	160s
3	Method	GET
4	Connections per seconds	200

E. System Test

At the system testing stage, the author performs tests that are used to analyze the results of security design on the system. System testing is carried out for each attack with different characteristics. The observations made have differences depending on the attack carried out. During the attack process, on the server side the IPS log file can be seen at alert_fast.txt which is located at /var/log/Snort/alert_fast.txt. Scenario testing is carried out to compare system performance against the two scenarios used and to know the characteristics of each attack.

III. Result and Discussion

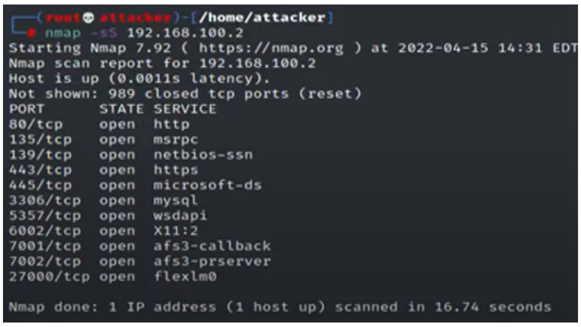
In this study, there were three attacks carried out by the attacker. Each attack has different characteristics such as how it works and the effect of the attack on the target. Based on the CIA Triad, information security is divided into three, namely Confidentiality, Integrity and Availability. The test results of this study of the characteristics of the attack can be seen in Table 2.

Table 2. Attacks Identifications with CIA Triads

Types of Attacks	**Aspects of CIA Triad**		
	Confidentiality	***Integrity***	***Availability***
Port Scanning	✓	✗	✗
Brute Force	✓	✓	✗
DDoS	✗	✗	✓

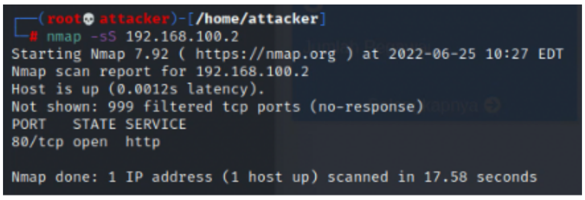
The port scanning attack is categorized as an attack on the privacy of the target party. This attack resulted in confidential information, namely the port status can be exposed or in other words attack the Confidentiality parameter of the web server. The second attack is brute force, where this attack is categorized as an attack on the Confidentiality and Integrity of the target, which in this study is the web server. Brute force in this study will match the possible words that have been made by the attacker to get a username and password. When the attacker has a username and password, the next step is that the attacker can log into the web server system which can result in the attacker having access rights to be able to change and even steal data. The last attack is DDoS where from the observations, it is an attack on the Availability aspect. Availability is an aspect where the server's ability to always provide services. During the DDoS attack, the target of the attack will experience paralysis so that when a legal user accesses the web server service, buffering occurs or takes time to be served.

Port Scanning attack testing in this study is based on two scenarios, namely before and after IPS is activated. Each scenario the author conducted the test five times and the author observed from the point of view of the attacker and the IPS server. The first scenario testing occurs when a Port Scanning attack is carried out by an attacker on a web server where the packages traffic through the Snort server has not been activated. The test results can be explained that the pattern of the five tests in the first scenario is the same, where the attacker can get information in the form of port status and services used from the web server while in the second scenario in five times of testing it can be concluded that when Snort IPS is activated, the attacker can no longer know the status of open ports and the services used. The attacker's appearance when getting information on the status of open ports and the services used on the web server can be seen in Fig. 8.



```
┌──(root💀attacker)-[/home/attacker]
└─# nmap -sS 192.168.100.2
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-15 14:31 EDT
Nmap scan report for 192.168.100.2
Host is up (0.0011s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
5357/tcp  open  wsdapi
6002/tcp  open  X11:2
7001/tcp  open  afs3-callback
7002/tcp  open  afs3-prserver
27000/tcp open  flexlm0

Nmap done: 1 IP address (1 host up) scanned in 16.74 seconds
```

Fig. 8. Port scanning test results before Snort activated.



```
┌──(root💀attacker)-[/home/attacker]
└─# nmap -sS 192.168.100.2
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 10:27 EDT
Nmap scan report for 192.168.100.2
Host is up (0.0012s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 17.58 seconds
```

Fig. 9. Port scanning test results after Snort activated.

The port scanning process is carried out for 16.74 seconds and displays information that there is 1 active host. In Fig. 9 the attacker performs a test for 17.21 seconds, which is carried out during scanning and then displays a notification in the form of ’Not shown: 1000 filtered TCP ports (no-response)’ which indicates that there is no response from the target to provide port information with the word otherwise the attacker does not have access to carry out attacks.

On the IPS server side, if it has not been activated, the administrator does not know if there is an attack on the attacker. From the point of view of the IPS server, when activated, the system will give an ’alert’ or warning that there is an attack on a user on the network. The display of alerts on the IPS server side during an attack can be seen in Fig. 10. In Fig. 10, you can see a message that matches the rule that has been created. The information message that appears is ”TCP Port Scanning!!!” with a reject action where the packages from the source will be rejected and the attacker will not get the desired information. IPS can monitor all traffic on the network so that in other words, all users on the network can be protected and the system can detect and block any discrepancies on the network based on the rule signature that has been created.

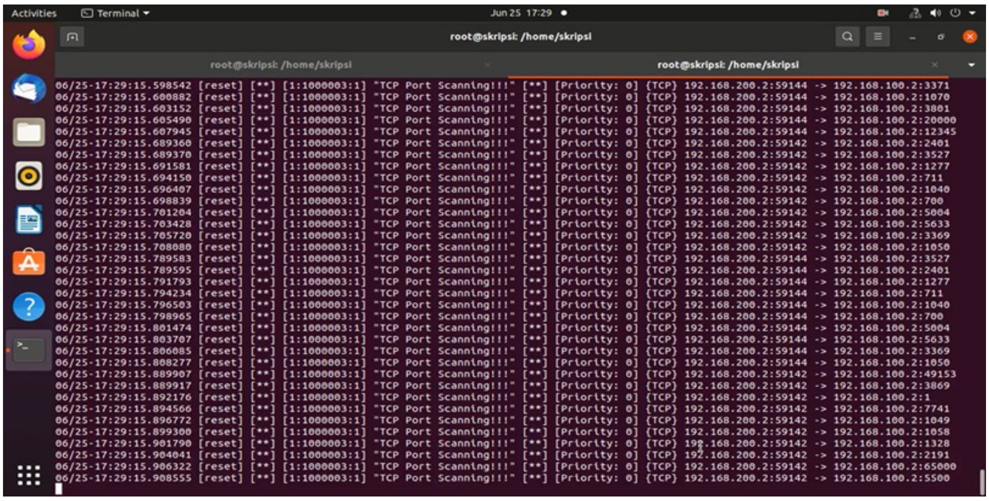
Brute force attack testing is based on two scenarios, namely before and after IPS is activated. Each scenario the author conducted the test five times and the author observed from the point of view of the attacker and the IPS server. The first scenario testing is when the attacker performs a brute force attack on a web server through an IPS server that has not been activated. The second scenario test is when the attacker performs a brute force attack on the web server through an IPS server that has been activated. The point of view that can be observed is the attacker’s point of view with the display parameters of the test results and from the IPS server’s point of view with the parameters of the attack alert results.

