Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.14144
e- ISSN : 2541-2019
p- ISSN : 2541-044X

Data Visualization for Building a Cyber Attack Monitoring Dashboard Based on Honeypot

I Gede Adnyana[1)*], Ayu Manik Dirgayusari[2)], Ketut Jaya Atmaja[3)]
[1,2,3)]Institut Bisnis dan Teknologi Indonesia, Indonesia
[1)]adnyana@instiki.ac.id, [2)]ayu.manik@instiki.ac.id, [3)]ketutjayaatmaja@instiki.ac.id

Submitted :Sep 29, 2024 | **Accepted** : Oct 28, 3034 | **Published** : Oct 30, 2024

Abstract: Computer networks are essential for modern life, enabling efficient global information exchange. However, as technology advances, network security challenges grow. To enhance security, honeypots are used alongside firewalls, mimicking legitimate systems to attract hackers and analyze their attack methods. In this research, Cowrie and Dionaea honeypots are implemented. Cowrie targets brute force attacks on SSH, while Dionaea detects port scanning and denial of service (DoS) attacks. These honeypots effectively capture and log malicious activities, providing insights into attack patterns. The collected data is analyzed using the ELK Stack, which offers real-time visualization of attack trends, frequency, and methods. This analysis helps security teams quickly identify and mitigate threats. The integration of honeypots with the ELK Stack significantly enhances network defense by improving detection, analysis, and response to cyber threats. The analysis of the results shows that both honeypots effectively capture and record malicious activities entering the network, providing critical insights into the attack patterns employed by attackers. Within just minutes of deployment, the honeypots logged over 1,000 attacks, predominantly originating from botnets attempting to exploit system vulnerabilities. The captured log data is processed through the ELK Stack, allowing for real-time visualization of attack patterns, including geographic origins, attack frequency, and methods used. This enables security teams to proactively identify trends, assess risks, and implement targeted mitigation strategies more efficiently.

Keywords: Cowrie; Dionaea; ELK Stack; Honeypot; Network Security

INTRODUCTION

In the rapidly evolving digital era, computer networks have become the backbone of various aspects of modern life, facilitating the fast and efficient exchange of information worldwide. As businesses, governments, and individuals increasingly rely on these networks for communication, commerce, and service delivery, the importance of securing them cannot be overstated. However, alongside these advancements, new challenges related to computer network security have emerged, making it a pressing concern for stakeholders in every sector. Computer network security has become a crucial aspect of information technology, aiming to protect systems, data, and network infrastructure from potential threats (Purba & Efendi, 2021).

One effective approach to enhancing computer network security is by installing or configuring a Honeypot that can collaborate with the operating system's built-in firewall. A Honeypot is a system designed to resemble a company's actual system, with the purpose of attracting hackers to attack it (Alzoubi & Alrashdan, 2022; Javadpour et al., 2024; Pratama et al., 2023). By creating a decoy that mimics real systems, Honeypots serve as traps for cybercriminals. This deception allows organizations to gather valuable insights into attack strategies, methodologies, and potential vulnerabilities. The Honeypot system aims to deceive hackers by presenting a decoy system, gathering information about the attacks they perform, and potentially uncovering the identity of the hackers (Natanegara et al., 2023; Sun et al., 2020; Wastumirad & Darmawan, 2021). This information is instrumental in understanding the tactics used by attackers, enabling organizations to strengthen their defenses against real threats. By studying the behaviors of hackers in a controlled environment, cybersecurity teams can better anticipate future attacks and refine their security protocols accordingly.

The information collected from the Honeypot is stored in the form of log data, including the attacker's IP address, the system ports targeted, the services attacked, and the time of the attack (Wibawa et al., 2020). Each attack log recorded by the Honeypot requires thorough analysis by an administrator. However, the complexity of the log data can make it difficult to interpret directly, particularly when dealing with large volumes of data generated over time. This complexity often leads to delays in response times to actual threats.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.14144
e- ISSN : 2541-2019
p- ISSN : 2541-044X

Therefore, to simplify the reading of the log data and facilitate effective decision-making, a monitoring dashboard is essential. This dashboard can visualize the information collected from the Honeypot in various forms, such as charts, graphs, or written explanations. By transforming complex log data into easily digestible visual formats, cybersecurity teams can quickly identify patterns, trends, and anomalies in attack behaviors, allowing for more proactive and informed responses to potential threats. This capability not only enhances overall network security but also contributes to a more resilient digital infrastructure capable of withstanding the evolving landscape of cyber threats.

