International Journal of Engineering, Technology and Natural Sciences
E-ISSN: 2685-3191 | P-ISSN: 2775-7706
Vol 5 No 2 (2023)

SIEM (Security Information Event Management) Model for Malware Attack Detection Using Suricata and Evebox

Hendra Setiawan [*, a,1] , Wiwin Sulistyo [b,2]
[a,b] Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Indonesia
[1] 672020602@student.uksw.edu*, [2] wiwinsulistyo@uksw.edu

Abstract

Malware or malicious software is software or program code specifically designed to damage software on a computer or perform malicious activities. Malware is spread over the internet and includes viruses and other forms of malware. Losses caused by malware can take the form of financial losses or disruptions to business processes. Prevention of malware attacks can be achieved by analyzing the malware to find out how it works and what its characteristics are. This information can be utilized to define an Indicator of Compromise (IOC), which is stored in a Cyber Threat Intelligence (CTI) system designed to be used as a source of information, such as the Intrusion Prevention System (IPS) Suricata. An Intrusion Detection System (IDS) can detect the presence of malware and can identify the same malware with the Signature Based Detection method. Furthermore, the database is stored by EveBox and organized to make it easier to read logs and alerts. All of these components are contained in the Security Information and Event Management (SIEM) model. The SIEM model can detect malware attacks based on their characteristics and store logs and alerts in real-time for deeper analysis by the Security Operations Center (SOC).

Keywords: Intrusion Prevention System, Evebox, Malware, SIEM, Suricata

I. INTRODUCTION

In the current era of digitalization, the internet has made it very easy for people in various fields. This is due to the convenience provided by the internet itself. We can see this convenience when we need information; by typing a keyword on an internet page, all the desired information can be presented from various sources. However, with this convenience, there are also negative effects, namely the presence of cybercrime [1]. The important thing to do is to anticipate the threat of unauthorized resource abuse. Cryptography, with data encryption and firewalls, is one example of a network security system that exists. Additionally, the network can be secured with the help of an Intrusion Detection System (IDS) and an Intrusion Prevention System (IPS) [2].

Therefore, the aspect of computer network security is one of the most important parts to ensure the integrity and validity of services for users. An attack on a computer web server can occur at any time. The first step in minimizing the occurrence of cyber attacks is the ability to detect attacks using Intrusion Detection Systems (IDS). Therefore, the accounting process must be able to have a real-time record of access performed on these objects [3].

Detecting the presence of malware can be done in several ways, such as by using the Security Information and Event Management (SIEM) model. SIEM is a monitoring system that can detect attacks or responses of a security system through log analysis of various events originating from various data sources of monitored web servers in real-time. It includes components such as Agents, Syslog, and Indicators of Compromise (IOC). IOC is a collection of attack information or cyber attacks written based on a specific format or language [4]. Information contained in an IOC can be obtained through malware analysis activities. Therefore, by utilizing the information in the IOC, SIEM can greatly assist organizations or companies in analyzing malware effectively, enabling better prevention and handling of cyber attacks, especially malware attacks [5].

Firewalls that utilize SIEM can detect and reject access that is suspected to be an attack (malware). Detection is based on network ports and protocols, as well as information contained in log files identified as true positive status. This study builds a SIEM model to have a network security system that is capable of detecting attacks based on signatures found in network traffic packets passing through. The SIEM has rules built using Suricata tools, which act as a scanner for suspicious traffic (Agent) and are then sent to Evebox, which acts as a central hub to collect, index, and process data and analyze security data to make decisions. Suricata is network software that can monitor logs and check every network node to verify if there is any suspicious traffic passing through. If any traffic is found passing through the web server, it will be dropped based on the rules created to prevent cyber

attacks. Suricata will act as a third-party supporting the firewall's ability to detect events in network traffic at all times. Meanwhile, Evebox serves as an event management system responsible for managing each event identified as an attack or not. Based on the test results, the SIEM model in this study is able to detect attacks that occur in real time, thus preventing malware attacks.

II. RESEARCH METHODS

SIEM (Security Information and Event Management) is a security system used to collect, analyze, and provide reports related to security incidents that occur on a network or system [6]. To carry out its functions effectively, SIEM uses a hierarchy consisting of several main components. The first of these is Security Information Management (SIM), which is responsible for collecting, storing, and analyzing security data from various sources, as well as providing related security information to monitor and address security threats that may occur on a network or system. The second component is Security Event Management (SEM), which analyzes data collected by SIM to identify significant security incidents that require prompt action.

To carry out its functions effectively, SIEM uses a hierarchy consisting of several main components, namely:

A. Collecting/Aggregating

At this level, SIEM collects data from various sources such as logs from operating systems, applications, networks, security sensors, and other devices. The collected data is processed and filtered before proceeding to the next stage.

B. Analyzing Signature Based

At this stage, SIEM analyzes the collected data and compares it with predetermined rules and policies. SIEM looks for signs of a threat or suspicious events and correlates events to identify complex attacks. SIEM also provides notifications to the administrator if necessary events occur.

