ComniTech: Journal of Computational Intelligence and Informatics
https://dx.doi.org/000000000
P-ISSN: xxxx-xxxx E-ISSN: xxxx-xxxx

Monitoring on Solid Foundation Servers Using Suricata Through Telegram Bot Notifications

***Muhammad Faishal Gusrianda*[1], *Fajrizal Fajrizal*[2], *Guntoro Guntoro**[3]

[1,2,3]Informatic Engineering, Faculty of Computer Science, Universitas Lancang Kuning, Pekanbaru, Indonesia.
*Corresponding Author.

Received 13/06/2024, Revised 20/06/2024, Accepted 25/06/2024, Published 30/06/2024

Abstract

This research was conducted to study the Telegram Bot-based Suricata technique, which is implemented in the application of data center server security where the data center is related to the center of the Drug and Narcotics Agency of the Republic of Indonesia. We collected data from observations and interviews conducted in the administration section of server data security. Furthermore, the data is analyzed to study the pattern of each technique used. Based on this pattern, a better technique is chosen for the security process to be carried out. Based on the experiments conducted, it is clear that each of these techniques has advantages and disadvantages. However, this Telegram Bot-Based Suricata technique is more recommended because it is simpler in implementation, the desired results can be achieved, and the system can be monitored anywhere.

Keywords: Suricata, Telegram Messenger, Linux Ubuntu Server, Putty, Curl

Introduction

Developments in the field of information technology (IT) are very rapid, facilitating access to information through internet technology and network infrastructure (Ismagilova et al., 2020). Although this development brings many benefits, it also has negative impacts, such as threats to information security (Gunduz & Das, 2020). Therefore, network security aspects are crucial to consider (Chowdhury & Gkioulos, 2021).

The Solid Foundation Pekanbaru Narcotics Rehabilitation Center has several servers equipped with a firewall security system. This system is used to support various institutional activities, such as websites and internal applications (David et al., 2022). These websites and applications are accessed by various users with various purposes and objectives, which in turn can affect server security (Vinoth et al., 2022). The more an application or website is used, the greater the threat to the system (Jaya et al., 2022).

Cybercrime is a negative consequence of information technology development (Khrypko et al., 2021). This cybercrime develops along with the rapid development of IT, often driven by economic motives and the misuse of information (Sviatun et al., 2021). One form of cybercrime that can occur is an attack on an agency or company's

ComniTech: Journal of Computational Intelligence and Informatics
https://dx.doi.org/000000000
P-ISSN: xxxx-xxxx E-ISSN: xxxx-xxxx

server system, which can damage information confidentiality, authenticity, and availability. This necessitates that network administrators regularly monitor the state of the networks they manage (Abbasi et al., 2021).

Common acts of cybercrime on servers include information theft or destruction, as well as data deletion (Andronie et al., 2021). If this cannot be identified early on, the resulting damage can be detrimental to the service provider, in this case, the Solid Foundation Pekanbaru Narcotics Rehab Center. Therefore, the purpose of this research is to make it easier for network administrators to monitor and check logs on the Solid Foundation server. With real-time monitoring, network administrators can take preventive action when an attack occurs on the server.

Materials and Methods

To prevent and minimize the possibility of attacks on the Solid Foundation Pekanbaru Narcotics Rehabilitation Center *server* , the authors design and try to implement *suricata (IDS)* into the network architecture of the Solid Foundation Pekanbaru Narcotics Rehabilitation Center *server* as shown below:

