SIEM and Threat Intelligence: Protecting Applications with Wazuh and TheHive

Jumiaty, Benfano Soewito
Computer Science Department-BINUS Graduate Program-Master of Computer Science,
Bina Nusantara University, Jakarta 11480, Indonesia

***Abstract*—The consequences of cyberattacks on enterprises are highly varied. DDoS assaults can render an organization's website inaccessible; SQL attacks can compromise the integrity of data in a database, and Brute Force attacks can lead to unauthorized users gaining control over a server or application. Hence, it is crucial for enterprises to be aware of these potential dangers and employ solutions capable of monitoring networks, apps, and servers. In this study, the author employs Wazuh, TheHive, Telegram, and CVSS. Wazuh functions as a tool for monitoring applications and identifying potential security risks. TheHive classifies threats according to their level of importance. Telegram is utilized for dispatching notifications to the administrator. The findings indicate that Wazuh can promptly identify security risks by verifying that the date and time configurations on each utilized server align with the Indonesian time standard. Several vulnerabilities in the applications were successfully detected. The Wazuh server monitors two specific apps, namely Kompetensi and ESPPD. Surveillance commenced on March 20, 2024, at 17:49 and concluded on June 20, 2024, at 01:10, effectively amassing a total of 16,580 logs. 11 essential alert categories require follow-up due to their potential to compromise the system's integrity, confidentiality, and availability. To validate the detection results, the Common Vulnerability Scoring System (CVSS) is used. The assessment of vulnerability levels varies depending on the Wazuh level and CVSS. This arises because CVSS assigns scores based on five exploitability characteristics and incorporates the expertise of specialists to determine the assessment category and evaluate the potential impact of a successful threat. The outcome of this assessment, involving professional expertise, is heavily influenced by the unique attributes of each company. As a result, even when evaluating the same threats, the assessment can yield varying results. Evaluations utilizing Wazuh and CVSS are highly efficient in determining the extent of discovered hazards. By integrating these two technologies, the produced findings become more accurate.**

Keywords*—Application server security; application vulnerability; threat detection; SIEM; Wazuh; TheHive; Telegram and CVSS

I. Introduction

The exponential growth of information technology will inevitably lead to a corresponding rise in cyber-attacks. Cyberattacks are directed on individuals and government organizations and enterprises [1]. Distributed Denial of Service (DDoS), SQL Injection, and Brute Force are some of the several types of cyberattacks [2]. Cyberattacks have severe consequences, including financial losses, exposure of personal information, harm to reputation, disruption of operations, and the expenses required to manage the event [3]. To mitigate cyber attacks, enterprises are required to establish robust risk management protocols to safeguard applications from such threats. Some risk management frameworks that can be utilized are NIST Cybersecurity, ISO 27000, ISA/IEC 62443, GDPR, and CIS Controls [4].

Ensuring the security of applications is a crucial component of preserving the overall security of a system. Two tools commonly employed for application monitoring are Open Source Security (OSSEC) and Security Information and Event Management (SIEM). SIEM offers immediate log analysis and management, facilitating prompt identification of threats and expedited reaction to security issues. Utilizing these tools can enhance businesses' ability to identify and mitigate possible security threats with more efficiency [5][6].

SIEM includes two categories of tools: commercial tools and open-source tools. Typically, paid SIEM systems have advanced functionalities, but they come with a higher cost. Some examples of paid SIEM tools include Splunk and IBM Qradar [7]. Nevertheless, the functionalities of open source SIEM can be highly efficient, but need more configuration and manual upkeep. When effectively managed, these open source solutions can offer robust and adaptable security measures customized to the organization's requirements.

Some of the open source SIEM tools include Open-source SIEM (OSSIM), Elasticsearch-Logstash-Kibana (ELK) stack, and Wazuh [8]. Wazuh assists in safeguarding the digital assets of businesses and individuals against security threats. The primary elements of Wazuh consist of the Wazuh indexer, Wazuh server, Wazuh dashboard, and Wazuh agent. The Wazuh agent is installed in the target application that is to be monitored [9]. The Wazuh platform includes SIEM management, allowing for real-time monitoring and detection of incidents through the analysis of event or activity reports within the application [10]. This implementation can further enhance the usefulness of Indeks KAMI as a result of SIEM's capability to assess system vulnerabilities, facilitate monitoring and auditing in relevant work units. This implementation has the potential to increase the value of the Information Security Index (KAMI) by utilizing the capabilities of SIEM to evaluate system vulnerabilities and optimize monitoring and auditing procedures in relevant work units [11]. Furthermore, SIEM has the capability to be integrated with SOAR and Honeyport in order to safeguard crucial assets inside an organization [12].

This research aims to identify application security vulnerabilities within an organizational unit by utilizing

Wazuh, which is seamlessly linked with TheHive. Wazuh serves the purpose of monitoring security threats, whereas TheHive serves the purpose of responding to incidents. Based on the analysis results, it can be inferred that Wazuh has a total of 4,372 rules and 16 levels of vulnerability. These rules will inevitably generate a substantial volume of logs on a daily basis. Not all of these logs pertain to the identification of security threats in applications. Only pertinent rules will be employed, and any rules that are not utilized will be disabled.

The Wazuh server collects logs from agents that are installed on the target monitoring application. The amount of agents that can be assigned to a Wazuh server is flexible and can be adjusted based on the organization's requirements. Wazuh agents are compatible with multiple operating systems, including Windows, Linux, Mac, Solaris, AIX, and hpUX [13].

The Wazuh Dashboard will present logs according to the agent. Administrators have the ability to view comprehensive information on threats, including the detection level of each agent. This will provide challenges for administrators to monitor concurrently. So the optimal approach to facilitate administrators' application monitoring is to establish integration Wazuh and TheHive. TheHive will collect logs from Wazuh and provide them on a unified dashboard page for all agents.

Integrating Wazuh with TheHive is a challenging task because of the absence of a shared network between the Wazuh server and TheHive server. To integrate Wazuh and TheHive on the same network, the zerotier custom platform is required as an additional step. Nevertheless, there are many advantages to be gained from effectively combining two technologies to be concurrently utilized in resolving issues within organizational units. In their research, Muhammad Alfian Fahrudi dan I Made Suartana [14] performed a three-stage testing process to integrate Wazuh with Telegram. The stages included vulnerability evaluation, injection attacks, and brute force. The findings demonstrated that the integration of Wazuh with Telegram enables the identification of potential risks and their subsequent transmission to the administrator through the Telegram application. However, the author's predicted detection time is surpassed due to the substantial data kept on the server, resulting in a lengthy procedure lasting approximately 10 minutes. Another research conducted by Muhammad Dehan Pratama, Fitri Nova and Deddy Prayama [15] the integration of wazuh and suricata. Suricata is utilized for threat detection, whereas Wazuh is employed to showcase the logs produced by Suricata on the Wazuh dashboard. The detection process is specifically aimed at identifying denial-of-service (DoS) assaults inside the flood attack category. Out of the five attack attempts, only two were successfully identified by Suricata. This detection rate was influenced by the use of an AWS Amazon server. Some attacks were promptly rejected by the server, preventing Suricata from detecting them.

Therefore, this study aims to combine the Wazuh and TheHive methods to identify and address instances of application security threats. Real-time threat detection will be implemented by Wazuh. To guarantee the real-time detection of threats, it is necessary to configure the time zone on each server. The identified threats will thereafter be transmitted to TheHive according to the pre-established threat level. Based on the identified dangers, TheHive will generate a case, which will then be partitioned into multiple tasks. By examining the specifics of the identified risks, the duties will be subsequently assigned to multiple teams, including the network security team, vulnerability management team, incident response team, and database security team. Following the successful creation of the case, TheHive will promptly transmit a notification to the application administrator using Telegram. The output generated by Wazuh and TheHive Integration will go through validation using the Common Vulnerability Scoring System (CVSS) 4.0 in order to assess the genuine validity and significance of the identified threats by the system. By including Wazuh, TheHive, telegram, and validation using CVSS, it is anticipated that applications inside these organisational units will be adequately protected against cyber threats.

II. Literature Review

Research reviews are performed to validate the chosen technique and uncover areas of research that have not been explored, so opening up new possibilities for this study. Stefan Stanković, Slavko Gajin, and Ranko Petrović [16], did research on the application of Wazuh for identifying security threats. They specifically focused on using Wazuh to identify attacks on web servers. Web servers are highly susceptible to a wide range of threats. Wazuh will provide a comprehensive and real-time display of the detected attacks. Wazuh is utilized not only for detecting security threats, but also for monitoring integrity, policies, and auditing systems.

Rio Pradana Aji, Yudi Prayudi and Ahmad Luthfi [17] conducted an additional study on Wazuh. They utilized Wazuh to enhance the website monitoring system by employing quantitative forensic investigation techniques to identify brute-force attacks. Wazuh assists businesses in the implementation of Centralized Log Management. Based on the research review, the utilization of Wazuh is currently restricted to the detection of a single form of assault, specifically brute force.