C. Reporting/Alerting:

At this level, SIEM generates a report on security incidents that have occurred and notifies administrators of attacks or suspicious activity. These reports and notifications help administrators determine what actions to take to improve the security of their system.

D. Lesson

At this stage, conclusions are drawn from the results of the logs obtained by Suricata, which are collected and analyzed by Evebox. This analysis can be used to mitigate similar attacks if they occur.

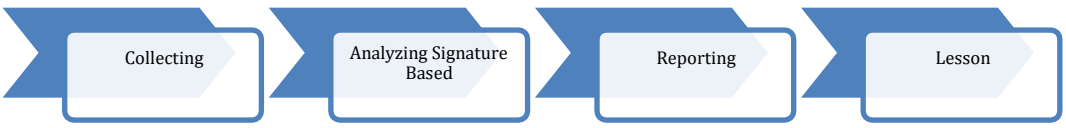

Figure 1. System hierarchy in SIEM

III. RESULT AND DISCUSSION

The SIEM model, designed in Figure 2, explains that when an attack occurs, SIM checks will be carried out based on rules made to detect an attack based on the attack's characteristics (signature) and types. Once the rules are detected, SEM will give a signal in the form of an alert/log that will be forwarded by SEM for log/alert management. The information will be presented in the form of a dashboard that shows where the attack occurred, the type of attack, and how to mitigate the attack. This information can help administrators determine how to respond to the attack and prevent similar attacks in the future. Additionally, MITRE ATT&CK (Adversary Technique Tactic & Common Knowledge) Framework will be used to provide knowledge about the techniques and tactics used by attackers to take over or damage a system. This information is listed in the Apply Mitigations Attack chart.

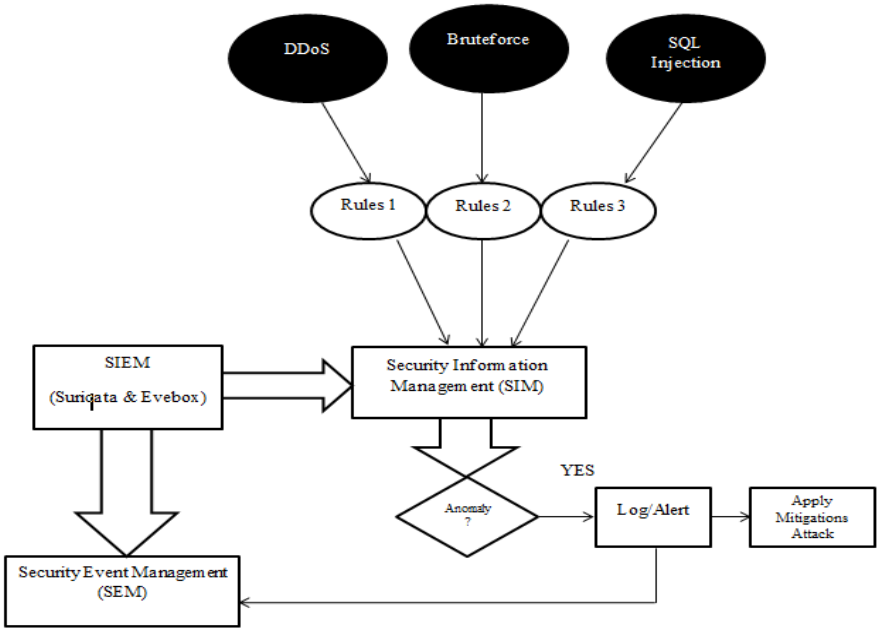

Figure 2. The built SIEM Model

Signature-based detection is a method used to identify the characteristics of network traffic and match them to known attack patterns. In SIEM, IPS/IDS are employed to safeguard real and client servers and the underlying networks used by Webservers. To create Suricata, packages and libraries are required, along with a package for Suricata rules that guide the operation of IPS/IDS according to the rules. These rules can be written in the form of scripts to detect ongoing intrusion attempts on the network and are crucial for Suricata. When established rules are adhered to, IPS can block packets using firewalls.

In Figure 3, a series of SIEM design topologies are depicted when an intrusion is detected outside the Webserver, through the router via the internet, Suricata performs a check. Then, traffic passing through IPS/IDS from the internet undergoes network packet inspection. When there is a conversation between the Internal Network and Webserver, a suspicious hash is identified, and every event detected by Suricata is recorded by Evebox, which functions as Event Management responsible for managing every event that occurs. Evebox can identify whether it is a pure attack (true positive) or suspected attack traffic (false positive) based on its ability to analyze the contents of alerts presented in detail in JSON (JavaScript Object Notation) format, making them easy to read.

