Integration of Wazuh and Suricata with Telegram for Enhanced Threat Detection and Multiple Attack Notifications

Noor Syahirah Abdullah, Nurhashikin Mohd Salleh*, Mohd Faizal Abdollah, Siti Rahayu Selamat

Fakulti Kecerdasan Buatan dan Keselamatan Siber, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

***Corresponding Author**

DOI: https://dx.doi.org/10.47772/IJRISS.2025.91100378

Received: 27 November 2025; Accepted: 03 December 2025; Published: 11 December 2025

ABSTRACT

The rise of connected devices over the internet has led to an increase in attacks on users, compromising their information exchange and revealing sensitive data. Modern cyber threats are becoming increasingly sophisticated and severe, taking advantage of security vulnerabilities in interconnected systems. With the growing complexity of cyber threats, effective threat detection systems are essential for maintaining network security. To improve the detection of various attack types and provide real- time warnings via Telegram, this project focuses on integrating Wazuh which is a security information and event management (SIEM) platform, with Suricata, a powerful network intrusion detection and prevention system (IDS/IPS). By offering a complete solution for log management and multi-attack detection, the integration seeks to strengthen an organization's entire security posture. From system analysis and design to implementation and testing, the process adheres to the Software Development Life Cycle (SDLC). To evaluate the effectiveness of the integrated system, several attack simulations were carried out, including DoS attacks (ICMP Ping and SYN flood), FTP brute-force attacks, and port-scanning activities. The system successfully detected all these attacks. This study highlights the strengths and limitations of integrating Wazuh with Suricata, providing valuable insights for future research aimed at developing more robust intrusion detection systems.

Keywords: Wazuh, Suricata, Security Information and Event Management (SIEM), Telegram

INTRODUCTION

The increasing sophistication and occurrence of attacks in today's digital landscape pose significant challenges for organisations working to secure their data and infrastructure. Modern cyber threats continue to evolve in severity and complexity, exploiting security weaknesses within interconnected systems. Multifaceted attacks are frequently difficult for traditional security systems to identify in real time, increasing risks and possible harm. Integrating reliable detection technologies with real-time alerting systems presents a viable way to overcome these obstacles. More than 230 active enemies have been identified by the 2024 Global Threat Report, which also notes that the fastest eCrime breakout time in 2023 was only 2 minutes and 7 seconds. This emphasizes how urgently quick detection and response skills are needed to lessen these quick invasions. According to the report, there has been a notable increase in covert cyber activity, including malware-free attacks, cloud breaches, and data theft. Despite improvements in detection systems, these patterns show that adversaries are constantly changing and adapting [1].

To overcome these obstacles, this project integrates Suricata, a network intrusion detection and prevention system, and Wazuh, an open- source security platform, with Telegram to provide real-time multi-attack notifications The open-source security platform Wazuh is proficient at compliance management, intrusion detection, and log monitoring. In contrast, Suricata is a high-performance intrusion detection and prevention system (IDS/IPS) that can detect threats by analyzing network data. By combining log-based and network- based insights, the utilization of these technologies provides a thorough method for threat identification. The effectiveness and timeliness of the response are just as crucial, even though detecting capabilities are crucial.

The cloud-based messaging app Telegram offers a quick and effective way to send out real-time alerts. Security teams may mitigate potential risks and respond to incidents more quickly by combining Telegram with Wazuh and Suricata, which allows for instant warnings regarding threats discovered.

The rest of this paper is structured as follows. Section II is the related work of the previous literature review. Section III outlines the methodology of this study. Section IV presents the results from the study, while Section V discusses implementation and implications of the results, as well as the limitations of the study. Section V also concludes the paper and provides several suggestions for future research.

Related Work

In this section, Wazuh, Suricata, and attacks will be explained.

Wazuh

Wazuh is an open-source programme that is used to preserve track on server resources, protect files with File Integrity Monitoring (FIM), and improve network security by immediately identifying and stopping potential threats [2]. It is an affordable incident detection solution that functions as a Security Information and Event Management (SIEM) tool that can identify threats and anomalies within a network's assets. Through the guidance of the application, one can establish a monitoring system that can identify anomalies, combine data from different sources, and create new detection rules [3]. Wazuh can be used in a real- world environment to test its ability to recognise and react to possible threats by simulating security incidents. Through the collection of real-time logs and the analysis of security history logs from various types of logs originating from various data sources on different devices, Security Information and Event Management (SIEM) is a technology that can detect various threats and incidents from security [4].

Instead of detecting DDoS attacks, it can be able to do detection of viruses, trojans, worms, ransomware, rootkits, backdoor, keylogger and spyware. Wazuh is incapable of detecting a wide range of malware by using its set of detection techniques such as log analysis, rootkit detection, file integrity monitoring and network intrusion detection. Through these methods, malware types can be identified which can help organizations to resolve the consequences of the attack. Wazuh Incident Response makes it possible to respond to security incidents in an automated and managed manner. It enables actions, such as banning malicious IP addresses, halting compromised services, or setting off alarms, when a threat is identified. The capacity to respond quickly and automatically is essential for reducing the effects of security breaches. Wazuh minimises possible harm to systems and networks by ensuring that events are handled promptly and effectively through integration with other technologies.