A study undertaken by Manju, Shanmugasundaram Hariharan, M. Mahasree, Andraju Bhanu Prasad and H.Venkateswara Reddy [18] investigated the detection of DDOS assaults by the integration of four tools: wireshark, snort, Wazuh, and splunk. Wireshark is utilized to conduct preliminary surveillance through the analysis of network traffic. In addition, the integration of snort with Wazuh will effectively identify and detect potential security threats. The output is transmitted to Splunk and will be presented in a way that is readily comprehensible to the administrator. The duration of this procedure is around five hours, resulting in a higher level of efficiency compared to the prior duration of seven to eight hours, resulting in a time savings of approximately three hours.

Additional study was carried out by Novianda Shafira Suryawatie Yomo, Ahmas Zafrullah Mardiansyah, and I Wayan Agus Arimbawa [19] utilising Security Information and Event Management (SIEM) with the Wazuh system. An experiment was conducted to assess the security of the University of Mataram academic information system by evaluating the Sql Injection attack utilising the Sql Injection payload inputted into the Burp Suite application.

Anand Groenewegen and Joris Shuko Janssen [20] conducted an evaluation and validation of TheHive Project, an open-source security Incident Response Platform. The objective of this study is to assess the level of maturity of TheHive Project as a security Incident Response Platform. Due to its comprehensive documentation, ease of management, and effectiveness in managing security incidents, TheHive Project is regarded as a mature security Incident Response Platform.

A study undertaken by Bharadwaj Mantha, Yeojin Jung, and Borja Garcia de Soto [21] employed the Common Vulnerability Scoring System (CVSS) to evaluate and quantify cyber vulnerabilities inside the construction sector. The CVSS system assigns a numerical score to individual vulnerability characteristics, therefore enabling the quantification of the security risk level for project participants including owners, contractors, and labour. The susceptibility of numerous leading construction firms was methodically evaluated using CVSS version 3.1, employing criteria including base, temporal, and environmental factors.

Based on the findings of the literature research, there are several efficient frameworks for real-time detection of threats to servers or apps. One such tool is Wazuh, which possesses the capability to be seamlessly incorporated with numerous other tools. This information provides background for the author to undertake research on the integration of Wazuh with TheHive for the purpose of detecting security threats and implementing incident reactions in the case of such threats. This integration not only identifies a single kind of threat, but is anticipated to identify all categories of threats that provide a risk to applications and servers. This research also has the implementation of a Telegram Bot to transmit real-time notifications to administrators. The last phase of the research is the verification process utilising CVSS 4.0.

III. Methodology

A. Phases of Research

This project involves the installation and evaluation of integrating Wazuh with TheHive to detect security vulnerabilities on application servers. This procedure involves configuring Wazuh, TheHive, and integrating Wazuh with thehive, as shown in Fig. 1. Subsequently, it is necessary to incorporate rulesets based on the specific requirements of the research. Wazuh is designed to carry out real-time monitoring of logs. Logs that are identified will be shown on the Wazuh dashboard for the purpose of analysis. Logs categorized as vulnerability levels 5 to 15 will be given to TheHive for additional analysis. TheHive will generate a case for the identified threat and break down the case into multiple tasks to be collectively analyzed with the security team, enabling administrators to respond promptly and effectively to the issue.

B. Network Topology

Fig. 2 depicts the functioning of the system, wherein the administrator gains access to the competency application and e-sppd through a VPN connection, using a unique login and password. Both applications require the installation of a wazuh agent, which is responsible for monitoring the logs of application activities. The application is susceptible to cyber attacks. The installed Wazuh agent on the application will transmit logs to the Wazuh server. In addition, the Wazuh server collects logs from the agent and presents them on the dashboard for the administrator to watch. Logs that are being monitored and have high vulnerability ratings will be transmitted to TheHive Server for additional examination. In addition, TheHive will generate a case based on the log and then split it into several tasks. Once the case is created, the administrator will automatically be notified via telegram.

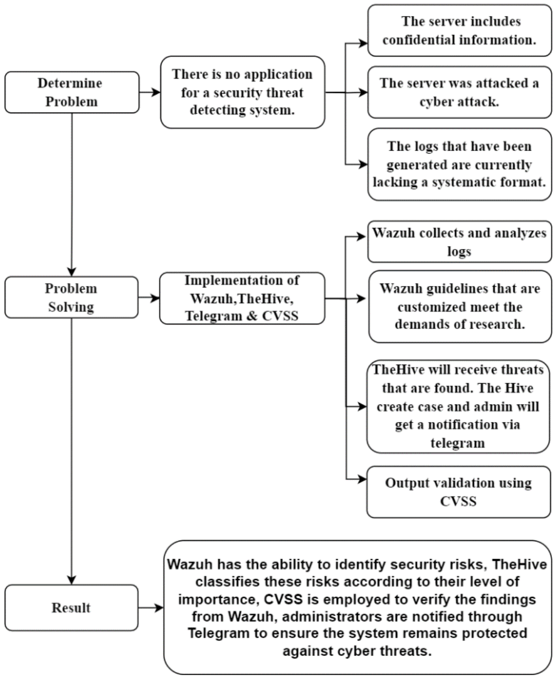

Determine Problem
There is no application for a security threat detecting system.
The server includes confidential information.
The server was attacked a cyber attack.
The logs that have been generated are currently lacking a systematic format.
Problem Solving
Implementation of Wazuh,TheHive, Telegram & CVSS
Wazuh collects and analyzes logs
Wazuh guidelines that are customized meet the demands of research.
TheHive will receive threats that are found. The Hive create case and admin will get a notification via telegram
Output validation using CVSS
Result
Wazuh has the ability to identify security risks, TheHive classifies these risks according to their level of importance, CVSS is employed to verify the findings from Wazuh, administrators are notified through Telegram to ensure the system remains protected against cyber threats.

Fig. 1. Phases of research.

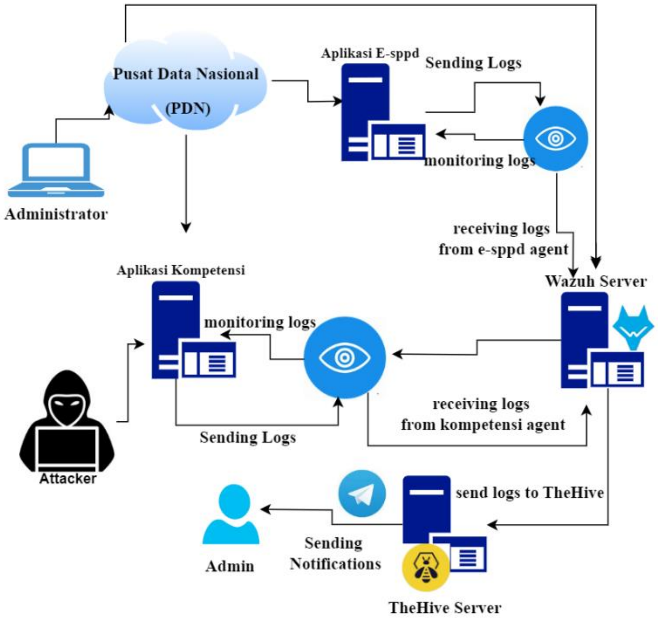


Fig. 2. Network topology.

C. Integration Flow of Wazuh and TheHive

Fig. 3 depicts the sequential steps involved in integrating Wazuh and TheHive. The initial step involves setting up the Wazuh server, which is then followed by configuring the thehive server. The subsequent step involves generating an integration file to facilitate the exchange of data between the two servers. Since Wazuh and TheHive are not connected to the same network, it is imperative to utilize zeroTier in order to establish a unified network for both. The integration file additionally establishes the threat level that will be transmitted to TheHive. TheHive will generate a case and partition it into multiple tasks to facilitate the analysis of the danger by administrators. The specifics of the Integration Flow are as follows: transmitted to TheHive. TheHive will generate a case and partition it into multiple tasks, hence facilitating the analysis of the threat for administrators.

