Analysis Performance Intrusion Detection System in Detecting Cyber-Attack on Apache Web Server

Doddy Teguh Yuwono[1], Setio Ardy Nuswantoro [2]
Department of Computerr Science, University of Muhammadiyah Palangkaraya[1]
Departement Information Technology Education Study Program,
University of Muhammadiyah Palangkaraya [2]
doddy.zha09@gmail.com[1], setioardy@gmail.com[2]

Article Info

Article history:

Received Okt 11, 2021
Revised Dec 12, 2021
Accepted Feb 22, 2022

Keyword:

Analyze
Detecting
Network
Security
Snort

ABSTRACT

Network security on the webserver is the most important part to ensure integrity and service for users. Web servers are often the target of attacks that result in data corruption. One of them is the SYN Flood attack, which is a type of Denial of Service (DOS) attack that provides massive SYN requests to the webserver. This research is to analyze attack indications and maintain system security from the threat of data flooding. One way to maintain a computer network security system is to use Snort as an IDS (Intrusion Detection System). Snort is software that functions to detect intrusions. Data packets passing through network traffic will be analyzed first. Data packets detected as intrusions will trigger an alert which is then stored in a log file. That way, network administrators can find out intrusions that occur on computer networks. The method of testing flood attack data is using the penetration testing method. The three test samples are data flooding attacks against ICMP, UDP, and TCP protocols. The results obtained when testing flooding attack data where detection sensors can detect all attacks and all attack samples, while the warnings generated by Snort are shown in a web form which can be seen in the detail of each attack that occurred.

Corresponding Author:

Doddy Teguh Yuwono
Departement of of Computerr Science
University of Muhammadiyah Palangkaraya
Jl. RTA. Milono KM 1.5 Palangka Raya, Palangka Raya, Kalimantan Tengah, Indonesia, 73112
Email: doddy.zha09@gmail.com

1. INTRODUCTION

In today's global era, Information Technology (IT) has developed rapidly, especially with the internet network that can facilitate communication with other parties. With easy access to this information, new problems arise, namely important information or data can be used by irresponsible parties for their benefit. So that a network security system becomes one of the important aspects.[1][2]

A network and internet server manager (system administrator) has responsibility for system security from time to time, ensuring that the system and network being managed are protected from various threats. School is one of the places where the use of the internet network is open to its users. These uses can be used properly and are not misused.[3] This resulted in a network system that should be used for learning but misused for other activities such as accessing social networks. In addition, the administrator must know something log that identifies an attack or network abuse.[4]

Therefore, a system is needed to handle network abuse or threats that will occur, namely by using the application, Intrusion Detection System (IDS) namely Snort and Snorby as monitoring of the resulting snort alerts. Snorby is a front-end web application (written in Ruby on Rails) for monitoring network security.

2. RESEARCH METHOD

The method used is the SPDLC (Security Policy Development Life Cycle) method because this method is appropriate to use in presenting the stages of system development related to network security following the research raised[5][6]. SPDLC is a method that defines a strategy for updating an organization's network system, the network system development cycle is defined in several phases. In the research of Veren Prisscilya and Tri Santoso, explaining the Security Policy Development Life Cycle (SDPLC) can help in the process of testing network and system security problems that are running[7]. According to Luay A. Wahsheh and Jim Alves Foss, the development of the SPDLC system was carried out by researching with 6 stages including identification, analysis, design, implementation, testing/auditing, analysis of results/evaluation[8][9].

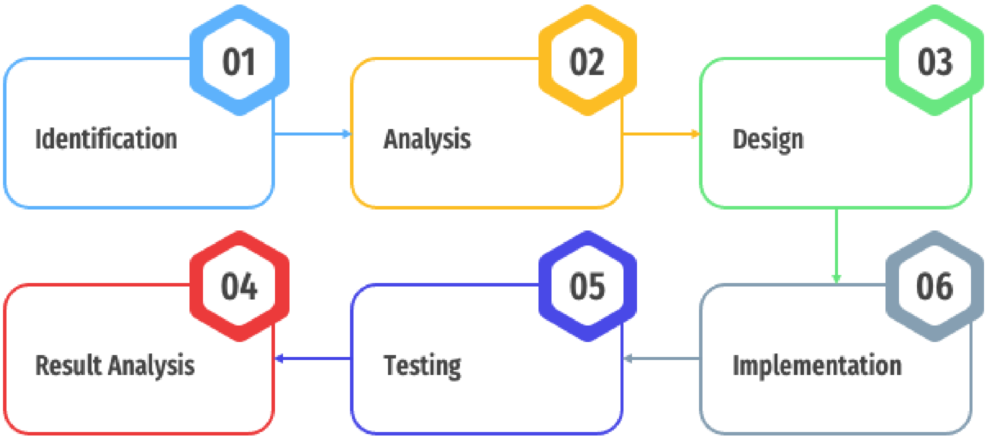


Figure 1. SPDLC Method

SPDLC is described as a stage starting from the evaluation stage which validates the effectiveness of the initial analysis stage[10][11]. Feedback from this evaluation can have an impact on changes in the architecture and technology used today[6][12]. The explanation of the stages in Figure 1 is as follows:

1. Identification: At this stage, the process of identifying problems is carried out which is used as the basis for journals and books to support research.
2. Analysis: At this stage, the author conducts an analysis based on the problems that have been described in the identification of problems, such as determining the software to be used, determining the topology related to the problem.
3. Design: At this stage the author makes a design such as the flow of detection of attacks, designing the topology that will be used.
4. Implementation: The author does the implementation based on the scenario that has been made, by installing and configuring all the software used and ready for testing.
5. Testing: At this stage, testing of the IDS, namely Snort, is carried out according to the parameters that have been determined and carried out from the IDS.
6. Results Analysis: The last stage is to describe the results obtained from the IDS test with analysis according to the parameters made so that conclusions are found from the IDS test.

2.1. Specifications Hardware and Software

Specifications of hardware and software used are, In this study, the hardware used is as follows at Table 1.

Table 1. Hadware

Brand	Macbook Pro 2015 (Attacker)	Acer E-14 (Attacker)	Asus Vivo-book (Server)
Processor	Intel Core i5	Intel Core i5 3.2 GHz	Intel Core i5 3.2 GHz
Memory	16 GB	4 GB DDR3	4 GB DDR3
Storage	256 GB	500 GB	500GB

In this research, the software used on the computer are is like Table 2.

Table 2. Software

Brand	Macbook Pro 2015 (Attacker)	Acer E-14 (Attacker)	Asus Vivo-book (Server)
OS	Catalina OS	Windows 10	Ubuntu Server 20 LTS
Software	• Zenmap • Loic • hping3 • Hydra	• Zenmap • Loic • hping3 • Medusa	• Snort • *Snorby*

2.2. Application

Design The system design that will be used to design a system that can detect intruders or attacks is the Intrusion Detection System (IDS), which previously required the tools or components needed to build the system which will work together to get maximum results.

1. **Snort**

A snort is security software that is very useful for observing activity in a computer network. Snort is an open-source GNU (General Public License), so it can be used freely and free of charge, the source code for Snort can also be obtained and modified yourself. In its use, Snort is still clean in the command line so it is quite inconvenient for users who are accustomed to using the Graphical User Interface (GUI).[13]

Snort can be a packet sniffer that allows Snort to read existing computer network traffic. Snort can also create logging (packet logger) of network traffic and alerts that occur, which allows an administrator to perform analysis to create a more secure computer network and can provide alerts if there is a suspicious activity.[14][12]

In addition, Snort has 2 techniques in recognizing intrusions that are happening on a computer network, the first using a signature in a database and matching it with existing network traffic. The second is with anomaly detection by comparing the network traffic being monitored with network traffic that usually occurs. With this capability, Snort can simplify the handling of computer network security. The snort program can be operated in three modes:

a. **Packet Sniffer**

Reads packets from the network and shows an uninterrupted stream from on the console (screen). [12] If you only want to see the header packets from TCP/IP on the screen, try using the command :

```
./snort -v
```

b. **Packet Logger**

Logs the packets to disk. If you want to keep a log of packets to disk, it is necessary to include a logging directory, i.e. where the log data is stored on it. [12] Using the following command Snort will automatically run in packet logging mode:

```
./snort –dev –l ./log
```

c. **NIDS (Network Intrusion Detection System)**
In this mode, snort will function to detect attacks carried out through computer networks. To enable the NIDS (Network Intrusion Detection System) network intruder detection system mode use the following command :

```
./snort –dev –l ./log –h 192.168.1.0/24 –c snort.conf
```

2. Snorby

Snorby is a front-end web application (written in Ruby on Rails) for monitoring network security related to systems network intrusion detection. The choice of using Snorby is because it has a good appearance and has functions that make it easier for administrators to tune the implemented rules.[15] To run Snorby we can write the command "bundle exec rails server –e production" in the terminal window ubuntu server. The following is a display snobby which can be seen in the following Figure 2.

