JIKO (Jurnal Informatika dan Komputer) Accredited KEMENDIKBUD RISTEK, No.105/E/KPT/2022
Vol. 6, No. 2, Agustus 2023, hlm. 122-130 p-ISSN: 2614-8897
DOI: 10.33387/jiko.v6i2.6385 e-ISSN: 2656-1948

THE IMPLEMENTATION OF NETWORK SERVER SECURITY SYSTEM USING HONEYPOT

Faldi[1], Dinamita Romadoni[2], Muhammad T Sumadi[3]

[1,2,3] Faculty of Science and Enginerring, Informatics Engineering Study Program, Universitas Muhammadiyah Kalimantan Timur, Samarinda, East Borneo, Indonesia
*Email: [1]fal146@umkt.ac.id, [2]1911102441097@umkt.ac.id, [3]mts653@umkt.ac.id

(Received: 20 July 2023, Revised: 26 July 2023, Accepted: 3 August 2023)

Abstract

Network Server security is essential to ensuring the integrity and availability of information systems. This research uses Honeypot technology to implement network Server security at the Muhammadiyah University of East Kalimantan. Honeypots attract the attention of attacks and monitor suspicious activities on the network. The research method used is NDLC (Network Development Life Cycle), which includes designing and implementing Honeypots and collecting and analyzing detected attack data. The research results show three attack techniques used in this study. First, the Slowloris attack with a Honeypot processing time of 2 seconds and Snort processing time of 180 seconds. Second, the GoldenEye attack with a Honeypot processing time of 2 seconds and a Snort processing time of 180 seconds. Third, the use of LOIC tools with a Snort processing time of 180 seconds. However, Honeypots have limitations in identifying Distributed Denial of Service (DDoS) attacks, as they focus more on penetration attempts or other suspicious activities.

**Corresponding Author: Faldi*

1. INTRODUCTION

With the increasing security threats to server networks from cyberattacks, organizations and government institutions are at risk of significant harm [1]. Servers are crucial for providing efficient and effective storage, management, and processing of data. Despite robust security systems, vulnerabilities still exist that can be exploited by internal and external threats [2], [3]. Therefore, it is essential to implement security measures to protect servers from potential harmful attacks [4], [5].

Network security for servers involves implementing robust measures and protocols to defend against a wide range of cyber threats, including unauthorized access, data breaches, denial-of-service (DoS) attacks, malware infiltration, and more [6]. The goal is to establish multiple layers of protection to create a formidable defense against potential vulnerabilities and cyber risks [7], [8].

Various aspects of network security for servers, including the use of firewalls, intrusion detection systems (IDS), encryption protocols, access controls, and the incorporation of honeypots as deceptive security measures [9], [10]. By comprehensively addressing these aspects, organizations can bolster their server network security and ensure their critical data and services' confidentiality, integrity, and availability [11].

One increasingly popular solution for detecting and mitigating server network attacks is using Honeypot technology [12]. Honeypots are designed to mimic real systems or services within a network and act as attractive targets for attackers [13]. Honeypot is an open-source system designed to attract the attention of attackers [14]. Honeypot systems can be in the form of fake servers or applications that appear active and connected to the internet [15]. When attackers attempt to breach them, the Honeypot system records the attackers' activities, such as the type of attack, tools used, and methods employed to compromise the server network [16]. This information is then sent to the network administrator to prevent similar attacks in the future [17].

The main objective of implementing honeypots is to divert attackers from the actual target and learn about the attack methods they employ [18]. This helps enhance the understanding of existing threats and improves the security of server networks [19]. Furthermore, this research combines honeypots with pfsense, which has snort installed on the package manager. This indicates that the study adopts a more holistic approach by leveraging multiple security tools and technologies to protect the server network from attacks. By focusing on the implementation of a network server security system using honeypots to detect and prevent network attacks [20]. This research is expected to make a significant contribution to enhancing server network security [21]. The integration of honeypots with pfsense and snort suggests that this study may offer a more effective and comprehensive approach to addressing security threats on the server.

2. RESEARCH METHOD

NDLC (Network Development Life Cycle) is a methodology used in computer network development that encompasses a series of stages or steps to be followed in order to build and develop a secure and efficient network. efektif [22]. The NDLC (Network Development Life Cycle) method is one of the approaches used to identify existing issues in servers. In Figure 1, there is a flow diagram illustrating the NDLC method.

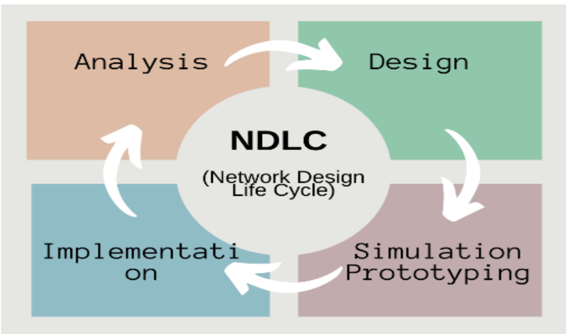


Figure 1. Research Method NDLC (Network Develompent Life Cycle)

1. Requirement Analysis: The requirement analysis phase aims to identify the devices and methods used for implementing Honeypot on the server network. The hardware requirements for creating a Honeypot include a computer with the following specifications: Processor: Intel(R) Core(TM) i5-10400 and RAM: 8.00 GB. The software requirements include the following:
   a. Oracle VM Virtualbox: This virtual machine software is used to run the server operating system.
   b. Kali Linux: It is utilized as an attacker system, employing attack methods such as Slowloris and GoldenEye.
   c. Ubuntu: This operating system is installed with Pentbox, which is used to run the Honeypot.
   d. PfSense Firewall: It is equipped with Snort for detecting attacks and blocking them.