LITERATURE REVIEW

Currently, research related to honeypot technology has seen significant growth; however, its primary focus is still on the implementation of honeypots for detecting attacks. The research conducted (Matin & Rahardjo, 2020) shows that the issue faced is the significant spread of malware, causing traditional analysis methods such as signature matching and heuristic methods to become ineffective. The objective of this research is to investigate the use of honeypots in machine learning for malware detection. The subsequent research conducted (Yang et al., 2023) discusses solutions to cybersecurity issues using honeypot technology. This research notes that the common defensive approaches in current cyber confrontations are often ineffective, thus necessitating the adoption of honeypots to shift the approach to a proactive one. The proposed system utilizes honeypots to observe the behavior and attack methods of intruders, aiming to enhance the protection of equipment and information. The use of high-interaction honeypots and a modular design is claimed to improve the effectiveness and usability of this honeypot technology. The results of the study indicate that the proposed method has significant advantages over traditional honeypot technologies and provides detailed data support for network security protection (Amal & Venkadesh, 2022; Mondal & Goswami, 2021; Yang et al., 2023). The research conducted (Gupta et al., 2023) discusses that the dependence on online activities creates demands on online applications to be more responsive and continuously adapt. A microservices-based architecture is proposed as a solution that enables the flexibility and speed required, but it also leads to more intensive cyberattacks. The article proposes an innovative honeypot design with a new monitoring setup to collect data from cyberattacks in microservices-based systems.

Previous research has shown efforts to enhance the effectiveness of honeypots in monitoring and detecting cyberattacks. While the reliability of honeypots in detecting attacks plays an important role, it is also essential to develop data presentation methods that can help administrators understand the available information. This is due to the default log data generated by honeypots, which often consists of raw data that requires specialized knowledge to interpret the meanings contained within it. Data visualization has the potential to present information about cyberattacks in a more intuitive manner that can be easily understood by administrators. This study focuses on developing data visualization techniques to build a cyberattack monitoring dashboard based on honepot technology. This initiative is expected to enhance administrators' ability to interpret and respond to cyber threats more efficiently.

Honeypot

A honeypot is a system that is built to closely resemble the original system, allowing it to deceive hackers into thinking they are attacking the actual target system. The primary goal of the honeypot system is to mislead hackers with a decoy system, as well as to gather information about the attacks carried out by the intruders and to obtain information about the hackers themselves (Wastumirad & Darmawan, 2021).

According (Ubaidillah et al., 2023) services on honeypots can be classified into three types :

1. Low Interaction Honeypot
   Low Interaction Honeypot is a type of honeypot service that has a low level of interaction with attackers on the system. This type of service is typically used to mimic a service on a system, such as HTTP, SMB, FTP, and others. One advantage of using a Low Interaction Honeypot is the ease of installation and configuration. Additionally, this service is suitable for detecting automated attacks from bots. However, a disadvantage of Low Interaction Honeypots is that if attackers do not use bots in their attack activities, this type of service is more easily detected by the attackers.
2. Medium Interaction Honeypot
   Medium Interaction Honeypot is a type of honeypot service that has a higher level of interaction compared to Low Interaction Honeypots. This type of service creates a fake operating system to interact directly with attackers. Consequently, every action taken by the attacker on the system is recorded by the honeypot, and this recorded data can be analyzed by network administrators. Medium Interaction Honeypots provide more comprehensive attack information than Low Interaction Honepots. However, the design of this honeypot service is quite complex and requires more effort in maintenance to ensure that attackers interacting with this service do not quickly realize that the attacked system is a decoy.
3. High Interaction Honeypot
   High Interaction Honeypot is a type of honeypot service that has the highest level of interaction compared to other types of honeypot services. This type of service replicates the entire original system on the server, allowing attackers to fully interact with the honeypot system without limitations. High Interaction Honeypots

*name of corresponding author

e- ISSN : 2541-2019
p- ISSN : 2541-044X

are typically used by network administrators to capture patterns of unknown cyberattacks from the internet, enabling these attacks to be studied and avoided in the future. However, the drawbacks of implementing this honeypot service include the complexity of design and configuration on the original system, as well as a higher level of risk because attackers can take control of the honeypot, which could pose a threat to the overall network system.

Cowrie

Cowrie is a type of honeypot designed to emulate the SSH (Secure Shell) services of a server. Cowrie is a medium interaction honeypot commonly used to detect brute force attacks targeting SSH and Telnet services (Tati Ernawati & Fikri Faiz Fadhlur Rachmat, 2021). After the SSH/Telnet service is attacked, Cowrie redirects the attacker to a fake honeypot service, leading the attacker to believe that their attack has been successful. However, the attacker is merely entering the trap set by Cowrie. Cowrie also features logging, which is the process by which Cowrie records or logs the activities performed by the attacker while interacting with the fake honeypot service. Through this logging feature, network administrators are able to identify and analyze every activity of the attacker on the fake honeypot (Mispriatin et al., 2022; Yudyanto et al., 2020).