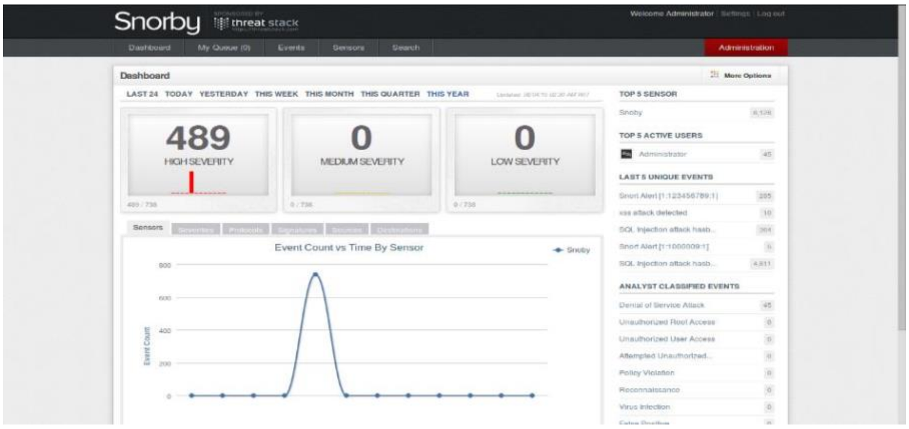


Figure 2. Snorby Web Display

2.3. Network Topology

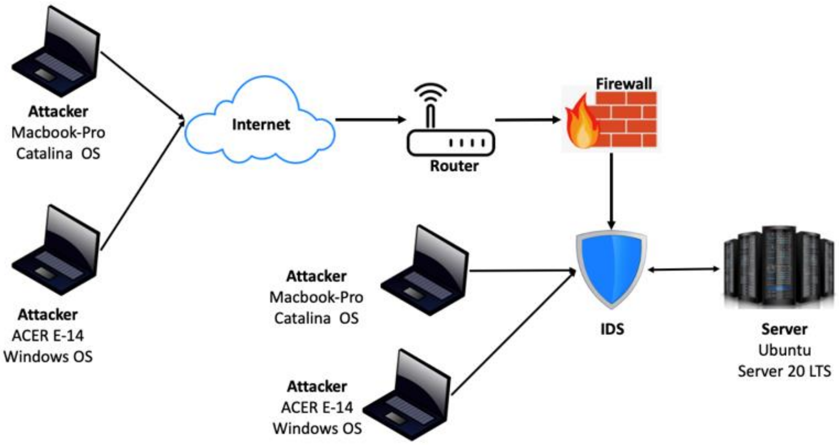


Figure 3. Network Topology

The system running by IDS has been configured with two interfaces, one of which is connected to the internal network and the other is connected to the external network. The scenarios that will be used are:

1. First Scenario PC Attacker will perform one ARP Spoofing attack.
2. The second scenario is that the PC Attacker performs two ARP Spoofing attacks at different times.
3. The third scenario, two PC Attackers perform ARP Spoofing attacks at the same time.

Attackers perform ARP Spoofing attacks using Loic and Hping3 tools that run on Windows and Macbooks. Low Orbit Ion Cannon (LOIC) is a simple flooding tool that can generate massive volumes of TCP, UDP, or HTTP traffic to subject a server to a heavy network load and The DDoS attack tool hping3 is a fairly basic command-line utility similar to the ping utilities. However, it offers more functionality than simply sending an ICMP echo request. In fact, hping3 can be used to send large volumes of TCP traffic to a target while spoofing the source IP addresses, making it appear to be random or even to originate from a specific, user-defined source. This powerful, robust tool is among Anonymous' current DDoS attack tools of choice.

4. The fourth scenario is to use SSH Brute Force.

In this test, we will use Hydra and Medusa Tools. Medusa is a tool that is used to carry out brute force attacks using a dictionary or list of passwords to enter a system and Hydra is a tool for hacking passwords using brute force, that is, using many dictionaries that contain to do the possible on passwords or usernames. used both have in common in the Brute Force process. SSH Brute Force is a technique that tries everything possible in carrying out an attack.

3. RESULTS AND ANALYSIS

3.1. Initial Network Testing

In the initial tests carried out on computers that were not installed by a snort and tried to attack the computer. The results obtained are:

1. When there is an attempted attack and network abuse, the computer does not know if someone is about to attack a computer that is not installed by snort.
2. Snort that has not been installed on the computer makes it impossible to read network traffic because to see the traffic of packets on the network, the snort must be installed first.

3.2. Final Network Testing

In the final test, the computer will be installed a snort system. The snort system will be tested with attempted attacks and will display alerts according to the threat and display detected network abuse. The following are the stages of testing:

Dos is a technique used by hackers to prevent legitimate users from accessing information or service by attacking computers and computer networks of the desired target. Testing on this type of attack using Loic and Hping3.