Wazuh's capabilities are improved through visualisation, which offers distinct, eye- catching depictions of security events and data. Users may efficiently analyse incident data, follow trends, and monitor real-time security metrics with the use of robust dashboards (like Kibana). Security analysts may find trends, investigate occurrences, and make defensible conclusions with the help of visualisation. It enhances system monitoring overall and makes threat detection easier by converting unintelligible data into understandable graphics.

Suricata

Suricata is free and open source that serves as both an intrusion prevention system (IPS) and an intrusion detection system (IDS) for networks [5]. It is known for its effectiveness in monitoring network traffic to detect and prevent potential cyber-attacks, offering customizable features like anomaly monitoring and integration with other security tools [6]. Suricata's performance has been compared to other systems like Snort in virtualized network environments, highlighting its strong capabilities in detecting and blocking incoming network attacks [7]. Suricata is an Intrusion Detection System (IDS) that looks for harmful activity in network data. By detecting risks and making it possible to isolate dangerous nodes within a Local Area Network (LAN) via Access Control Lists, it plays a critical role in cybersecurity [8]. It uses multithreading specified for high-speed traffic handling which efficiently deals with massive amounts of data, improving performance and detection capabilities [7]. Several features of Suricata that able to enhance network security represented by [9]:

- Threat Detection

Suricata is a reliable option for contemporary network intrusion detection systems since it can analyse numerous threat kinds and adjust to diverse network environments. Basically, it continuously looks for harmful patterns in network traffic. This traffic is compared against a large database of pre-established Suricata rules and known attack signatures that gives the ability to recognise malware, attempted exploits, and unusual network activities.

- Deep Packet Inspection (DPI)

The ability of Suricata to inspect data packets which analyses both source and destination empower itself to identify any hidden threats surrounded by either encrypted traffic or files being transmitted. The high accuracy rates of machine learning models show that DPI improves anomaly detection capabilities, surpassing conventional signature-based techniques.

- Protocol Analysis

Suricata comes out with the ability to analyse a large-scale network protocol which represent on ways of variety types of communication work. This enables it to spot suspicious activity inside protocols, such as odd data transfers or efforts to take advantage of flaws in particular communication techniques. Suricata has been adapted to analyse SOME/IP traffic, enabling it to detect replay attacks and header anomalies, thereby improving security in vehicular networks.

- Network Traffic Baselining

Suricata specialized in creating a baseline for general traffic visualization. A machine learning engine can use the data generated by Suricata to learn the usual patterns and spot notable variations that could point to a possible attack by tracking activity over time.

- Threat Hunting

Suricata can be incorporated into wider threat hunting frameworks in terms of MITRE ATT&CK, which assists in identifying techniques used by threat actors [10] [11]. Security experts can benefit from Suricata comprehensive logs and analysis features which allow them to proactively search for hidden risks within the network, look at unusual activities, and spot trends using Suricata's data.

Attacks

Several attacks need to be simulated to the integrated system which are Denial-of-Service (DoS) attack, Brute-force attack and Port Scanning attack. The goal of a DoS attack is to overload a system or network with traffic so that authorised users are unable to access it. This kind of attack evaluates how well the system can identify and stop unusual traffic patterns. By methodically guessing passwords or credentials, brute-force attacks repeatedly try to obtain unauthorized access to accounts or systems. The system's ability to recognize and react to questionable login attempts will be evaluated through testing for brute-force attacks. Finally, port scanning is a reconnaissance method in which attackers look for open ports and services on a target system to find possible weaknesses.

METHODLOGY

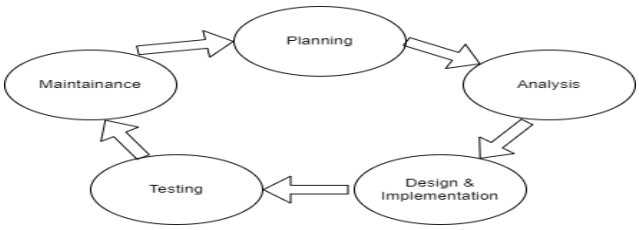

Figure 1: Research Methodology

Figure 1 depicts a Software Development Life Cycle (SDLC), which is a procedure for organizing, developing, testing, and implementing information systems, is depicted in the diagram. Its five primary stages are grouped in a cyclical flow that represents the iterative process of system development [12].

Planning Phase

During the planning stage, the project's groundwork is created to guarantee a systematic and efficient system development process. The project scope, which includes determining the goals, contribution, and questions, is defined at the start of this phase. The main objectives are to study the capabilities of the integration of Wazuh and Suricata in addition to identifying the design that can be implemented which produces an effective integration system to be evaluated. Having conversations with the project supervisor is essential for making sure the project meets technical and academic requirements while anticipating and resolving any issues. To guarantee clarity and viability, the project's goals and scope were clarified during these meetings.