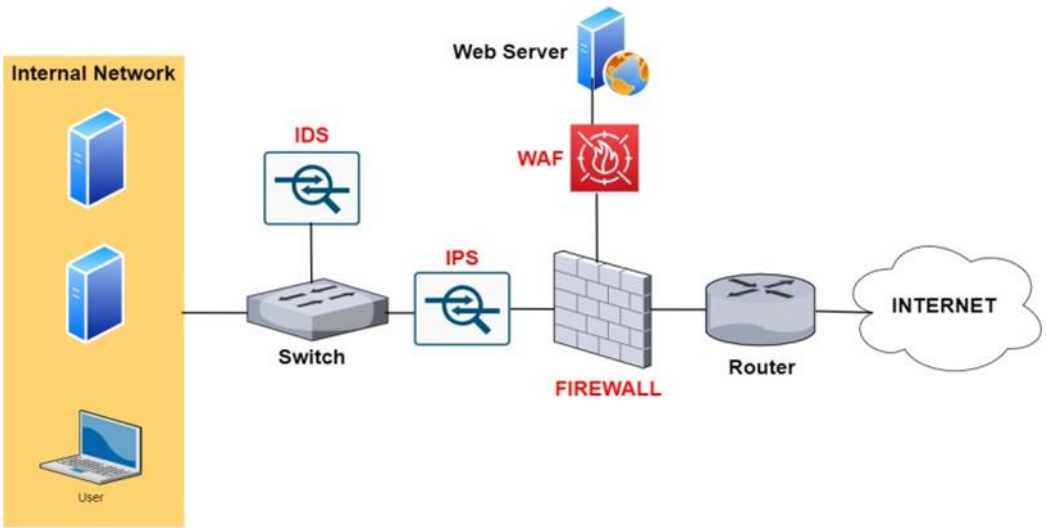

Figure 3. Network security topology using Suricata on Webserver

The structure of rules in Suricata includes the commands to be executed, which are explained in the rule header section. The criteria for matching rules with data packets are contained in the rule header, while warnings and instructions about packets that should generate an alert are included in the rule options. Suricata's ability to detect network-based threats is determined by the contents of the rule header section :

Table 1. Components and explanation of rules

Component	Example	Explanation
rule header	Alert ip ip an y any -> any any	Contains Actions to be taken, address, source and destination, and network traffic direction.
Rule option	(msg:"GPL ATTACK_RESPONSE ID CHECKRETURNED ROOT;…)	This includes the message to be displayed, details of the packet contents, warning type, source ID, and additional details such as references to rules or vulnerabilities.
Rule location	/nsm/server_data/securityonion/rules/,,,,	The location of the rule in the security file structure and the specified rule file is indicated by Suricata.

Table 2. Explanation of Header Rules

Rules Header	Penjelasan
Action	The type of action chosen will affect SIEM's response to the packet, such as whether SIEM will issue a warning (alert) or let the packet pass, or even block the packet (reject or drop).
Protocol	protocol describes the protocol that is seen by the rule.
Address	the first address describes the origin of the IP data packet and the second address describes the purpose of IP data packets.
Port	the first port describes the port the data packet comes from and the second port describes the purpose of the data packet port.
Direction	direction describes the destination of data packets.

After creating the ruleset, a scenario is created to launch the Cyber Kill Chain on the server to test the SIEM's ability to detect one of the stages of an attack. An alert is triggered by the IPS indicating the detection of a Trojan malware during the Weaponize stage and Exploitation stage before the malware establishes a Command and Control (CnC) connection and compromises the server.

During this stage, rules are created to enable Suricata to detect attacks. Program code 1 shows the identification rules for DDoS attacks by simulating some initial attacks on the SIEM system running on the Webserver. Specifically, it involves flooding the system with requests, commonly done by botnets, to find a vulnerable host. Once a vulnerable host is identified, Trojan-type malware is inserted in the many packets sent during the DDoS attack. This malware causes the attacker's remotely controlled host to initiate an attack on the target server, which results in an increase in traffic that can overwhelm the server. The rules for detecting DDoS attacks, which generate alerts in the form of the message "ET TROJAN BACKDOOR.Win32.Pushdo.s Checkin", are shown in program code image 1. These rules are used by Suricata to identify DDoS attacks involving trojan malware, and the alerts generated by Suricata are indexed by Evebox for analysis based on the malware's characteristics (signature).

Program Code

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN Backdoor.Win32.Pushdo.s Checkin";
flow:established,to_server; content:"POST"; http_method; content:"Mozilla/4.0 (compatible|3b
20|MSIE 6.0|3b 20|Windows NT 5.1|3b 20|SV1)"; http_user_agent; http_accept; content:"*/*";
http_accept_lang; content:"en-us"; http_content_type; content:"application/octet-stream";
http_header_names; content:"|0d 0a|Accept|0d 0a|Accept- Language|0d 0a|Content-Type|0d
0a|Content-Length|0d 0a|User-Agent|0d 0a|Host|0d 0a|"; content:!"Referer";
flowbits:set,ET.Pushdo.S; threshold: type threshold,track by_src,count 1,seconds 60; threshold:
type limit,track by_src,count 1,seconds 600; classtype:trojan-activity; sid:2016867;
```

Program Code 1. DDoS Attack Detection rules

To test the second attack namely Bruteforce, a method of attack carried out using a trial and error method to force login recursively, with the input length and symbol combination method to attempt to gain unauthorized access to the system illegally, rules are created to detect such attacks. In program code 2, the rules are designed to identify Bruteforce attacks by their characteristics (signature), resulting in an alert in the form of the message "ET ATTACK_RESPONSE FTP inaccessible directory access COM1" dropped by Suricata. This message indicates that an attacker is trying to gain illegal access through the FTP (File Transfer Protocol) protocol, which is often used to access each directory or data store that is directly connected to the Webserver.

