Early Intrusion Detection System (IDS) using Snort and Telegram approach

Aan Erlansari [1], Funny Farady Coastera [2], Afief Husamudin [3]

[1,2,3]Infomatika, Faculty of Engineering, University of Bengkulu.
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA
(Tel: 0736-341022; fax: 0736-341022)

[1] Aan_erlanshari@unib.ac.id
[2]ffaradyc@unib.ac.id
[3]Afiefh17@gmail.com

Abstract— Computer network security is an important factor that must be considered. Guaranteed security can avoid losses caused by attacks on the network security system. The most common prevention against network attacks is to place an administrator, but problems will arise when the administrator is not supervising the network, so to overcome these problems a system called IDS (Intrusion Detection System) can detect suspicious activity on the network through automating the work functions of an administrator. Snort is one of the software that functions to find out the intrusion. Data packets that pass through network traffic will be analyzed. Data packets detected as intrusion will trigger alerts which are then stored in log files. Thus, administrators can find out intrusions that occur on computer networks, and the existence of instant messaging applications can help administrators to get realtime notifications, one of which is using the Telegram application. The results of this study are, Snort able to detect intrusion of attacks on computer networks and the system can send alerts from snort to administrators via telegram bot in real-time.

Keywords— IDS (Intrusion Detection System), Monitoring, Network Security, Real-time, Snort, Telegram

I. INTRODUCTION

Security could be a huge issue for all networks in today's enterprise domain. Hackers and intruders have created several fortunate efforts to bring down company organization and network services. Several strategies are developed to secure the network infrastructure and communication over the web, among them the utilization of firewalls, encryption, and virtual non-public networks. Intrusion detection could be a comparatively new addition to such techniques.

Intrusion Detection System began disclosure over the foremost recent number of years. Utilizing interruption location techniques, you will be able to gather and use knowledge from sorts of disruptions and see whether or not someone is trying to assault your system or specific hosts. The data gathered on these lines are often used to solidify your system security, even as for legitimate functions. Various weak appraisal instruments are too accessible inside the advertising that may be used to survey distinctive types of security gaps show in your organization.

The suggested work is to scale back the malicious activities by characteristic the intruders early in network done through the observance of the node behavior/features with Snort and wire. During this paper, we tend to designed efficient intrusion detection. The system contains 3 phases like feature choice, outlier detection, and classification. The primary contribution of this paper is that the introduction of a brand new feature choice formula referred to as intelligent complete feature choice that helpful|is beneficial|is helpful} for recommending the useful options. The second phase of this paper is the introduction of a brand new detection methodology referred to as an entropy-based weighted outlier detection methodology for removing the useless records. The third contribution of this paper is that the use of the prevailing classification formula referred to

ISSN 2442-7888 (online) DOI 10.24167/Sisforma

as showing an intelligence layered approach for effective classification. The most advantage of this projected work is to pick out the helpful options that are helpful to boost the classification (intrusion detection) accuracy.

The rest of this paper is managed as follows: Section two provides the literature survey and system development model. Section three demonstrates the results and discussion. Section four provides a conclusion and also the future works

II. LITERATUR REVIEW

The early analysis was explicit that Intrusion Detection System (IDS) datasets that were created in university Lincoln Laboratories are wont to assess the performance of Snort [1][2]. The analysis of snort is completed supported the detection rate. It's been found and concludes that snort detection rate is required to boost and additionally the false alert ought to be reduced to boost the general performance of snort. Here, Snort is evaluated on week three, week 4, and week five knowledge. The week three data is attack free and able to train Snort. Week four and week five data include attacks and are utilized in the testing part. Throughout the testing part, Snort generates many alarms:

Table 1 event logged by the snort

Day	1	2	3	4	5
No of even	18557	6392	2092	3490	7780

Nevertheless, Rishab stated [3] that snort can show all matches packet outlined by the administrator. The data stored in MySQL database that we have created a UI to show all the required data regarding the alert generated. The knowledge includes supply IP, Destination IP, Alert generated, Date, and Time of once the packet was received.

Natawat [4] in his research built a snort system using IDS as a result of Intrusion detection systems are efficient network security tools for detective work and observance network traffic knowledge. They generate associate alert once abnormal behavior patterns are matched to existing rules. However, as a result of the IDS could have high false positive and false negative values, we have proposed another system, incorporating data processing of the association rules inside the Snort IDS. The system was completely tested and compared to the first Snort IDS Rules also as icmp.rules and ICMP-info.rules inside the Snort IDS, the system proven to be more useful and more precise.

