JOSCEX Journal of Soft Computing Exploration

Implementation of signature-based intrusion detection system using SNORT to prevent threats in network servers

Pahala Bima Pramudya[1], Alamsyah[2]

[1,2]Department of Computer Science, Universitas Negeri Semarang, Indonesia

Article Info

Article history:

Received Sep 10, 2022
Revised Sep 24, 2022
Accepted Sep 29, 2022

Keywords:

Network security
Intrusion detection system
SNORT
Signature-based

ABSTRACT

Security is an important factor in today's digital era. In a network, implementing a security system is the focus of a network developer. One of the most basic network securities is in the form of access. To manage the security of a system must be known in advance who is involved in the system and what activities are carried out. Just like a security alarm, which monitors work conditions, this is the function of the Intrusion Detection System (IDS). IDS has several effective methods for detecting threats, one of which is the Signature-based method. IDS can be implemented through the open-source SNORT application, and the method works with rules which are commands to IDS to recognize various attacks. IDS rules will be included in the signature matching process, which means matching between rules and incoming attacks and views of both protocols, then the IDS will generate alerts that contain notifications. This study conducted a reading of the MIT-DARPA 1999 dataset on 1,252,412 packages and tested alerting with Network Scanning and DoS attacks. Analyze Package Data runs at a speed of 83,494 packets /second and gets a true positive percentage reaching 100% and an accuracy of 98.10%.

Corresponding Author:

Pahala Bima Pramudya,
Department of Computer Science,
Universitas Negeri Semarang,
Sekaran, Gunungpati, Semarang, Indonesia.
Email: pahalabima@students.unnes.ac.id

1. INTRODUCTION

Network security is becoming increasingly important in this technology age. The more various threats that will come into the network, the more complex also a network will be built. Security is based on attack, which means security on a network that is built based on the threat of future attacks, and the attacks that are carried out. The best application of network security is prevention before an attack occurs, in this case, it certainly requires a lot of anticipation, and one of the most important is installing a security alarm [1], [2]. This security alarm is called the Intrusion Detection System (IDS). IDS will be the first very important security in a network server, whose function is to monitor. In many cases, IDS also responds to traffic that is not normal /anomalous by blocking the user or Internet Protocol (IP) addres that attempts to access the network [3], [4].

The proposed procedure for IDS is with the application of SNORT. SNORT can analyze network traffic data in real time by utilizing various existing attack rules. SNORT components are packet decoder, preprocessor, detection engine, logging and alerting system, and output module. The SNORT using rule structure consists of two logical parts. The first part is the header rules, while the second part is the rules option [5]. The rules have to be cleverly crafted so they can be applied to multiple attacks. Rules can detect one type or several types of intrusion activity [6]–[8].

The detection method that will be used in implementing this IDS is Signature-based, or protocol based. In contrast to the Anomaly-based method, it studies unusual events in the network and considers them to be attacked, whereas this method has a fairly high false positive rate, inability to read unknown attacks, and insufficient investigative capacity [9]. The Anomaly method, which has been added to the deep learning attack recognition system, still has a weakness in the form of a low packet reading speed [10]. This signature method is quite helpful in implementing IDS which requires high accuracy using which the work of the network traffic is checked against pre-configured and predefined attack patterns known as Signatures.

Van et al. [10] conducted deep learning research on Anomaly-based IDS. The title of their research is "An anomaly-based Network Intrusion Detection System using Deep learning". Research with deep learning examines that the Anomaly-based IDS method has many shortcomings such as a false positive rate, so it is necessary to use deep learning techniques such as Restricted Boltzmann Machines (RBM) and Autoencoder with the KDDCup99 dataset. The results of the study with 4,900,000 single connection vectors KDDCup99 were executed in 161,695s (SRBM) and 240,133s (SAE). This means that the implementation of Anomaly-based IDS can be more effective if you add deep learning techniques, which also still have a bit of a time problem. So that the application of basic network security, using the Signature method is more effective. Li et al. [11] in their research entitled "Designing collaborative blockchained signature-based intrusion detection in IoT environments" examined IDS design in an IoT environment by choosing to use the Signature method because the Anomaly method has a false positive rate. This means that in designing using the IoT, the Anomaly method has not been able to be used because of these shortcomings.