1. The rules used to detect Dos using :

   alert tcp any any -> any any (msg:"Posible DDoS attack attemp -- 1";flow:to_server,established ;flags:PA ; threshold: type threshold, track by_src,count 1000,seconds 10;classtype:attempted-dos;sid:1000011;

2. The rules used to detect Dos using Hping3 :

   alert tcp any any -> any any (msg:"DDos attack hping3";flow:to_server; flags:S; threshold: type threshold, track by_src, seconds 10, count 1000; classtype:attempted-dos; sid:1000012; rev:1;)

The rule states that snort will warn packets using the TCP protocol with source and destination IP any and source port and destination port any. Based on the first rule Snort will write the message Possible DDoS attack attempt –1 and based on the second rule is DDoS attack hping3. The first rule has an id number of 1000011 and the second rule has an id of 1000012 and both rules are version 1.

rules in this rule, the factors that determine Snort to detect a threat are flags and threshold. Flags are control bits that indicate or indicate different connection states or information on how a packet should be handled. In the first rule, Snort will alert if there are packets with flags P (push) and A (ACK) and in the second rule will give alerts if there are packets with flags S (SYN). A threshold is used to reduce the number of events logged from a rule. The threshold command limits the number of events that are logged during a given time interval. The first and second rules mean that Snort will record if for 10 seconds there are at least 1000 events based on the same source IP. After Snort takes notes, the time will be repeated from 0.

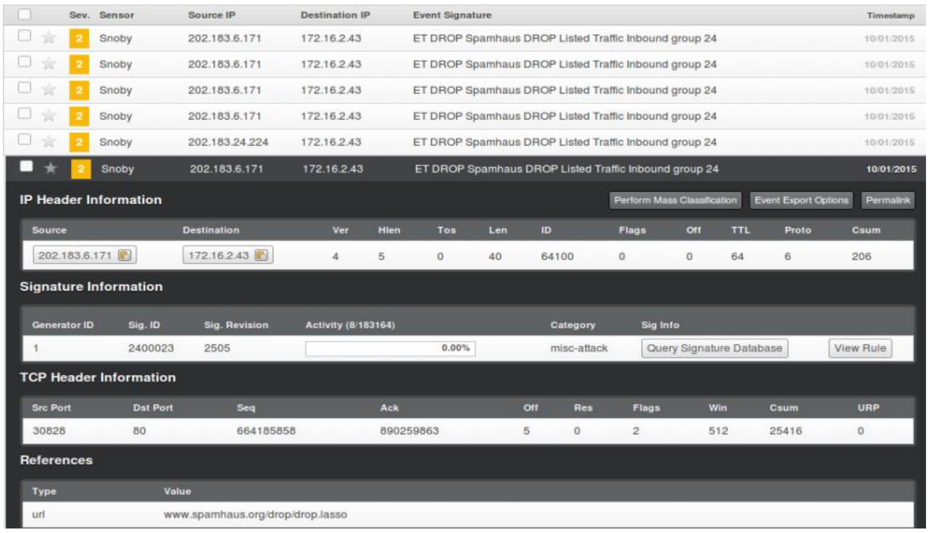
After testing, a comparison is also made between the rules applied and the rules that have been found previously. The following are the results of the comparison of these rules, the following is a comparison of the rules that have been found previously with the rules that have been applied which can be seen in Table 3.

Table 3. Comparison of the DoS

Rule	Loic	Hping 3	
		Rand Source	Same Source
1	Yes	Yes	Yes
2	No	Yes	Yes

From table 3, it can be seen that rule number 1 (the rule that is applied) can detect the attack being tested from both Loic and Hping3 tools. Making rule number 1 is done by applying the previously found rule number 2 in table 3 (can be seen in the appendix in the section on the Denial of Service comparison rule) and testing. The test results are also used as the basis for the tuning rule. Tuning is done by looking at the factors that influence this attack. In this case, flags are generated in the TCP header that can be seen when indicating the number of times the event occurred in a unit of time.

The following is a display of Dos test results using Loic and Hping3 which can be seen in Figure 4 and Figure 5.



□		Sev.	Sensor	Source IP	Destination IP	Event Signature	Timestamp
□	☆	2	Snoby	202.183.6.171	172.16.2.43	ET DROP Spamhaus DROP Listed Traffic Inbound group 24	10/01/2015
□	☆	2	Snoby	202.183.6.171	172.16.2.43	ET DROP Spamhaus DROP Listed Traffic Inbound group 24	10/01/2015
□	☆	2	Snoby	202.183.6.171	172.16.2.43	ET DROP Spamhaus DROP Listed Traffic Inbound group 24	10/01/2015
□	☆	2	Snoby	202.183.6.171	172.16.2.43	ET DROP Spamhaus DROP Listed Traffic Inbound group 24	10/01/2015
□	☆	2	Snoby	202.183.24.224	172.16.2.43	ET DROP Spamhaus DROP Listed Traffic Inbound group 24	10/01/2015
□	★	2	Snoby	202.183.6.171	172.16.2.43	ET DROP Spamhaus DROP Listed Traffic Inbound group 24	10/01/2015