A. Intrusion Detection System

The intrusion detection system (IDS) is often defined as a tool or associate application that detects malicious activities or policy violations inside the network. IDS has been widely utilized in recent years united of the most network security parts. The target of this study is to search out the best-fit approach that might considerably scale back the number of options. Besides, the approach would result in high classification accuracy with less process time [5]. In order to avoid computer users from malicious effects, IDS (intrusion detection system) is is meant to seem out network activities and manufacture alerts to several persons like administrative and others [6]. IDS is used for two purposes: one methodology is used to identify known attacks, and the other method is used for unknown attacks. The implementation of the second technique isn't simple, and therefore the system ought to come with the correct learning and testing method.

B. Snort

Snort is that the form of the Intrusion Detection System that's used for scanning databases flowing on the network [7], [8]. Snort logically divided into multiple components. Snort logically divided into multiple parts. These parts work along to find specific attacks and generate output into a needed format from the detection system [9]. A snort-based IDS consists of following major components as shown in the table:

Table 2 Components of Snort

Name	**Description**
Packet Decoder	Prepares packet for processing
Preprocessors on Input Plugins	Used to normalized protocol header, detect anomalies, packet-reassembly, and TCP stream re-assembly

ISSN 2442-7888 (online) DOI 10.24167/Sisforma

Detection Engine	Applies rules to packets
Logging and Alerting System	Generates alert and log messages
Output Modules	Process alerts and logs and generate a final output

Figure 1 shows the Snort IDS rules generator procedure. The association rules of the defined parameter's entries are used to generate the Snort IDS rules by the MinSIC module for detecting network probe attacks.

Figure 1 Snort IDS generator rules

The Snort IDS rules exhibitions were assessed by utilizing the precision comparisons method. Appropriate parameters, as seen in Table 2, layout the Grunt IDS rules as appeared in Figure 1.

C. Scanning

Port scanning is a process of finding out which ports are open on a particular host or all hosts on a network[9]. The first step in any intruder activity is usually to find out what services are running on a network. Once an intruder has found this information, attacks for known vulnerabilities for these services are tried. The portscan preprocessor is designed to detect port scanning activities. The preprocessor can be used to log the port scanning activities to a particular location in addition to standard logging. Hackers can use multiple port scanning methods. Refer to man pages or documentation of the Nmap utility [10] to learn more about port scanning methods. The Nmap utility is a widely used tool for port scanning.

The following is the general format of the preprocessor used in the snort.conf file.

preprocessor portscan: <address> <ports> <time period> <file>

There are four arguments to the preprocessor.

• The address range of IP addresses to monitor is a single IP address or a network address. The range is specified using the CIDR block.

• The number of ports accessed within a certain period can be specified. For example, a number 5 means that if five ports are scanned within the time

period specified, an alert is generated.

• The period is the number of seconds that defines the period used for the threshold.

• The path of the file name where the activity should be logged

D. Rapid Prototyping Model

Rapid prototyping is an effective approach to requirements validation and evolution via an executable model of a software system to demonstrate concepts, discover requirements errors and find possible fixing solutions, and discover and discover missing requirements [11][12], [13][14]. Besides the implementation of main system functionalities, a prototype has a User Interface (UI) that allows the client to validate their requirements visually, make it easy to find out faults in the requirements, and then fix them[15][16].

III. METHOD

In this section, we present system development methods. As mentioned in the previous chapter (Literatur Review), this research built on prototype software development model, the phase of a prototype are:

a. analysis

At this stage, the activities carried out are to analyze the purposes contained in the existing problems. Researchers define the whole object of which is to identify all the requirements, among others: (a) analysis of the problem, (b) analysis of needs, (c) an analysis of the user

b. Design

The activities carried out are researchers create design drawings in the wake of the network topology. This topology design created using Microsoft Visio

c. Implementation

The activities carried out are the development of the overall device and troubleshooting plan. At this stage in the form of an actual implementation. The result of this phase is a device that is ready to be tested.

ISSN 2442-7888 (online) DOI 10.24167/Sisforma

d. Evaluation

The security monitoring network attack simulation. The author tested a new system simulation for the user to determine whether the new system simulation gives satisfactory results. If so, then it will proceed to the next stage of the use of the new system. If not, the author will make revisions to return to stages 1, 2, and 3 with a better understanding of the needs of the user.

e. Results

Network monitoring systems have been successful. In this step, it is necessary for the maintenance of the new system so that the system can run properly and the user can feel the comfort in using the network attack monitoring system.