The test results can be concluded that the five tests in the first scenario have similarities, namely in the first scenario when IPS has not been activated the attacker can see or get information in the form of a username and password Login Form from the web server while the results of the test after IPS is activated, the attack is successfully blocked by Snort. Attackers cannot access any information and are immediately disconnected. Fig. 11 is the result of an attack on the attacker’s side where the results obtained are username and password to log into the web app which is the content of the web server. In the second scenario, when the attacker attacks the web server login form with activated IPS, the results will be nil. On the attacker side will display a response in the form of STATUS output, which indicates that the experiment is repeated continuously without any response until the attacker he stops it. The display of the results of the second brute force attack can be seen in Fig. 12.

Based on Fig. 10, the results of the test where the username obtained is ”admin” and the password obtained is ”1”. Attacker performs matching 900 times with a calculation time of 52 seconds, the time and number of attempts can change based on the number of words made by the attacker to be used as a possible match. When IPS has not been activated, if there is an attack on the web server side there is no protection so that the attack can easily get information in the form of a username and password from the web server Login Form. When the attacker can enter the web server system, the attacker can perform the next action, such as adding, deleting, changing and even stealing personal data on the web server. It can be seen in Fig. 13 when the data is still normal and there has been no change while in Fig. 14 it is the result of the attacker’s actions by adding and deleting employee data. Actions from attackers that can result in losses for web server administrators.

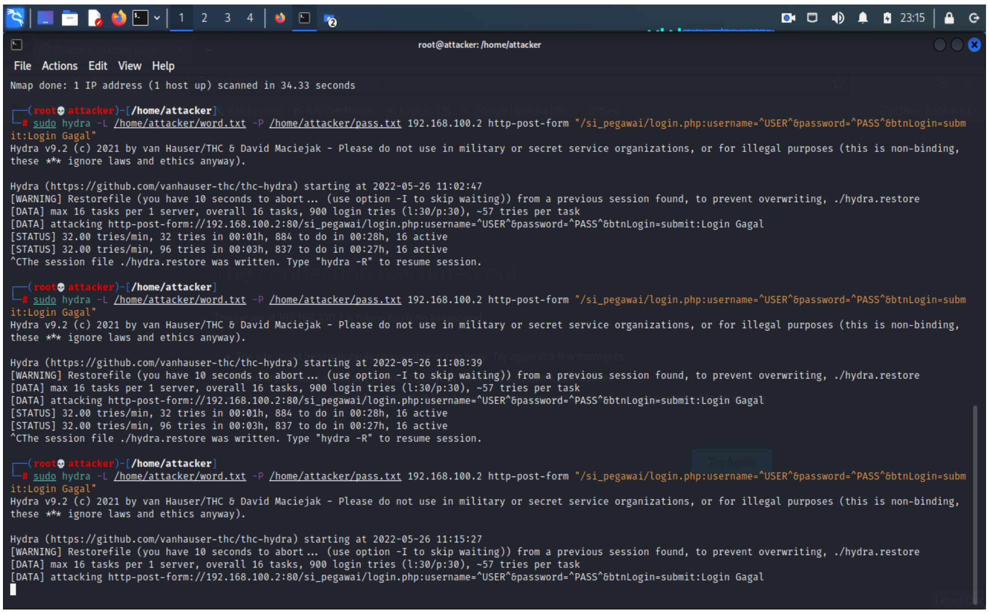
On the IPS server side shown in Fig. 15, Snort immediately gives a warning to the network administrator in the form of a notification in accordance with the message that has been set in the rule and drop the packages sent by the attacker and send a TCP RST packages or by disconnecting the attack packages so that the attacker does not get the information he is looking for. The warning will continue to appear in real time as long as the attack is still being carried out by the attacker. The attack alert log will also be stored in Alert_fast.txt but will also be deleted automatically if the log file storage is full and overwritten with other attack alerts.

When testing DDoS attacks, slow HTTP attacks using the GET method are used. In these attacks, the attacker will carry out the attack by gradually and continuously sending incomplete HTTP header packages to the web server while maintaining the connection session. This test is based on two scenarios, namely before and after IPS is activated. Each scenario the author conducted five tests and observed from the point of view of the attacker, IPS server, legal user, and web server. Testing the Slow HTTP attack in the first and second scenarios is when the attacker performs a Slow HTTP attack on a web server where traffic packages go through before and after the IPS server is activated. On the attacking side, the authors get a test result with the attack packages status parameters which are the first to fifth tests with each using an IPS scenario that has not been activated, IPS is activated with a drop rule on Snort and IPS is activated with a reject rule on Snort. Fig. 16, Fig. 17, and Fig 18 are one of the five results of testing the status of attack packages using two attackers where the most visible difference is in the closed line and service available line. In the scenario without IPS being activated, the closed line widens as there is no resistance from the web server, allowing the attacker to easily carry out his operations in accordance with the conditions where Slow HTTP attacks are characterized by carrying


Activities
Terminal
Jun 25 17:29
root@skripsi: /home/skripsi
root@skripsi: /home/skripsi
root@skripsi: /home/skripsi