IP Header Information

Source	Destination	Ver	Hlen	Tos	Len	ID	Flags	Off	TTL	Proto	Csum
202.183.6.171	172.16.2.43	4	5	0	40	64100	0	0	64	6	206

Signature Information

Generator ID	Sig. ID	Sig. Revision	Activity (8/183164)	Category	Sig Info
1	2400023	2505	0.00%	misc-attack	Query Signature Database View Rule

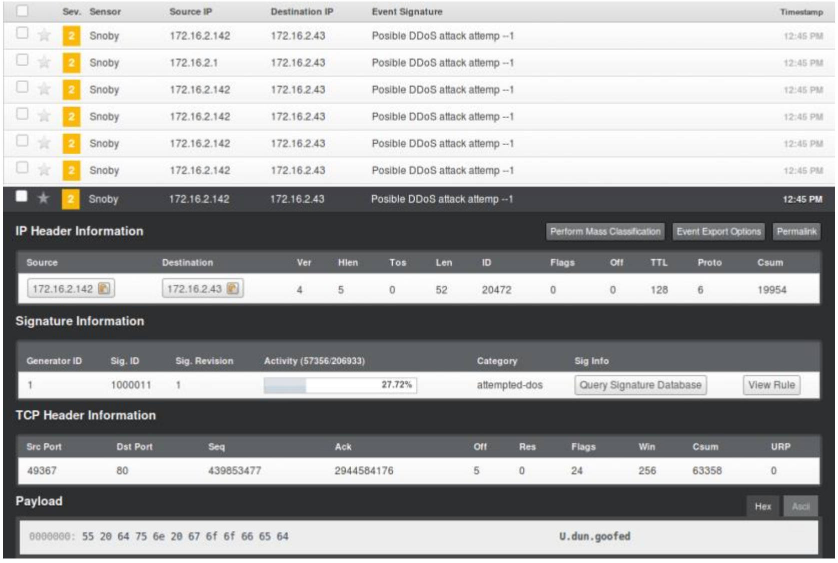
TCP Header Information

Src Port	Dst Port	Seq	Ack	Off	Res	Flags	Win	Csum	URP
30828	80	664185858	890259863	5	0	2	512	25416	0

References

Type	Value
url	www.spamhaus.org/drop/drop.lasso

Figure 4. Hping3 DoS and details on the attack



□		Sev.	Sensor	Source IP	Destination IP	Event Signature	Timestamp
□	☆	2	Snoby	172.16.2.142	172.16.2.43	Posible DDoS attack attemp --1	12:45 PM
□	☆	2	Snoby	172.16.2.1	172.16.2.43	Posible DDoS attack attemp --1	12:45 PM
□	☆	2	Snoby	172.16.2.142	172.16.2.43	Posible DDoS attack attemp --1	12:45 PM
□	☆	2	Snoby	172.16.2.142	172.16.2.43	Posible DDoS attack attemp --1	12:45 PM
□	☆	2	Snoby	172.16.2.142	172.16.2.43	Posible DDoS attack attemp --1	12:45 PM
□	☆	2	Snoby	172.16.2.142	172.16.2.43	Posible DDoS attack attemp --1	12:45 PM
□	★	2	Snoby	172.16.2.142	172.16.2.43	Posible DDoS attack attemp --1	12:45 PM

IP Header Information

Source	Destination	Ver	Hlen	Tos	Len	ID	Flags	Off	TTL	Proto	Csum
172.16.2.142	172.16.2.43	4	5	0	52	20472	0	0	128	6	19954

Signature Information

Generator ID	Sig. ID	Sig. Revision	Activity (57356/206933)	Category	Sig Info	
1	1000011	1	27.72%	attempted-dos	Query Signature Database	View Rule

TCP Header Information

Src Port	Dst Port	Seq	Ack	Off	Res	Flags	Win	Csum	URP
49367	80	439853477	2944584176	5	0	24	256	63358	0

Payload

```
0000000: 55 20 64 75 6e 20 67 6f 6f 66 65 64                    U.dun.goofed
```

Figure 5. Loic Dos and details on attacks

SSH Brute Force is a technique that tries everything possible in carrying out an attack. SSH Brute Force is divided into standard SSH Brute Force and dictionary-based attacks. Testing on this type of attack using Hydratools and Medusa. The following is a rule used to detect SSH Brute Force actions on a computer network.