Dionaea

Dionaea is a type of honeypot that has a low level of interaction with attackers. It is also a honeypot developed from the Nephentes honeypot (Rahman et al., 2020). Dionaea is typically used to emulate several system services, including FTP (File Transfer Protocol), TFTP (Trivial File Transfer Protocol), SMB (Server Message Block), and other file-sharing services (Wastumirad & Darmawan, 2021). Dionaea can also detect malware attacks (Yugitama et al., 2020). Python is the programming language used by Dionaea to create scripts, and libemu is employed to detect shellcode. It also supports IPv6 and TLS. Dionaea can identify and evaluate payloads to generate malware replicas. Libemu is used by dionaea to detect payloads.

ELK (Elasticsearch, **Logstash**, and **Kibana) Stack**

Elasticsearch

Elasticsearch is an application used for analyzing and searching data (Yugitama et al., 2020). This application is also freely available and open-source on the internet. Elasticsearch is also referred to as a database that uses document-oriented storage, commonly known as NoSQL. NoSQL databases have a data storage format in the form of .JSON or JavaScript Object Notation. The advantage of NoSQL databases is their extremely fast data retrieval, as each column is indexed automatically (Stoleriu et al., 2021). The success of data retrieval depends on a process known as data ingestion. This process involves processing raw data through steps such as parsing, normalization, and adding supplementary data before the indexing process. Additionally, Elasticsearch is also scalable, meaning it can be adjusted according to the system's needs (Zmaranda et al., 2021).

Logstash

Logstash is an application used to process and filter data before it is sent to the Elasticsearch application for indexing (Yudhistira & Fitrisia, 2023). The data obtained by Logstash can come from various data sources, such as Beats, Amazon S3, and Kafka. When handling each piece of data received from these sources, Logstash requires a processing pipeline or configuration document consisting of three stages: input, output, and filter. The architecture of these stages can be achieved through the provided plugins. Each of these plugins assists the Logstash application in retrieving data as needed based on a schedule, outputting log results as documents, and more (Yudhistira & Fitrisia, 2023).

Kibana

Kibana is an application used to analyze and visualize data obtained through the Elasticsearch application. The Kibana application can read data in the form of structured or unstructured text, numerical data, time series data, geospatial data, logs, metrics, security events, and more. This application can be accessed via a web browser. Features of Kibana can present data in the form of charts, metrics, tables, images, and others. Another feature of Kibana is the dashboard feature, which can be used to aggregate visualized data into a single main page (Stoleriu et al., 2021).

Filebeat

Filebeat is an application used to send log data to either Logstash or Elasticsearch. This application acts as an agent installed on the server or data source system. With Filebeat, log data can be sent to the Logstash shipper. The Logstash shipper is where the Logstash application retrieves log data from various sources, one of which is the Filebeat application. Subsequently, each piece of data will be parsed and sent to Elasticsearch. Parsing is the process of breaking down incoming data into smaller, more structured components. The data that has gone through the parsing process in Elasticsearch will be used by Kibana for visualization (Sholihah et al., 2020).

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.14144
e- ISSN : 2541-2019
p- ISSN : 2541-044X

METHOD

This research explores honeypot technology and relevant data visualization techniques to produce innovative solutions that meet the needs for monitoring cyber attacks. The following is a flowchart of the research, illustrating the stages of the research process to be conducted.

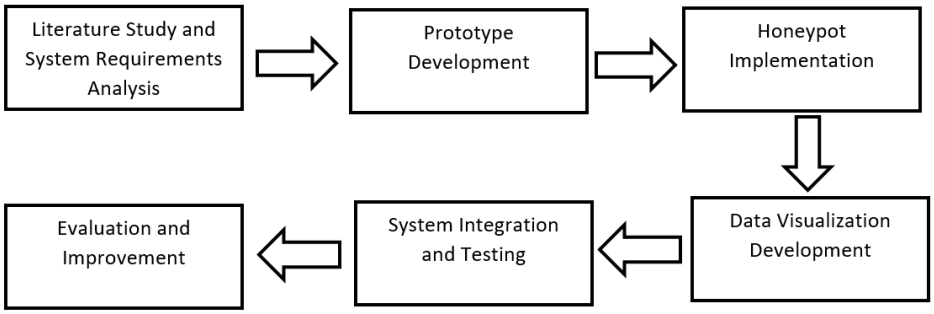

Fig. 1 Research stages

In Fig. 1, the stages of the research can be explained, starting with the literature study and system needs analysis. This involves conducting relevant literature reviews in the context of developing a honeypot-based cyber attack monitoring dashboard and analyzing requirements to understand the fundamental requirements and user needs related to the honeypot-based monitoring dashboard.