These software and hardware components are essential for setting up the Honeypot system and conducting the necessary attack simulations and security monitoring.

2. Design: After completing the requirement analysis, the next stage is network design and topology. The design phase aims to provide an overview of the implementation to be carried out. Below is the Honeypot network scheme, which can be seen in Figure 2.

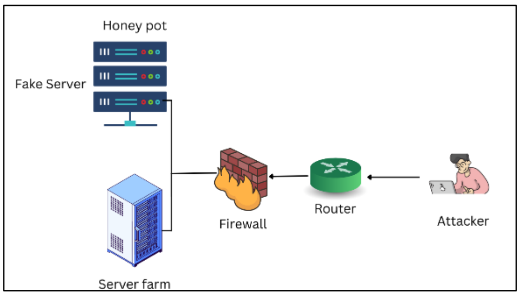


Figure 2. Honeypot technology design

In Figure 2, it is explained that when an attacker attempts to enter the internet network, the firewall redirects the traffic, causing the attack to enter the Honeypot system, which serves as a location for capturing and recording the attacker's activities. The previous analysis also required a network topology, which can be seen in Figure 3, showing the interconnected devices within the network.

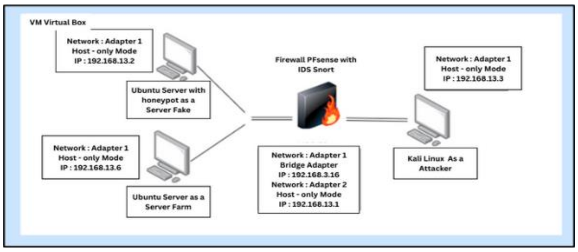


Figure 3. Network Topology

In the topology, there are four virtual machines connected to each other using Host-Only and Bridged Adapter. Kali Linux, Ubuntu, and Ubuntu Server can communicate with each other through the Host-Only network. PfSense, which has Snort installed, is used for detecting and preventing attacks. PfSense acts as a gateway with a WAN connection linked to the host network and a LAN connection connected to other virtual machines via the Host-Only adapter.

3. Simulation Prototype: At this stage, a simulation is conducted based on the designed architecture. The simulation can be observed in the image below, depicted in Figure 4.

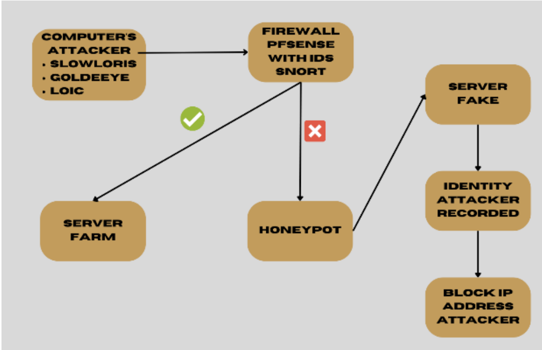


Figure 4. Network Topology

In Figure 4, the simulation process for this research is presented. Based on Figure 4, the testing process begins with launching Slowloris, GoldenEye, and LOIC attacks, aiming to render a legitimate server inaccessible to authorized users. Subsequently, the attacker launches these attacks on the Honeypot system, which is equipped with security measures, and the PfSense firewall, capable of detecting incoming attacks on the server system. If the IP address originates from a legitimate user, it is directed to the genuine server. However, if the IP address is from an unauthorized or invalid user, it is redirected to the Honeypot to trap the attacker using a fake server. The Honeypot records and detects the attacker's identity and activities, and the PfSense firewall can block the attacker's identity and withhold incoming attack packets aimed at the server system.

4. Implementation: The implementation phase involves the actual deployment of all the designed components. This stage includes the installation of equipment, configuration of software and hardware, and integration with existing systems.

3. RESULT AND DISCUSSION

In this stage, we will discuss the implementation process of Honeypot, PfSense Firewall, and TCP, UDP, and HTTP attacks.

3.1 Web Server

The operating system used is Ubuntu Server 22.04.2 LTS. The web server creation is conducted to test the effectiveness of Honeypot as a simulated web server. Below is the image of the web server that has been created, as shown in Figure 5.

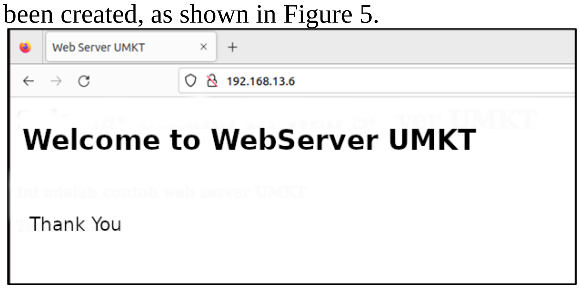


Figure 5. WebServer View

The web server has the IP address 192.168.13.6 as the main server's address. This IP address has been carefully chosen and is separate from the Honeypot's IP address used for the simulated system. By using different IP addresses, the Honeypot and web server can minimize the risk of attacks from attackers.