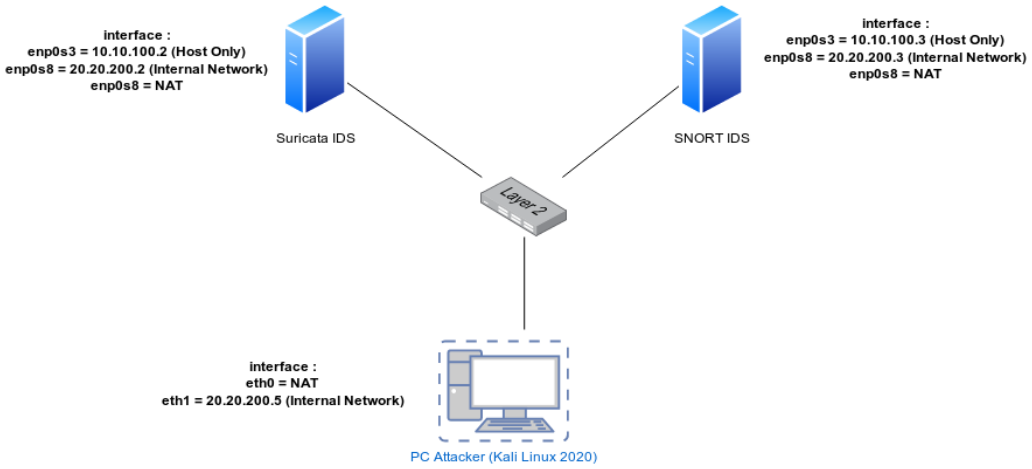


Figure 1. Proposed Server Topology Using Suricata (IDS)

By adding a *logsuricata server* to an existing *server*, the *server administrator* can monitor *logs* that are identified as attack *traffic* to the *server*.

However, this is still limited because the administrator must log in to the *server* and check the *logs* manually, so the author tries to integrate the *logsuricata* using *Telegram bots* so that the administrator can see *log* activities automatically and in *real time*.

ComniTech: Journal of Computational Intelligence and Informatics
https://dx.doi.org/000000000
P-ISSN: xxxx-xxxx E-ISSN: xxxx-xxxx

Telegram and Suricata Integration Flowchart

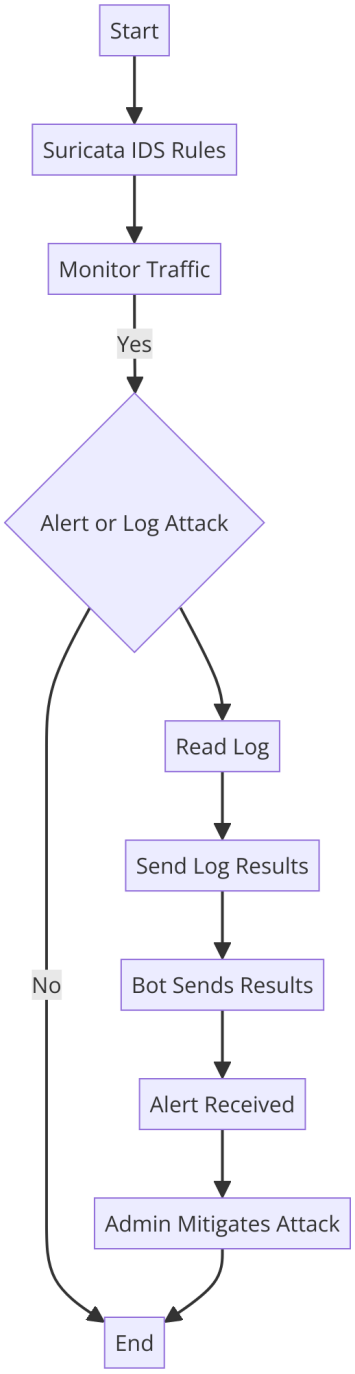


Figure 2. Flowchat How Suricata (IDS) and Telegram Bots Work


ComniTech: Journal of Computational Intelligence and Informatics
https://dx.doi.org/000000000
P-ISSN: xxxx-xxxx E-ISSN: xxxx-xxxx

The explanation in the *flowchat* image is that when there is *traffic or packets* to the *server*, the *suricata* will first match the available rules. If the *traffic or packets* come in accordance with the *rules* belonging to the suricata, the suricata will display a *log alert* in the filefast.logidirectory */var/log/suricata/fast.log,* and if they don't match, the suricata will ignore them.

The author designs a bah *script or shell script* to make it easier to view *suricata alert logs.* In the *script,* the *suricata log* will be processed and read every line, and then the *log* will be sent via a telegram bot intermediary to the network *administrator*. Telegram*bot* has an API with an ID token that can be used to create your own bot and supports many programming languages.