Next, in the prototype development stage, an initial prototype of the cyber attack monitoring dashboard will be built using data visualization technology and integration with the honeypot. This prototype will include graphical representations of the cyber attack data collected by the honeypot, as well as an intuitive user interface.

In the honeypot implementation stage, the honeypot will be set up and configured to collect cyber attack data from relevant environments. This process involves adjusting the honeypot configuration to meet research needs and allow for accurate data collection.

During the data visualization development stage, effective data visualizations will be developed to convert the cyber attack data obtained from the honeypot into informative and easily understandable visual representations on the dashboard.

In the system integration and testing stage, the dashboard prototype will be integrated with the data obtained from the honeypot, and the functionality of the dashboard will be tested for monitoring and analyzing cyber attacks. Testing will be conducted to ensure the dashboard's performance and reliability.

Finally, in the evaluation and improvement stage, the performance of the dashboard will be evaluated based on targeted outcomes such as response speed, accuracy, and ease of use. If necessary, improvements and refinements to the dashboard will be made based on the evaluation results.

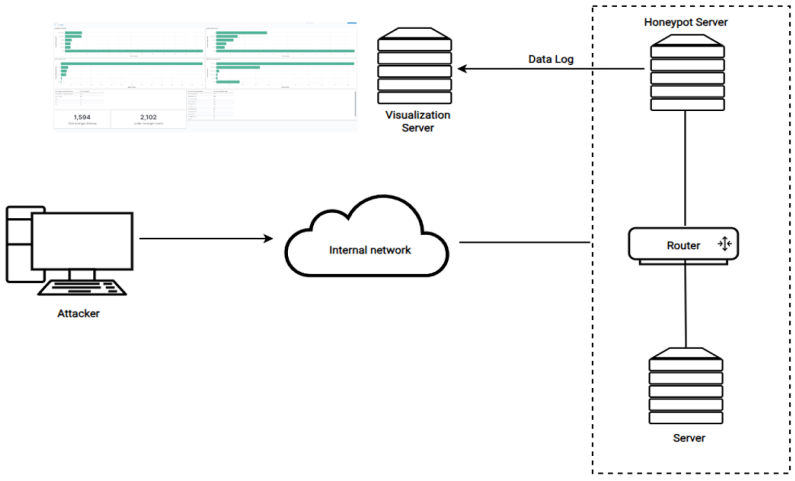

Fig. 2 Honeypot topology and data visualization

In Fig. 2, an illustration shows incoming attacks targeting the main server within the internal network. However, using diversion techniques, these incoming attacks are directed to a honeypot server. The honeypot acts as if it is the original server but actually functions as a trap to monitor the attacker's activities without disturbing the actual server. This technique is highly effective in protecting the system from threats, as attackers are unaware that they are interacting with a honeypot rather than the main server.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.14144
e- ISSN : 2541-2019
p- ISSN : 2541-044X

Every activity detected by the honeypot is automatically logged in an attack log. This log data is then processed and displayed in graphical form to facilitate attack analysis. Additionally, written descriptions of each attack incident are provided for deeper explanations. This information is crucial for the cybersecurity team in understanding attack patterns, identifying types of threats, and taking appropriate mitigation steps to prevent future attacks.

RESULT

Running a Honeypot Server

On the honeypot server, the first honeypot that needs to be run is Cowrie. The Cowrie honeypot must be executed through the Cowrie user created during the honeypot installation. Cowrie cannot be run by any user other than the Cowrie user itself. The command to run the Cowrie honeypot can be seen as follows:

```
$ cowrie/bin/cowrie start
```

Fig.3 The command to run Cowrie

The next honeypot that needs to be run is the Dionaea honeypot. The command to run the Dionaea honeypot can be seen as follows:

```
# /opt/dionaea/bin/dionaea -l all,-debug -L '*'
```

Fig.4 The command to run Dionaea

Based on Fig. 4, this command will run Dionaea from the directory "/opt/dionaea/bin/." The "-l all" option is used to display all logging activities from Dionaea, "-debug" is the option to disable debug logging, and "-L '*'" is the option to write all data results during an attack into the Dionaea log file.