3. The The rules used to detect Brute Force Attack

   alert tcp any any -> $HOME_NET 22 (msg:"Potential SSH Brute Force Attack";flow:to_server; flags:A; threshold:type threshold, track by_src, count 15, seconds 60;classtype:attempted-dos;sid:1000020;rev:1;)

This rule provides information that Snort will detect all data packets using the TCP protocol from any source IP to the home network IP and from any port to port 22. Based on this rule, Snort will group threats detected in attempted dos. The first rule has an id of 1000020 and is a version 1 rule.

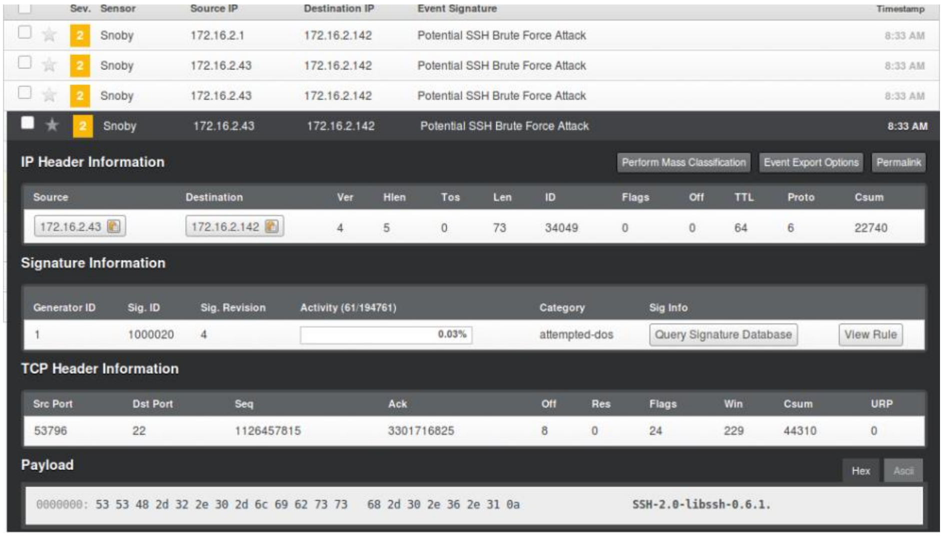
In this rule, what determines Snort can detect the threats being tested are flags and threshold. Flags are control bits that indicate or indicate different connection states or information on how a packet should be handled. In the first rule, Snort will make an alert if there is a packet with flags A (ACK). The threshold is used to reduce the number of events logged from a rule. The threshold command limits the number of events that are logged during a given time interval. The type threshold in the rule means that Snort will record if for 60 seconds there are at least 15 events.

After testing, a comparison is also made between the applied rules and the rules that have been found previously. The following are the results of the comparison of these rules, the following is a comparison of the rules that have been found previously with the rules that have been applied which can be seen in Table 4.

Table 4. Comparison of the DoS

Rule	Hydra	Medusa	Remarks
1	Yes	Yes	No Error
2	Yes	Yes	There is an Error

From Table 4 it can be seen that rule number 1 (the rule that is applied) can detect the attacks being tested from either the Hydra or Medusa tools. Making rule number 1 is done by applying the previously found rule number 2 in table 2 (can be seen in the appendix in the SSH Brute Force comparison rule section) and testing. The test results are also used as the basis for the tuning rule. Tuning is done by looking at the factors that influence this attack. In this case, flags are generated in the TCP header which can be seen when indicating the number of times the event occurred in a unit of time. The following is a display of the results of the SSH Brute Force test display using Hydra and Medusa which can be seen in Figure 6 and Figure 7.



□		Sev.	Sensor	Source IP	Destination IP	Event Signature	Timestamp
□	☆	2	Snoby	172.16.2.1	172.16.2.142	Potential SSH Brute Force Attack	8:33 AM
□	☆	2	Snoby	172.16.2.43	172.16.2.142	Potential SSH Brute Force Attack	8:33 AM
□	☆	2	Snoby	172.16.2.43	172.16.2.142	Potential SSH Brute Force Attack	8:33 AM
□	★	2	Snoby	172.16.2.43	172.16.2.142	Potential SSH Brute Force Attack	8:33 AM

IP Header Information

Source	Destination	Ver	Hlen	Tos	Len	ID	Flags	Off	TTL	Proto	Csum
172.16.2.43	172.16.2.142	4	5	0	73	34049	0	0	64	6	22740

Signature Information

Generator ID	Sig. ID	Sig. Revision	Activity (61/194761)	Category	Sig Info	
1	1000020	4	0.03%	attempted-dos	Query Signature Database	View Rule