Having separate IP addresses for the Honeypot and the main web server adds an extra layer of security to the network. It helps ensure that attackers targeting the Honeypot won't accidentally impact the real web server, and vice versa. This practice is a common security measure to isolate potentially vulnerable systems from critical production servers, reducing the potential for damage or unauthorized access during the testing and monitoring process [23].

3.2 Honeypot

The operating system used is Ubuntu Server 22.04.2 LTS. The creation of the Honeypot is carried out within Pentbox, which provides a set of integrated computer security tools in one package. One of the features of Pentbox is Honeypot. By using Honeypot, one can learn the methods and techniques used by attackers [12], [24], [25].

Pentbox's Honeypot feature allows users to deploy a simulated system that attracts and traps attackers. By monitoring the activities of the attackers on the Honeypot, network administrators and security experts can gain valuable insights into the types of attacks being used, the attacker's tactics, and potential vulnerabilities in their network defences [26], [27]. This knowledge can be instrumental in improving overall network security and developing better countermeasures to protect against real-world attacks [28].

Figure 6. Honeypot Configuration in PentBox

After running Pentbox 1.8, several options will be displayed. Since you intend to use Honeypot, select "Network Tools" (Option 2) and then choose "Honeypot" (Option 3) as shown in Figure 6. After setting up Pentbox and directing it to the Honeypot system, the next step is to configure the Honeypot to open port 80 to capture and identify attacks.

Configuring the Honeypot involves setting up a service or application that listens on port 80, which is typically used for web traffic (HTTP). By opening this port on the Honeypot, it creates an attractive target for attackers, making them believe it is a legitimate web server. The Honeypot will record the activities of any attackers who attempt to interact with the open port, providing valuable information on their tactics and techniques.

However, it's essential to implement proper security measures and restrict access to the actual production server to minimize the risk of attackers exploiting the Honeypot or impacting the real network. The data collected from the Honeypot can help strengthen network defenses and improve overall security.

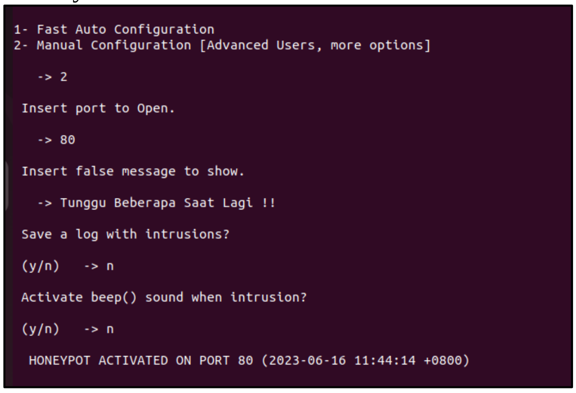


Figure 7. Configure port in honeypot

Setting up the Honeypot to open port 80 and inputting the message "wait a few more moments" on the website provided by Pentbox can be achieved through the configuration of the Honeypot software

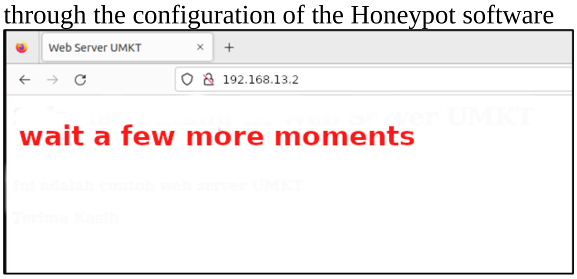


Figure 8. Honeypot webpage

3.3 Slowloris Testing

Testing using the Slowloris attack with the Denial of Service (DoS) method aims to send numerous HTTP connections or invalid requests to the target network with IP 192.168.13.2, causing it to be unable to process requests from legitimate users and resulting in server downtime. Honeypot and PfSense will detect the attack from the server.

Figure 9. TCP Attack using Slowloris

Based on Figure 9, the successful attack was executed with the command *#./Slowloris -dns 192.168.13.2*. The meanings of each parameter in the command are as follows:

Table 1. TCP Attack Testing Analysis

Parameter	Description
#	is the symbol or term used to represent the superuser or administrative access in Unix and Linux operating systems.
./Slowloris	This is the name of the Slowloris executable file or script being executed.
-dns	This parameter indicates that the attack is targeting a specific domain or IP address (in this case, 192.168.13.2) to perform the DoS attack.
192.168.13.2	This is the target IP address where the Slowloris attack is being directed.

In Table 1, the parameters for conducting the Slowloris attack are listed. The attack will then commence by sending incomplete HTTP requests to the target server while keeping the connections open. The goal is to fill up all available connections on the server, causing it to become unresponsive and unable to serve requests from legitimate networks.

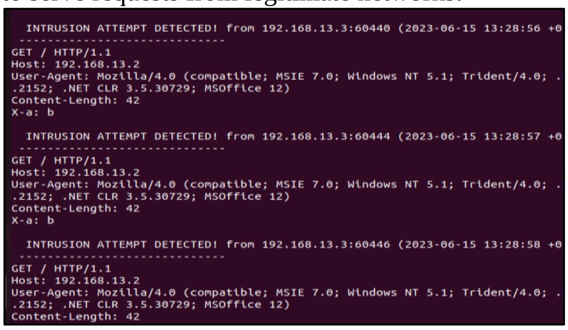
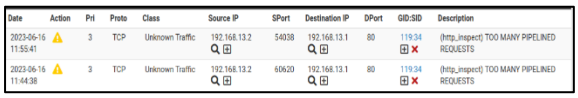


Figure 10. Honeypot log

In Figure 10, the log displays the identified attack captured by the Honeypot. It provides information about the attacker, such as the time of the attack, the attacker's IP address, and the operating system used by the attacker. This information can be crucial for analyzing the attack patterns and understanding the tools and methods utilized by the attacker, which can further assist in enhancing network security and implementing appropriate countermeasures.



Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2023-06-16 11:55:41		3	TCP	Unknown Traffic	192.168.13.2	54038	192.168.13.1	80	119:34	(http_inspect) TOO MANY PIPELINED REQUESTS
2023-06-16 11:44:38		3	TCP	Unknown Traffic	192.168.13.2	60620	192.168.13.1	80	119:34	(http_inspect) TOO MANY PIPELINED REQUESTS

Figure 11. Alert TCP Attack on IDS Snort Pfsense

In Figure 11, the results of the identified Slowloris attack are displayed in Snort on PfSense. The attack is automatically logged in the Snort Alerts on PfSense, providing information such as the time of the attack, the protocol used, the source IP address, the destination IP address, the port utilized, and a description of the attack. The details of the Snort alerts are available in Table 3.

Having the Snort system in place helps to detect and respond to various types of attacks in real-time, including the Slowloris attack. The information captured by Snort enables network administrators to take appropriate actions to mitigate the effects of the attack and strengthen the overall network security.

3.4 GoldenEye Testing

The next testing involves using GoldenEye with the Distributed Denial of Service (DDoS) attack method. This attack floods the target system with a massive amount of network traffic, aiming to overwhelm the system and render it unable to function normally.

GoldenEye is designed to perform DDoS attacks, and it can utilize various techniques to generate a high volume of network traffic, such as HTTP GET and POST requests. By overwhelming the target system's resources, the DDoS attack disrupts its ability to respond to legitimate user requests, leading to service outages or slowdowns.

During this testing, the effectiveness of the system's defense mechanisms, including the Honeypot and PfSense with Snort, in mitigating and detecting the GoldenEye DDoS attack will be evaluated. This evaluation is crucial for enhancing the network's resilience against DDoS attacks and ensuring the continuity of services even under such hostile conditions.

Figure 12. TCP attack using golden eye

In Figure 12, the process shows the GoldenEye attack conducted using Kali Linux against port 80 on IP Address 192.168.13.2. The test for the GoldenEye attack was performed with the following command: # ./GoldenEye.py http://192.168.13.2// -s 10 -m random. Here's the meaning of each parameter :

Table 2. TCP attack testing analysis

Parameter	Description
#	is the symbol or term used to represent the superuser or administrative access in Unix and Linux operating systems.
./*GoldenEye*.py	This is the name of the GoldenEye script or executable file being run.
http://192.168.13.2/	This parameter specifies the target URL for the attack. In this case, it is the target IP address 192.168.13.2 with the "http://" protocol and double slashes "//" indicating the root directory of the website.
-s 10	This parameter represents the number of concurrent threads used for the attack. In this case, 10 threads will simultaneously send attack requests to the target.
-m random	This parameter specifies the method of the attack, in this case, the attack method used is "random."

In Table 2 are the results of the analysis for each parameter used in the GoldenEye attack, which aims to send random fake requests to the target server. As a result, each attack will be blocked by Snort on PfSense.

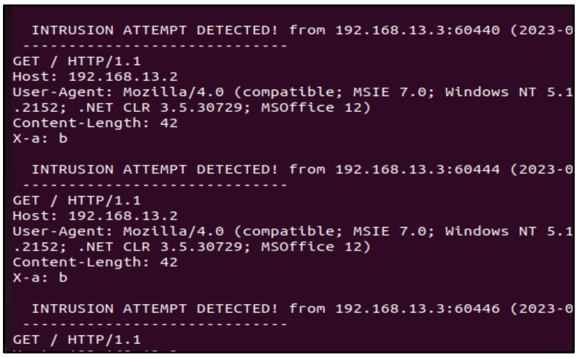


Figure 13. Honeypot log

In Figure 13, the result shows the attack conducted using Kali Linux with the GoldenEye attack method, and it triggered an alert on the Honeypot system. The Honeypot detected and recorded the malicious activity generated by the GoldenEye attack, providing valuable information about the attacker's techniques and methods.

The Honeypot's capability to capture and analyze such attacks is essential in understanding the various tactics used by attackers and strengthening the overall network security. The recorded data can be used for further analysis, improving defense strategies, and enhancing the network's resilience against future attacks.

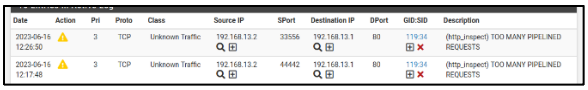


Figure 14. Alert TCP Attack on IDS Snort PfSense

In Figure 14, there is an attack using GoldenEye, and it automatically triggers an alert in Snort on PfSense. The alert contains information such as the time of the attack, the protocol used, the source IP address, the destination IP address, the port utilized, and a description of the attack. The details of the Snort alerts are available in Table 3.

3.5 LOIC (Low Ion Cannon) Testing

The next testing involves an attack using the LOIC (Low Orbit Ion Cannon) tool with the Distributed Denial of Service (DDoS) attack method.