After both honeypots are running, the next application that needs to be started is Filebeat. To run the Filebeat application, navigate to the Filebeat directory, and then use the following command to execute the application:

```
# cd /home/honeydev/filebeat/filebeat-7.17.19-linux-x86_64
# sudo ./filebeat -e
```

Fig. 5 The command to run Filebeat

Running the Monitoring Server

On the monitoring server, the ELK Stack applications are run in the following order: Elasticsearch, Kibana, and Logstash. The commands used to run these three applications are as follows:

```
# systemctl start elasticsearch
# systemctl start kibana
# systemctl start logstash
```

Fig. 6 The command to run ELK Stack

Cowrie Log Data

Cowrie is a type of honeypot specifically designed to mimic SSH (Secure Shell) services on a server. This honeypot falls into the medium interaction category and is commonly used to detect brute force attacks targeting SSH and Telnet services. Each SSH connection attempt from brute force attacks can be viewed in the log data "cowrie.json" located in the directory "/cowrie/var/log/cowrie." For more details, see Fig. 7

```
root@server1:/home/cowrie/cowrie/var/log/cowrie# ls -l
total 1968
-rw-rw-r-- 1 cowrie cowrie 772689 Sep 25 10:18 cowrie.json
-rw-rw-r-- 1 cowrie cowrie      0 Sep 11 15:13 cowrie.json.2024-09-11
-rw-rw-r-- 1 cowrie cowrie   5315 Sep 12 14:03 cowrie.json.2024-09-12
-rw-rw-r-- 1 cowrie cowrie 307672 Sep 23 16:48 cowrie.json.2024-09-23
-rw-rw-r-- 1 cowrie cowrie 598280 Sep 25 10:24 cowrie.log
-rw-rw-r-- 1 cowrie cowrie   1507 Sep 11 15:14 cowrie.log.2024-09-11
-rw-rw-r-- 1 cowrie cowrie   6511 Sep 12 14:11 cowrie.log.2024-09-12
-rw-rw-r-- 1 cowrie cowrie 290449 Sep 23 16:48 cowrie.log.2024-09-23
```

Fig. 7 Cowrie log data

The Cowrie log contains several important pieces of information, including the source IP, source port, timestamp, username and password used in brute force attempts, and various other details. A snippet of the log from Cowrie can be seen in Fig. 8

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.14144
e- ISSN : 2541-2019
p- ISSN : 2541-044X

Fig. 8 Snippet of the Cowrie log

Dionaea Log Data

Dionaea, as one type of honeypot, has the capability to detect and receive various types of attacks, including port scanning and denial of service (DoS). When an attacker performs a port scan, Dionaea can record the activity by collecting information about the accessed ports and the attacker's interaction patterns. Additionally, this honeypot can also receive denial of service attacks, where the attack aims to flood the network or system with excessive traffic to cause disruption or service failure.

The logs from the Dionaea honeypot contain various important pieces of information useful for attack analysis. Among this information are timestamps, indicating when the attack occurred, the source IP of the attacker, identifying the origin of the attack, and the ports that were targeted. A snippet of the log from Dionaea can be seen in Fig. 9.

Fig. 9 Snippet of the Dionaea log

Data Visualization ELK Stack

The visualization of log data from the ELK Stack is performed on a web browser of the host by entering the IP address and port of Kibana from the monitoring server. Fig. 10 shows the initial appearance of the ELK Stack when accessed for the first time through a web browser.

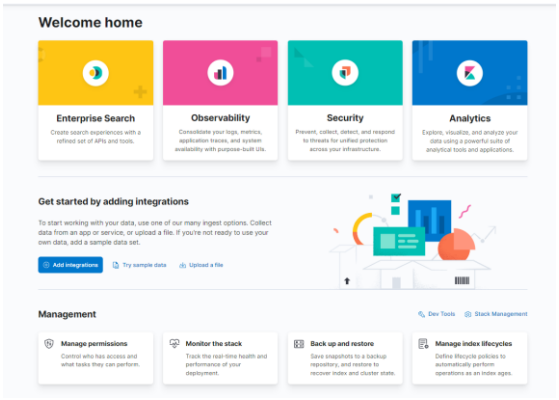

Fig. 10 Initial appearance of the ELK Stack

Several configurations need to be set up before visualizing data in Kibana, including creating index patterns for the Cowrie and Dionaea indices. An index pattern is a feature used to select the data index that will be used for visualization. The index pattern ensures that the visualization results display only the selected data based on the name without affecting other data indices. Figure 11 shows the details of creating the index pattern for the Dionaea log data. The "Timestamp Field" section is the configuration used to filter data based on time.