```
Program Code
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ET ATTACK_RESPONSE FTP innaccessible directory access COM1"; flow:established,to_server; content:"POST"; http_method; content:"Mozilla/4.0 (compatible|3b 20|MSIE 6.0|3b 20|Windows NT 5.1|3b 20|SV1)"; http_user_agent; http_accept; content:"*/*"; http_accept_lang; content:"en-us"; http_content_type; content:"application/octet-stream"; http_header_names; content:"|0d 0a|Accept|0d 0a|Accept- Language|0d 0a|Content-Type|0d 0a|Content-Length|0d 0a|User-Agent|0d 0a|Host|0d 0a|"; content:!"Referer"; flowbits:set,ET.SQLI; threshold: type threshold,track by_src,count 1,seconds 60; threshold: type limit,track by_src,count 1,seconds 600; classtype:string-detect; sid:2000499;
```

Program code 2. Bruteforce attack detection rules

The last type of attack, known as SQL Injection, is a SQL command technique that uses manipulation of SQL command logic to gain access to databases and other critical data. This test will determine how well Suricata rules protect against various SQL Injection attacks. In program 3, rules are created to detect SQL Injection attacks by applying filters to metacharacters such as ("&, ;, `, ', , ", |, *, ?, ~," etc.) that are used in the syntax of a SQL query so that they cannot easily be turned into an instruction for accessing data by an attacker.

```
Program Code
Drop tcp any any -> IP Internal $HTTP_PORTS (msg:”SQL Injection Attack”;flow:established to server; content:”id; nocase; http_url;pcre:”/(and\W+select) | (union.*select); classtype:web-application-attack; sid:2000534
```

Program code 3. SQL Injection attack detection rules

In the testing process, the Suricata system that has been designed according to the created testing scenario will be tested to ensure its performance. When the testing scenario conditions are met, the response time parameter value is taken when the attack occurs to assess the security system's performance from a user's perspective in general. The performance of the SIEM system is also measured by how well the rules are used to handle various types of attacks carried out by the attacker, and it can also be seen from the response time parameter. Figure 4 shows the event detected by Suricata when responding to a network that sends packets continuously at a predetermined time.

```
Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=33800 chksum=0
--------------------------------------------------------------------------------
Count:1 Event#3.49724
ET TROJAN Backdoor.Win32.Pushdo.s Checkin
192.168.1.96 -> 72.4.145.131
IPVer=4 hlen=5 tos=0 dlen=983 ID=0 flags=0 offset=0 ttl=0 chksum=7058
Protocol: 6 sport=49928 -> dport=80

Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=8552 chksum=0
--------------------------------------------------------------------------------
Count:1 Event#3.49725
ET TROJAN Backdoor.Win32.Pushdo.s Checkin
192.168.1.96 -> 46.30.213.157
IPVer=4 hlen=5 tos=0 dlen=295 ID=0 flags=0 offset=0 ttl=0 chksum=62477
Protocol: 6 sport=49929 -> dport=80