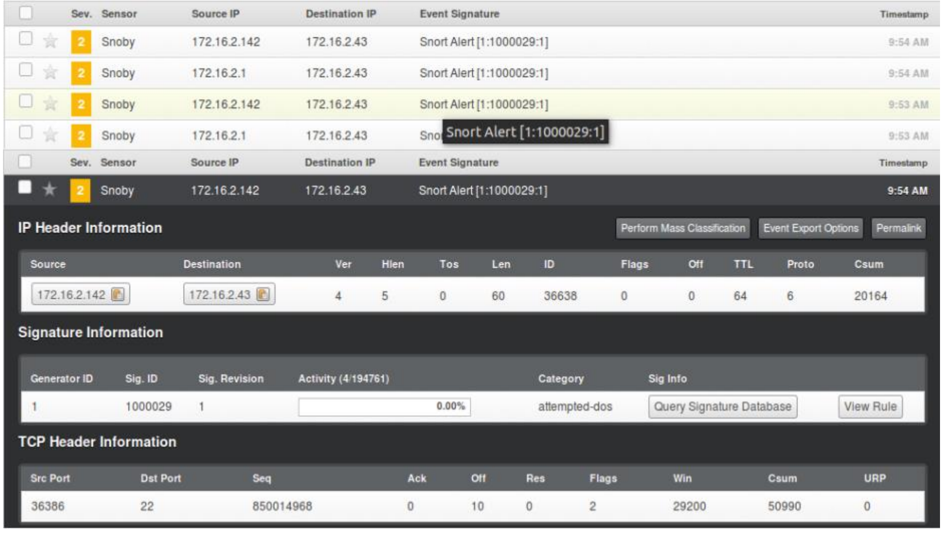
TCP Header Information

Src Port	Dst Port	Seq	Ack	Off	Res	Flags	Win	Csum	URP
53796	22	1126457815	3301716825	8	0	24	229	44310	0

Payload

```
0000000: 53 53 48 2d 32 2e 30 2d 6c 69 62 73 73   68 2d 30 2e 36 2e 31 0a          SSH-2.0-libssh-0.6.1.
```

Figure 6. Hydra SSH Brute Force and attack details



□		Sev.	Sensor	Source IP	Destination IP	Event Signature	Timestamp
□	☆	2	Snoby	172.16.2.142	172.16.2.43	Snort Alert [1:1000029:1]	9:54 AM
□	☆	2	Snoby	172.16.2.1	172.16.2.43	Snort Alert [1:1000029:1]	9:54 AM
□	☆	2	Snoby	172.16.2.142	172.16.2.43	Snort Alert [1:1000029:1]	9:53 AM
□	☆	2	Snoby	172.16.2.1	172.16.2.43	Sno Snort Alert [1:1000029:1]	9:53 AM
□		Sev.	Sensor	Source IP	Destination IP	Event Signature	Timestamp
□	★	2	Snoby	172.16.2.142	172.16.2.43	Snort Alert [1:1000029:1]	9:54 AM

IP Header Information

Source	Destination	Ver	Hlen	Tos	Len	ID	Flags	Off	TTL	Proto	Csum
172.16.2.142	172.16.2.43	4	5	0	60	36638	0	0	64	6	20164

Signature Information

Generator ID	Sig. ID	Sig. Revision	Activity (4/194761)	Category	Sig Info	
1	1000029	1	0.00%	attempted-dos	Query Signature Database	View Rule

TCP Header Information

Src Port	Dst Port	Seq	Ack	Off	Res	Flags	Win	Csum	URP
36386	22	850014968	0	10	0	2	29200	50990	0

Figure 7. Medusa SSH Brute Force and attack details

4. CONCLUSION

Based on the test results and previous results, the following conclusions were drawn:

1. Testing and implementation of the rules used can detect all types of attacks being tested.
2. The ability of each rule to detect attacks based on the precision rate and recall rate produces a value of 1 for each type of attack tested. This shows the rule can detect 100% of the tested threats and does not produce false positives.
3. The warnings generated by Snort are shown in a web form which can be seen in the detail of each attack generated.

ACKNOWLEDGEMENTS

In this research, the writer gives thanks to the Universitas Muhammadiyah Palangkaraya, which has provided financial support. As well as the LP2M University of Muhammadiyah Palangkaraya who have participated as mentor who support the smooth implementation of this research.

REFERENCES

[1] U. L. Yusuf Abdulloh, Joko Triyono, “Pengaruh Penempatan Snort Terhadap Keamanan Jaringan (Studi Kasus Laboratorium Vi Jaringan Kampus 3 Ist Akprind Yogyakarta),” *Jarkom*, vol. 8, no. 1, pp. 10–19, 2020.

[2] Y. Arta, A. Syukur, and R. Kharisma, “Simulasi Implementasi Intrusion Prevention System (IPS) Pada Router Mikrotik,” *It J. Res. Dev.*, vol. 3, no. 1, pp. 104–114, 2018, doi: 10.25299/itjrd.2018.vol3(1).1346.