When the network *administrator* gets a notification from the *Suricata Telegram* bot, the admin can quickly respond and take preventive action so that the impact of *cybercrime* can be minimized. The benefit of the notification integration is that the network administrator can monitor malicious traffic or packages heading to the server from any location.

Hardware Requirements

The following are computer hardware specifications used as Prototype Testing :

Computer/Laptop (Host) Specifications

a. Processor :Dual core 1.8 GHz
b. Memory (RAM): 4 GB
c. Hard disk : 320 GB
d. Ethernet Controller : Intel NM10 and D-link RTL8139
e. Monitor: 14 inc

Suricata Node Specifications

a. Processor (Core): Dual core 1.8 GHz
b. Memory (RAM): 1.5 GB
c. Hard disk: 20 GB
d. Ethernet Controller: NAT & Custom (Vmnet1)

Linux Kali Node Specification (Attacker)

a. Processor (Core) 1
b. Memory (RAM): 2 GB
c. Hard disk: 20 GB
d. Ethernet Controller: NAT & Custom (Vmnet1)

Software Requirements

The following are the software requirements and application packages used to support Suricata-based Server Monitoring at the Solid Foundation Pekanbaru Narcotics Rehabilitation Center:

a. Linux Ubuntu Server
b. Suricata
c. curl
d. bash

ComniTech: Journal of Computational Intelligence and Informatics
https://dx.doi.org/000000000
P-ISSN: xxxx-xxxx E-ISSN: xxxx-xxxx

While the software used on the Host Computer is as follows:

a. Windows10 Pro x64
b. Vmware Workstation 14
c. Putty
d. Telegram Messaging

IP Address Design

In order for a device to be connected to the network, it must have a special address commonly called the Internet Protocol (IP Address), the ip address will be configured on the suricata server and also the attacker 's computer . There is also an ip address for the suricata server as follows:

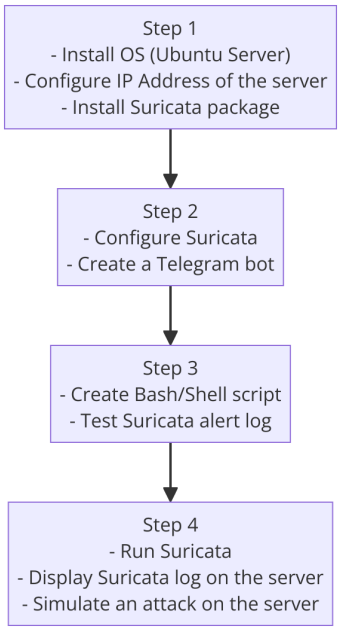
IP Address : Ethet1 Suricata Server: DHCP
Ether2 : 192.168.1.1 /24
Gateway : -

In the ip address server design above, the author uses 2 Network Adapters on ether1 used to connect the server to the internet and this is needed to send logs via telegram bot and Ether2 is used to connect to the computer (attacker/tester). connecting with the computer (attacker/tester). While the ip address used on the attacker 's computer is:

IP Addreess : 192.168.1.2/24 SubnetMask : 255.255.255.0 Gateway

Flow of the Suricata-Based Monitoring Design Process

To simplify the design process, a structured flow of the stages to be carried out is needed. So that the design and implementation process is more directed, as in the following figure:



ComniTech: Journal of Computational Intelligence and Informatics
https://dx.doi.org/000000000
P-ISSN: xxxx-xxxx E-ISSN: xxxx-xxxx

Figure 3. Suricata Design Process

Results and Discussion

At the implementation and testing stage, the author will use several penetration testing tools, including the nmap, nikto, and hydra tools. nmap functions to scan the port on the destination server, while nikto functions to scan the vulnerability on the webserver, and hydra is used to bruteforce attack passwords. In this research, the author uses brute force to attack SSH passwords.