To determine the tools, hardware, software, and configuration required for system implementation, a thorough requirements analysis is carried out. This entails using Suricata for network-based intrusion detection and assessing Wazuh's suitability and capabilities for log management, monitoring, and intrusion detection. To improve threat awareness and response, Telegram will also be recognized as the medium for real-time multi-attack notifications. Deliverables and milestones for every stage of the Software Development Life Cycle (SDLC) are used to create the project timetable. To make sure the project stays on course, important tasks including system design, tool configuration, attack simulation, and testing are planned with suitable due dates. Resources are distributed according to the needs of the system, including a virtualized testing environment, attack simulation datasets, and monitoring tools.

Analysis Phase

Analysis phase is carried out to gain a comprehensive understanding of the system that will be constructed, a complete and comprehensive assessment of the project requirements, tools, and objectives. The functionality of Wazuh and Suricata as essential elements for threat detection is examined in this phase, along with how they might be combined with Telegram to provide real- time alerts. This phase's main objective is to examine the chosen tools' capabilities and make sure they complement the project's goals, particularly in terms of identifying various assaults including port scanning, brute-force attacks, and denial of service (DoS). This phase starts with an in-depth literature review of Wazuh and Suricata, studying their features, limitations and strength. In addition, Telegram is analyzed as a medium of communication to generate real-time alerts. The analysis highlights verifying that notifications are delivered promptly and in a structured format.

A requirement gathering procedure is carried out to determine the hardware, software, and network specifications required to construct and test the system to support the analytical phase even further. This entails determining the resources needed to create a virtualized testing environment in which Wazuh and Suricata will function without a hitch. Virtual machines, attack simulation, network setups, and enough processing power to manage logs and traffic data are essential elements.

To guarantee appropriate simulation and detection techniques, the sorts of attacks to be evaluated are thoroughly examined. Analyzing traffic patterns to spot irregularities such abnormally high request volumes is the main goal of a study on Denial of Service (DoS) attacks. Monitoring login attempts is part of the analytic process for brute-force attacks in order to identify recurring, unsuccessful authentication attempts that could be signs of an attack. The establishment of rules to detect unauthorized scanning activities is made possible by the final examination of port scanning, which aims to understand how attackers find open ports and vulnerabilities.

Choosing the metrics and assessment standards that will be applied to measure the system's efficacy is another task for this phase. To assess how well the system detects and reacts to the simulated attacks, metrics including detection accuracy, response time, false positives/negatives, and notification delivery time are set up. The project needs, resources, and tools are completely understood at the end of the analysis phase, and a thorough implementation plan is created. This stage guarantees that every part is in line, possible problems are found, and the system design can move forward with a strong base.

Design and Implementation Phase

After analyzing all the requirements needed, the design phase summarizes the outcomes of the analysis into a structured blueprint which consists of system architecture, components, and workflows. The design will be planned in detail to ensure the successful integration of Wazuh, Suricata, and Telegram for enhanced threat detection and real-time notifications. After design has been completed, Wazuh, Suricata, and Telegram must be configured and deployed in a controlled environment for testing and assessment throughout the installation phase.

Testing Phase

In the testing phase, the system is thoroughly evaluated to ensure that the integration of Wazuh, Suricata, and Telegram works effectively in detecting multiple attacks and sending real- time notifications. This phase focuses on validating the system's ability to detect Denial of Service (DoS) attacks, brute-force attacks, and port scanning within a controlled environment. The primary objectives of this phase are to assess the system's accuracy, efficiency, and reliability while identifying any issues or areas for improvement. Setting up a virtualized environment to mimic actual attack scenarios is the first step in the testing process. Nmap is used for port scanning, Hydra or other brute-force tools are used to mimic repeated unsuccessful login attempts, and tools like hping3 are used to create DoS assaults. To test the system's ability to detect, record, and react to harmful behavior, each attack type is meticulously set up. Results will be obtained based on the detection time in Wazuh dashboard, Suricata and time of alerts generated in Telegram.

Maintenance Phase

Maintenance phase focuses on ensuring efficiency and effectiveness of the system after it has been implemented. It consists of monitoring the system's performance and configuring updates when necessary. Continuous system monitoring refers to when the system has effectively identified the attack which are DoS attack, Brute-force attack and Port Scanning according to the telegram that works in real-time.

RESULT & ANALYSIS

After simulating the attacks, results were obtained through Wazuh Dashboard and Telegram, focusing on evaluating the detection capabilities of the integrated system. The findings are divided into some subtopics according to several attack scenarios, such as ICMP Ping, SYN flood, FTP Brute-force and Port Scanning. Each section thoroughly examines system logs, generated alerts, and the time taken for notifications to be delivered via the Telegram notification system. Graphs and tables are included where necessary to provide clearer insight into the results.

ICMP Ping Flood Result

The hping3 tool was used to mimic an ICMP flood attack by sending a large number of ICMP echo requests to the target victim.