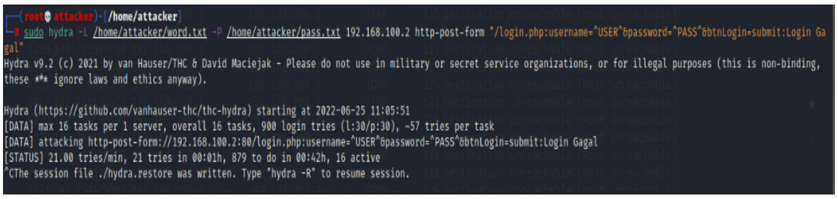
```
06/25-17:29:15.598542 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:3371
06/25-17:29:15.600882 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:1070
06/25-17:29:15.603152 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:3801
06/25-17:29:15.605490 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:20000
06/25-17:29:15.607945 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:12345
06/25-17:29:15.689360 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:2401
06/25-17:29:15.689370 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:3527
06/25-17:29:15.691581 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:1277
06/25-17:29:15.694150 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:711
06/25-17:29:15.696407 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:1040
06/25-17:29:15.698839 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:700
06/25-17:29:15.701204 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:5004
06/25-17:29:15.703428 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:5633
06/25-17:29:15.705720 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:3369
06/25-17:29:15.708080 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:1050
06/25-17:29:15.789583 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:3527
06/25-17:29:15.789595 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:2401
06/25-17:29:15.791793 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:1277
06/25-17:29:15.794234 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:711
06/25-17:29:15.796503 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:1040
06/25-17:29:15.798965 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:700
06/25-17:29:15.801474 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:5004
06/25-17:29:15.803707 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:5633
06/25-17:29:15.806085 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:3369
06/25-17:29:15.808277 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59144 -> 192.168.100.2:1050
06/25-17:29:15.889907 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:49153
06/25-17:29:15.889917 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:3869
06/25-17:29:15.892176 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:1
06/25-17:29:15.894566 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:7741
06/25-17:29:15.896772 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:1049
06/25-17:29:15.899300 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:1058
06/25-17:29:15.901790 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:1328
06/25-17:29:15.904041 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:2191
06/25-17:29:15.906322 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:65000
06/25-17:29:15.908555 [reset] [**] [1:1000003:1] "TCP Port Scanning!!!" [**] [Priority: 0] {TCP} 192.168.200.2:59142 -> 192.168.100.2:5500
```

Fig. 10. Alert display on Snort to port scanning.



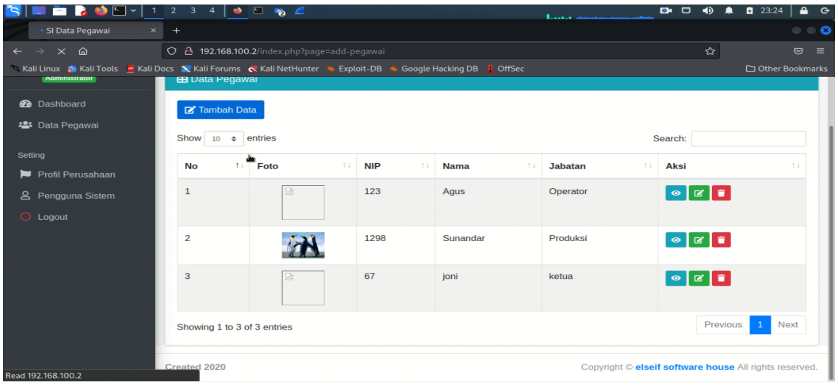
```
Nmap done: 1 IP address (1 host up) scanned in 34.33 seconds

┌──(root💀attacker)-[/home/attacker]
└─# sudo hydra -L /home/attacker/word.txt -P /home/attacker/pass.txt 192.168.100.2 http-post-form "/si_pegawai/login.php:username=^USER^&password=^PASS^&btnLogin=subm
it:Login Gagal"
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding,
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-26 11:02:47
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 900 login tries (l:30/p:30), ~57 tries per task
[DATA] attacking http-post-form://192.168.100.2:80/si_pegawai/login.php:username=^USER^&password=^PASS^&btnLogin=submit:Login Gagal
[STATUS] 32.00 tries/min, 32 tries in 00:01h, 884 to do in 00:28h, 16 active
[STATUS] 32.00 tries/min, 96 tries in 00:03h, 837 to do in 00:27h, 16 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.

┌──(root💀attacker)-[/home/attacker]
└─# sudo hydra -L /home/attacker/word.txt -P /home/attacker/pass.txt 192.168.100.2 http-post-form "/si_pegawai/login.php:username=^USER^&password=^PASS^&btnLogin=subm
it:Login Gagal"
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding,
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-26 11:08:39
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 900 login tries (l:30/p:30), ~57 tries per task
[DATA] attacking http-post-form://192.168.100.2:80/si_pegawai/login.php:username=^USER^&password=^PASS^&btnLogin=submit:Login Gagal
[STATUS] 32.00 tries/min, 32 tries in 00:01h, 884 to do in 00:28h, 16 active
[STATUS] 32.00 tries/min, 96 tries in 00:03h, 837 to do in 00:27h, 16 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.

┌──(root💀attacker)-[/home/attacker]
└─# sudo hydra -L /home/attacker/word.txt -P /home/attacker/pass.txt 192.168.100.2 http-post-form "/si_pegawai/login.php:username=^USER^&password=^PASS^&btnLogin=subm
it:Login Gagal"
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding,
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-26 11:15:27
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 900 login tries (l:30/p:30), ~57 tries per task
[DATA] attacking http-post-form://192.168.100.2:80/si_pegawai/login.php:username=^USER^&password=^PASS^&btnLogin=submit:Login Gagal
```

Fig. 11. Brute force test results before IPS activated.



```
┌──(root💀attacker)-[/home/attacker]
└─# sudo hydra -L /home/attacker/word.txt -P /home/attacker/pass.txt 192.168.100.2 http-post-form "/login.php:username=^USER^&password=^PASS^&btnLogin=submit:Login Ga
gal"
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding,
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-06-25 11:05:51
[DATA] max 16 tasks per 1 server, overall 16 tasks, 900 login tries (l:30/p:30), ~57 tries per task
[DATA] attacking http-post-form://192.168.100.2:80/login.php:username=^USER^&password=^PASS^&btnLogin=submit:Login Gagal
[STATUS] 21.00 tries/min, 21 tries in 00:01h, 879 to do in 00:42h, 16 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.
```

Fig. 12. Brute force test results after IPS activated.



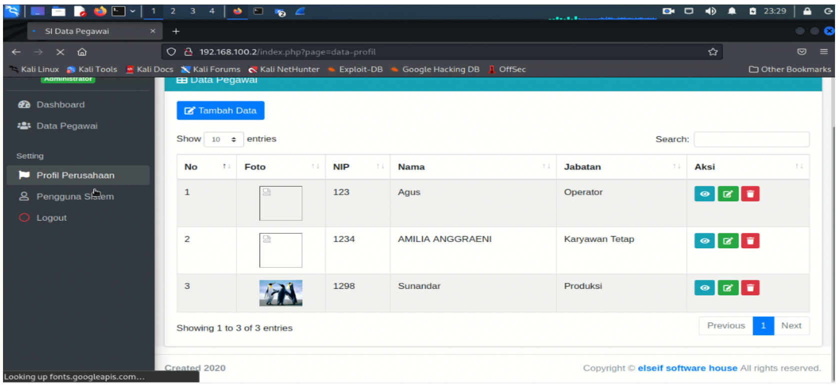
No	Foto	NIP	Nama	Jabatan	Aksi
1		123	Agus	Operator	
2		1298	Sunandar	Produksi	
3		67	joni	ketua	

Showing 1 to 3 of 3 entries

Created 2020

Copyright © elseif software house All rights reserved.