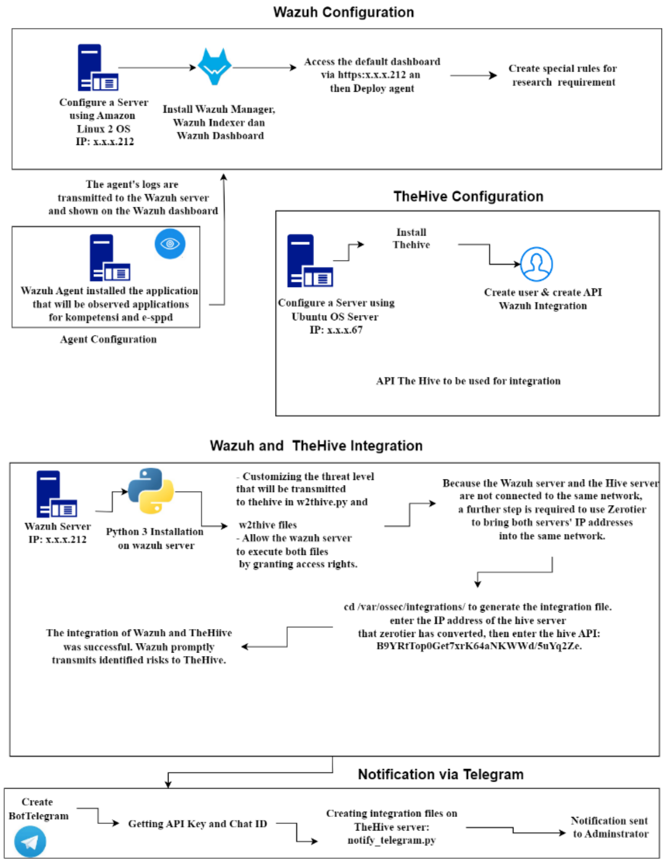

Wazuh Configuration
Configure a Server using Amazon Linux 2 OS IP: x.x.x.212
Install Wazuh Manager, Wazuh Indexer dan Wazuh Dashboard
Access the default dashboard via https:x.x.x.212 an then Deploy agent
Create special rules for research requirement
The agent's logs are transmitted to the Wazuh server and shown on the Wazuh dashboard
Wazuh Agent installed the application that will be observed applications for kompetensi and e-sppd
Agent Configuration
TheHive Configuration
Install Thehive
Configure a Server using Ubuntu OS Server IP: x.x.x.67
Create user & create API Wazuh Integration
API The Hive to be used for integration
Wazuh and TheHive Integration
Wazuh Server IP: x.x.x.212
Python 3 Installation on wazuh server
- Customizing the threat level that will be transmitted to thehive in w2thive.py and w2thive files
- Allow the wazuh server to execute both files by granting access rights.
Because the Wazuh server and the Hive server are not connected to the same network, a further step is required to use Zerotier to bring both servers' IP addresses into the same network.
cd /var/ossec/integrations/ to generate the integration file. enter the IP address of the hive server that zerotier has converted, then enter the hive API: B9YRtTop0Get7xrK64aNKWWd/5uYq2Ze.
The integration of Wazuh and TheHiive was successful. Wazuh promptly transmits identified risks to TheHive.
Notification via Telegram
Create BotTelegram
Getting API Key and Chat ID
Creating integration files on TheHive server: notify_telegram.py
Notification sent to Adminstrator

Fig. 3. System overview.

Based on Fig. 3, it can be concluded that the process details include:

- A server is configured with Amazon Linux OS 2 for the purpose of running Wazuh. This server consists of the Wazuh Manager, Wazuh Indexer, and Wazuh Dashboard. To access the Wazuh dashboard, use the designated IP address. Wazuh agents will be deployed on the application that needs to be monitored. The purpose of the agent is to transmit logs to the server, which will subsequently be exhibited on the Wazuh dashboard.
- The Ubuntu OS will host the installation of TheHive server. Once the installation is finished, the dashboard of TheHive can be accessed by using a designated IP address. First, add TheHive user to the system. Then, proceed with the creation of an API for the integration process.
- The integration between Wazuh and TheHive is accomplished using Python 3. The Wazuh server will have Python 3 installed.
- Wazuh and TheHive need to be connected within the same network utilizing zero tiers.
- The integration file will utilize the API provided by TheHive server and the IP generated by zero tier.
- Wazuh will send the logs to TheHive. The logs received by TheHives will be uploaded to the case, at the same time the administrator will receive detailed notifications of the threat.

IV. Result

A. Configuration and Modification of Wazuh Rules

System configuration and integration refer to Fig. 3. To implement Wazuh and TheHive, two servers are needed, namely the Wazuh server and TheHive server. The wazuh server will be installed on the Amazon Linux 2 operating system and TheHive server using ubuntu operating system. To start monitoring applications using agents, the Wazuh server needs to add as many agents as the number of applications to be monitored. The Wazuh server will monitor logs from two agents, the competency agent and the e-sppd agent. An illustration of the Wazuh server can be seen in Fig. 4.

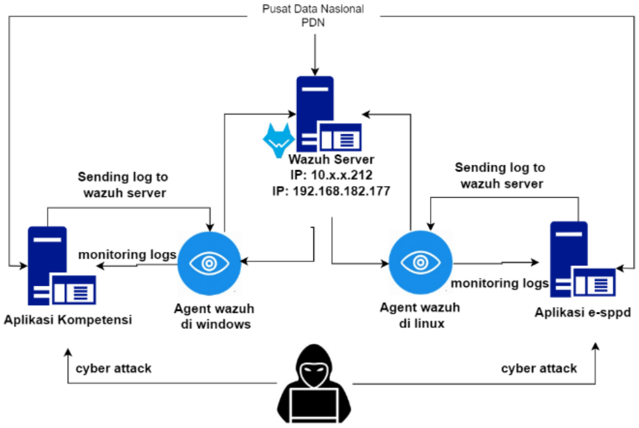


Fig. 4. Server Wazuh illustration.

Configuration stage of the wazuh server:

- Installation of Wazuh package on Amazon Linux server 2 : curl -sO https://packages.wazuh.com/4.4/wazuh-install.sh && sudo bash ./wazuh-install.sh -a
- Checking the Wazuh package that has been installed on the amazon linux server 2: sudo yum list-installed | grep Wazuh.

- After the installation process is complete, the Wazuh dashboard can be accessed using a dedicated ip via https://10.30.x.212.
- The next step is to add an agent. Add agent is done on the server side and the application side will be monitored. On the server side, add agent is done through the Wazuh dashboard then select deploy new agent and adjust the operating system used by the application that will be paired with the agent. The first agent is a competency application using the windows operating system, the agent installation stage from the windows-based application side can be done using windows powershell, the instructions are as follows:
    - Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.7.1-1.msi -OutFile ${env.tmp}\wazuh-agent; msiexec.exe /i ${env.tmp}\wazuh-agent /q WAZUH_MANAGER='10.30.x.212' WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='kompetensi' WAZUH_REGISTRATION_SERVER='10. 30.x.212'
    - NET START WazuhSvc

The second agent is the e-sppd application that uses the Linux operating system, the agent installation stage from the Linux-based application side can be done using the Linux terminal, the instructions are as follows:

- wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.1-1_amd64.deb && sudo WAZUH_MANAGER='10.30.x.212' WAZUH_AGENT_NAME='esppd' dpkg -i ./wazuh-agent_4.7.1-1_amd64.deb
- sudo systemctl daemon-reload
- sudo systemctl enable wazuh-agent
- sudo systemctl start wazuh-agent

Threats will be identified by the installed Wazuh agent using Wazuh rules. Wazuh's 16 rule classification tiers are organized according to how seriously systems and applications are threatened. These levels are numbered 0 through 15, with each level denoting a distinct degree of intensity. Every rule is intended to identify a range of potentially harmful or suspicious activity on a network or system, from very minor risks to more significant and destructive assaults. Wazuh gives administrators flexibility in responding to various threats by offering 16 levels of rule classification. This allows administrators to tailor actions based on the severity and criticality of each security event that is identified. Furthermore, users may more easily group and arrange security responses thanks to a clear hierarchy in rule classification, which increases handling efficiency for both threats and security incidents as a whole.

Wazuh has 4. 372 ID rules in addition to rule classification, which are grouped according to their functions: syslog, firewall, ids (intrusion detection system), web-log, squid (proxy server), Windows, Wazuh, sysmon (system monitor), powershell, cloudflare (web application firewall), audit detections, Amazon Security Lake, ms-graph (Microsoft Graph), multiverse, sshd (Secure Shell Daemon), fireeye, and unbound (DNS Server). These groups make it simple to arrange and sort rules according to the kind of log or activity being monitored.

In accordance with the requirements of research, Wazuh also makes it easier to add additional decoders and rules. Using the Wazuh dashboard, accessible via the Management menu, one may create custom rules by first selecting rules and then searching for "local_rules.xml" and the specific custom rule that one wants.

B. *Installation Stage of TheHive Server:*

- wget -q -O /tmp/install.sh https://archives.strangebee.com/scripts/install.sh ; sudo -v ; bash /tmp/install.sh. then select 2 which are Install TheHive.
- After the installation process is complete, Thehive dashboard can be accessed using a dedicated ip via http://10.10.x.67:9000/login.
- The next step is to add user mimi@thehive.local and create API Key B9YRtTop0Get7xrK64aNKWWd/5uYq2Ze which will be used in the integration process. after the integration process is complete, the user mimi@thehive.local will receive logs from the Wazuh.