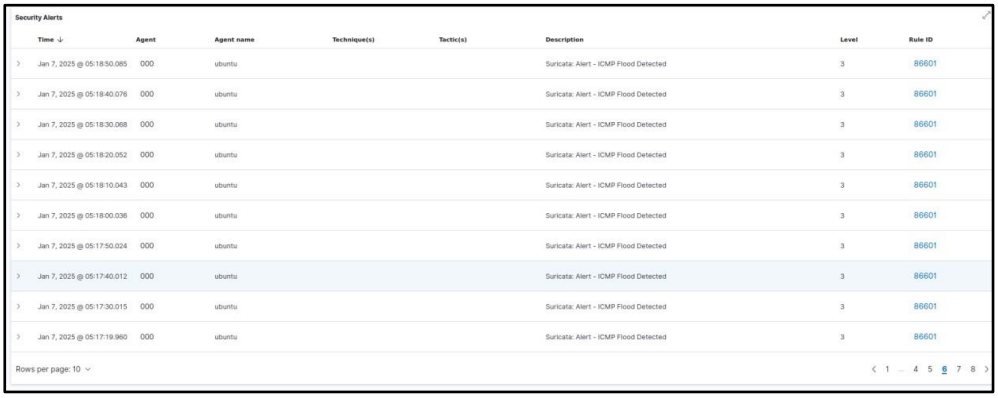

Figure 2: ICMP Ping logs

Figure 2 presents list of alerts generated after the ICMP ping attack have been simulated. The logs were mostly generated based on Suricata rules detection as it mentioned "Suricata Alert" in the description above which represents ICMP Flood Detected.

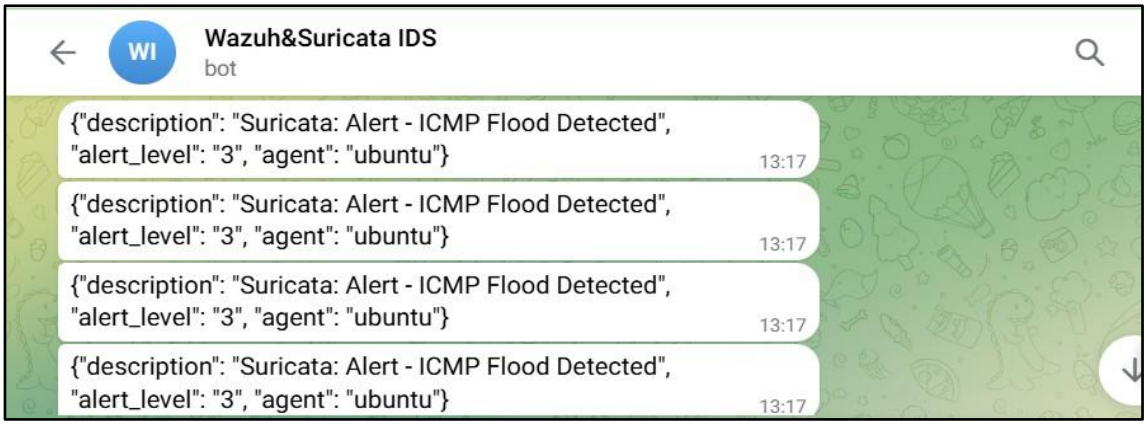

Figure 3: ICMP Ping in the Telegram bot

Figure 3 represents alerts generated by Telegram bot after ICMP ping has been launched by the attacker. The alert showed a description of the attack which is 'ICMP Flood Detected', alert level and agent involved.

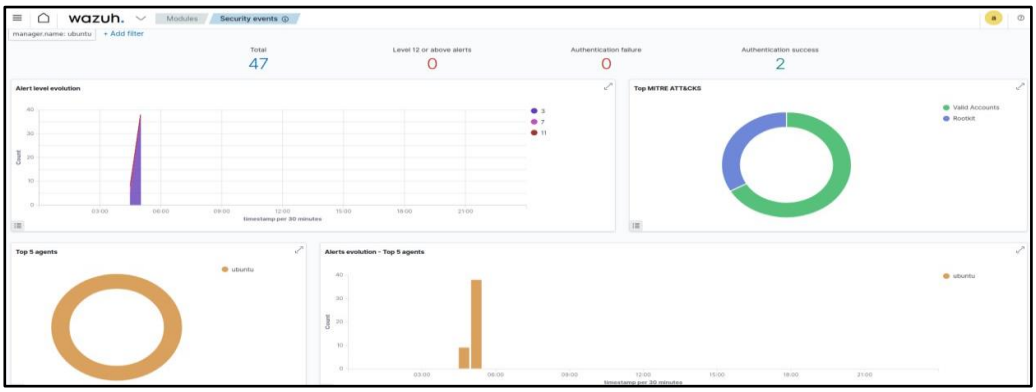

Figure 4: ICMP Ping visualization

Wazuh Dashboard plays a role in displaying the graph based on the behavior of the attack as shown in Figure 4. The alert evolution graph which has been set to a timestamp per 30 minutes shows a significant spike in activity around 05:00, according to the time ICMP ping flood attack that has been launched, which is 05:17 AM.