NMAP Testing

The author scans the server port using nmap as shown below:

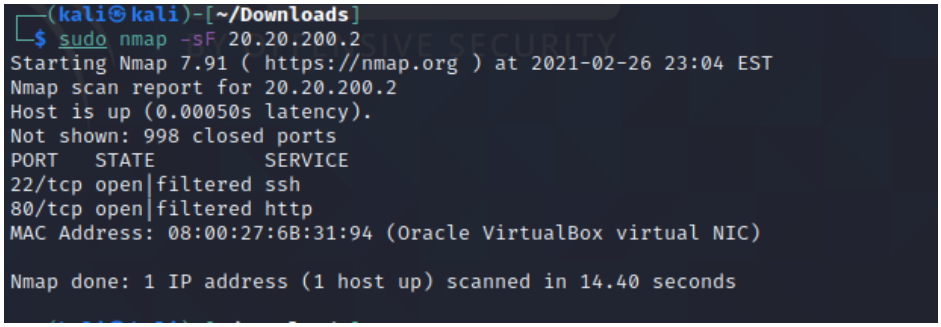
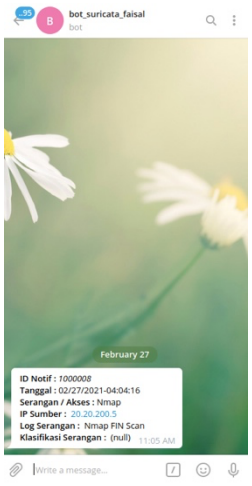


Figure 4. NMAP Testing

When the nmap scanning process runs, the suricata log will detect it, and then it will be saved in the fast.log file. After successfully saving, the notification will be sent to Telegram; in this case, the author sends the log to the group that the author has prepared, as shown in Figure 5.



ComniTech: Journal of Computational Intelligence and Informatics
https://dx.doi.org/000000000
P-ISSN: xxxx-xxxx E-ISSN: xxxx-xxxx

Figure 5. Log Suricata Nmap ke Telegram

Nikto Testing

The author will scan the webserver using the nikto tool as shown below:

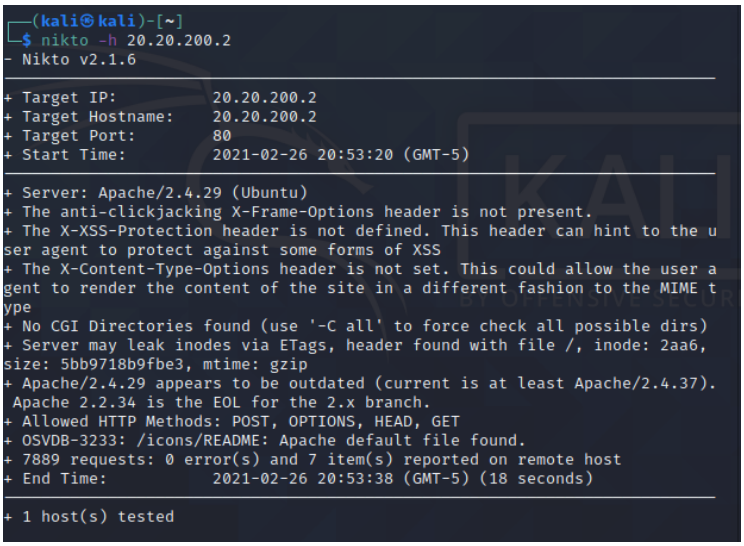
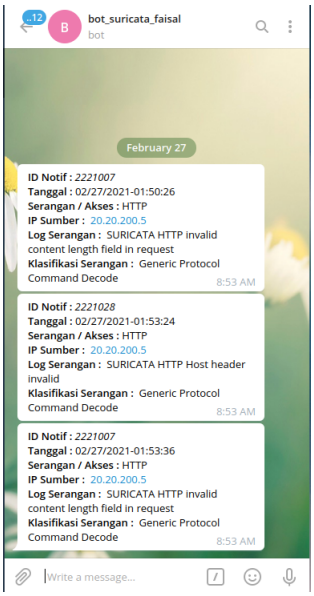


Figure 6. Nikto Testing

As in the nmap test, attacks that are successfully detected through the server port scanning process using nikto will be sent to a telegram notification as shown. 7



ComniTech: Journal of Computational Intelligence and Informatics
https://dx.doi.org/000000000
P-ISSN: xxxx-xxxx E-ISSN: xxxx-xxxx