Seq=0 Ack=0 Off=5 Res=0 Flags=******** Win=0 urp=29178 chksum=0
--------------------------------------------------------------------------------
Count:1 Event#3.49726
ET TROJAN Backdoor.Win32.Pushdo.s Checkin
192.168.1.96 -> 46.30.213.157
IPVer=4 hlen=5 tos=0 dlen=877 ID=0 flags=0 offset=0 ttl=0 chksum=61895
Protocol: 6 sport=49929 -> dport=80
```

Figure 4. Alerts triggered from the Suricata ruleset

When Suricata detects an attack and generates an alert based on the rules that have been created, Evebox manages these alerts so that they can be analyzed in terms of the attack's content on events that have been collectively stored based on the date, time, and type of attack that Suricata detected using Signature-based methods. On Evebox, as shown in Figure 5, there are three tables that contain a Timestamp which indicates when the attack occurred in real-time, followed by Source/Dest which is an abbreviation of Source (the IP address where the attack originated) and Destination (the IP address of the target), and Signature, which is a matching method used to detect attacks based on the characteristics recognized by Suricata and identify the category of the attack. Evebox then provides an indicator of the attack level based on the color displayed on the Evebox user interface, such as blue (low), yellow (medium), and red (high).

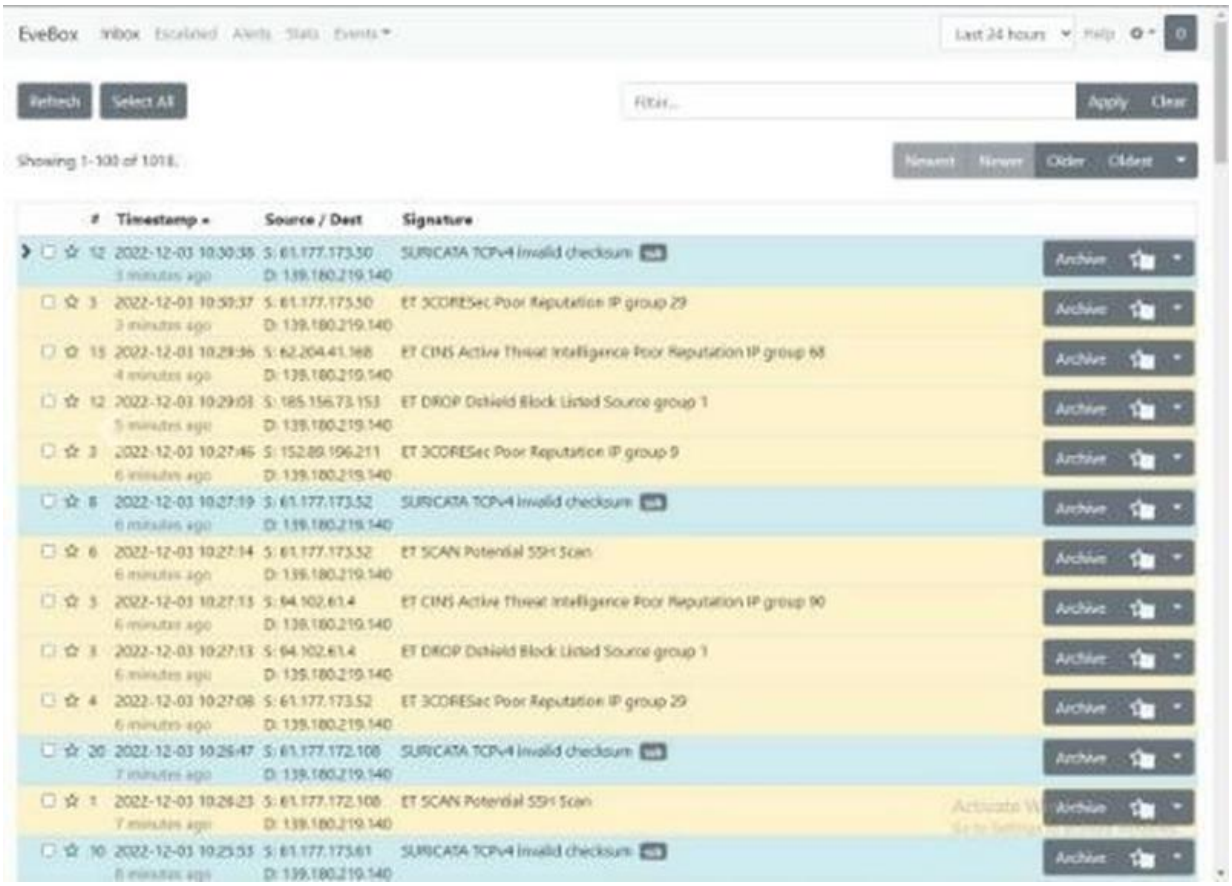

Figure 5. Evebox capture results when getting alerts

On figure 6, it shows the event management results from evebox which have been collected from December 1-2, 2022, detecting a potentially "high" severity attack marked in red, which is an attempted SQL Injection on the web server at 14:58 for 3 times. The result from Evebox, which is a website for detecting alerts and event management on Suricata detection, proves that the malware attempted an SQL Injection attack through the open port 3306 to gain illegal access.

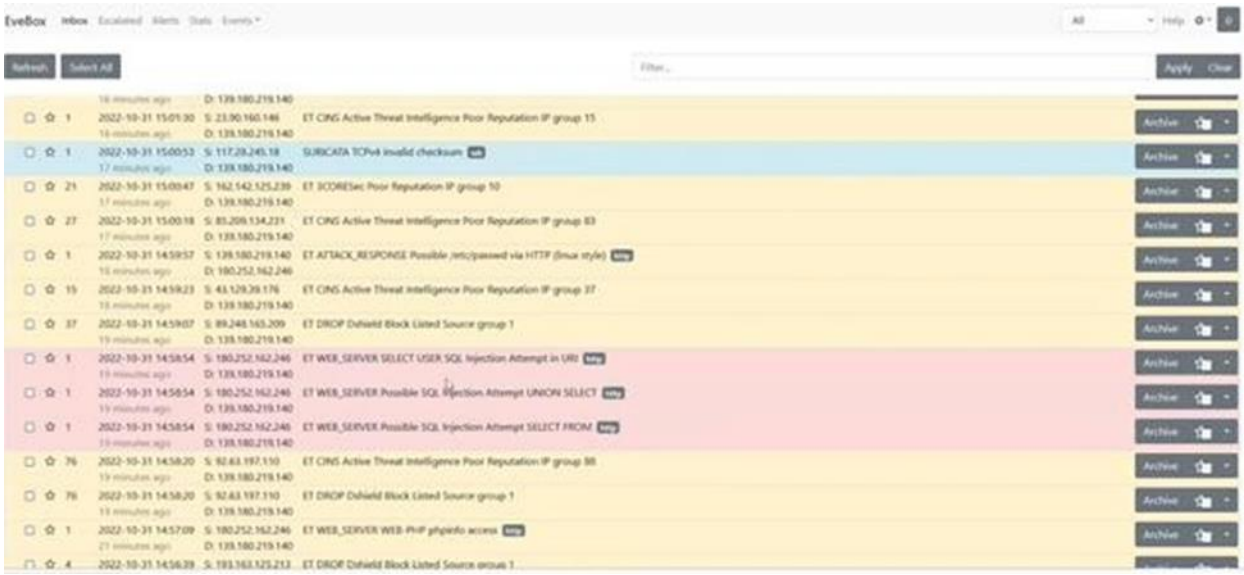

Figure 6. Display on Evebox