SYN Flood Result

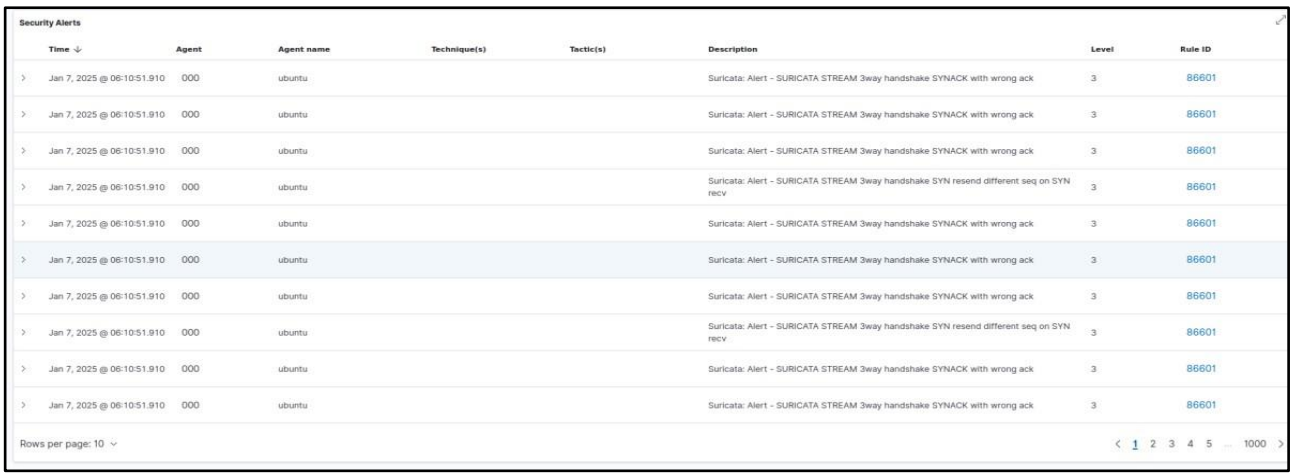

Figure 5: SYN Flood logs

Figure 5 shows a comprehensive table of security alerts produced by Wazuh, all of which came from the same agent called Ubuntu. The fact that each alert was recorded at 06:10:51.910 on January 7, 2025, suggests that several things happened at once. "Suricata Alert – SURICATA STREAM 3-way handshake SYN ACK with wrong ack" is the description of the alerts, which were caused by the network security monitoring tool Suricata and indicate irregularities in the TCP handshake procedure. Additionally, "SYN resend different seq on SYN recv" is mentioned in certain alerts, suggesting possible packet sequencing problems throughout the connection procedure. Every alert has the same severity rating, which is 3, indicating a low to medium threat level with the ID of the rule: 86601.