The objective is to send a massive number of requests to the target server, flooding it with excessive traffic and overloading the server's resources. As a result, the server becomes unable to serve legitimate user requests, leading to service disruption or unresponsiveness.

LOIC is designed to perform DDoS attacks and is capable of launching simultaneous attacks from multiple sources. By coordinating these attacks, LOIC can generate a high volume of network traffic directed at the target server. This influx of requests exhausts the server's processing capabilities, making it incapable of handling legitimate user requests and causing service downtime.

During this testing, the effectiveness of the network's defense mechanisms, including the Honeypot and PfSense with Snort, in detecting and mitigating the LOIC DDoS attack will be assessed. Understanding how the network handles such attacks is crucial for enhancing its resilience against DDoS threats and ensuring continuous service availability for legitimate users.

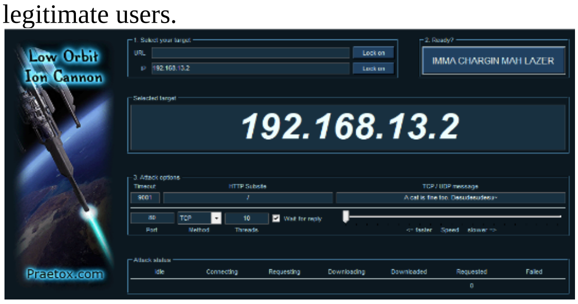
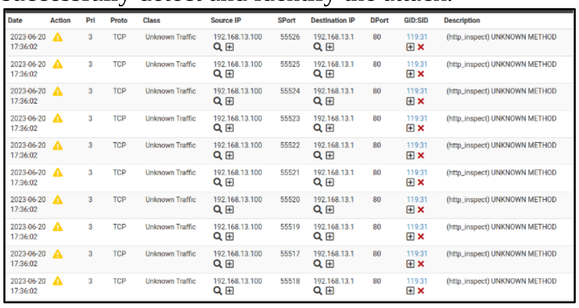


Figure 15. TCP attack using LOIC

In Figure 15, the process illustrates the HTTP attack conducted using LOIC against port 80 on IP Address 192.168.13.2. However, in this scenario, the Honeypot was unable to identify the attack. On the other hand, Snort (as shown in Figure 16) was able to successfully detect and identify the attack.



Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2023-06-20 17:36:02		3	TCP	Unknown Traffic	192.168.13.100	55526	192.168.13.1	80	119:31	(http_inspect) UNKNOWN METHOD
2023-06-20 17:36:02		3	TCP	Unknown Traffic	192.168.13.100	55525	192.168.13.1	80	119:31	(http_inspect) UNKNOWN METHOD
2023-06-20 17:36:02		3	TCP	Unknown Traffic	192.168.13.100	55524	192.168.13.1	80	119:31	(http_inspect) UNKNOWN METHOD
2023-06-20 17:36:02		3	TCP	Unknown Traffic	192.168.13.100	55523	192.168.13.1	80	119:31	(http_inspect) UNKNOWN METHOD
2023-06-20 17:36:02		3	TCP	Unknown Traffic	192.168.13.100	55522	192.168.13.1	80	119:31	(http_inspect) UNKNOWN METHOD
2023-06-20 17:36:02		3	TCP	Unknown Traffic	192.168.13.100	55521	192.168.13.1	80	119:31	(http_inspect) UNKNOWN METHOD
2023-06-20 17:36:02		3	TCP	Unknown Traffic	192.168.13.100	55520	192.168.13.1	80	119:31	(http_inspect) UNKNOWN METHOD
2023-06-20 17:36:02		3	TCP	Unknown Traffic	192.168.13.100	55519	192.168.13.1	80	119:31	(http_inspect) UNKNOWN METHOD
2023-06-20 17:36:02		3	TCP	Unknown Traffic	192.168.13.100	55517	192.168.13.1	80	119:31	(http_inspect) UNKNOWN METHOD
2023-06-20 17:36:02		3	TCP	Unknown Traffic	192.168.13.100	55518	192.168.13.1	80	119:31	(http_inspect) UNKNOWN METHOD

Figure 16. Alert TCP Attack LOIC on IDS Snort PfSense

In Figure 16, there is an attack using TCP, and it automatically triggers an alert in Snort on PfSense. The alert contains information such as the time of the attack, the protocol used (TCP), the source IP address, the destination IP address, the port utilized, and a description of the attack. The details of the Snort alerts are available in Table 3.

3.6 *Firewall* Snort on *PfSense*

Then, in figures 11, 14, and 16, each parameter of the TCP Snort attack alert on PfSense is explained as follows:

Table 3. Alert Analysis on Snort PfSense

Parameter	Description
Date	Shows the date and time of the attack occurrence.
Pri (Priority)	Indicates the severity level of the threat; the higher the priority value, the more serious the threat.
Protocol	Shows the network protocol type as TCP.
Class	Indicates the detected attack category.
Source IP	Shows the source IP address that triggered the threat alert.
Sport (Source Port)	Port from which the attacker originates the attack towards the target server.
Destination IP	IP address indicating the device or server receiving the related network traffic.
DPort (Destination Port)	Port targeted by the attacker to send attack packets.
Description	Information about the ongoing attack.