Based on the use of IDS in the network and the application of good methods: “The anomaly method still lacks a high false-positive rate, few researchers focus on the use of signature-based rules. There are some limited studies testing IDS with the Anomaly and deep learning methods. because such approaches are more accurate in detecting known attacks, compared to anomaly-based approaches, and typically no heavy computations are required [12]. Therefore, this research demands taking advantage of rules as signatures to prevent threats in the server network. This research aims to obtain a system with good precautionary measures”.

2. METHOD

The method is applied to solve problems including procedures, measuring, and analytical methods. The method used in this research is signature-based IDS. Signature-based IDS uses rules as a signature which will later be matched with incoming attack protocols [13]. In this research, the system development uses its external library called DPX as in dynamic preprocessor library which is simpler and focuses on signature based only.

2.1. Research Methods

In this study, a combination of SNORT and dynamic preprocessor as a signature machine is applied to analyze the MIT-DARPA 1999 dataset. Dynamic preprocessor as the library is used to load the SNORT data and run as IDS to read the MIT-DARPA 1999 dataset. 566 signature-based method is used to classify the dangerous package. The classification results are evaluated based on a confusion matrix by calculating the accuracy of the rules [14]. The flowchart of the method used in this study is shown in Figure 1.

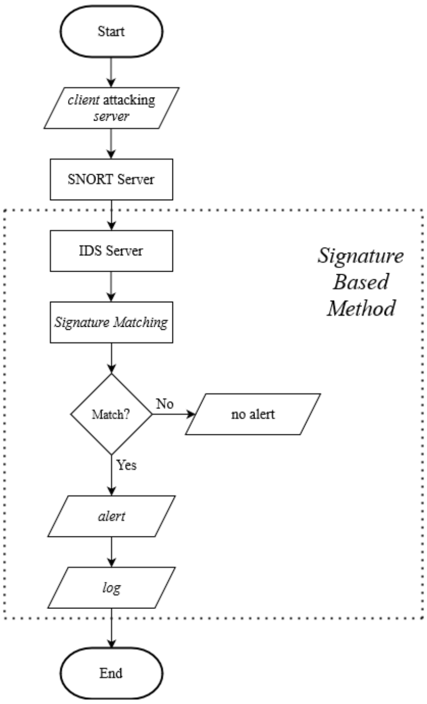

Figure 1. Flowchart of method

2.2. Preparing Dataset

The dataset is used to test the system with attack patterns collected in the tcpdump data file. The dataset used is taken from the evaluation results of the Intrusion Detection Defense Advanced Research Projects Agency from the MIT official website and is named 1999 DARPA Datasets in the last week of the research, week 5. There are two parts to the 1999 DARPA Intrusion Detection Evaluation: offline evaluation and real-time evaluation. The intrusion detection system is tested in an offline evaluation using network traffic and audit logs collected on a simulated network [15]. The system processes data in batch mode and identifies attack sessions amidst normal activity. The sampling method used in this research is Network Scanning and DoS (Denial of Services), the type of attack is a port scan that is open on the server and has the opportunity to break into the system, as well as attacks on servers on the network by using up resources by the server [16].

2.3. SNORT Stage

SNORT is available under the General Public License (GNU). SNORT is an open-source lightweight IDS developed by Martin Rosh in 1998, SNORT is a NIDS that uses a series of signatures to identify different attacks [6]. SNORT contains a set of rules through which rules control traffic, examine packets and detect DDoS attacks [17] When SNORT detects an incursion, it generates an alert to notify the network admin. Commonly applied in units of packets' header, statistical information (packet size), and payload information. Thus, it has a unique feature of high detection rate and accuracy. However, it cannot detect novel attacks and, at the same time, it requires expert knowledge to create and update rules frequently, which is both costly and faulty [18].