C. *Integration of Wazuh and TheHive Servers*

The Integration Stages are:

- Installing python 3 on Wazuh server
- Install the Hive Python using PIP (Python Package Index)
- create custom-w2thive.py. and custom-w2thive file which will be used for Wazuh and TheHive integration.
- Customize the custom-w2thive.py and determine the logs that will be sent by Wazuh to TheHive are logs with at least rules level 5 to level 15.
- Continued by customizing the custom-w2thive file.
- Provide access to Wazuh to run the custom-w2thive.py file. and w2thive.:
    - sudo chmod 755 /var/ossec/integrations/custom-w2thive.py
    - sudo chmod 755 /var/ossec/integrations/custom-w2thive

  - sudo chown root:wazuh-user/var/ossec/integrations/custom-w2thive.py
  - sudo chown root:wazuh-user /var/ossec/integrations/custom-w2thive

- The Wazuh Server and TheHive Server are not connected to each other within the same network. An effective resolution is to employ zerotier. The procedure consists of the following steps:
  - Create an network name : mythesis_network
  - Create network_ID : 363c67c55a3f3ff1.
  - Select the format of the desired IP, namely: 192.168.x.x.
  - Install zerotier on both server: curl -s htttps://install.zerotier.com.
  - Connect thehive server with zerotier network ID, with the: zerotier-cli join 363c67c55a3f3ff1.
- After the Wazuh server and Wazuh thehive are in the same network and get the IP from the zero tier. The next step is to create integration files. Integration files can be created through the Wazuh dashboard on the Management menu and then select configuration. IP yang di input pada file merupakan IP dari zerotier. The API used is an API from TheHive server.
- The integration process is completed, TheHive will receive logs from any server starting from rule level 5 to rule Level 15.

D. Threat Detection Results Based on Logs

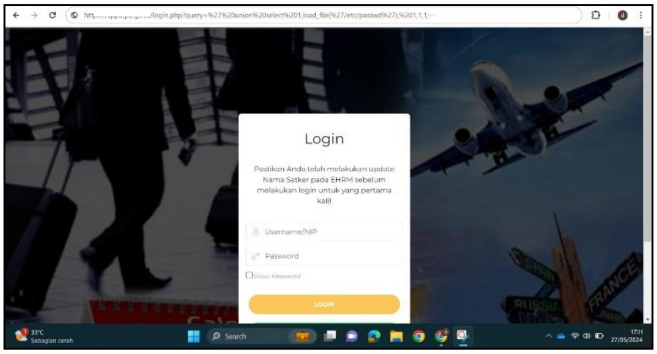
Based on the application threat detection results from the Wazuh server, there is a discrepancy in the timing between the attack attempt and the log generated by the server. This occurs due to a discrepancy in the date settings between the Wazuh server and the Wazuh dashboard. Presented in Fig. 5 are the unsynchronised date settings.

```
[root@wazuh-server wazuh-user]# date
Tue May 28 16:19:13 UTC 2024
[root@wazuh-server wazuh-user]# sudo hwclock --show
2024-05-28 18:18:20.045819+0000
```

Fig. 5. Unsynchronized time display.

The date instruction displays the date and time of the Wazuh server, while the sudo hwclock --show instruction displays the actual date and time. The following is a Sql Injection experiment by inputting the command : https://sppd.pu.go.id/login.php?query=%27%20union%20select%201,load_file(%27/etc/passwd%27),%201,1,1;-- on the web url that will be accessed.

The Sql Injection test in Fig. 6 was conducted on May 27, 2024, at 17:11 but Wazuh recorded the attack at 16:5516.55. Fig. 7 shows Sql injection detection.



htt[illegible]/login.php?query=%27%20union%20select%201,load_file(%27/etc/passwd%27),%201,1,1;--

Login

Pastikan Anda telah melakukan update Nama Satker pada EHRM sebelum melakukan login untuk yang pertama kali!

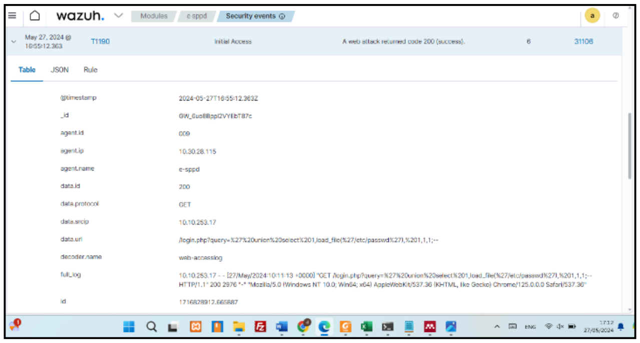
Username/NIP

Password

□ Show Password

LOGIN

Fig. 6. Sql injection testing.



May 27, 2024 @ 16:55:12.363 T1190 Initial Access A web attack returned code 200 (success). 6 31106

@timestamp	2024-05-27T16:55:12.363Z
_id	GW_6uoBBppl2VYEbT87c
agent.id	009
agent.ip	10.30.28.115
agent.name	e-sppd
data.id	200
data.protocol	GET
data.srcip	10.10.253.17
data.url	/login.php?query=%27%20union%20select%201,load_file(%27/etc/passwd%27),%201,1,1;--
decoder.name	web-accesslog
full_log	10.10.253.17 - - [27/May/2024:10:11:13 +0000] "GET /login.php?query=%27%20union%20select%201,load_file(%27/etc/passwd%27),%201,1,1;-- HTTP/1.1" 200 2976 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36"
id	1716828912.665887

Fig. 7. Sql injection detection.

The attack was carried out using IP 192.168.56.1 but the detection results were noted ip10.x.x.17. If viewed from the IP recorded by the Wazuh as if the attempted attack came from within the organization. This is because of the error in the organization's internal network configuration, this finding will be an input to the internal network improvement. The solution used for time synchronization on wazuh servers and wazuh dashboards is as follows:

- Install ntp: sudo yum install ntp -y
- Change ntp server: Sudo nano /etc/ntp.conf .
- Install ntpdate: sudo yum install ntpdate -y
- Synchronize system time with ntp server: sudo ntpdate -u pool.ntp.org
- If still not synchronized, verify the server time zone: timedatectl.
- Set the time zone to match the server location: sudo timedatectl set-timezone Asia/Jakarta
- Then re-synchroniz: sudo ntpdate -u pool.ntp.org

Furthermore, in addition to performing Sql Injection tests. The Wazuh server additionally monitors two specific apps, namely the competence application and the esppd application. The surveillance commenced on March 20, 2024, at 17:49 and concluded on June 20, 2024, at 01:10, effectively amassing a total of 16,580 logs. There are 11 essential alert categories that require follow-up due to their potential to disturb the system's integrity, confidentiality, and availability. Table I displays a summary of the 11 alert categories.

TABLE I. DETECTED THREATS

No	Alert	Level	Rule.mitre.technique	Amount
1	URL too long. Higher than allowed on most browsers. Possible attack.	13	Endpoint Denial of Service	2
2	Multiple web server 400 error codes from same source ip.	10	Vulnerability Scanning/Reconnaissance	5.388
3	High amount of POST requests in a small period of time (likely bot).	10	Network Denial of Service	857
4	Multiple web server 500 error code (Internal Error).	10	-	2
5	Multiple Windows logon failures.	10	Brute Force	11
6	SQL injection attempt.	7	Exploit Public-Facing Application	100
7	Listened ports status (netstat) changed (new port opened or closed).	7	Netstat Listening Ports	727
8	Host-based anomaly detection event (rootcheck).	7	-	171
9	Integrity checksum changed.	7	Stored Data Manipulation/ Impact	288
10	File deleted.	7	File Deletion/Data Destruction	38
11	Common web attack	6	Process Injection, File and Directory Discovery, Exploit Public-Facing Application	192

Referring to the information shown in Table I, Wazuh will identify a total of 11 risks. These threats will be thoroughly detailed, together with their respective impacts on servers and applications.

1) URL too long. Higher than allowed on most browsers. Possible attack, Identifies a potential attack in which an attacker can take advantage of a vulnerability by transmitting an excessively lengthy URL to the target. The significance of this danger on the system's confidentiality, integrity, and availability is substantial.

2) Multiple web server 400 error codes from same source ip executed across a network that does not necessitate specific circumstances or access privileges and does not need user engagement. Although server availability will be affected by this threat, data confidentiality and integrity will remain unaffected. Yet, these vulnerability screening operations can also serve as the initial stage of more severe attacks.

3) High amount of POST requests in a small period of time (likely bot). The process is executed over the network using POST requests that can be automated by bot programming languages. The present attack does not affect the principles of secrecy and integrity. Nevertheless, this attack will significantly affect availability by inundating the server with a substantial volume of requests, therefore introducing a denial of service (DoS) risk.

4) Multiple web server 500 error code (Internal Error).The exploitation of these vulnerabilities occurs via networks with modest complexity and does not necessitate privileges or user involvement. The aforementioned form of attack has minimal effect on the confidentiality and integrity of data, but significantly affects the availability of services.