In Table 3, each parameter is analyzed to provide information about the detected attacks in the PfSense security system. By analyzing the characteristics of the attacks, PfSense can effectively block those attacks to protect the server network.

The analysis of the detected attacks helps PfSense in identifying the attack patterns, sources, and methods used by the attackers. With this knowledge, PfSense can implement appropriate rules, filters, and countermeasures to block and mitigate similar attacks in the future. By actively responding to threats, PfSense enhances the overall security posture of the network and ensures the protection of the server and its resources from potential threats.

3.7 *Results* of Attack Data

After conducting attack testing to assess the performance of the Honeypot and Snort Firewall on PfSense in identifying, detecting, and blocking attacks, the following data was obtained from the server network:

Table 4. Result on data from attack testing

Attack Tools	Times notification received on honeypot	Times notification received on Snort
Slowloris	2s	180s
Golden Eye	2s	120s
LOIC	-	180s

In Table 4, the results of the attack testing conducted on the server are presented. Based on the performed testing, three types of attacks were identified:

1) Slowloris: Detected by Honeypot with a processing time of 2 seconds and by Snort with a processing time of 180 seconds.
2) GoldenEye: Detected by Honeypot with a processing time of 2 seconds and by Snort with a processing time of 120 seconds.
3) LOIC: Detected by Snort with a processing time of 180 seconds, with a time difference of 60 seconds for Snort to identify the attack.

Honeypot is designed to attract and capture suspicious activities, but it may not detect all types of attacks, such as the LOIC attack. Honeypots are more focused on detecting penetration attempts or other suspicious activities.

Each type of attack may have distinct characteristics, and the choice of defense mechanisms, like Snort in this case, plays a crucial role in identifying and mitigating different attack types. By utilizing both Honeypot and Snort on PfSense, the network's overall security is enhanced, as they complement each other in capturing various types of threats and attacks, thereby fortifying the network's defenses against potential risks.

4. CONCLUSION

After conducting the research, analyzing the data, and discussing the findings, the following conclusions have been drawn:

1. Based on the conducted testing, Honeypot proved to be effective in detecting Slowloris and GoldenEye attacks, but it was not efficient in detecting DDoS attacks executed using specialized software like LOIC.
2. PfSense Firewall was not able to identify DDoS attacks comprehensively, but it provided other relevant information regarding the ongoing attacks.
3. Honeypot performed well in detecting threat packets, and it required only 2 seconds, while Snort took approximately 180 seconds, depending on the internet connection.
4. Honeypot's alerts worked effectively and provided real-time information, whereas Snort alerts took some time to deliver the information.

In summary, the research highlights the strengths and limitations of the Honeypot and Snort Firewall implementations. Honeypot was successful in detecting specific types of attacks but had limitations in identifying DDoS attacks with specialized tools. On the other hand, Snort showed effectiveness in identifying various attacks, including DDoS, but it might have longer processing times compared to Honeypot. Understanding these strengths and weaknesses is essential for designing a comprehensive network security strategy that combines various tools and techniques to protect the network from different types of threats.

5. REFERENCES

[1] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments," *Energy Reports*, vol. 7, pp. 8176–8186, Nov. 2021, doi: 10.1016/j.egyr.2021.08.126.

[2] F. Akpan, G. Bendiab, S. Shiaeles, S. Karamperidis, and M. Michaloliakos, "Cybersecurity Challenges in the Maritime Sector," *Network*, vol. 2, no. 1, pp. 123–138, Mar. 2022, doi: 10.3390/network2010009.

[3] S. Usman, M. Jamil, and A. Fuad, "Implementasi Teknologi Cloud Private Network Berbasis Teknologi Virtualisasi," *JIKO (Jurnal Inform. dan Komputer) UNKHAIR*, vol. 2, no. 2, pp. 56–60, 2019.

[4] C. S. Kruse, B. Smith, H. Vanderlinden, and A. Nealand, "Security Techniques for the Electronic Health Records," *J. Med. Syst.*, vol. 41, no. 8, p. 127, Aug. 2017, doi: 10.1007/s10916-017-0778-4.

[5] F. Simson, "Comparative Analysis of Quality of Service Performance of Video Streaming Services Using OSPF and EIGRP Networks," *JIKO (Jurnal Inform. dan Komputer)*, vol. 6, no. 1, pp. 36–42, Apr. 2023, doi: 10.33387/jiko.v6i1.5826.

[6] S. K. Khan, N. Shiwakoti, P. Stasinopoulos, and Y. Chen, "Cyber-attacks in the next-generation cars, mitigation techniques, anticipated readiness and future directions," *Accid. Anal. Prev.*, vol. 148, p. 105837, Dec. 2020, doi: 10.1016/j.aap.2020.105837.

[7] F. Cremer *et al.*, "Cyber risk and cybersecurity: a systematic review of data availability," *Geneva Pap. Risk Insur. - Issues Pract.*, vol. 47, no. 3, pp. 698–736, Jul. 2022, doi: 10.1057/s41288-022-00266-6.

[8] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Cybersecurity Threats, Countermeasures and Mitigation Techniques on the IoT: Future Research Directions," *Electronics*, vol. 11, no. 20, p. 3330, Oct. 2022, doi: 10.3390/electronics11203330.