The overall process of the research development method displayed in Figure 2.

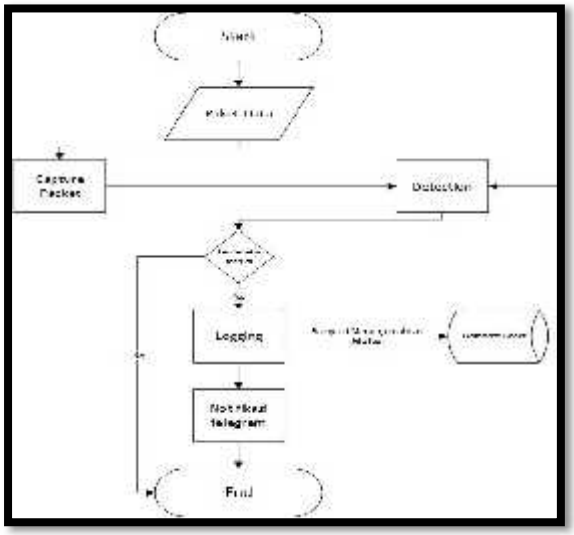

Figure 2. System workflow

IV. RESULTS AND DISCUSSION

In this part, we display the result of IDS using Snort and Telegram.

A. Network architecture

Network architecture was designed based on the general framework of the system before the evaluation. To link all the network, this procedure clarified how the network addressing system and router configuration:

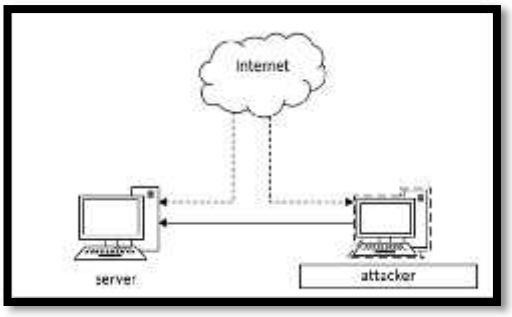

Figure 3. Network architecture

Figure 2 describes a simulation of monitoring evaluation between the attacker and the embedded Snort server. This simulation conducted in a local network and using Virtual Box.

B. Snort configuration

Snort configuration started with installing the supporting application

Table 2 installation supporting application

#apt-get install -y build-essential libpcap-dev libpcre3-dev libdumbnet-dev bison flex zlib1g-dev liblzma-dev openssl libssl-dev

The following command in table 2 is essential to call the library inside Ubuntu repository, proceed with the next phase is Snort configuration.

Table 3 snort configuration

//Membuat *user* dan group snort
#groupadd snort
#*user*add snort
//membuat direktori dan file untuk *rule*
#mkdir /etc/snort
#mkdir /etc/snort/*rules*
#touch /etc/snort/*rules*/nmap.*rules*
#touch /etc/snort/*rules*/ftp.*rules*
#touch /etc/snort/*rules*/ssh.*rules*
#touch /etc/snort/*rules*/ddos.*rules*
#touch /etc/snort/sid-msg.map
#mkdir /etc/snort/preproc_*rules*
#mkdir /usr/local/lib/snort_*dynamicrules*
#mkdir /etc/snort/so_*rules*
//membuat direktori untuk logging snort
#mkdir /var/log/snort
#mkdir /var/log/snort/archived_logs
//memberikan hak akses
#chmod -R 5775 /etc/snort
#chmod -R 5775 /var/log/snort
#chmod -R 5775 /var/log/snort/archived_logs
//mengganti kepemilikan folder
#chown -R snort:snort /etc/snort
#chown -R snort:snort /var/log/snort
#chown -R snort:snort /usr/local/lib/snort_*dynamicrules*

ISSN 2442-7888 (online) DOI 10.24167/Sisforma

```
//Salin konfigurasi file
#snort-2.9.8.3/etc/
#sudo cp *.conf* /etc/snort #sudo cp *.map /etc/snort #sudo cp *.dtd /etc/snort
```

This function begins with creating a user and snort group, creating rules directory, creating logging snort directory, giving access to snort directory, and copying file configuration. The snort configuration phase proceeds with Installation and Configuration Barnyard2, Implementation Database and the last is Snort Rule Implementation.