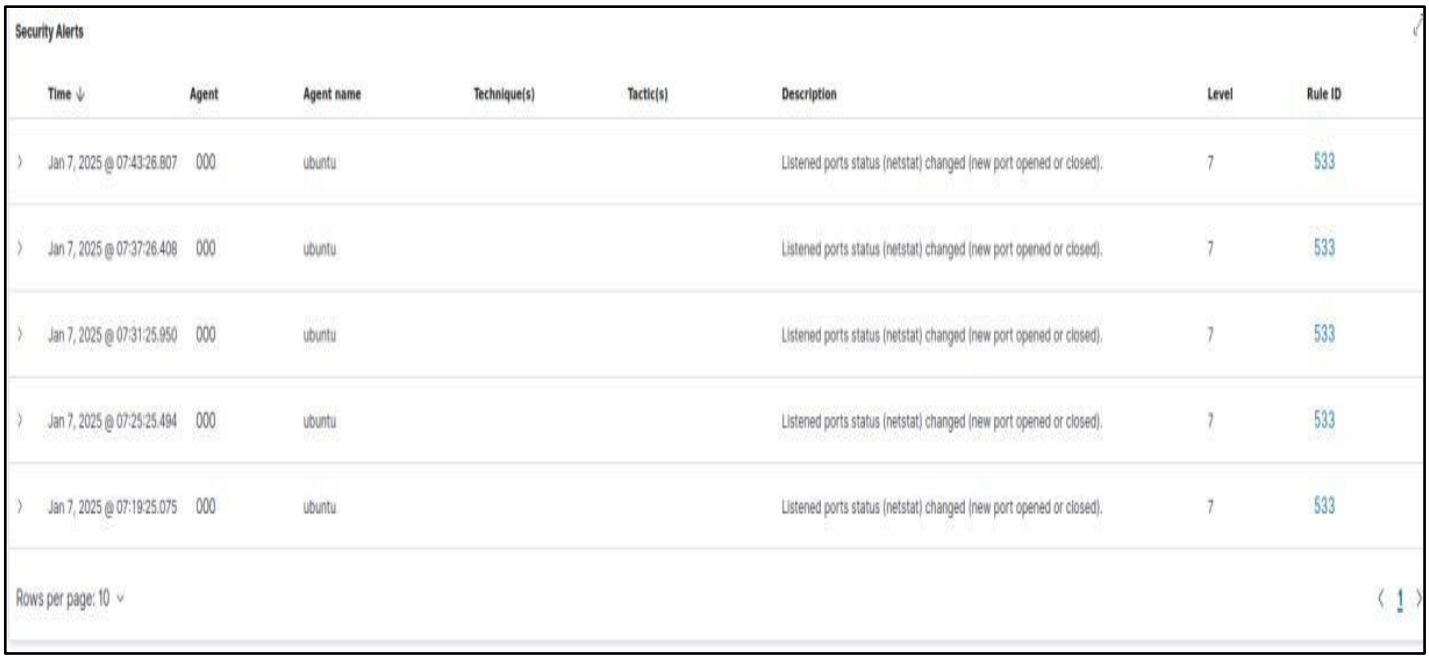

Figure 6: Open port alerts

According to the SYN Flood attack launched, the alerts have the potential to the discovery of new security flaws by revealing unauthorized open ports as shown in Figure 6.

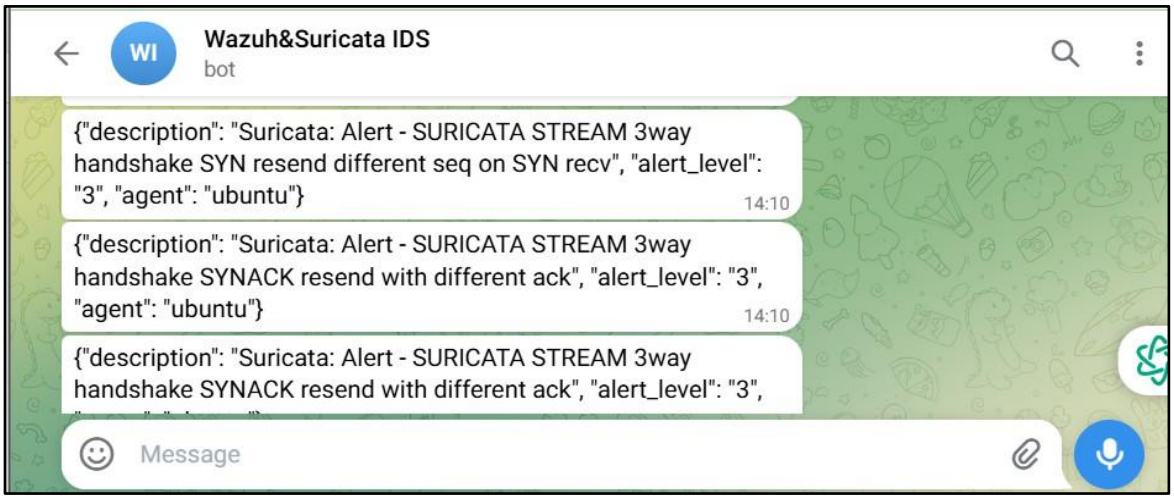

Figure 7: SYN Flood Telegram Notification

Figure 7 highlights Telegram notifications that provide instant threat notice by showing a sequence of real-time security alerts sent to a Telegram bot called Wazuh & Suricata IDS. "SYN resend with a different sequence on SYN recv" and "SYNACK resend with a different acknowledgement" are among the abnormalities in the TCP 3-way handshake procedure that are regularly reported by the alerts. A severity level of three, which denotes a low to medium threat, is given to each alert. The timestamps (14:10) show that several alarms were triggered nearly simultaneously, and all alerts are produced by the same agent, known as Ubuntu.

Figure 8: SYN flood visualization

Figure 8 above illustrates the Wazuh Dashboard visualization of the SYN flood attack which was launched around 06:10am. With the alert evolution graph set to timestamp every 30 seconds, there is an apparent spike in activity at 06:10.

FTP Brute-force Result

```
syahirah@fyp:~$ hydra -l user123 -P passwordlist.txt ftp://192.168.247.141
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, o
r for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-01-08 03:08:32
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to
 prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 2006208 login tries (l:1/p:2006208), ~125388 tries per task
[DATA] attacking ftp://192.168.247.141:21/
[STATUS] 259.00 tries/min, 259 tries in 00:01h, 2005949 to do in 129:05h, 16 active
[STATUS] 267.67 tries/min, 803 tries in 00:03h, 2005405 to do in 124:53h, 16 active
[21][ftp] host: 192.168.247.141   login: user123   password: abc123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-01-08 03:15:30
syahirah@fyp:~$
```

Figure 9: FTP Brute-force was launched

Figure 10: Wazuh Dashboard security alerts

Figure 11: Wazuh Dashboard visualization

A Wazuh dashboard in the Figure 11 above showcasing real-time monitoring findings from a security incident involving the number of alarms (118), critical alerts with level 12 or higher (0), authentication failures (95), and authentication successes (0) which are among the important security metrics that are summarized at the top. A potential brute force assault is suggested by the large number of authentication failures without any successful authentication attempts. An obvious rise in alerts occurs between 03:05:00 and 03:10:00, as shown in the "Alert level evolution" figure, suggesting a concentrated time of questionable behavior. The alert levels, which represent different degrees of severity, run from 3 to 10. This is obvious that primary tactics detected include Password Guessing and Brute Force, confirming the nature of the attack.