In one of the alerts that identified a Port Scanning Attack to lauch SQL Injection attack by Evebox, when the archive was opened, it was explained that on figure 7, on December 2, 2022 at 8:53 AM from source IP xx.xx.xxx.xx to the destination IP xxx.xxx.xx.140 through an unused port for inter-network communication, which is 3306 with TCP/IP protocol to try to gain access to the internal webserver. The attack was recognized based on its characteristic using Signature-based method. The result from Evebox, which is a website for alert detection and event management in Suricata detection, proved that the malware performed a Bruteforce attack through the open port 3306 to gain illegal access.

Figure 7. Display of evebox on SQL port Bruteforce attacks

Then in Figure 8. In the form of Bruteforce attack information presented in the form of a JSON script on evebox that tries to enter by force through port 22 SSH (secure shell) to gain access illegally. The Bruteforce attack was detected on December 3, 2022 at 03:30:37 by identifying its characteristics in the signature and id lines and it was known that the attack was from the IP source xx.xxx.xx.50 to the webserver IP xxx.xx.xxx.xxx .

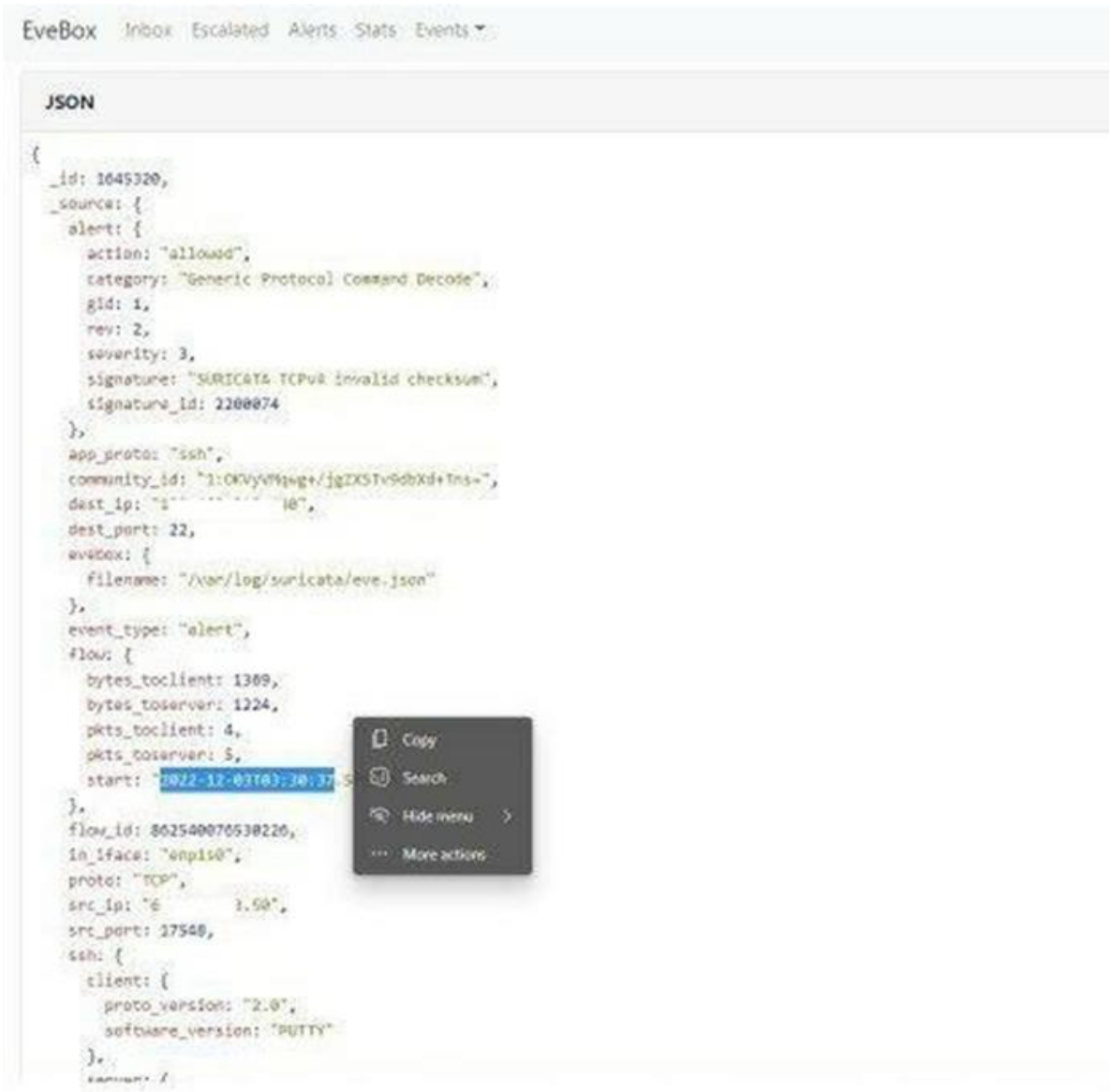

Figure 8. Analysis of Bruteforce attacks on SSH ports

The last attack, which flooded packet data containing several types of malware, was identified through one of the alerts in Table 3. One of the logs was taken, and its hash value was found to be for the push.do Trojan malware with SHA 256 and SHA1 values. VirusTotal provided information that the malware file is an executable with a size of 836kb and is highly dangerous since it can execute attacks once it has infiltrated the web server, without waiting for the attacker to activate it, which can compromise network security. Figure 9 displays the result of the hash value analysis on VirusTotal, which indicates that out of the 67 antivirus programs, 50 classified the sample malware as a Trojan and thus highly dangerous.