[9] X. Yang, J. Yuan, H. Yang, Y. Kong, H. Zhang, and J. Zhao, "A Highly Interactive Honeypot-Based Approach to Network Threat Management," *Futur. Internet*, vol. 15, no. 4, p. 127, Mar. 2023, doi: 10.3390/fi15040127.

[10] *Virtualization for Security*. Elsevier, 2009.

[11] U. Tariq, I. Ahmed, A. K. Bashir, and K. Shaukat, "A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review," *Sensors*, vol. 23, no. 8, p. 4117, Apr. 2023, doi: 10.3390/s23084117.

[12] M. Mesbah, M. S. Elsayed, A. D. Jurcut, and M. Azer, "Analysis of ICS and SCADA Systems Attacks Using Honeypots," *Futur. Internet*, vol. 15, no. 7, p. 241, Jul. 2023, doi: 10.3390/fi15070241.

[13] W. A. Sulaksono and C. E. Suharyanto,

"Implementasi Honeypot Sebagai Sistem Keamanan Jaringan Pada Virtual Private Server," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 1, pp. 90–95, 2020.

[14] H. Artail, H. Safa, M. Sraj, I. Kuwatly, and Z. Al-Masri, "A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks," *Comput. Secur.*, vol. 25, no. 4, pp. 274–288, Jun. 2006, doi: 10.1016/j.cose.2006.02.009.

[15] W. Ahmad, M. A. Raza, S. Nawaz, and F. Waqas, "Detection and Analysis of Active Attacks using Honeypot," *Int. J. Comput. Appl.*, vol. 184, no. 50, pp. 27–31, Mar. 2023, doi: 10.5120/ijca2023922624.

[16] C. Sanders and J. Smith, "Using Canary Honeypots for Detection," in *Applied Network Security Monitoring*, Elsevier, 2014, pp. 317–338.

[17] H. Wafi, A. Fiade, N. Hakiem, and R. B. Bahaweres, "Implementation of a modern security systems honeypot Honey Network on wireless networks," in *2017 International Young Engineers Forum (YEF-ECE)*, May 2017, pp. 91–96, doi: 10.1109/YEF-ECE.2017.7935647.

[18] S. Maesschalck, V. Giotsas, B. Green, and N. Race, "Don't get stung, cover your ICS in honey: How do honeypots fit within industrial control system security," *Comput. Secur.*, vol. 114, p. 102598, Mar. 2022, doi: 10.1016/j.cose.2021.102598.

[19] P. Williams, I. K. Dutta, H. Daoud, and M. Bayoumi, "A survey on security in internet of things with a focus on the impact of emerging technologies," *Internet of Things*, vol. 19, p. 100564, Aug. 2022, doi: 10.1016/j.iot.2022.100564.

[20] Li Li, Hua Sun, and Zhenyu Zhang, "The research and design of honeypot system applied in the LAN security," in *2011 IEEE 2nd International Conference on Software Engineering and Service Science*, Jul. 2011, pp. 360–363, doi: 10.1109/ICSESS.2011.5982237.

[21] M. I. Al-Ghamdi, "WITHDRAWN: Effects of knowledge of cyber security on prevention of attacks," *Mater. Today Proc.*, Apr. 2021, doi: 10.1016/j.matpr.2021.04.098.

[22] N. Nurdadyansyah and M. Hasibuan, "Perancangan Local Area Network Menggunakan NDLC Untuk Meningkatkan Layanan Sekolah," pp. 342–346, 2021.

[23] A. B. Feroz Khan and A. G, "A Multi-layer Security approach for DDoS detection in Internet of Things," *Int. J. Intell. Unmanned Syst.*, vol. 9, no. 3, pp. 178–191, Jun. 2021, doi: 10.1108/IJIUS-06-2019-0029.

[24] A. Mairh, D. Barik, K. Verma, and D. Jena, "Honeypot in network security," in *Proceedings of the 2011 International Conference on Communication, Computing & Security - ICCCS '11*, 2011, p. 600, doi: 10.1145/1947940.1948065.

[25] M. Mohammadzad and J. Karimpour, "Using rootkits hiding techniques to conceal honeypot functionality," *J. Netw. Comput. Appl.*, vol. 214, p. 103606, May 2023, doi: 10.1016/j.jnca.2023.103606.

[26] M. R. Amal and P. Venkadesh, "H-DOCTOR: Honeypot based firewall tuning for attack prevention," *Meas. Sensors*, vol. 25, no. December 2022, 2023, doi: 10.1016/j.measen.2022.100664.

[27] D. Fraunholz, M. Zimmermann, S. D. Anton, J. Schneider, and H. Dieter Schotten, "Distributed and highly-scalable WAN network attack sensing and sophisticated analysing framework based on Honeypot technology," in *2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence*, Jan. 2017, pp. 416–421, doi: 10.1109/CONFLUENCE.2017.7943186.

[28] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, and J. Schönfelder, "A Survey on Honeypot Software and Data Analysis," Aug. 2016, [Online]. Available: http://arxiv.org/abs/1608.06249.

[29] A. J. Alhasan and N. Surantha, "Evaluation of Data Center Network Security based on Next-Generation Firewall," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 9, pp. 518–525, 2021, doi: 10.14569/IJACSA.2021.0120958.

[30] Tati Ernawati and Fikri Faiz Fadhlur Rachmat, "Keamanan Jaringan dengan Cowrie Honeypot dan Snort Inline-Mode sebagai Intrusion Prevention System," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, pp. 180–186, 2021, doi: 10.29207/resti.v5i1.2825.