5) Multiple Windows logon failures. These vulnerabilities are used on networks with modest complexity and do not necessitate any rights or user involvement. Upon successful execution of this Brute Force attack, the consequences for system confidentiality and integrity are significant, since the attacker gains the ability to view and alter sensitive data. Nevertheless, the effect on the availability of the system is really minimal.

6) The exploitation of Sql Injection attempts over the network can be achieved with minimal complexity and without the need for privileges or user involvement. Should the assault prove successful, the consequences for data confidentiality and integrity are much more pronounced, as the attacker gains access to and can alter sensitive data within the database. However, the effect on system availability is rather minimal, unless the attack results in a server breakdown or overload.

7) Listened ports status (netstat) changed (new port opened or closed). This vulnerability is leveraged across the network with minimal complexity and demands minimal rights without user involvement, rendering it insignificant to system availability, confidentiality, and integrity. Nevertheless, a modification in the state of a port can create new opportunities for attacks, hence enabling an attacker to gain unauthorized access or interfere with services operating on that specific port.

8) Host-based anomaly detection event (rootcheck). This vulnerability is leveraged via local access with minimal complexity and demands minimal rights without actual user involvement. NTFS Alternate Data Streams that are deemed suspicious have the potential to conceal harmful files or content that can be exploited by an attacker. While the effect on data integrity will be significant, the effect on system confidentiality and availability will be quite lesser.

9) Integrity checksum changed. The exploitation of these vulnerabilities occurs via local access with minimal complexity and necessitates elevated privileges without any user involvement. Alteration of system files, such as /usr/bin/cloud-init-per, can significantly affect the integrity and availability of the system, although the effect on system secrecy is rather little.

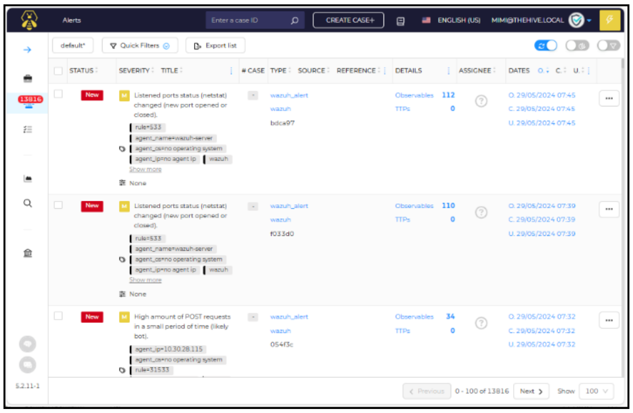
10) File deleted. This vulnerability through local access, this vulnerability demands high privileges without user interaction and is characterised by little complexity. Deleting certificate key files, such as /etc/ssl/pu_go_id/cert3.key, can significantly affect system integrity and availability by compromising the validity of certificates and causing disruptions to services that rely on such certificates. The effect on the confidentiality of the system is really minimal.

*11)*Common web attack. This vulnerability is exploited over the network with low complexity and without requiring privileges or user interaction. This attack attempts to access sensitive files such as /etc/passwd, which can have a high impact on system confidentiality and integrity if successful. However, the impact on system availability is relatively low.

The findings of this work demonstrate that the integration of wazuh with Thehive effectively detects a range of security concerns. Whereas research undertaken by Muhammad Dehan Pratama, Fitri Nova and Deddy Prayama [15] Wazuh and Suricata focuses on detecting a certain kind of threat, namely The detection procedure is deliberately designed to identify denial-of-service (DoS) attacks that fall within the flood attack category. Meanwhile, the research undertaken by Rio Pradana Aji, Yudi Prayudi and Ahmad Luthfi [17] Wazuh is limited to the identification of a singular type of attack, namely brute force.

E. *Threat Analysis*

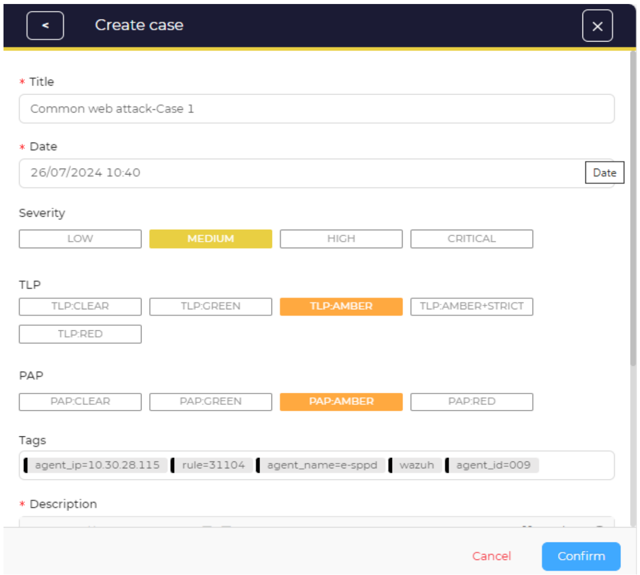
Wazuh effectively transmits the identified alert records to TheHive. In addition, the logs will undergo further analysis. Fig. 8 displays the Dashboard view of TheHive.



STATUS	SEVERITY	TITLE	# CASE	TYPE	SOURCE	REFERENCE	DETAILS		ASSIGNEE	DATES O. C. U.
New	M	Listened ports status (netstat) changed (new port opened or closed). rule=533 agent_name=wazuh-server agent_os=no operating system agent_ip=no agent ip wazuh Show more None	-	wazuh_alert	wazuh	bdca97	Observables TTPs	112 0	?	O. 29/05/2024 07:45 C. 29/05/2024 07:45 U. 29/05/2024 07:45
New	M	Listened ports status (netstat) changed (new port opened or closed). rule=533 agent_name=wazuh-server agent_os=no operating system agent_ip=no agent ip wazuh Show more None	-	wazuh_alert	wazuh	f033d0	Observables TTPs	110 0	?	O. 29/05/2024 07:39 C. 29/05/2024 07:39 U. 29/05/2024 07:39
New	M	High amount of POST requests in a small period of time (likely bot). agent_ip=10.30.28.115 agent_os=no operating system rule=31533	-	wazuh_alert	wazuh	054f3c	Observables TTPs	34 0	?	O. 29/05/2024 07:32 C. 29/05/2024 07:32 U. 29/05/2024 07:32

< Previous 0 - 100 of 13816 Next > Show 100

Fig. 8. Dashboard view of TheHive.



* Title
Common web attack-Case 1

* Date
26/07/2024 10:40

Severity
LOW | MEDIUM | HIGH | CRITICAL

TLP
TLP:CLEAR | TLP:GREEN | TLP:AMBER | TLP:AMBER+STRICT | TLP:RED

PAP
PAP:CLEAR | PAP:GREEN | PAP:AMBER | PAP:RED

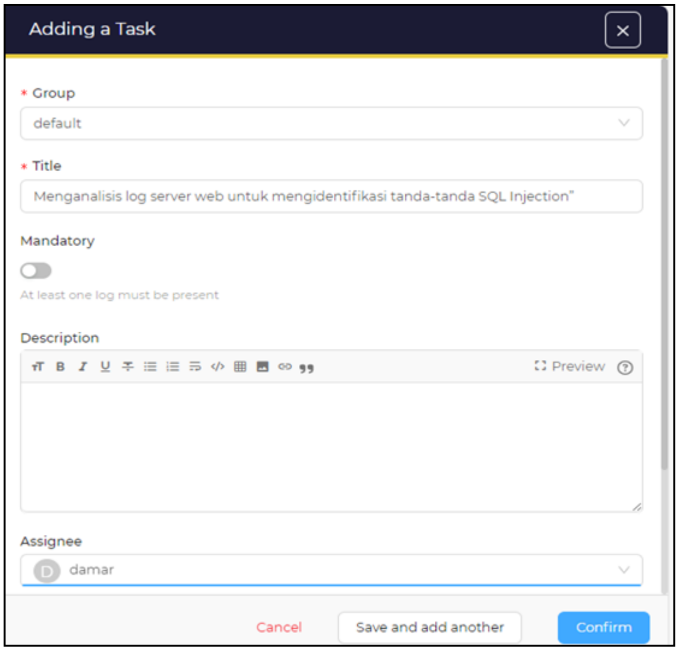
Tags
agent_ip=10.30.28.115 | rule=31104 | agent_name=e-sppd | wazuh | agent_id=009

* Description

Fig. 9. The process of creating a case.

TheHive will generate cases based on the collected logs and allocate them into tasks for each team. By dividing the responsibilities in this way, the analysis process can be expedited and enhanced, allowing for a prompter response to identified security threats. Fig. 9 displays the case view, where as Fig. 10 shows the tasks.

The generated case, including the additional tasks, will be assigned to many teams, such as the network security team, vulnerability management team, incident response team, and database security team, to jointly coordinate and carry out a comprehensive investigation of the risk. The generated case is depicted in Fig. 11. A successful case was set up on July 26, 2024 at 10:55 (WIB).