Snort Rules are the rules that are used to detect intrusion and suspicious activities. Here are a few snort rules from the official website which is implemented in this study:

a. Rule Port Scanning

```
Alert tcp $EXTERNAL_NET any -> $HOME_NET 80
(msg:"nmap FIN Scan"; flags:F; sid:1000008; rev:1;)
```

Figure 4. Port Scanning scheme

b. FTP Access Rule

```
Alert Tcp Any Any -> 192.168.43.75 21 (Msg Possible
Ftp Brute Force"; Flow:To_Server; Flags:S;
Threshold:Type Threshold, Type Both, Count 2,
Seconds 60; Metadata:Service Ftp;
Classtype:Attempted-Admin; Sid:10000011; Rev:001; )
```

Figure 5. FTP access Rule

c. Ssh Access Rule

```
Alert Tcp Any Any -> 192.168.43.75 22 (MsgpossibleSsh Brute
Force"; Flow:To_Server; Flags:S; Threshold:Type Threshold,
Type Both, Count 2, Seconds 60; Metadata:Service Ssh;
Classtype:Attempted Dos; Sid:10000001; Rev:001;)
```

Figure 6. SSH Access Rule

d. DDoS Rule

```
Alert Udp $External_Net Any -> 192.168.43.75 80 (Msg:"Ddos
Udp"; Flow:Stateless; Threshold:Type Both,Track By Dst,
Count 70, Seconds 10; Classtype:Bad-Unknown; Gid:1;
Sid:10000012; Rev:1;)
```

Figure 7. DDoS Rule

e. Rule Ping of Death

```
Alert Icmp Any Any -> $Home_Net Any (Msg:"Icmp Test
Detected Potential Ping Of Death"; Gid:1; Sid:10000034;
Rev:1; Classtype:Icmp-Event;)
```

Figure 8. Rule of Death

C. Testing simulation

Snort testing pattern conducted in a local network with designed testing mode toward few services in the server. The test performed under simulation mode and real test mode to compare effectivity between these two modes.

a. Port scanning result

Figure 9. Port scanning using NMAP

Port scanning test performed using a FIN scan to find out which port was opened, in this case, we discovered FTP:21, SSH:22, and HTTP:80 was opened.

b. FTP access test

FTP access test performed with “brute force” attack and the target is 192.168.143.127/24.

Figure 10. FTP access attack

IDS system automatically acknowledges the intrusion because it identified to access FTP without authorization from eligible IP and defined it as an attack.

c. SSH test

IDS detected any intrusion through Port:22 and IP: 192.168.43.75. The test also performed using “brute force” attack, using

ISSN 2442-7888 (online) DOI 10.24167/Sisforma

username ”root” and password taken from file pass.txt.

Figure 11. SSH brute force attack

The target of intrusion is IP 192.168.43.75 and recognized with SID 10000006 and coming from IP Address 192.168.2.128.

d. DDoS test

Unlike other tests, the DDoS test using “hping3” as a tool installed on the attacker computer. The evaluation target is 192.168.43.75 with *port* 80 and protocol UDP.

Figure 12. DDoS attack

Intrusion detected with SID 10000012 through port:80 and IP Address: 192.168.43.127 targeted 192.168.43.75 and port 80.

No	Skenario Pengujian Sistem	Uji Coba	Hasil yang diharapkan	Hasil pengujian sistem	Kesimpulan
1	Port Scanning	Nmap -sS -sV	Terdeteksi	Terdeteksi	Sukses
2	FTP	Brute Force	Terdeteksi	Terdeteksi	Sukses
3	SSH	Brute Force	Terdeteksi	Terdeteksi	Sukses
4	DDos	DDos UDP -p 80	Terdeteksi	Terdeteksi	Sukses
5	IP Server 192.168.43.75	Ping Of Death	Terdeteksi	Terdeteksi	Sukses

Figure 13. Evaluation system

From all performed tests, we got a result as expected. The system detected the intrusion done by the attacker. Intrusion detection by the rules made from port scanning, FTP Access, SSH access, and DDoS.

D. Telegram Notification

Notification from a server to Telegram was configured before the testing.

Figure 14. Detection information

For every test performed, Telegram receives a notification toward any intrusion that came to a server, e.g: intrusion from DDoS test, Telegram automatically notified as displayed in figure 12.

Figure 15. Telegram notification

This notification directly sent to an administrator to inform an activity from unknown sources.