Fig. 13. Web server display under normal condition.



Show 10 entries Search:

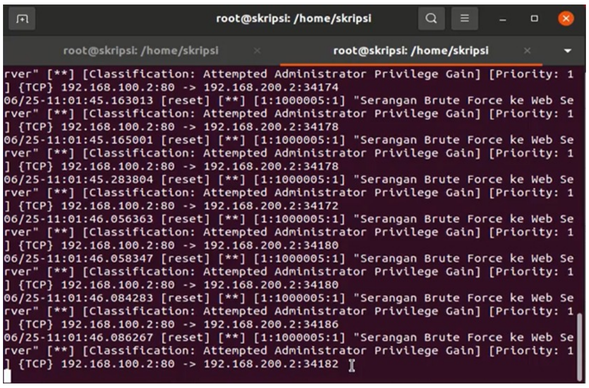
No	Foto	NIP	Nama	Jabatan	Aksi
1		123	Agus	Operator	
2		1234	AMILIA ANGGRAENI	Karyawan Tetap	
3		1298	Sunandar	Produksi	

Showing 1 to 3 of 3 entries

Created 2020 Copyright © elseif software house All rights reserved.

Looking up fonts.googleapis.com...

Fig. 14. Attacker action display after performing attack.


root@skripsi: /home/skripsi
root@skripsi: /home/skripsi
root@skripsi: /home/skripsi

```
rver" [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1
] {TCP} 192.168.100.2:80 -> 192.168.200.2:34174
06/25-11:01:45.163013 [reset] [**] [1:1000005:1] "Serangan Brute Force ke Web Se
rver" [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1
] {TCP} 192.168.100.2:80 -> 192.168.200.2:34178
06/25-11:01:45.165001 [reset] [**] [1:1000005:1] "Serangan Brute Force ke Web Se
rver" [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1
] {TCP} 192.168.100.2:80 -> 192.168.200.2:34178
06/25-11:01:45.283804 [reset] [**] [1:1000005:1] "Serangan Brute Force ke Web Se
rver" [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1
] {TCP} 192.168.100.2:80 -> 192.168.200.2:34172
06/25-11:01:46.056363 [reset] [**] [1:1000005:1] "Serangan Brute Force ke Web Se
rver" [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1
] {TCP} 192.168.100.2:80 -> 192.168.200.2:34180
06/25-11:01:46.058347 [reset] [**] [1:1000005:1] "Serangan Brute Force ke Web Se
rver" [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1
] {TCP} 192.168.100.2:80 -> 192.168.200.2:34180
06/25-11:01:46.084283 [reset] [**] [1:1000005:1] "Serangan Brute Force ke Web Se
rver" [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1
] {TCP} 192.168.100.2:80 -> 192.168.200.2:34186
06/25-11:01:46.086267 [reset] [**] [1:1000005:1] "Serangan Brute Force ke Web Se
rver" [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1
] {TCP} 192.168.100.2:80 -> 192.168.200.2:34182
```

Fig. 15. Alert display on Snort to brute force attacks.

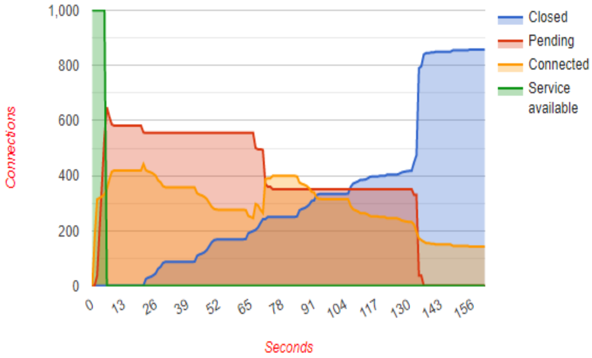


Fig. 16. Attack graph before IPS activated.

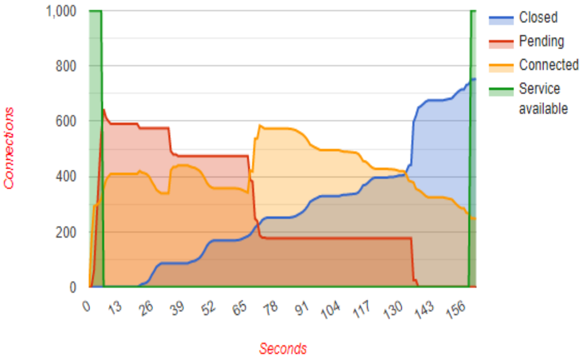


Fig. 17. Attack graph after IPS activated with drop rule.

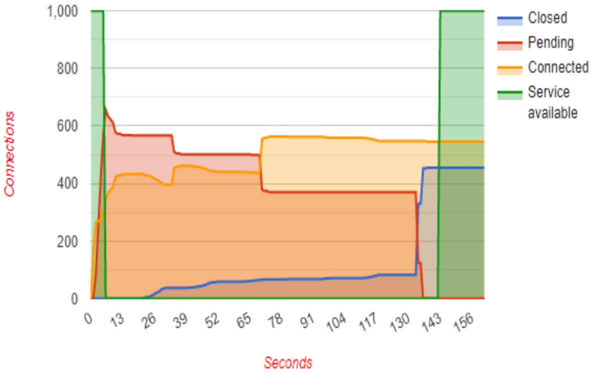
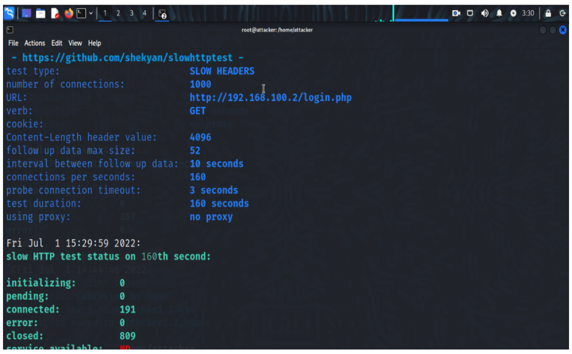


Fig. 18. Attack graph after IPS activated with reject rule.