[3] A. L. Ginting, J. Napitupulu, and J. Jamaluddin, “Sistem Monitoring Pendeteksian Penyusup Menggunakan Snort pada Jaringan Komputer Fakultas Ekonomi Universitas Methodist Indonesia,” pp. 83–87, 2018, doi: 10.31227/osf.io/w5gt7.

[4] Y. Arta, “Implementasi Intrusion Detection System Pada Rule Based System Menggunakan Sniffer Mode Pada Jaringan Lokal,” *It J. Res. Dev.*, vol. 2, no. 1, pp. 43–50, 2017, doi: 10.25299/itjrd.2017.vol2(1).979.

[5] Firmansyah and M. Wahyudi, “Analisis Performa Access Control List menggunakan Metode Firewall Policy Base Performance Analysis of the Access Control List Using the Firewall Policy-Based Method Article Info ABSTRAK,” *Matrik J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 20, no. 2, pp. 283–292, 2021, doi: 10.30812/matrik.v20i1.1068.

[6] A. H. Hambali and S. Nurmiati, “Implementasi Intrusion Detection System (IDS) Pada Keamanan PC Server Terhadap Serangan Flooding Data,” *Sainstech J. Penelit. dan Pengkaj. Sains dan Teknol.*, vol. 28, no. 1, pp. 35–43, 2018, doi: 10.37277/stch.v28i1.267.

[7] V. Prisscilya and T. Santoso, “Implementasi Keamanan Jaringan Menggunakan Intrusion,” *J. Inf. Technol.*, pp. 1–8, 2021.

[8] J. D. Santoso, “Keamanan Jaringan Nirkabel Menggunakan Wireless Intrusion Detection System,” *Infos*, vol. 1, no. 3, pp. 44–50, 2019.

[9] E. Stephani, Fitri Nova, and Ervan Asri, “Implementasi dan Analisa Keamanan Jaringan IDS (Intrusion Detection System) Menggunakan Suricata Pada Web Server,” *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 1, no. 2, pp. 67–74, 2020, doi: 10.30630/jitsi.1.2.10.

[10] M. Hanindia and P. Swari, “Intrusion Detection System (Ids) Menggunakan Raspberry Pi 3 Berbasis Snort Studi Kasus : Stmik Stikom Indonesia,” *J. SCAN*, vol. XV, pp. 2–7, 2020.

[11] W. Yunanri and Y. B. Fitriana, “Analisis Network Security Komputer Tingkat Desa Menggunakan Metode Security Policy Development Life Cycle (SPDLC),” vol. 1, no. 2, pp. 11–21, 2021.

[12] A. Aminanto and W. Sulistyo, “Simulasi Sistem Keamanan Jaringan Komputer Berbasis IPS Snort dan Honeypot Artilery,” *Aiti*, vol. 16, no. 2, pp. 135–150, 2020, doi: 10.24246/aiti.v16i2.135-150.

[13] M. Rahouti, K. Xiong, N. Ghani, and F. Shaikh, “SYNGuard: Dynamic threshold-based SYN flood attack detection and mitigation in software-defined networks,” *IET Networks*, vol. 10, no. 2, pp. 76–87, 2021, doi: 10.1049/ntw2.12009.

[14] B. Fachri and F. H. Harahap, “Simulasi Penggunaan Intrusion Detection System (IDS) Sebagai Keamanan Jaringan dan Komputer,” *J. Media Inform. Budidarma*, vol. 4, no. 2, p. 413, 2020, doi: 10.30865/mib.v4i2.2037.

[15] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, “Machine learning-based IoT-botnet attack detection with sequential architecture,” *Sensors (Switzerland)*, vol. 20, no. 16, pp. 1–15, 2020, doi: 10.3390/s20164372.

BIOGRAPHY OF AUTHORS

Doddy Teguh Yuwono obtained Bachelor Degree in Technical Information from STMIK Palangkaraya in 2016, obtained Master Degree in Magister Technical Information from University of Ahmad Dahlan Yogyakarta in 2019. He has been a Lecturer with the Faculty of Engineering anda Informatics, Department of Computerr Science, University of Muhammadiyah Palangkaraya, since 2019. His current research interests include Artificial intelligence, Cyber Security, Digital Forensics and Information System.

Setio Ardy Nuswantoro obtained Bachelor Degree in Technical Information from University of Ahmad Yani Yogyakarta in 2016, obtained Master Degree in Magister Technical Information from University of Islamic Indonesia in 2020. He has been a Lecturer with the Faculty of Teacher Training and Education, Department Information Technology Education Study Program, University of SMuhammadiyah Palangkaraya, since 2020. His current research interests include Big Data, Data Mining, System Analysis and Software Development.