Table 3. Hash value detection results from Pushdo Trojan

Hash Value	Type Hash
73a81543fc49c5e6e66814da0e9a493db1b3ee028dd52fb2c88695b0160821c5	SHA 256
1597afd25aef1f8cda67355aa3bfd06b06ef3440	SHA 1

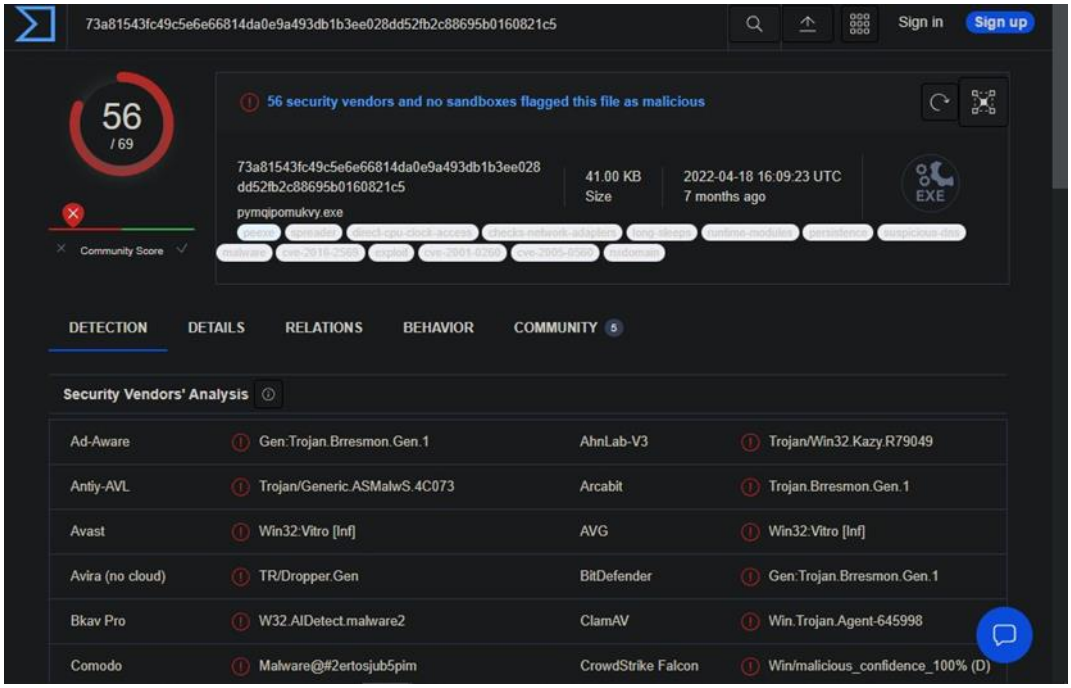

Figure 9. Analysis of malware in virustotal

The second result is to ensure that the SHA 256 hash value 73a81543fc49c5e6e66814da0e9a493db1b3ee028dd52fb2c88695b0160821c5, through Hybrid Analysis, proves that submitting the same value provides information that the 41kb executable file is categorized as Trojan.Bresmon.Generic malware with a Malicious label or danger level of 87%.

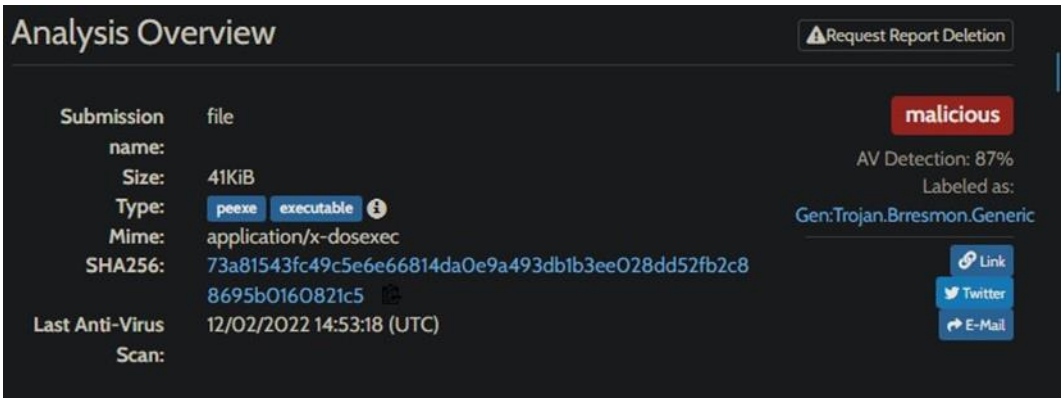

Figure 10. Analysis of types of hybrid Malware analysis

IV. CONCLUSION

Overall, the SIEM model designed for malware attack detection using Suricata and Evebox is capable of detecting various types of cyber attacks that are commonly used to gain unauthorized access. The model provides conclusions in the form of IOC (Indicator of Compromise), which contains information on the attacks that have been logged, including the time the attack occurred in real-time. By detecting the attack characteristics (signature-based) based on predefined rules, the model can prevent malware attacks from occurring and provide initial mitigation when the same malware attack occurs in the future. In summary, the SIEM model can effectively detect and respond to malware attacks, thereby enhancing the security of an organization's network.

V. SUGGESTIONS AND OPINIONS

It would be nice for further research to use a combination of Behavior Based methods so that SIEM can identify unusual attacks and add combinations of rules so that the alert results obtained are more likely to be True Positive and more sensitive to more dangerous combinations of attacks and improve the rules for effectiveness in analyzing results obtained.

VI. ACKNOWLEDGEMENT

The author would like to express their gratitude to Mrs. Desti Mualfah S.Kom., M.Kom., the mentor of the Talent Scouting Academy internship program.