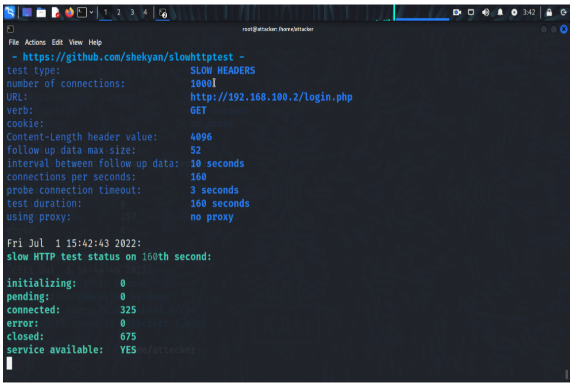
out attacks gradually, continually, and maintaining connection session time until closing the session by its own. In scenarios with IPS activated, the closed line will not reach the highest point, because before all closed attack connections they will be terminated first by IPS with a drop or reject rule on Snort. The Service Available line has a difference from the time of testing, namely when the scenario with IPS has not been activated where until the specified time there is no Service Available line which indicates that legal users cannot access the web server until the time of the attack is complete. In the scenario with IPS activated by using the drop rule, the Service Available line takes a longer time than using the reject rule, but both can withstand Slow HTTP attacks.

In IPS Scenario before it is activated, legal users cannot access the web server for 160 seconds or testing time, but at 165 seconds the web server can run smoothly according to the characteristics of the attack, namely when all attack connections are closed, the web server can return to serve. In the IPS scenario activated with a drop rule on Snort and Snort activated with a reject rule, both legal users can access the web server but at different times. The time required by IPS with the drop rule on Snort is much longer, which is 160 seconds compared to the reject rule on Snort with the required time of 145 seconds. The results of the DDoS test on the attacker can be seen in Fig. 19, Fig. 20, and Fig. 21.



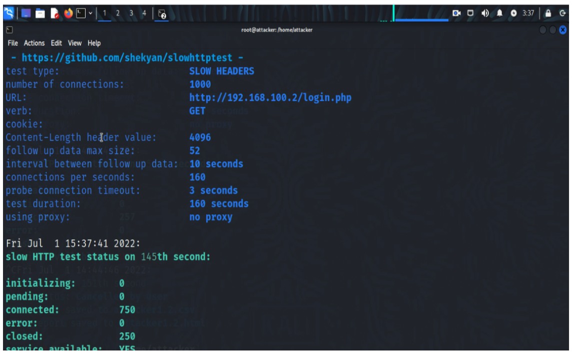
```
 - https://github.com/shekyan/slowhttptest -
test type:                        SLOW HEADERS
number of connections:            1000
URL:                              http://192.168.100.2/login.php
verb:                             GET
cookie:
Content-Length header value:      4096
follow up data max size:          52
interval between follow up data:  10 seconds
connections per seconds:          160
probe connection timeout:         3 seconds
test duration:                    160 seconds
using proxy:                      no proxy

Fri Jul  1 15:29:59 2022:
slow HTTP test status on 160th second:

initializing:       0
pending:            0
connected:          191
error:              0
closed:             809
service available:  NO
```

Fig. 19. DDoS test results on attacker before IPS active.



```
 - https://github.com/shekyan/slowhttptest -
test type:                        SLOW HEADERS
number of connections:            1000
URL:                              http://192.168.100.2/login.php
verb:                             GET
cookie:
Content-Length header value:      4096
follow up data max size:          52
interval between follow up data:  10 seconds
connections per seconds:          160
probe connection timeout:         3 seconds
test duration:                    160 seconds
using proxy:                      no proxy

Fri Jul  1 15:42:43 2022:
slow HTTP test status on 160th second:

initializing:       0
pending:            0
connected:          325
error:              0
closed:             675
service available:  YES
```

Fig. 20. DDoS test results on attacker after active IPS with drop rule.



```
 - https://github.com/shekyan/slowhttptest -
test type:                        SLOW HEADERS
number of connections:            1000
URL:                              http://192.168.100.2/login.php
verb:                             GET
cookie:
Content-Length header value:      4096
follow up data max size:          52
interval between follow up data:  10 seconds
connections per seconds:          160
probe connection timeout:         3 seconds
test duration:                    160 seconds
using proxy:                      no proxy

Fri Jul  1 15:37:41 2022:
slow HTTP test status on 145th second:

initializing:       0
pending:            0
connected:          750
error:              0
closed:             250
service available:  YES
```

Fig. 21. DDoS test results on attacker after active IPS with reject rule.

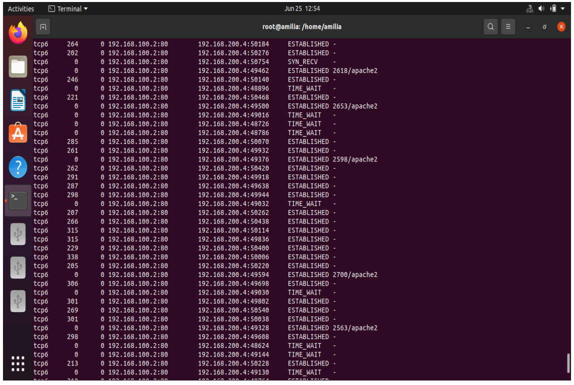
For legal users as long as the attacker attacks without activating Snort then when trying to access the web server, the connection will be disconnected. The web server is paralyzed because it tries to serve all requests from both the attacker and legal users. This causes legal users to experience buffering when accessing web server services or takes time to access services. When legal users access to web server services as long as the attacker attacks by activating Snort, the web server side can provide services to legal user requests. This is because IPS has succeeded in blocking the attack so that legitimate requests can be served. In this study, the results from the three scenarios still require time to be able to access web server services, but when the IPS server is activated, the time required is faster than before the IPS server was activated.

Table 3 is the result of observations on the CPU and RAM that can be seen from the web server side. The observed results are in accordance with table 3.4 that the average CPU usage of the five tests for the IPS scenario before being activated is 58% and the average RAM usage is 1.54 Gb. This is one of the main factors that the web server cannot serve requests so that users cannot access the web server services. When testing the attack before IPS is activated, there is a significant change in CPU and RAM usage. While the observations in the IPS scenario after being activated by using two actions have the same result that the average CPU usage from the five tests is 48.8% and RAM is 1.54 GB. CPU usage decreased by 9.2% from the difference in results before and after IPS was enabled. During the attack testing process when IPS is enabled, CPU and RAM usage has decreased and has a more stable value in a certain number range than before IPS was activated. This test can be concluded that the web server is successfully protected from attacks so that it can serve requests from legal users and can access web server services.