* Group
default

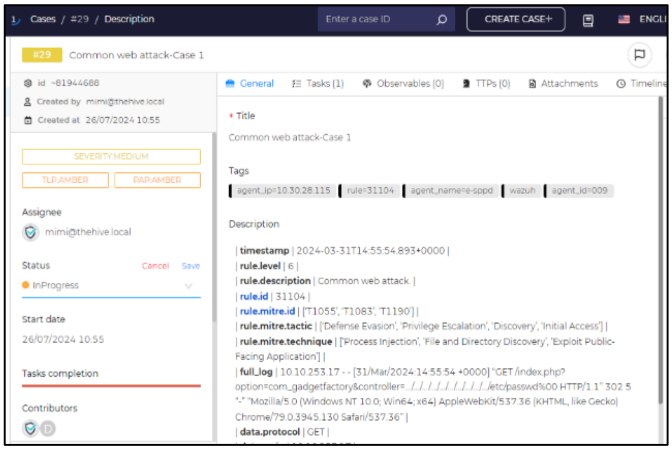
* Title
Menganalisis log server web untuk mengidentifikasi tanda-tanda SQL Injection"

Mandatory
At least one log must be present

Description

Assignee
damar

Fig. 10. The process of adding tasks for cases.



#29 Common web attack-Case 1

id ~81944688
Created by mimi@thehive.local
Created at 26/07/2024 10:55

SEVERITY:MEDIUM
TLP:AMBER PAP:AMBER

Assignee
mimi@thehive.local

Status
InProgress

Start date
26/07/2024 10:55

Tasks completion

Contributors

* Title
Common web attack-Case 1

Tags
agent_ip=10.30.28.115 | rule=31104 | agent_name=e-sppd | wazuh | agent_id=009

Description

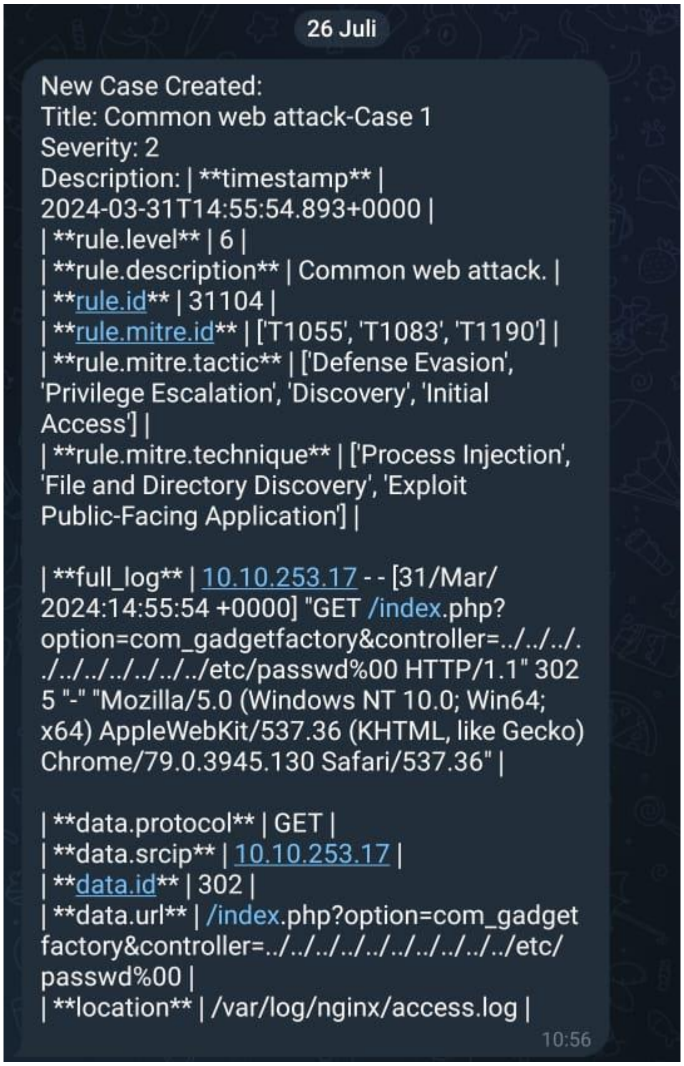
```
| timestamp | 2024-03-31T14:55:54.893+0000 |
| rule.level | 6 |
| rule.description | Common web attack. |
| rule.id | 31104 |
| rule.mitre.id | ['T1055', 'T1083', 'T1190'] |
| rule.mitre.tactic | ['Defense Evasion', 'Privilege Escalation', 'Discovery', 'Initial Access'] |
| rule.mitre.technique | ['Process Injection', 'File and Directory Discovery', 'Exploit Public-
Facing Application'] |
| full_log | 10.10.253.17 - - [31/Mar/2024:14:55:54 +0000] "GET /index.php?
option=com_gadgetfactory&controller=../../../../../../../../../../etc/passwd%00 HTTP/1.1" 302 5
"-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/79.0.3945.130 Safari/537.36" |
| data.protocol | GET |
```

Fig. 11. Cases that have been created.

Once the case and task have been established, the subsequent action involves notifying the administrator through the use of a telegraph. The following is a comprehensive breakdown of the steps:

- Develop a Telegram bot utilizing BotFather.
- Enter the command /start, followed by /newbot, and then provide a name for your bot, for as ThesisMimi_bot. Obtain the API Key: 7346165341:AAFqNsOpcqyg8vJ8ScMzYMQrWE8L33hpHb4.
- Enter the word "hello" in the chat with ThesisMimi_bot to obtain the Chat_ID. The result is as follows: {"ok":true,"result":[{"update_id":294625573,"message":{"message_id":2,"from":{"id":1134260586,"is_bot":false,"first_name":"Mimi","language_code":"id"},"chat":{"id":1134260586,"first_name":"Mimi","type":"private"},"date":1718511323,"text":"hello"}}]}
- The subsequent action involves generating an integration file on Thehive server, specifically named as the notify_telegram.py file.

An illustration of the notification received by the application administrator or server is shown in Fig. 12.



```
26 Juli

New Case Created:
Title: Common web attack-Case 1
Severity: 2
Description: | **timestamp** |
2024-03-31T14:55:54.893+0000 |
| **rule.level** | 6 |
| **rule.description** | Common web attack. |
| **rule.id** | 31104 |
| **rule.mitre.id** | ['T1055', 'T1083', 'T1190'] |
| **rule.mitre.tactic** | ['Defense Evasion',
'Privilege Escalation', 'Discovery', 'Initial
Access'] |
| **rule.mitre.technique** | ['Process Injection',
'File and Directory Discovery', 'Exploit
Public-Facing Application'] |

| **full_log** | 10.10.253.17 - - [31/Mar/
2024:14:55:54 +0000] "GET /index.php?
option=com_gadgetfactory&controller=../../../.
./../../../../../../etc/passwd%00 HTTP/1.1" 302
5 "-" "Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/79.0.3945.130 Safari/537.36" |

| **data.protocol** | GET |
| **data.srcip** | 10.10.253.17 |
| **data.id** | 302 |
| **data.url** | /index.php?option=com_gadget
factory&controller=../../../../../../../../../etc/
passwd%00 |
| **location** | /var/log/nginx/access.log |
                                          10:56
```

Fig. 12. Notification via telegram.

The analysis of Fig. 11 cases that were created on July 26, 2024 at 10:55 (WIB) and Fig. 12 Notification via telegram on July 26, 2024 at 10:56 (WIB) reveals that the time interval between case creation and notification reception by the administrator is a mere one minute. These results demonstrate that the system exhibits a rapid and effective reaction in identifying and issuing alerts to the administrator about possible risks or incidents of security identified.

Whereas research undertaken by Muhammad Alfian Fahrudi dan I Made Suartana [14] The findings demonstrated that the integration of Wazuh with Telegram enables the identification of potential risks and their subsequent transmission to the administrator through the Telegram application. However, this resulted in a lengthy procedure lasting approximately 10 minutes. Meanwhile, the research undertaken by Manju, Shanmugasundaram Hariharan, M. Mahasree, Andraju Bhanu Prasad, and H. Venkateswara Reddy [18] examined the identification of Distributed Denial of Service (DDoS) assaults by combining four tools: wireshark, snort, Wazuh, and spearhead. The duration of this operation was approximately five hours.

F. *Validation of Results using the Common Vulnerability Scoring System (CVSS)*

Exploitability metrics and impact metrics are the primary factors utilized to evaluate the severity of vulnerabilities. Exploitability metrics assess the technical components of vulnerability exploitation, including Attack Vector (AV), Attack Complexity (AC), Attack Requirements (AT), Privileges Required (PR), and User Interaction (UI). The complete description of the Exploitability metrics feature may be found in Table III Impact metrics are employed to evaluate the consequences that would occur if a vulnerability is successfully exploited. These metrics encompass the impact on the confidentiality, integrity, and availability of the vulnerable system. The effect metrics element will be comprehensively explained in Table IV. Table II shows qualitative severity rating scale.