REFERENCES

[1] Ramli, M., & Soewito, B. (2023). Monitoring dan Evaluasi Keamanan Jaringan Dengan Pendekatan System Information and Security Management (SIEM). Faktor Exacta, 16(1).

[2] Anugrah, F. T., Ikhwan, S., & AG, J. G. (2022). Implementasi Intrusion Prevention System (IPS) Menggunakan Suricata Untuk Serangan SQL Injection. Techné: Jurnal Ilmiah Elektroteknika, 21(2), 199-210.

[3] Mualfah, D., & Riadi, I. (2017). Network forensics for detecting flooding attack on web server. International Journal of Computer Science and Information Security, 15(2), 326.

[4] Widiasari, I. R. (2022). "Siasat" Uksw (Universitas Kristen Satya Wacana) Website Security Analysis Using Owasp (Open Web Application Security Project). Jurnal Teknik Informatika (Jutif), 3(3), 763-770.

[5] M. Syani, "Implementasi Intrusion Detection System (IDS) Menggunakan Suricata Pada Linux Debian 9 Berbasis Cloud Virtual Private Servers (VPS)," Jurnal Infokar, vol. 1, no. 1, pp. 13-20, 2020.

[6] B. S. Anggoro and W. Sulistyo, "Implementasi Intrusion Prevention System Suricata dengan Anomaly-Based untuk Keamanan Jaringan PT. Grahamedia Informasi," Seminar Nasional APTIKOM (SEMNASTIK), pp. 280-288, 201

[7] Pamungkas, M. H., & Chandra, D. W. (2022). Analisis Pola dan Dampak Serangan Cryptojacking dengan Menggunakan Metode Analisis Dinamis dan Analisis Statis. JURIKOM (Jurnal Riset Komputer), 9(5), 1511-1519.

[8] Sopaheluwakan, C. R., & Chandra, D. W. (2020). Anti-WebShell PHP Backdoor Scanner pada Linux Server. ILKOM Jurnal Ilmiah, 12(2), 143-153.

[9] Tallane, R. B., & Chandra, D. W. (2022). Implementation Of Intrusion Detection System (Ids) Using Security Onion. Syntax Literate; Jurnal Ilmiah Indonesia, 7(10), 14947-14959.

[10] Aulianita, R., & Martiwi, R. (2021). PENGGUNAAN METODE IDS DALAM IMPLEMENTASI FIREWALL UNTUK PENCEGAHAN SERANGAN Distributed Denial Of Service (DDoS) PADA JARINGAN. Jusikom: Jurnal Sistem Komputer Musirawas, 6(2), 94-104.