Table 3. Attacks Identifications with CIA Triads

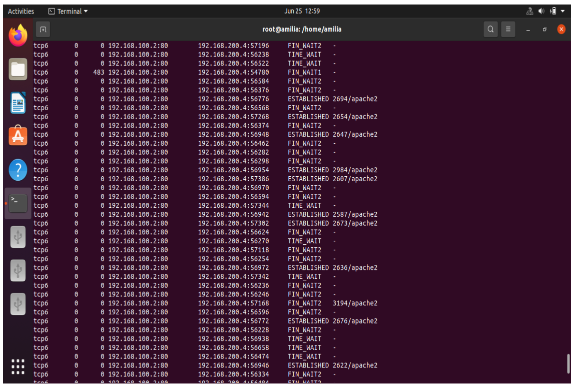
No	Observation Result					
	Before IPS activated		After IPS Activated			
			Action Drop		Action Reject	
	Memory (GB)	CPU (%)	Memory (GB)	CPU (%)	Memory (GB)	CPU (%)
1	1.3	65	1.3	57	1.3	56
2	1.6	57	1.6	45	1.6	50
3	1.6	56	1.6	49	1.6	46
4	1.6	55	1.6	48	1.6	47
5	1.6	57	1.6	45	1.6	45
Av.	**1.54**	**58**	**1.54**	**48**	**1.54**	**48**

On the web server side, observations on netstat which is used to monitor network connections that are running were also conducted. When making observations, the authors compared the IPS scenario before it was activated, IPS was activated with a drop rule on Snort and IPS was activated with a reject rule on Snort. The difference between the three scenarios is in the status of each obtained. When the IPS scenario before it is activated, it is seen that it will result in LAST_ACK and TIME_WAIT packages statuses that dominate at the end of the test where TIME_WAIT means that the web server is still waiting for a socket close to handle packages that are still on the network and LAST_ACK means that the remote is in a shutdown state, the socket is complete and closed the connection session as well as waiting for ACK. When IPS is activated, the drop rule in Snort results in TIME_WAIT and FIN_WAIT2 statuses that dominate at the end of the test where TIME_WAIT means that the web server is still waiting for the socket to close to handle packages that are still on the network and FIN_WAIT2 means the socket is closed but waiting for the remote shutdown side. When IPS is activated with a drop rule on Snort, it produces TIME_WAIT and FIN_WAIT2 statuses that dominate at the end of the test where TIME_WAIT means that the web server is still waiting for the socket to close to handle packets that are still on the network and FIN_WAIT1 means the socket is closed and the remote is already in a shutdown state. The results of the test of the IPS scenario before it is activated can be seen in Fig. 22, the IPS scenario is activated by the drop rule in Snort in Fig. 23 and IPS is activated by the reject rule in Snort in Fig. 24.



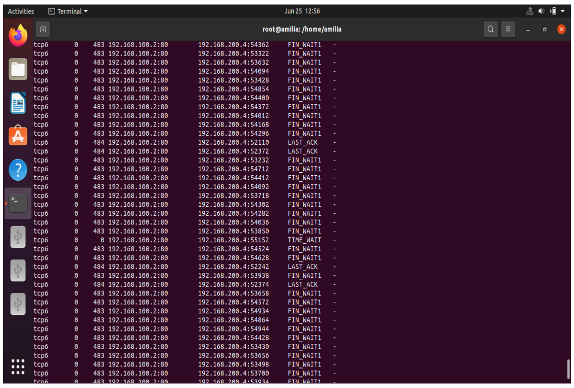
```
tcp6     264      0 192.168.100.2:80        192.168.200.4:50184     ESTABLISHED -
tcp6     202      0 192.168.100.2:80        192.168.200.4:50276     ESTABLISHED -
tcp6       0      0 192.168.100.2:80        192.168.200.4:50754     SYN_RECV    -
tcp6       0      0 192.168.100.2:80        192.168.200.4:49462     ESTABLISHED 2618/apache2
tcp6     246      0 192.168.100.2:80        192.168.200.4:50140     ESTABLISHED -
tcp6       0      0 192.168.100.2:80        192.168.200.4:48896     TIME_WAIT   -
tcp6     221      0 192.168.100.2:80        192.168.200.4:50468     ESTABLISHED -
tcp6       0      0 192.168.100.2:80        192.168.200.4:49500     ESTABLISHED 2653/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:49016     TIME_WAIT   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:48726     TIME_WAIT   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:48786     TIME_WAIT   -
tcp6     285      0 192.168.100.2:80        192.168.200.4:50070     ESTABLISHED -
tcp6     261      0 192.168.100.2:80        192.168.200.4:49932     ESTABLISHED -
tcp6       0      0 192.168.100.2:80        192.168.200.4:49376     ESTABLISHED 2598/apache2
tcp6     262      0 192.168.100.2:80        192.168.200.4:50420     ESTABLISHED -
tcp6     291      0 192.168.100.2:80        192.168.200.4:49918     ESTABLISHED -
tcp6     287      0 192.168.100.2:80        192.168.200.4:49638     ESTABLISHED -
tcp6     298      0 192.168.100.2:80        192.168.200.4:49944     ESTABLISHED -
tcp6       0      0 192.168.100.2:80        192.168.200.4:49032     TIME_WAIT   -
tcp6     207      0 192.168.100.2:80        192.168.200.4:50262     ESTABLISHED -
tcp6     266      0 192.168.100.2:80        192.168.200.4:50438     ESTABLISHED -
tcp6     315      0 192.168.100.2:80        192.168.200.4:50114     ESTABLISHED -
tcp6     315      0 192.168.100.2:80        192.168.200.4:49836     ESTABLISHED -
tcp6     229      0 192.168.100.2:80        192.168.200.4:50400     ESTABLISHED -
tcp6     338      0 192.168.100.2:80        192.168.200.4:50006     ESTABLISHED -
tcp6     205      0 192.168.100.2:80        192.168.200.4:50220     ESTABLISHED -
tcp6       0      0 192.168.100.2:80        192.168.200.4:49594     ESTABLISHED 2700/apache2
tcp6     306      0 192.168.100.2:80        192.168.200.4:49698     ESTABLISHED -
tcp6       0      0 192.168.100.2:80        192.168.200.4:49030     TIME_WAIT   -
tcp6     301      0 192.168.100.2:80        192.168.200.4:49802     ESTABLISHED -
tcp6     269      0 192.168.100.2:80        192.168.200.4:50540     ESTABLISHED -
tcp6     301      0 192.168.100.2:80        192.168.200.4:50038     ESTABLISHED -
tcp6       0      0 192.168.100.2:80        192.168.200.4:49328     ESTABLISHED 2563/apache2
tcp6     298      0 192.168.100.2:80        192.168.200.4:49608     ESTABLISHED -
tcp6       0      0 192.168.100.2:80        192.168.200.4:48624     TIME_WAIT   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:49144     TIME_WAIT   -
tcp6     213      0 192.168.100.2:80        192.168.200.4:50228     ESTABLISHED -
tcp6       0      0 192.168.100.2:80        192.168.200.4:49130     TIME_WAIT   -
```