The vulnerability assessment results using CVSS 4.0 will be summarised in Table VI, which includes the Exploitability Metrics from Table III and the Impact Metrics from Table IV. This table additionally incorporates the outcomes of danger detection determined by the rules and degrees of the Wazuh. Prior to that, it is necessary to categorise the 15 levels of risk into five CVSS categories in order to facilitate the comprehension of each identified threat. Table V displays a classification of 15 Wazuh levels into five CVSS categories.

TABLE II. QUALITATIVE SEVERITY RATING SCALE

Category	Score CVSS
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

TABLE III. EXPLOITABILITY METRICS

Metric Name	Metric Value	Description
Attack Vector (AV)	Network (N)	Vulnerabilities that can be exploited remotely over a network
	Adjacent (A)	The vulnerable system is inside the protocol stack, but the attack is limited to the protocol level on logically adjacent topologies.
	Local (L)	Attackers access vulnerable systems through local.
	Physical (P)	This attack requires the attacker to physically touch or manipulate the vulnerable system.
Attack Complexity (AC)	Low (L)	Attackers do not require specific targeting to take advantage of the vulnerability.
	High (H)	Attackers must have additional methods available to bypass existing security systems.
Attack Requirements (AT)	None (N)	Attack success is independent of the deployment and execution conditions of the vulnerable system or there are no attack requirements.
	Present (P)	The success of an attack depends on the conditions that require the preparation of specific targets that must be met in order to achieve vulnerability exploitation.
Privileges Required (PR)	None (N)	No privileges are required for an attacker to successfully exploit the vulnerability.
	Low (L)	Attackers need privileges, but can only access non-sensitive data
	High (H)	Attackers require privileges such as administrator who has full access to vulnerable systems.
User Interaction (UI)	None (N)	Vulnerable systems can be exploited without interaction from the user. remote attackers can send packets to the target system.
	Passive (P)	Successful exploitation of these vulnerabilities requires limited interaction by the targeted user with the vulnerable system. For example: utilizing a modified website to display malicious content when the page is rendered.
	Active (A)	Successful exploitation of these vulnerabilities requires the targeted user to perform specific interactions. For example: importing files into the vulnerable system in a specific way.

TABLE IV. IMPACT METRICS

Metric Name	Metric Value	Description
Confidentiality Impact to the Vulnerable System (VC)	High (H)	Confidential information/data is leaked. the information disclosed has immediate and serious consequences. the attacker has control of the information.
	Low (L)	The attacker has access to some information/data, but the attacker has no control over the information obtained. Leaked information has little impact
	None (N)	The confidentiality of the system is still maintained.
Integrity Impact to the Vulnerable System (VI)	High (H)	Complete loss of integrity. The attacker has full access to the data. Attackers can alter or delete data, leading to immediate and serious consequences on the system.
	Low (L)	Allows modification of data, but the attacker does not have full control over the data. The attacker does not have full control over the data, so it does not have an immediate and serious impact on the Vulnerable System.
	None (N)	System integrity is still maintained.
Availability Impact to the Vulnerable System (VA)	High (H)	Makes the service unavailable
	Low (L)	Performance degrades or interruptions in accessing the system occur. attackers do not have the ability to completely deny service to legitimate users
	None (N)	No impact on availability in Vulnerable Systems.

TABLE V. GROUPING OF 15 WAZUH LEVELS INTO FIVE CVSS CATEGORIES

Wazuh Level	CVSS Score	Description
Level 0	None (Score: 0.0)	It has nothing to do with security.
Level 2	Low (Score: 1.0 - 3.9)	Events that have only a minor impact on security but are not considered a significant threat.
Level 3		
Level 4		
Level 5		
Level 6	Medium (Score: 4.0 - 6.9)	Events that have a moderate impact on security could be a greater threat.
Level 7		
Level 8		
Level 9	High (Score: 7.0 - 8.9)	Events that have a major impact on security and can indicate an active attack or a serious problem requiring immediate action.
Level 10		
Level 11		
Level 12	Critical (Score: 9.0 - 10.0)	The events have had a huge impact on security. Indicates a serious attack to deal with.
Level 13		
Level 14		
Level 15		

A comparative graph of the two tools is created based on the data provided in Table VI. Fig. 12 depicts a comparison between the threat level provided by Wazuh and the CVSS score for different categories of security alerts. The graph displays two values for each sort of alert: the Wazuh level and the CVSS score.

TABLE VI. RECAPITULATION OF WAZUH LEVEL AND CVSS SCORE

No	Alert	Wazuh Level	CVSS Vector		CVSS Severity Level	
			Exploitability	Impact	Score	Descrip
1	URL too long. Higher than allowed on most browsers. Possible attack.	13	AV:N; AC:L; AT:N; PR:N; UI:A.	VC: H; VI: H; VA: H .	8.6	High
2	Multiple web server 400 error codes from same source ip.	10	AV:N; AC:L; AT:N; PR:N; UI:N.	VC:N; VI:N; VA:H.	8.8	High
3	High amount of POST requests in a small period of time (likely bot).	10	AV:N ; AC:L ; AT:N ; PR:N ; UI:N.	VC:N; VI:N; VA:H.	8.7	High
4	Multiple web server 500 error code (Internal Error).	10	AV:N; AC:L; AT:N; PR:N; UI:N.	VC:L; VI:L; VA:H .	8.8	High
5	Multiple Windows logon failures.	10	AV:N; AC:L; AT:N; PR:N; UI:N.	VC:H; VI:H; VA:L.	9.3	Critical
6	SQL injection attempt.	7	AV:N; AC:L; AT:N; PR:N; UI:N.	VC:H; VI:H; VA:L.	9.3	Critical
7	Listened ports status (netstat) changed (new port opened or closed).	7	AV:N; AC:L; AT:N; PR:L ; UI:N .	VC:L; VI:L; VA:L.	6.9	Medium
8	Host-based anomaly detection event (rootcheck).	7	AV:L; AC:L; AT:N; PR:L; UI:N.	VC:L; VI:H; VA:L.	6.9	Medium
9	Integrity checksum changed.	7	AV:L; AC:L; AT:N; PR:H; UI:N.	VC:L; VI:H; VA:H.	6.8	Medium
10	File deleted.	7	AV:L; AC:L; AT:N; PR:H; UI:L .	VC:L ; VI:H; VA:H.	6.8	Medium
11	Common web attack	6	AV:N; AC:L; AT:N; PR:N ; UI:N.	VC:H ; VI:H; VA:L .	9.3	Critical

Description:

- Attack Vector : AV
- Attack Complexity : AC
- Attack Requirements : AT
- Privileges Required : PR
- User Interaction : UI
- Confidentiality Impact : VC
- Integrity Impact : VI
- Availability Impact : VA

- Network : N
- for Attack Vector
- None : N
- for Attack Requirements, Privileges Required, User Interaction
- High : H
- Low : L

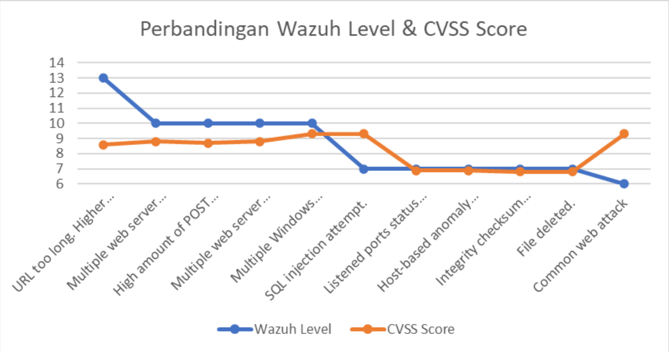


Fig. 13. Comparison chart of wazuh level and CVSS score.

Based on Fig. 13, it can be inferred that there are variations between the Wazuh level and CVSS Score, particularly in relation to the excessive length of the threat URL. Exceeds the maximum limit set by most browsers. There is a potential security breach involving multiple unsuccessful attempts to log in to Windows, an attempt to exploit SQL vulnerabilities, and a common type of attack targeting online applications. The disparity in vulnerability level assessment arises due to the utilisation of CVSS, which assigns a score based on five Exploitability characteristics. This process incorporates the expertise of professionals who determine the assessment category and take into account the potential impact in the event of a successful threat. The outcomes of this evaluation, which involves the expertise of professionals, vary based on the unique attributes of each organisation, although evaluating the same threats.

In general, evaluations utilising Wazuh and CVSS are highly efficient in identifying the severity of identified security risks. By integrating these two instruments, the acquired outcomes become more precise. The purpose of this assessment procedure is to systematically validate the hazards encountered, facilitating the implementation of appropriate mitigation measures. By integrating Wazuh with TheHive and utilising CVSS, organizations can enhance their ability to detect, assess, and address security issues, leading to improved application security.