The SNORT components consist of (a) packet sniffer: SNORT uses Data AcQuisition (DAQ) library to read packets, (b) packet decoder: Used to decode Layer 2 (data link layer) and Layer 3 (network layer) protocols, mainly focusing on TCP / IP protocol settings, (c) preprocessor: a preprocessor is a component or plug-in that presents a package to a detection engine in a contextually relevant way, (d) detection engine: Its role is to detect whether there is intrusion activity in a packet. It consists of two components, a Rules builder, and an Inspection component, (e) logging and alerting systems for better performance enhancement, logging can also be turned off completely while leaving alerts enabled, (f) output modules: these handles write tasks and display events. It supports different output formats. It can send output to a file or syslog.

Snort is logically divided into some parts. SNORT is an open source IDS for monitoring and preventing attacks on computer security and produces the output to be a required format of the detection system [19]. When working as an IDS, SNORT can detect attacks or data packets that enter computer networks and warn network administrators to take precautions. SNORT implementation is carried out in the Ubuntu 14.04 operating system, by installing the SNORT library. In SNORT, the rule will also be applied to form a database of threat recognition from outside parties. In SNORT, the management and development of database rules are carried out so that IDS plays a maximum role in recognizing attacks and taking prevention.

2.4. Signature Based

IDS systems can generally be classified into signature-based, and anomaly based, according to their detection mechanism. Signature-based IDS attacks look for certain patterns that represent known threats, while anomaly-based IDS detects deviations from predefined normal behavior profiles, which can be caused by previously known or unknown attacks [20], [21]. Anomaly-based IDSs often generate high levels of false alerts due to the difficulty of creating an accurate profile. Signature-based or misuse IDS uses a variety of techniques to find similarities between system behavior and previously known attacks stored in the database. Signature-based intrusion detection is more efficient in finding attacks on networks with fewer features and less modeling time [22].

There are several main components of the SNORT signature-based builder, namely package decoder, preprocessors, detection engine, logging alerting system, and output modules. The first step in detecting network attacks is to capture network traffic using libpcap as the packet capture library. This component will separate the data packet through the ethernet card which will then be used by SNORT. And then enter the Packet Decoder, this component retrieves data in layer 2 sent by the previous component (Packet Capture Library). Then Preprocessor, an analysis of the package is carried out before it is processed by the next component. The analysis process carried out can be marked, grouped, or terminated because the package received is incomplete. The detection engine can be said to be the heart of SNORT. Packages received here after going through several stages of the process will be compared with existing or predetermined rules. The rule here contains a signature which is a category of attack. Output, after the Detection Engine process, is carried out, the results are in the form of reports and alerts [23].

2.5. Designing Systems

The first step in implementing signature-based IDS is to build a virtual machine PC client-server network with a real PC, with a virtual machine as a server using the ubuntu 14.04 LTM linux operating system because the operating system and its version are the most compatible operating systems with SNORT as the IDS applications. The PC server uses 2GB RAM and 10GB hard drive, only as the IDS server. Then for the client operating system using windows 10 with 4GB of RAM.

Next, the install stage of the IDS package rebuilds a plain server that has been set by IP to become an IDS server that is ready for the system testing phase with attacks. The install IDS package stage consists of several steps, namely install MySQL, data acquisition, and barnyard2.

Preparing SNORT as an IDS application by configuring it. Several parts need to be configured in SNORT to act as NIDS or network intrusion detection system, such as the database, SNORT config file, and SNORT rules. In writing rules, there is a protocol in the components of each rule itself. Like action, as an action taken when the attack in question appears, it can be in the form of an alert or notifying or logging or saving a warning. Protocols, as a type of protocol from incoming attacks, can be filled like TCP, UDP, or ICMP. Address port, as the port where the attack might occur. Direction is the direction and purpose of the attack to be detected, such as an incoming attack on the server or an outgoing attack. Address port, as a port again as before. Message will be displayed when the intended attack appears, and as a classification of the type of attack, and the level of danger.

2.6. Testing Systems

The stage of conducting a dataset detection test and network attack on a network server that has IDS installed as an attack notification alarm. namely by trying to send a disturbance packet to the SNORT IDS server, to know the advantages and disadvantages of the system that has been made and whether it works properly and effectively.