Fig. 22. Netstat result display on web server before IPS activated.



```
tcp6       0      0 192.168.100.2:80        192.168.200.4:57196     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56238     TIME_WAIT   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56522     TIME_WAIT   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54780     FIN_WAIT1   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56584     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56376     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56776     ESTABLISHED 2694/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:56568     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:57268     ESTABLISHED 2654/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:56374     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56948     ESTABLISHED 2647/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:56462     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56282     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56298     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56954     ESTABLISHED 2984/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:57386     ESTABLISHED 2607/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:56970     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56594     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:57344     TIME_WAIT   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56942     ESTABLISHED 2587/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:57302     ESTABLISHED 2673/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:56624     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56270     TIME_WAIT   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:57118     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56254     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56972     ESTABLISHED 2636/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:57342     TIME_WAIT   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56236     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56246     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:57168     FIN_WAIT2   3194/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:56596     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56772     ESTABLISHED 2676/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:56228     FIN_WAIT2   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56938     TIME_WAIT   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56658     TIME_WAIT   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56474     TIME_WAIT   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:56946     ESTABLISHED 2622/apache2
tcp6       0      0 192.168.100.2:80        192.168.200.4:56334     FIN_WAIT2   -
```

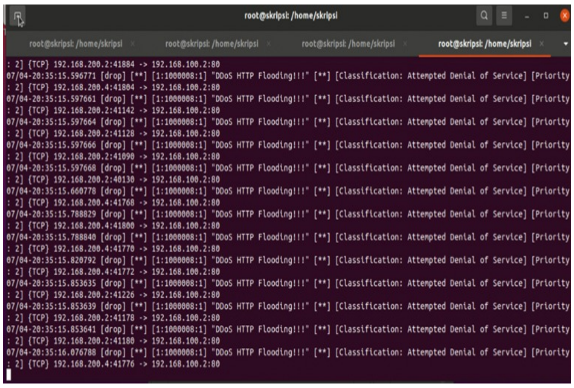
Fig. 23. Netstat result display on web server after IPS activated with drop rule.



```
tcp6       0    483 192.168.100.2:80        192.168.200.4:54362     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53322     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53632     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54094     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53428     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54854     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54400     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54372     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54012     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54168     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54296     FIN_WAIT1   -
tcp6       0    484 192.168.100.2:80        192.168.200.4:52110     LAST_ACK    -
tcp6       0    484 192.168.100.2:80        192.168.200.4:52372     LAST_ACK    -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53232     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54712     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54412     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54092     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53718     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54302     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54282     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54036     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53850     FIN_WAIT1   -
tcp6       0      0 192.168.100.2:80        192.168.200.4:55152     TIME_WAIT   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54524     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54628     FIN_WAIT1   -
tcp6       0    484 192.168.100.2:80        192.168.200.4:52242     LAST_ACK    -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53938     FIN_WAIT1   -
tcp6       0    484 192.168.100.2:80        192.168.200.4:52374     LAST_ACK    -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53658     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54572     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54934     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54864     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54944     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:54428     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53430     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53656     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53498     FIN_WAIT1   -
tcp6       0    483 192.168.100.2:80        192.168.200.4:53700     FIN_WAIT1   -
```

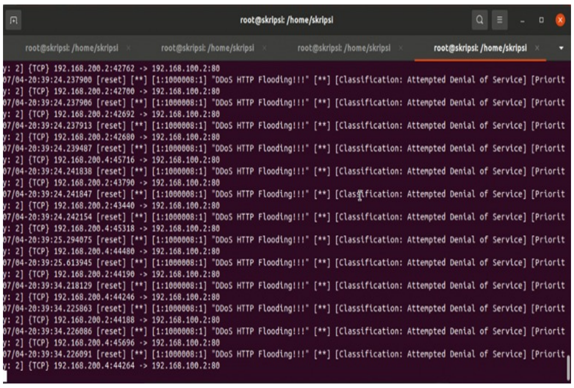
Fig. 24. Display Netstat results on the web server after IPS activated with reject rule.

On the Snort side, the test results can be seen in Fig. 25 and Fig. 26. The test results can be concluded that Snort can prevent attacks from attackers even though it still takes time and protects the web server side so that users legally can still access the service from the web server. Snort will send alerts or warnings to network administrators when there is a package mismatch with the rules that have been created. When using the drop rule, IPS will only drop packages that match the rule, but when using the reject rule, IPS will drop packages that match the rule and send TCP RST packages or termination of connection sessions on attack packages. The warning will continue to be displayed as long as an attack is launched.


root@skripsi: /home/skripsi
root@skripsi: /home/skripsi
root@skripsi: /home/skripsi
root@skripsi: /home/skripsi
root@skripsi: /home/skripsi

```
: 2] {TCP} 192.168.200.2:41884 -> 192.168.100.2:80
07/04-20:35:15.596771 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.4:41804 -> 192.168.100.2:80
07/04-20:35:15.597661 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.2:41142 -> 192.168.100.2:80
07/04-20:35:15.597664 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.2:41128 -> 192.168.100.2:80
07/04-20:35:15.597666 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.2:41090 -> 192.168.100.2:80
07/04-20:35:15.597668 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.2:40130 -> 192.168.100.2:80
07/04-20:35:15.660778 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.4:41768 -> 192.168.100.2:80
07/04-20:35:15.788829 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.4:41800 -> 192.168.100.2:80
07/04-20:35:15.788840 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.4:41770 -> 192.168.100.2:80
07/04-20:35:15.820792 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.4:41772 -> 192.168.100.2:80
07/04-20:35:15.853635 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.2:41226 -> 192.168.100.2:80
07/04-20:35:15.853639 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.2:41178 -> 192.168.100.2:80
07/04-20:35:15.853641 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.2:41180 -> 192.168.100.2:80
07/04-20:35:16.076788 [drop] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priority
: 2] {TCP} 192.168.200.4:41776 -> 192.168.100.2:80
```

Fig. 25. Display alert on snort against DDoS attacks with drop rule.