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.14144
e- ISSN : 2541-2019
p- ISSN : 2541-044X

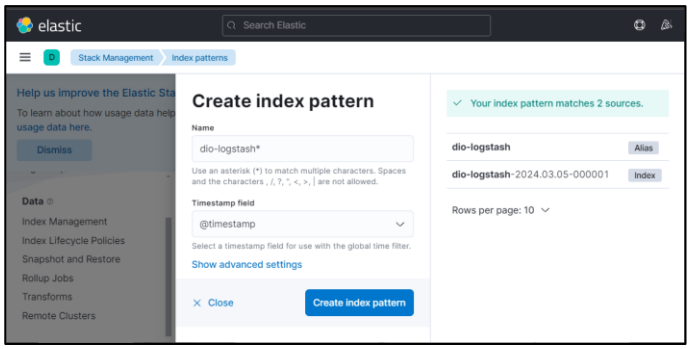

Fig. 11 Creating Index Patterns in Kibana

The index patterns created for Dionaea and Cowrie will map the fields that can be used as references for visualization. These fields can be directly utilized on the dashboard when creating data visualizations.

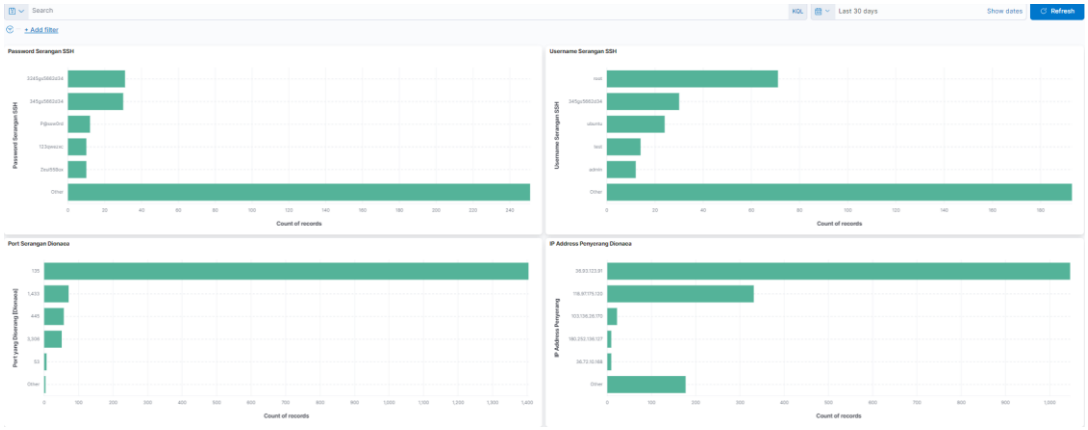

Fig. 12 Cowrie and Dionaea log data graph

Fig. 12 displays several graphs illustrating the results of attack data analysis on a honeypot. The first graph in the upper left corner shows the statistics of the Most Common SSH Passwords, where the passwords most frequently used in brute force attack attempts are clearly visible with high frequencies, particularly for several common passwords that attackers often use. Meanwhile, the graph on the upper right displays the Most Common SSH Usernames, indicating that attacks often target accounts with easily guessable usernames, such as 'root' and 'admin', with a significant number of attacks.

At the bottom of the figure, the left graph shows the Most Targeted Ports, where certain ports are more frequently attacked, especially port 22, which is used for SSH services. On the right, the final graph illustrates the Most Active Attacker IP Addresses, indicating some of the most active attacker IPs in the network. Overall, this data provides a comprehensive overview of the attack patterns conducted via SSH services, including brute force methods, the most frequently targeted ports, and the origins of the attacking IPs.

Fig. 13 shows the attack statistics detected by the Dionaea and Cowrie honeypots. In the upper left corner, the Top 5 Commands in Cowrie SSH table displays the five most frequently attempted commands by attackers through the SSH service, such as `cd ~; chattr -ia .ssh; lockr -ia .ssh`, with the highest number of attempts being 31 times. These actions are typically aimed at modifying access and taking control of the SSH directory on the attacked server. On the right, there is the Top 10 Cowrie Attacker IPs table, which records the IP addresses of the most active attackers, with IP `103.125.235.24` leading with 517 attack attempts, followed by IPs `193.70.87.152` and `217.107.34.149`.