Port Scanning Result

```
root@fyp:~# nmap -A 192.168.247.141
Starting Nmap 7.80 ( https://nmap.org ) at 2025-01-08 13:45 UTC
WARNING: Service 192.168.247.141:9200 had already soft-matched rtsp, but now soft-matched sip; igno
ring second value
Nmap scan report for 192.168.247.141
Host is up (0.0016s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE   VERSION
21/tcp   open  ftp       vsftpd 3.0.5
22/tcp   open  ssh       OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http      Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
443/tcp  open  ssl/https
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 401 Unauthorized
|     osd-name: ubuntu
|     x-frame-options: sameorigin
|     content-type: application/json; charset=utf-8
|     cache-control: private, no-cache, no-store, must-revalidate
|     set-cookie: security_authentication=; Max-Age=0; Expires=Thu, 01 Jan 1970 00:00:00 GMT; Secur
e; HttpOnly; Path=/
|     content-length: 77
|     Date: Wed, 08 Jan 2025 13:45:42 GMT
|     Connection: close
|     {"statusCode":401,"error":"Unauthorized","message":"Authentication required"}
|   GetRequest:
|     HTTP/1.1 302 Found
```

Figure 12: Port scanning was launched

Figure 13: Port Scanning alerts appear in Wazuh Dashboard

Several security alarms that Suricata identified are shown in Figure 13, with particular attention paid to possible network scans. Every warning is associated with rule ID 86601, which signifies that the action is consistent with the signature for "ET SCAN Possible Nmap User-Agent Observed." The same agent, "ubuntu," with ID 000, is linked to all the alerts. Every alert has a constant time of detection, which was logged at 13:46:04.768 on January 8, 2025, indicating a spike in activity at that exact instant. Each alert has a severity rating of 3, which denotes a moderate level of threat.

Figure 14: Wazuh Dashboard visualization

A dashboard from Wazuh that summarizes the security events observed for a system under monitoring is shown in Figure 14. None of the 54 warnings that emerged were able to meet the crucial criterion of level 12 or above. The timeline graph in the "Alert level evolution" section indicates that most alerts are classified at levels 3, 5, and 7, with a substantial alert spike occurring around 13:46. This suggests a spiking in questionable activities at that period.

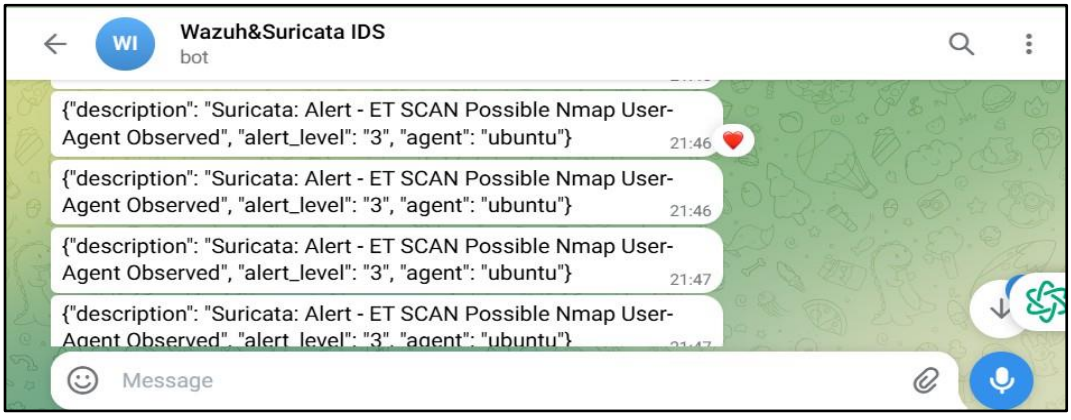

Figure 15: Alerts generated in the Telegram

Based on Figure 15 above, a sequence of real-time security warnings produced by the Wazuh & Suricata IDS Telegram can be observed. The structured JSON format of each alert message highlights important elements including the agent, alert level, and description. According to the description, Suricata found a "ET SCAN Possible Nmap User-Agent Observed," indicating that the Nmap tool was probably used to execute possible network scanning activities. With an alert level of 3, all incidents are classified as low to medium severity. Furthermore, a "ubuntu" agent is linked to the identified activity, suggesting that the alarms were sent off by keeping an eye on a computer running the Ubuntu operating system. Between 21:46 and 21:47, the warnings were issued continually, indicating a brief spike in scanning activity. The summarization of these attacks is shown in Table 1.