root@skripsi: /home/skripsi
root@skripsi: /home/skripsi
root@skripsi: /home/skripsi
root@skripsi: /home/skripsi
root@skripsi: /home/skripsi

```
y: 2] {TCP} 192.168.200.2:42762 -> 192.168.100.2:80
07/04-20:39:24.237900 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.2:42700 -> 192.168.100.2:80
07/04-20:39:24.237906 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.2:42692 -> 192.168.100.2:80
07/04-20:39:24.237913 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.2:42680 -> 192.168.100.2:80
07/04-20:39:24.239487 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.4:45716 -> 192.168.100.2:80
07/04-20:39:24.241838 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.2:43790 -> 192.168.100.2:80
07/04-20:39:24.241847 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.2:43440 -> 192.168.100.2:80
07/04-20:39:24.242154 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.4:45318 -> 192.168.100.2:80
07/04-20:39:25.294075 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.4:44480 -> 192.168.100.2:80
07/04-20:39:25.613945 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.2:44190 -> 192.168.100.2:80
07/04-20:39:34.218129 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.4:44246 -> 192.168.100.2:80
07/04-20:39:34.225863 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.2:44188 -> 192.168.100.2:80
07/04-20:39:34.226086 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.4:45696 -> 192.168.100.2:80
07/04-20:39:34.226091 [reset] [**] [1:1000008:1] "DDoS HTTP Flooding!!!" [**] [Classification: Attempted Denial of Service] [Priorit
y: 2] {TCP} 192.168.200.4:44264 -> 192.168.100.2:80
```

Fig. 26. Display alert on Snort against DDoS attacks with reject rule.

IV. CONCLUSION

In this paper, we have discussed the implementation research of IPS for analyzing CIA Triads against network security attacks on web servers. The study shows that IPS can block the port information from the attacker when IPS is activated. The IPS also reduce the effect of DDoS attack by dropping before reaching the server. Nevertheless, the implementation of Snort and IPTables as IPS in this study still has shortcomings and the accuracy of the rules, especially DDoS attacks, so in future research, it is expected to formulate more specific rules with attack characteristics and always update the rules. Future research can also replace using other snort detection methods such as anomalies based on NIPS or by combining other technologies such as SDN, load balancers, or cloud computing.

ACKNOWLEDGMENT

This research was supported by the Institute for Research and Community Service (LPPM) Institut Teknologi Telkom Purwokerto.

REFERENCES

[1] D. Novianty and D. Prastya, ”Jumlah pengguna internet di indonesia capai 202,6 juta orang,” suara.com, Feb. 15, 2021.

[2] kompas.com, ”BSSN Catat Adanya 88, 4 Juta Serangan Siber Selama Pandemi Corona,” kompas.com, Apr. 23, 2020.

[3] B. Wijaya and A. Pratama, ”Deteksi penyusupan pada server menggunakan metode intrusion detection system (IDS) berbasis snort,” *SISFOKOM (Sistem Informasi dan Komputer)*, vol. 9, no. 1, pp. 97–101, 2020.

[4] Y. Yanti and R. Effendi, ”Analisa sistem keamanan jaringan komputer firewall menggunakan shorewall pada PT. indofarma global medika,” *Jurnal TEKSAGRO*, vol. 1, no. 2, pp. 14–21, 2020.

[5] A. R. Kelrey and A. Muzaki, ”Pengaruh ethical hacking bagi keamanan data perusahaan,” *CyberSecurity dan Forensik Digital*, vol. 2, no. 2, pp. 77–81, 2019.

[6] Saritha, B. R. Reddy, and A. S. Babu, ”Prediction of DDoS attacks using machine learning and deep learning algorithms,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 4, pp. 4860–4867, 2019.

[7] M. Arman, ”Metode pertahanan web server terhadap distributed slow HTTP DoS attack,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, pp. 56–70, 2020.

[8] A. Evgeny V, N. Arina V, and K. Irina S, ”Port scanning detection based on anomalies,” in *2017 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, Nov. 2017.

[9] F. A. Prayogo and I. R. Widiasari, ”Perancangan Sistem Pencegahan Serangan Bruteforce pada Jaringan Wireless,” Feb. 2017.

[10] H. Alamsyah, Riska, and A. A. Akbar, ”Analisa keamanan jaringan menggunakan network intrusion detection and prevention system,” *JOINTECS (Journal of Information Technology and Computer Science)*, vol. 5, no. 1, pp. 17–24, 2020.

[11] R. S. Putra, R. Mayasari, and N. B. A. Karna, ”Implementasi dan analisis keamanan jaringan virtual HIPS snort pada layanan web server dengan penyerangan DoS dan DDoS,” in *e-Proceeding of Engineering*, vol. 5, no. 3, 2018, pp. 4958–4965.

[12] R. Suwanto, I. Ruslianto, and M. Diponegoro, ”Implementasi intrusion prevention system (IPS) menggunakan snort dan IPTABLE pada monitoring jaringan lokal berbasis website,” *Jurnal Komputer dan Aplikasi*, vol. 7, no. 1, pp. 97–107, 2019.

[13] Y. W. Pradipta and Asmunin, ”Implementasi intrusion prevention system (IPS) menggunakan snort dan iptables berbasis linux,” *Jurnal Manajemen Informatika*, vol. 7, no. 1, pp. 21–28, 2017.

[14] E. S. J. Atmadji, B. M. Susanto, and R. Wiratama, ”Pemanfaatan iptables sebagai intrusion detection system (IDS) dan intrusion prevention system (IPS) pada linux server,” *TEKNIKA*, vol. 6, no. 1, pp. 19–23, 2017.

[15] M. Katoningati, I. Gunawan, R. I. Salsabila, and A. E. D. Melania, ”Analisis layer aplikasi (protokol HTTP) menggunakan wireshark,” *JES (Jurnal Elektro Smart)*, vol. 1, no. 1, pp. 13–15, 2021.

[16] D. B. Rendro, Ngatono, and W. N. Aji, ”Analisis monitoring sistem keamanan jaringan komputer menggunakan software NMAP (studi kasus di SMK negeri 1 kota serang),” *Jurnal Prosisko*, vol. 7, no. 2, pp. 108–115, 2020.

[17] A. A. Kurniawan and Y. Nugroho, ”Upaya penetrasi dengan enumeration menggunakan hydra,” *Journal of Technology and Informatics (JoTI)*, vol. 1, no. 1, pp. 61–64, 2019.

[18] M. M. Rasheed, A. K. Faieq, and A. A. Hashim, ”Development of a new system to detect denial of service attack using machine learning classification,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 1068–1072, 2021.

[19] F. R. Nurdiana, I. Gunawan, R. C. Viollita, M. A. Faizal, and D. Nurcahyadi, ”Analisis keamanan Wifi menggunakan wireshark,” *JES (Jurnal Elektro Smart)*, vol. 1, no. 1, pp. 10–12, 2021.

[20] Security Policies User Guide for Security Devices, Sunnyvale, California: Juniper Networks, Inc., 2022.