Top 5 Sintaks dalam SSH Cowrie	Total Percobaan
cd ~; chattr -ia .ssh; lockr -ia .ssh	31
lockr -ia .ssh	31
test	10
00	2
22	2

Top 10 IP Penyerang Cowrie	Total Percobaan Serangan
103.125.235.24	517
193.70.87.152	338
217.107.34.149	320
42.51.32.210	278
78.187.21.105	262
71.128.32.25	236
212.102.51.36	38
5.182.211.148	30
8.216.95.92	25
47.108.76.140	20
Other	26

1,594
Total Serangan Dionaea

2,102
Jumlah Serangan Cowrie

Fig 13. Attack Statistics on Cowrie and Dionaea

*name of corresponding author

e- ISSN : 2541-2019
p- ISSN : 2541-044X

In Fig. 13, there are two important indicators. Total Attacks on Dionaea reached 1,594, indicating the number of attacks detected by the Dionaea honeypot, which typically includes various types of attacks such as port scanning and denial of service (DoS). Meanwhile, Total Attacks on Cowrie recorded a total of 2,102 attacks detected by the Cowrie honeypot, most of which were brute force attacks through the SSH service. This data provides a broad overview of the attack activities successfully identified by both honeypots.

DISCUSSIONS

The implementation of honeypots using Cowrie and Dionaea aims to divert cyberattacks from the main server to the honeypot server. Cowrie is specifically designed to detect attacks focused on the SSH protocol, particularly brute force attacks. Meanwhile, Dionaea serves to capture attacks related to port scanning and denial of service (DoS). Both honeypots work efficiently to capture and log malicious activity entering the network, allowing system administrators to understand the patterns and methods used by attackers.

The log data from the attacks captured by the honeypots is automatically sent to a monitoring server for further analysis. In this case, the ELK Stack system (Elasticsearch, Logstash, and Kibana) is used to visualize the data. The ELK Stack enables the security team to view attack trends in graphical form and provides detailed information about the origin of attacks, methods used, and targets attacked. With clear visualizations, the security team can more easily analyze attack data in real time and plan necessary mitigation steps.

Within minutes of operation, the honeypot server received over 1,000 incoming attacks through both Cowrie and Dionaea. Most of these attacks are indicated to originate from a botnet, where bots automatically attempt to exploit vulnerabilities within the system. Through data analysis visualized using the ELK Stack, the security team can delve deeper into the attack patterns, including the geographic origin of the attacks, frequency of attempts, and techniques used, thereby enabling more proactive and targeted preventive measures.

CONCLUSION

The conclusion of this study shows that the implementation of honeypots using Cowrie and Dionaea is effective in diverting and detecting cyberattacks. Cowrie successfully detected brute force attacks on the SSH protocol, while Dionaea efficiently captured port scanning and denial of service (DoS) attacks. The attack log data collected from both honeypots were then analyzed using the ELK Stack, which provided real-time data visualization and helped the security team understand attack patterns, the frequency of attempts, and techniques used, enabling more proactive and targeted preventive measures. Furthermore, the data visualization from the ELK Stack facilitates faster and more accurate decision-making in attack mitigation efforts and overall network security improvements.

ACKNOWLEDGMENT

This work was supported by the Indonesian Ministry of Education and Culture through the Research Grant PDP No. 110/E5/PG.02.00.PL/2024, 2927/LL8/AL.04/2024, 011/INSTIKI.R4.D1/PM.03/06.2024.

REFERENCES

Alzoubi, W. A., & Alrashdan, M. T. (2022). The effect of using honeypot network on system security. *International Journal of Data and Network Science*, *6*(4), 1413–1418. https://doi.org/10.5267/j.ijdns.2022.5.010

Amal, M. R., & Venkadesh, P. (2022). Review of cyber attack detection: Honeypot system. *Webology*, *19*(1), 5497–5514. https://doi.org/10.14704/WEB/V19I1/WEB19370

Gupta, C., Van Ede, T., & Continella, A. (2023). HoneyKube: Designing and Deploying a Microservices-based Web Honeypot. *Proceeding - 44th IEEE Symposium on Security and Privacy Workshops, SPW 2023*. https://doi.org/10.1109/SPW59333.2023.00005

Javadpour, A., Ja'fari, F., Taleb, T., Shojafar, M., & Benzaïd, C. (2024). A comprehensive survey on cyber deception techniques to improve honeypot performance. *Computers & Security*, *140*(1), 103792. https://doi.org/10.1016/j.cose.2024.103792

Matin, I. M. M., & Rahardjo, B. (2020). The Use of Honeypot in Machine Learning Based on Malware Detection: A Review. *2020 8th International Conference on Cyber and IT Service Management, CITSM 2020*. https://doi.org/10.1109/CITSM50537.2020.9268794

Mispriatin, M., Ginting, J. G. A., & Arifwidodo, B. (2022). Analisis Kinerja Honeypot Dionaea Dan Cowrie Dalam Mendeteksi Serangan. *Prosiding Seminar Nasional Teknoka*, *6*, 170–178. https://doi.org/10.22236/teknoka.v6i1.448