V. CONCLUSION

Based on the conducted research, it can be inferred that:

1) Wazuh is utilised for the purpose of identifying and recognising potential threats, whilst TheHive is employed to scrutinise and assess the identified hazards. Each of these instruments is customised based on the specific study requirements. This integration also includes supplementary tools, including ZeroTier, which is used to establish a virtual network that links the two entities. Through this combination, Wazuh and TheHive effectively identify and respond to security issues in real-time, promptly notifying administrators via Telegram.

2) All servers should have their date and time settings synchronised to the Indonesian time zone. Synchronisation is crucial as it can impact the formatting of the detection result output in terms of date and time. By adjusting the time parameters appropriately, the resulting information will be more precise.

3) Deploying Wazuh and TheHive on distinct networks offers further advantages to the organisation. By having distinct server locations, the preservation of detection logs is secure, even in the event of one server experiencing a failure. Although integration may pose early challenges, it has the benefit of ensuring the security of log data in case of server issues.

4) Implement Secure Coding Practices, namely employing secure code to mitigate application vulnerabilities. The detection of threats in the form of multiple occurrences of web server 500 error code (Internal Error) twice, and web server 400 error code 2,444 times, suggests that system administrators need to exercise greater vigilance and caution while writing code for the system/application. This highlights a flaw in the code that has to be promptly addressed in order to avoid potential vulnerabilities that could be exploited by attackers.

5) Implement input validation to mitigate the risk of SQL injection, as well as enforce restrictions on the length of URLs received by the server to prevent potential buffer overflow or denial-of-service (DoS) attacks. This step is crucial in order to mitigate potential risks, such as encountering a URL that exceeds the maximum allowable length. Exceeds the maximum limit supported by most browsers. There is a potential attack and an attempt to do SQL injection.

6) Implementing Multi-Factor Authentication (MFA) to monitor user access and ensure authentication and authorisation. The discovered threats involved many instances of failed login attempts, which may suggest unauthorised access attempts or brute force attacks.

7) Assessments combining Wazuh and CVSS 4.0 are highly efficient in identifying the severity of identified threats. By integrating these two instruments, the outcomes achieved become more dependable and precise. The purpose of this assessment technique is to offer a concise and systematic evaluation of the encountered threats.

8) Possibly, TheHive might be integrated with network monitoring tools and intrusion detection systems like Suricata and Zeek (Bro) to enable the comparison of acquired findings for a more thorough study. The speed of technological advancement directly correlates with the increasing magnitude of cyber dangers. In order to enhance their preparedness, security teams should strive to incorporate tools that help streamline their tasks and provide robust security for servers and applications. Suricata and Zeek are mutually synergistic systems, with Suricata functioning as a signature-based detection and prevention system, Zeek providing behavior-based in-depth analysis, and TheHive serving as an incident management platform. The integration of the three components provides organisations with a more comprehensive threat detection system.

9) In the future, it is anticipated that all applications within the Organisational Unit would be able to incorporate wazuh and thehive for the purpose of monitoring servers and apps. This integration will facilitate administrators in monitoring the system, expediting preventive measures, and ensuring the maintenance of system security. This will enhance the efficacy of security management and offer superior safeguarding against prospective risks.

REFERENCES

[1] S. A. Utari, V. Ardia, Jamiati, and D. Fitria, “How an Organization Should Implement Risk Communication in Response to Cyber Attack in Indonesia,” J. Educ., vol. 05, no. 04, pp. 14314–14328, 2023, [Online]. Available: http://jonedu.org/index.php/joe

[2] N. Singh, “Sql i – a w,” vol. 2, no. 6, pp. 42–46, 2012.

[3] A. A. Putra, O. D. Nurhayati, and I. P. Windasari, “Perencanaan dan Implementasi Information Security Management System Menggunakan Framework ISO/IEC 20071,” J. Teknol. dan Sist. Komput., vol. 4, no. 1, p. 60, 2016, doi: 10.14710/jtsiskom.4.1.2016.60-66.

[4] V. Mahendra and B. Soewito, “Penerapan Kerangka Kerja NIST Cybersecurity dan CIS Controls sebagai Manajemen Risiko Keamanan Siber,” Techno.Com, vol. 22, no. 3, pp. 527–538, 2023, doi: 10.33633/tc.v22i3.8491.

[5] Ronal Hadi, Y. Yuliana, and H. A. Mooduto, “Deteksi Ancaman Keamanan Pada Server dan Jaringan Menggunakan OSSEC,” JITSI J. Ilm. Teknol. Sist. Inf., vol. 3, no. 1, pp. 8–15, 2022, doi: 10.30630/jitsi.3.1.58.

[6] M. Ramli et al., “Monitoring dan Evaluasi Keamanan Jaringan dengan Pendekatan Security Information and Security Management (SIEM),” vol. 16, no. 1, pp. 1979–276, 2023, doi: 10.30998/faktorexacta.v16i1.16534.

[7] S. S. Sekharan and K. Kandasamy, “Profiling SIEM tools and correlation engines for security analytics,” Proc. 2017 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2017, vol. 2018-Janua, pp. 717–721, 2017, doi: 10.1109/WiSPNET.2017.8299855.

[8] M. Sheeraz et al., “Effective Security Monitoring Using Efficient SIEM Architecture,” Human-centric Comput. Inf. Sci., vol. 13, p. 17, 2023.

[9] Wazuh, “Installing the Wazuh central components,” Wazuh. https://documentation.wazuh.com/current/installation-guide/index.html. (accessed May 23, 2024).

[10] N. F. Pratama, “Perancangan Sistem Deteksi Dini Keamanan Informasi DISKOMINFO Kabupaten Bandung,” JATISI (Jurnal Tek. Inform. dan Sist. …, vol. 10, no. 1, pp. 808–820, 2023.

[11] C. Arfanudin, B. Sugiantoro, and Y. Prayudi, “Analisis Serangan Router Dengan Security Information and Event Management Dan Implikasinya Pada Indeks Keamanan Informasi Analysis of Router Attack With Security Information and Event Management and Implications in Information Security Index,” CyberSecurity dan Forensik Digit., vol. 2, no. 1, pp. 2615–8442, 2019.

[12] M. Hafiz and B. Soewito, “Information Security Systems Design Using SIEM, SOAR and Honeypot,” J. Pendidik. Tambusai, vol. 6, no. 2, pp. 15913–15926, 2022.

[13] Wazuh, “Agen Wazuh.” https://documentation.wazuh.com/current/installation-guide/wazuh-agent/index.html#wazuh-agent (accessed May 23, 2024).

[14] M. A. Fahrudi and I. M. Suartana, “Integrasi End-point Security Berbasis Agent dan Bot Messenger untuk Deteksi dan Monitoring Serangan pada

Web Server secara Real-time," J. Informatics Comput. Sci., vol. 04, pp. 275–282, 2023, doi: 10.26740/jinacs.v4n03.p275-282.

[15] Fitri Nova, M. D. Pratama, and D. Prayama, "Wazuh sebagai Log Event Management dan Deteksi Celah Keamanan pada Server dari Serangan Dos," JITSI J. Ilm. Teknol. Sist. Inf., vol. 3, no. 1, pp. 1–7, 2022, doi: 10.30630/jitsi.3.1.59.

[16] Stefan Stanković, Slavko Gajin, and Ranko Petrović, "A Review of Wazuh Tool Capabilities for Detecting Attacks Based on Log Analysis," IX Int. Conf. IcETRAN, no. june, pp. 6–9, 2022.

[17] R. Pradana Aji, Y. Prayudi, and A. Luthfi, "Analysis of Brute Force Attack Logs Toward Nginx Web Server on Dashboard Improved Log Logging System Using Forensic Investigation Method," J. Tek. Inform., vol. 4, no. 1, pp. 39–48, 2023, doi: 10.52436/1.jutif.2023.4.1.644.

[18] H. V. Reddy, "Intrusion Detection System Using Customized Rules for Snort," Int. J. Manag. Inf. Technol., vol. 15, no. 3, pp. 01–14, 2023, doi: 10.5121/ijmit.2023.15301.

[19] N. Shafira Suryawatie Yomo, A. Zafrullah Mardiansyah, and I. Wayan Agus Arimbawa, "Deteksi Serangan SQL Injection Menggunakan Security Information and Event Management (SIEM) Wazuh," pp. 1–9, 2019, [Online]. Available: http://eprints.unram.ac.id/41453/

[20] A. Groenewegen and J. S. Janssen, "TheHive Project: The maturity of an open-source Security Incident Response platform," no. July, 2021.

[21] M. Bharadwaj R.K., J. Yeojin, and G. D. S. Borja, "Implementation of the Common Vulnerability Scoring System to Assess the Cyber Vulnerability in Construction Projects," no. June, pp. 117–124, 2020, doi: 10.3311/ccc2020-030.