V. CONCLUSION

Snort supported IDS to detect early intrusion and notified to the server administrator through Telegram application. The IDS system could detect for Port scanning attack, FTP brute force, SSH brute force, and DDoS attack. For further research, we suggest adding more rules scenario on Snort to get better security in the network system.

REFERENCES

[1] N. D. Patel, P. Vrushank, and S. Prof, “An analysis of Network Intrusion Detection System using SNORT,” vol. 1, no. 3, pp. 3–5, 2013.

[2] E. Risyad, M. Data, and E. S. Pramukantoro, “Perbandingan Performa Intrusion Detection System (IDS) Snort Dan Suricata Dalam Mendeteksi Serangan TCP SYN Flood,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 9, pp. 2615–2624, 2018.

[3] S. R. Pampattiwar and P. A. Z. Chhangani, “Hybrid Intrusion Detection System Using Snort,” *Int. Res. J. Eng. Technol.*, vol. 4, no. 4, pp. 1–6, 2017, [Online]. Available: https://www.irjet.net/archives/V4/i4/IRJET-V4I4439.pdf.

[4] N. Khamphakdee, N. Benjamas, and S. Saiyod, “Improving intrusion detection system based on snort rules

ISSN 2442-7888 (online) DOI 10.24167/Sisforma

for network probe attacks detection with association rules technique of data mining," *J. ICT Res. Appl.*, vol. 8, no. 3, pp. 234–250, 2015, doi: 10.5614/itbj.ict.res.appl.2015.8.3.4.

[5] M. H. Kamarudin, C. Maple, and T. Watson, "Hybrid feature selection technique for intrusion detection system," *Int. J. High Perform. Comput. Netw.*, vol. 13, no. 2, p. 232, 2019, doi: 10.1504/ijhpcn.2019.097503.

[6] A. Kumar and P. Shanmugavadivu, *Space of RGB-H-CMYK*, vol. 1, no. February. Springer Singapore, 2019.

[7] D. Day and B. Burns, "A Performance Analysis of Snort and Suricata Network Intrusion Detection and Prevention Engines," *ICDS 2011, Fifth Int. Conf. Digit. Soc.*, no. c, pp. 187–192, 2011, [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=icds_2011_7_40_90007.

[8] "Snort_Open_Source_Network_Security.Pdf." .

[9] library of congress cataloging-in-publication Data, *Intrusion Detection Systems with Snort Advanced IDS Techniques Using Snort , Apache , MySQL , PHP , and ACID B RUCE P ERENS ' O PEN S OURCE S ERIES*. 2003.

[10] N. Org, "Port Scanning Techniques," 2012. http://nmap.org/book/man-port-scanning-techniques.html (accessed Jan. 21, 2020).

[11] J. Detand, R. Bastiaens, B. Grimonprez, and O. Rysman, "The role of prototyping in product development," *Proc. 4th Int. PMI Conf.*, no. January, p. 8, 2010, [Online]. Available: https://www.researchgate.net/publication/240045178_The_role_of_prototyping_in_product_development.

[12] M. Carr and J. Verner, "Prototyping and Software Development Approaches," *Prototyp. Softw. Dev. Approaches*, no. 3, pp. 1–16, 2004, [Online]. Available: https://www.google.com.my/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CCkQFjAA&url=http://www.cb.cityu.edu.hk/is/getFileWorkingPaper.cfm?id=55&ei=73eDU6aCBo7PlAXNooGQBg&usg=AFQjCNFCEFbDyv9tNk_YuH0VpPfavJPs2A&sig2=wimyHPVpHpp.

[13] R. Ganpatrao Sabale, "Comparative Study of Prototype Model For Software Engineering With System Development Life Cycle," *IOSR J. Eng.*, vol. 02, no. 07, pp. 21–24, 2012, doi: 10.9790/3021-02722124.

[14] B. Murdoch and A. Lee, "Evaluating the Simulation of Rapid Application Development."

[15] Y. Yang, X. Li, W. Ke, and Z. Liu, "Automated Prototype Generation From Formal Requirements Model," *IEEE Trans. Reliab.*, pp. 1–25, 2019, doi: 10.1109/tr.2019.2934348.

[16] M. S. P. Aditama, S. A. Wicaksono, and F. Pradana, "Pembangunan Sistem Informasi Kenaikan Jabatan Fungsional Dosen Universitas Brawijaya," vol. 2, no. 10, pp. 3538–3544, 2018, [Online]. Available: http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/2661/1002.