Figure 7. Log Suricata Nikto ke Telegram

Hydra Testing

Next is the author's test to bruteforce attack for ssh service using the hydra tool, in this case the test will involve a dictionary that can be downloaded at the following link : https://github.com/jeanphorn/wordlist/blob/master/ssh_ passwd.txt. Next, open the linux terminal and type the command as shown below:

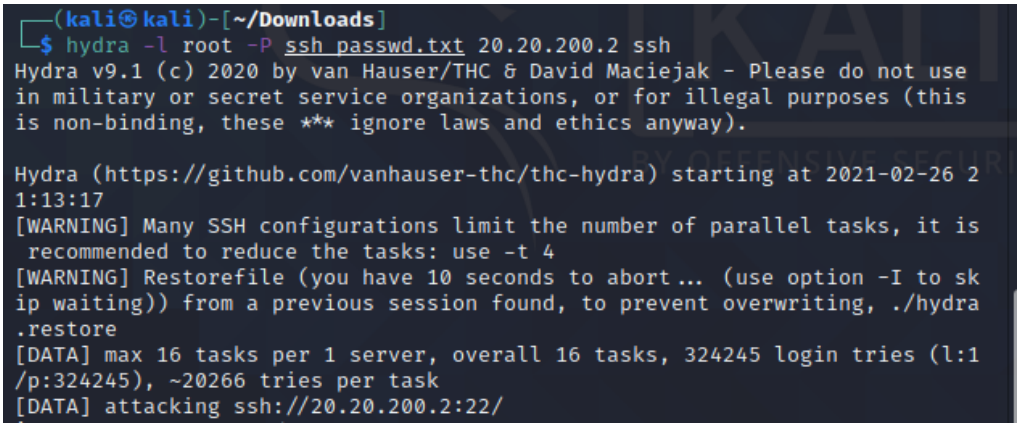


Figure 8. Bruteforce Hydra Testing

The explanation above is the root user on the target server then ssh_password.txt is a dictionary of ssh passwords that have been downloaded at the link above, then at the end of the command there is the target server ip and service name that will be bruteforced. Similar to the previous test using nmap and nikto, the bash script brutoforce hydra log is sent via notification to telegram as shown. 9

ComniTech: Journal of Computational Intelligence and Informatics
https://dx.doi.org/000000000
P-ISSN: xxxx-xxxx E-ISSN: xxxx-xxxx

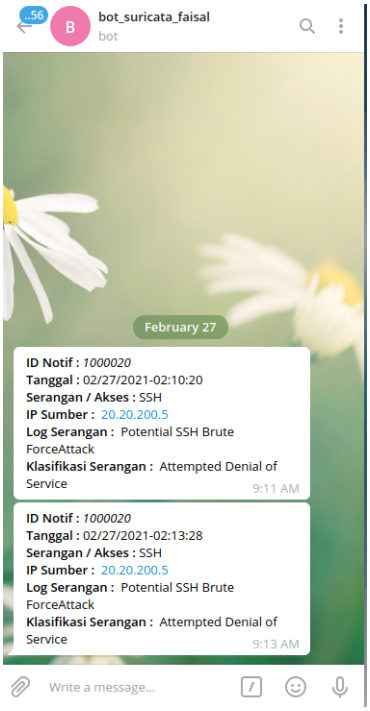


Figure 9. Log Suricata Hydra to Telegram

From some of the tests above, it can be seen that the Suricata configuration successfully detects logs from penetrating attack tools such as port scanning, etc. Then these logs are not only detected but also successfully sent automatically to the telegram group when an attack occurs on the server. This is certainly very helpful and makes it easier for network administrators to monitor activity and logs on the server. From the three tests that have been carried out, if loaded into a table, it will be like a table. 1.