Mondal, A., & Goswami, R. T. (2021). Enhanced Honeypot cryptographic scheme and privacy preservation for an effective prediction in cloud security. *Microprocessors and Microsystems*, *81*(1), 103719. https://doi.org/10.1016/j.micpro.2020.103719

Natanegara, T., Muhyidin, Y., & Singasatia, D. (2023). IMPLEMENTASI HONEYPOT COWRIE DAN SNORT SEBAGAI ALAT DETEKSI SERANGAN PADA SERVER. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(3), 1871–1877. https://doi.org/10.36040/jati.v7i3.6989

Pratama, M. A., Setiawan, H., & Mair, Z. R. (2023). Implementasi Honeypot Sebagai Pendeteksi Serangan Pada

*name of corresponding author

Virtual Private Server (VPS). *Jurnal Software Engineering and Computational Intelligence*, *1*(1), 26–39. https://doi.org/10.36982/jseci.v1i1.3045

Purba, W. W., & Efendi, R. (2021). Perancangan dan analisis sistem keamanan jaringan komputer menggunakan SNORT. *AITI*, *17*(2), 143–158. https://doi.org/10.24246/aiti.v17i2.143-158

Sholihah, W., Pripambudi, S., & Mardiyono, A. (2020). Log Event Management Server Menggunakan Elastic Search Logstash Kibana (ELK Stack). *JTIM : Jurnal Teknologi Informasi Dan Multimedia*, *2*(1), 12–20. https://doi.org/10.35746/jtim.v2i1.79

Stoleriu, R., Puncioiu, A., & Bica, I. (2021). Cyber Attacks Detection Using Open Source ELK Stack. *Proceedings of the 13th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2021*. https://doi.org/10.1109/ECAI52376.2021.9515120

Sun, Y., Tian, Z., Li, M., Su, S., Du, X., & Guizani, M. (2020). Honeypot identification in softwarized industrial cyber–physical systems. *IEEE Transactions on Industrial Informatics*, *17*(8), 5542–5551. https://doi.org/10.1109/TII.2020.3044576

Ubaidillah, U., Taryo, T., & Hindasyah, A. (2023). Analisis dan Implementasi Honeypot Honeyd Sebagai Low Interaction Terhadap Serangan Distributed Denial Of Service (DDOS) dan Malware. *JTIM : Jurnal Teknologi Informasi Dan Multimedia*, *5*(3), 208–217. https://doi.org/10.35746/jtim.v5i3.405

Wastumirad, A. W., & Darmawan, M. I. (2021). Implementasi Honeypot Menggunakan Dionaea Dan Kippo Sebagai Penunjang Keamanan Jaringan Komunikasi Komputer. *Jurnal Teknologi*, *9*(1), 80–91. https://doi.org/10.31479/jtek.v9i1.119

Wibawa, G. H. P., Sasmita, I. G. M. A., & Raharja, I. M. S. (2020). Analisis Data Log Honeypot Menggunakan Metode K-Means Clustering. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, *8*(1), 13–21. https://doi.org/10.24843/jim.2020.v08.i01.p02

Yang, X., Yuan, J., Yang, H., Kong, Y., Zhang, H., & Zhao, J. (2023). A Highly Interactive Honeypot-Based Approach to Network Threat Management. *Future Internet*, *15*(4), 127. https://doi.org/10.3390/fi15040127

Yudhistira, A., & Fitrisia, Y. (2023). MONITORING LOG SERVER DENGAN ELASTICSEARCH, LOGSTASH DAN KIBANA (ELK). *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, *8*(1), 124–134. https://doi.org/10.36341/rabit.v8i1.2975

Yudyanto, N., Syaifuddin, S., & Azhar, Y. (2020). Integrasi Modern Honey Network Dengan Grafana Untuk Visualisasi. *Jurnal Repositor*, *2*(10), 1380–1389. https://doi.org/10.22219/repositor.v2i10.1047

Yugitama, R., Kartika Rachman, P. P., & Sulistyo, S. (2020). EFISIENSI MONITORING HONEYPOT DENGAN MENGGUNAKAN VISUALISASI DAN OTOMATISASI LAPORAN LOG SERANGAN. *JURNAL IT*, *10*(3), 1–14. https://doi.org/10.37639/jti.v10i3.138

Zmaranda, D. R., Moisi, C. I., Győrödi, C. A., Győrödi, R. Ș., & Bandici, L. (2021). An analysis of the performance and configuration features of MySQL document store and elasticsearch as an alternative backend in a data replication solution. *Applied Sciences*, *11*(24), 11590. https://doi.org/10.3390/app112411590

*name of corresponding author