[31] D. Desmira, D. Apriana, and M. Avicena H.B.H, "Analisa Jaringan Local Area Network Pada Laboratorium Komputer SMK Informatika Kota Serang," *INSANtek*, vol. 3, no. 1, pp. 23–31, 2022, doi: 10.31294/instk.v3i1.532.

[32] L. Dian, "Analisis Implementasi Honeypot Dan IDS Pada Jaringan Hospot Sebagai Penunjang Keamanan Jaringan Di Kopkar BGA Dengan Menggunakan Honeyd Dan Snort," pp. 1–10, 2022.

[33] A. Zainuddin, L. Affandi, and A. D. Susilo, "Honeypot Dan Ids Di Kampus Stmik Ppkia Pradnya Paramita Malang," *J. Teknol. Inf.*, vol. 5, no. 2, pp. 107–111, 2014.

[34] S. Akbar *et al.*, "Analisis Performansi Intrusion Detection System , Firewall , Honeypot Dan Load Balancer Dalam Rangka Mitigasi Serangan Dos Dan Ddos Pada Lpse Kab . Luwu Timur Performance Analysis of Intrusion Detection System , Firewall , Honeypot and Load Balancer To Mi," 2016.

[35] Astrid Noviriandini, Hermanto Hermanto, Diah Ayu Ambarsari, and Didy Eriawan, "Analisis Management Bandwidth Dan Firewall Dengan

Router Mikrotik Pada Pt. Bca Multifinance," *J. Tek. dan Sci.*, vol. 1, no. 3, pp. 40–45, 2022, doi: 10.56127/jts.v1i3.466.

[36] S. M. Sulaman, "An Analysis and Comparison of The Security Features of Firewalls and IDSs," 2011.

[37] M. Arman and N. Rachmat, "Implementasi Sistem Keamanan Web Server Menggunakan Pfsense," *Jusikom J. Sist. Komput. Musirawas*, vol. 5, no. 1, pp. 13–23, 2020, doi: 10.32767/jusikom.v5i1.752.

[38] A. P. Sari and N. Kemala, "PERANCANGAN JARINGAN VIRTUAL PRIVATE NETWORK BERBASIS IP SECURITY MENGGUNAKAN ROUTER MIKROTIK," vol. 7, no. 2, pp. 150–164, 2020.

[39] M. Iqbal, A.- Arini, and H. B. Suseno, "Analisa Dan Simulasi Keamanan Jaringan Ubuntu Server Dengan Port Knocking, Honeypot, Iptables, Icmp," *Cyber Secur. dan Forensik Digit.*, vol. 3, no. 1, pp. 27–32, 2020, doi: 10.14421/csecurity.2020.3.1.1933.

[40] K. Al Fikri and Djuniadi, "Keamanan Jaringan Menggunakan Switch Port Security," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 2, pp. 302–307, 2021.

[41] K. Y. Layuk, "Analisis Keamanan Jaringan Web Server Menggunakan Suricata Pada Sekolah Menengah Pertama Negeri 1 Palopo Krismadia Yanti Layuk," Universitas Cokroamimnoto Palopo, 2021.

[42] N. Fitriana and F. N. Khasanah, "Honeypot Menggunakan Honeyd Sebagai Solusi Keamanan Jaringan Dari Aktivitas Serangan," vol. 5, no. 2, pp. 143–152, 2018.

[43] A. S. Imron Kadafi Hariri, "Pemantauan Dan Analisis Performa Sistem Honeypot Dengan Simple Network Management Protocol (SNMP)," vol. 2, no. 1, 2021.

[44] 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.

[45] M. H. Siregar and R. Dermawati, "Implementasi Honeypot Pada Jaringan Internet Labor Fakultas Teknik Uniks Menggunakan Dionaea Sebagai Keamanan Jaringan," *Edutic - Sci. J. Informatics Educ.*, vol. 7, no. 1, pp. 20–30, 2020, doi: 10.21107/edutic.v7i1.8660.

[46] W. Wilman, I. Fitri, and N. D. Nathasia, "Port Knocking Dan Honeypot Sebagai Keamanan Jaringan Pada Server Ubuntu Virtual," *J I M P - J. Inform. Merdeka Pasuruan*, vol. 3, no. 1, pp. 27–33, 2018, doi: 10.37438/jimp.v3i1.86.

[47] M. T. Alshammari, "Editorial Preface From the Desk of Managing Editor… Associate Editors," *IJACSA - Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 9, pp. 204–208, 2019.

[48] D. Bayu Rendro and W. Nugroho Aji, "Analisis Monitoring Sistem Keamanan Jaringan Komputer Menggunakan Software Nmap (Studi Kasus Di Smk Negeri 1 Kota Serang)," *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 7, no. 2, pp. 108–115, 2020.

[49] K.-C. Lu *et al.*, "Evaluation and Build to honeypot System about SCADA Security for Large-Scale IoT Devices," *J. Robot. Netw. Artif. Life*, vol. 6, no. 3, pp. 157–161, 2019, doi: 10.2991/jrnal.k.191202.008.

[50] D. Desmira *et al.*, "Implementasi Honeypot Sebagai Sistem Keamanan Jaringan Pada Virtual Private Server," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 1, pp. 90–95, 2020, doi: 10.30630/jitsi.1.2.10.