The test is carried out in two stages, the firs test system with the MIT-DARPA 1999 dataset and compare the packet read results with the wireshark IP traffic monitor application. Second, testing for server disturbances with the port scanner technique using Nmap to find out which ports are open on the server and testing for disturbances on the server using the DoS technique.

3. RESULTS AND DISCUSSIONS

The results of collecting the test dataset using the MIT-DARPA 1999 dataset and the test data will then be executed through a windows 10 host with the network scanning and denial of services network attack methods. Table 3 contains the dataset MIT-DARPA 1999.

Table 3. Dataset MIT-DARPA 1999

Total Pckg	*Pckg/sec*	*IP4*	*ICMP*	*UDP*	*TCP*
1252412	83494	1179384	10865	14139	1153947

To find out that this signature method is capable of being the main alarm in a complex system, a comparison is made with previous research, with information as follows True Positive (TP) is the attack package detected by the IDS attack. Then, False Positive (FP) is normal packet detected by IDS attack. True Negative (TN) is normal packets detected by IDS normal packets. False Negative (FN) is attack packet detected normal packet by IDS. Calculations are carried out to determine the True Positive Percentage (TPP), and the accuracy of the system. According to Khampkhakdee [5] the formula of TPP and accuracy can be shown in equations (1) and (2).

$$\text{TPP} = \frac{\text{TP x 100}}{(\text{TP+FN})} \quad (1)$$

$$\text{Accuracy} = \frac{(\text{TP+TN})\text{x 100}}{(\text{TP+TN+FP+FN})} \quad (2)$$

In this study, testing was carried out with the additional application wireshark as an IP traffic monitor application to find out if there are packets that are not detected by IDS and get the following figures. True Positive (TP) = 1254744, False Positive (FP) = 48492, True Negative (TN) = 1252412, False Negative (FN) = 0.

IDS signature-based, which was tested, reads network traffic packets that were included in the model which received a presentation of 1.457% of the 1,252,412 incoming packets, 100% alerting of 18,285 incoming attacks or malicious packets, and 98.10% accurate.

4. CONCLUSION

In this study, testing was carried out by analyzing the attack package and alerting the MIT-DARPA 1999 dataset with attacks in the form of network scanning and denial of services on the ubuntu 14.04 linux operating system network server with the implementation of signature-based intrusion detection system security using SNORT. The results obtained from this study are in the form of a TPP level or the correctness of reading the original attack and the detection accuracy level of IDS SNORT using the signature method which can properly detect with a TPP level reaching 100% and an accuracy of 98.10%.

REFERENCES

[1] A. Li, X. Li, Y. Pan, and W. Zhang, "Strategies for network security," *Sci. China Inf. Sci.*, vol. 58, no. 1, pp. 1–14, 2015.

[2] I. Oladeji, P. Makolo, R. Zamora, and T. T. Lie, "Density-based clustering and probabilistic classification for integrated transmission-distribution network security state prediction," *Electr. Power Syst. Res.*, vol. 211, no. June, p. 108164, 2022.

[3] A. Boulaiche and K. Adi, "An auto-learning approach for network intrusion detection," *Telecommun. Syst.*, vol. 68, no. 2, pp. 277–294, 2018.

[4] F. Righetti, C. Vallati, M. Tiloca, and G. Anastasi, "Vulnerabilities of the 6P protocol for the Industrial Internet of Things: Impact analysis and mitigation," *Comput. Commun.*, vol. 194, no. July, pp. 411–432, 2022.

[5] N. Khamphakdee, N. Benjamas, and S. Saiyod, “Improving intrusion detection system based on snort rules for network probe attacks detection with association rules technique of data mining,” *J. ICT Res. Appl.*, vol. 8, no. 3, 2015.

[6] A. Chahal and R. Nagpal, “Performance of snort on Darpa dataset and different false alert reduction techniques,” in *Proc. 3rd Int. Conf. Elect., Electron., Eng. Trends, Commun., Optim. Sci.(EEECOS)*, 2016, pp. 1–8.

[7] S. A. R. Shah and B. Issac, “Performance comparison of intrusion detection systems and application of machine learning to Snort system,” *Futur. Gener. Comput. Syst.*, vol. 80, pp. 157–170, 2018.