Table 1 Testing Results of Suricata and Telegram-Based Monitoring

Name IDS	Attack Type	Tools	Rule Detection Capability	Detection Time (ms)
Suricata	Scanning Port, http	Nmap	Yes	13.52
	Ddos Attack	Nikto	Yes	02.17
	Telnet	Hydra	Yes	10.89
Snort	Scanning Port	Nmap	Yes	14.50
	Scanning hhtp	Nikto	No	-
	Teltet	Hydra	Yes	10.95

ComniTech: Journal of Computational Intelligence and Informatics
https://dx.doi.org/000000000
P-ISSN: xxxx-xxxx E-ISSN: xxxx-xxxx

Conclusion

In accordance with the discussion that has been described, the conclusions that can be drawn are: Suricata-based server. Server monitoring will minimize attacks and threats from attackers that have the potential to damage and disrupt the system on the server. Testing Suricata-based server monitoring using pentest tools such as nmap, nikto, and hydra has been successful in testing by sending notifications to telegrams so that network administrators can take early preventive action when getting reports from telegram notifications, as long as they are within internet range. For the future, this suricata-based monitoring system, in its development update, can add IPS (Intrusion Prevent System) features so that suricata can directly block activities from malicious traffic. Thesuricata log is created on its own server and combined with a firewall device so that the process of sending telegram logs is not interrupted by an attack on the server. Building a more interactive telegram bot so that network administrators can create specific commands that they want to display on the server. Develop with other programming languages, such as Python, so that the features of the bot can be more varied and can be further developed.

References

Ismagilova, E., Hughes, L., Rana, N. P., & Dwivedi, Y. K. (2020). Security, Privacy and Risks Within Smart Cities: Literature Review and Development of a Smart City Interaction Framework. Information Systems Frontiers, 24(2), 393–414. https://doi.org/10.1007/s10796-020-10044-1

Gunduz, M. Z., & Das, R. (2020). Cyber-security on smart grid: Threats and potential solutions. Computer Networks, 169, 107094. https://doi.org/10.1016/j.comnet.2019.107094.

Chowdhury, N., & Gkioulos, V. (2021). Cyber security training for critical infrastructure protection: A literature review. Computer Science Review, 40, 100361. https://doi.org/10.1016/j.cosrev.2021.100361

David, D. S., Anam, M., Kaliappan, C., Selvi, S. a. M., Sharma, D. K., Dadheech, P., & Sengan, S. (2022). Cloud Security Service for Identifying Unauthorized User Behaviour. Computers, Materials & Continua/Computers, Materials & Continua (Print), 70(2), 2581–2600. https://doi.org/10.32604/cmc.2022.020213

Vinoth, S., Vemula, H. L., Haralayya, B., Mamgain, P., Hasan, M. F., & Naved, M. (2022). Application of cloud computing in banking and e-commerce and related security threats. Materials Today: Proceedings, 51, 2172–2175. https://doi.org/10.1016/j.matpr.2021.11.121.

Jaya, I. K. N. A., Dewi, I. a. U., & Mahendra, G. S. (2022). Implementation of Wireshark Application in Data Security Analysis on LMS Website. Journal of Computer Networks, Architecture and High Performance Computing, 4(1), 79–86. https://doi.org/10.47709/cnahpc.v4i1.1345

Khrypko, S., Matveev, V., Nykytchenko, O., Stefanova, N., Ishchuk, A., Ishchuk, O., & Bondar, T. (2021). Cybercrime in the Economic Space: Psychological Motivation and Semantic-Terminological Specifics. https://doi.org/10.22937/ijcsns.2021.21.11.18

Sviatun, O. V., Goncharuk, O. V., Roman, C., Kuzmenko, O., & Kozych, I. V. (2021). Combating Cybercrime: Economic and Legal Aspects. WSEAS Transactions on Business and Economics, 18, 751–762. https://doi.org/10.37394/23207.2021.18.72.

Abbasi, M., Shahraki, A., & Taherkordi, A. (2021). Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey. Computer Communications, 170, 19–41. https://doi.org/10.1016/j.comcom.2021.01.021

Andronie, M., Lăzăroiu, G., Iatagan, M., Uță, C., Ștefănescu, R., & Cocoșatu, M. (2021). Artificial Intelligence-Based Decision-Making Algorithms, Internet of Things Sensing Networks, and Deep Learning-Assisted Smart Process Management in Cyber-Physical Production Systems. Electronics, 10(20), 2497. https://doi.org/10.3390/electronics10202497.