Table 1: Analysis of the attack's detection

Type of Attack	Attack Time	Detection		Detection Time	
		Yes	No	Wazuh Dashboard	Telegram
DoS (ICMP Ping)	10:49 AM	/		10:49 AM	10:49 AM
DoS (SYN Flood)	03:16 PM	/		03:17 PM	03:17 PM
FTP Brute- force	12:21PM	/		12:21PM	12:21PM
Port Scanning	21:45 PM	/		21:46 PM	21:46 PM

Table 1 above describes analysis of the detection of all attacks that have been launched such as DoS (ICMP Ping and SYN Flood) attacks, FTP brute-force, and port scanning are among them. The time of occurrence and detection status are noted for every assault. The DoS (ICMP Ping) attack was discovered instantly on both platforms simultaneously at 10:49 AM. Likewise, the DoS (SYN Flood) assault happened at 3:16 PM and was discovered on both platforms one minute later at 3:17 PM. When the FTP Brute- force assault happened at 12:21 PM, both platforms were able to identify it simultaneously and immediately. Although Telegram quickly reported the Port Scanning assault, which occurred at 21:45 PM, the Wazuh Dashboard logged its discovery at 21:46 PM.

DISCUSSION

The combination of Wazuh, Suricata, and Telegram for improved cyber threat detection and real-time alerting is what this project has to offer. The project helps to effectively monitor and identify a variety of assaults, including DoS attacks (ICMP Ping and SYN Flood), FTP Brute-force, and port scanning, by effectively integrating these platforms. A layer of real-time alerting is added by using Telegram for multi-attack alerts, which speeds up reaction times to possible attacks. Along with identifying areas where setups and synchronization may be improved, the project also helps to understand the advantages and disadvantages of each platform, especially about the detection of various attack types. Additionally, the examination of detection timing differences offers important information about how to improve the integration procedure for improved real-time reporting and platform coordination. This research integrates attack detection, log management, and rapid notifications into a single, integrated solution, offering a fundamental framework for enhancing cybersecurity monitoring systems.

One key direction for future research is to expand the types of attacks evaluated. Although the current study focused on common attacks such as DoS and FTP brute force using supported devices, assessing a wider range of cyberthreats would provide a more complete understanding of the system's capabilities. Future tests could include more advanced attack vectors such as SQL injection, malware infections, ransomware, and insider threats to determine how effectively the integration handles complex scenarios.

ACKNOWLEDGEMENT

The authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM) for the support.

REERENCES

1. Adam, M. (2024). CrowdStrike 2024 Global Threat Report: Adversaries Gain Speed and Stealth. Available at: https://www.crowdstrike.com/en-us/blog/crowdstrike-2024-global-threat-report// [Accessed at 23 November 2024]
2. Nova, F., Pratama, M. D., & Prayama, D. (2022). Wazuh Sebagai log event management Dan Deteksi Celah Keamanan Pada server dari serangan dos. JITSI: Jurnal Ilmiah Teknologi Sistem Informasi, 3(1), 1-7.
3. Steenwinckel, B., De Paepe, D., Vanden Hautte, S., Heyvaert, P., Bentefrit, M., Moens, P., ... & Ongenae, F. (2021). FLAGS: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning. Future Generation Computer Systems, 116, 30-48.
4. Vielberth, M. (2021). Security information and event management (SIEM). In Encyclopedia of Cryptography, Security and Privacy (pp. 1-3). Berlin, Heidelberg: Springer Berlin Heidelberg.
5. Veerasingam, P., Abd Razak, S., Abidin, A. F. A., Mohamed, M. A., & Satar, S. D. M. (2023). INTRUSION DETECTION AND PREVENTION SYSTEM IN SME'S LOCAL NETWORK BY USING SURICATA. Malaysian Journal of Computing and Applied Mathematics, 6(1), 21-30.
6. Ghazi, D. S., Hamid, H. S., Zaiter, M. J., & Behadili, A. S. G. (2024). Snort versus suricata in intrusion detection. Iraqi Journal of Information and Communication Technology, 7(2), 73-88.
7. Ghazi, D. S., Hamid, H. S., Zaiter, M. J., & Behadili, A. S. G. (2024). Snort versus suricata in intrusion detection. Iraqi Journal of Information and Communication Technology, 7(2), 73-88.
8. Sree, T., Harsha, Y. S. S., & Rajagopalan, N. (2024, July). Suricata-Based Intrusion Detection and Isolation System for Local Area Networks. In 2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT) (pp. 1-5). IEEE.
9. Dallon, R. (2024). What is Continuous Security Monitoring?. Available at: https://www.stamus-networks.com/blog/what-is-continuous-security-monitoring [Accessed at 23 November 2024]
10. Ammi, M., & Jama, Y. M. (2023). Cyber Threat Hunting Case Study using MISP. J. Internet Serv. Inf. Secur., 13(2), 1-29.
11. Nour, B., Pourzandi, M., & Debbabi, M. (2023). A survey on threat hunting in enterprise networks. IEEE communications surveys & tutorials, 25(4), 2299-2324.
12. Pargaonkar, S. (2023). A comprehensive research analysis of software development life cycle (SDLC) agile & waterfall model advantages, disadvantages, and application suitability in software quality engineering. International Journal of Scientific and Research Publications, 13(8), 120-124.