[8] J. S. Abbasi, F. Bashir, K. N. Qureshi, M. Najam ul Islam, and G. Jeon, “Deep learning-based feature extraction and optimizing pattern matching for intrusion detection using finite state machine,” *Comput. Electr. Eng.*, vol. 92, no. July 2020, p. 107094, 2021.

[9] R. Samrin and D. Vasumathi, “Review on anomaly based network intrusion detection system,” in *2017 int. conf. electr. electron. commun. comput. optim. tech. (ICEECCOT)*, 2017, pp. 141–147.

[10] N. T. Van and T. N. Thinh, “An anomaly-based network intrusion detection system using deep learning,” in *2017 int. conf. syst. sci. eng. (ICSSE)*, 2017, pp. 210–214.

[11] W. Li, S. Tug, W. Meng, and Y. Wang, “Designing collaborative blockchained signature-based intrusion detection in IoT environments,” *Futur. Gener. Comput. Syst.*, vol. 96, pp. 481–489, 2019.

[12] P. Ioulianou, V. Vasilakis, I. Moscholios, and M. Logothetis, “A signature-based intrusion detection system for the internet of things,” *Inf. Commun. Technol. Form*, 2018.

[13] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, “Rule generation for signature based detection systems of cyber attacks in iot environments,” *Bull. Networking, Comput. Syst. Softw.*, vol. 8, no. 2, pp. 93–97, 2019.

[14] B. Prasetiyo, Alamsyah, M. A. Muslim, and N. Baroroh, “Evaluation performance recall and F2 score of credit card fraud detection unbalanced dataset using SMOTE oversampling technique,” *J. Phys. Conf. Ser.*, vol. 1918, no. 4, p. 42002, 2021.

[15] Alamsyah, B. Prasetiyo, and M. N. Ardian, “Enhancement security AES algorithm using a modification of transformation ShiftRows and dynamic S-box,” *J. Phys. Conf. Ser.*, vol. 1567, no. 3, p. 32025, 2020.

[16] M. A. Muslim and B. Prasetiyo, “Implementation twofish algorithm for data security in a communication network using library chilkat encryption activex,” *J. Theor. Appl. Inf. Technol.*, vol. 84, no. 3, p. 370, 2016.

[17] Z. Hassan, R. Odarchenko, S. Gnatyuk, A. Zaman, and M. Shah, “Detection of distributed denial of service attacks using snort rules in cloud computing & remote control systems,” in *2018 IEEE 5th Int. Conf. Methods Syst. Navig. Motion Control (MSNMC*, 2018, pp. 283–288.

[18] E. Jaw and X. Wang, “A novel hybrid-based approach of snort automatic rule generator and security event correlation (SARG-SEC),” *PeerJ Comput. Sci.*, vol. 8, p. e900, 2022.

[19] A. Erlansari, F. F. Coastera, and A. Husamudin, “Early Intrusion Detection System (IDS) using Snort and Telegram approach,” *SISFORMA*, vol. 7, no. 1, pp. 21–27, 2020.

[20] Y. Wang, W. Meng, W. Li, J. Li, W.-X. Liu, and Y. Xiang, “A fog-based privacy-preserving approach for distributed signature-based intrusion detection,” *J. Parallel Distrib. Comput.*, vol. 122, pp. 26–35, 2018.

[21] P. S. K. Oberko, V. H. K. S. Obeng, H. Xiong, and S. Kumari, “A survey on attribute-based signatures,” *J. Syst. Archit.*, vol. 124, no. July 2021, p. 102396, 2022.

[22] K. Rai, M. S. Devi, and A. Guleria, “Decision tree based algorithm for intrusion detection,” *Int. J. Adv. Netw. Appl.*, vol. 7, no. 4, p. 2828, 2016.

[23] M. Merouane, “An approach for detecting and preventing DDoS attacks in campus,” *Autom. Control Comput. Sci.*, vol. 51, no. 1, pp. 13–23, 2017.

DOI: 10.52465/joscex.v3i2.80