Journal of Technology and Computer (JOTECHCOM)
Vol. 3, No. 1, February 2026, pp. 16~23
ISSN: 3048-0477 (Media Online)

Event-Driven Intrusion Detection and Response Automation Using n8n Workflow Platform

Nauval Alfarizi[1], Rivaldi Rivaldi[2]
[1,2]Department of Computer and Network Engineering, Sekolah Menengah kejuruan Telkom 2 Medan
[1]nauvalalfarizi026@gmail.com, [2]rvaldi26@gmail.com

Article Info

Article history:

Received January 18, 2026
Revised February 02, 2026
Accepted February 09, 2026

Keywords:

Automation
n8n
Workflow
Log Monitoring
Brute Force Attak

ABSTRACT

This study introduces a server security monitoring system that uses events to detect SSH brute-force attacks. It uses automatic log analysis and sends real-time alerts. To test how well the system works, an experiment was conducted simulating attacks against an SSH service (port 22) without a firewall. Three different situations were tested: normal access, detecting unusual activity, and high-stress attacks. Under normal conditions, the system saw very little traffic: 233 packets, an average of 19 packets per second, and 38 kbps, indicating little impact and no false alarms. As the attacks grew more intense, network traffic increased significantly, reaching 96,997 packets and 76.5 MB of data during high-stress attacks, with an average speed of 1,132 kbps. All 500 brute-force attempts were found and recorded. By combining automated workflows with real-time Telegram alerts, administrators can get timely warnings. The results show that the system is effective, can handle large amounts of data, and is dependable for real-time SSH attack detection and server security monitoring.

Corresponding Author:

Nauval Alfarizi
Sekolah Menengah Kejuruan Telkom 2 Medan
nauvalalfarizi026@gmail.com

1. INTRODUCTION

Technological developments worldwide are characterized by the emergence of various technologies, such as communication networks (e.g., 5 G), intelligent robots, and the Internet of Things (IoT) in life [1]. Information technology began to develop since the era of written and printed media, then continued to develop until a time when society gradually began to become familiar with long-distance information technology, which was the beginning of the emergence of fast information technology such as telephones, radio, television, and computers, marking the beginning of the technology known as the internet [2]. However, behind all of this, there is a significant impact from the massive development of existing technology. If companies fail to address weak data protection, it can pose a serious threat to their business cycle. Gaps in data protection systems can lead to various data security threats for companies [3].

Anomaly-Based Intrusion Detection Systems have been widely studied for their ability to identify previously unknown attacks, commonly referred to as *zero-day attacks*, and to support the discovery of new security rules by detecting observed deviations in system behavior. In this context, rule-based intrusion detection approaches remain relevant due to their ability to define security policies with high precision using *crisp* (hard) rules or through more flexible *fuzzy-rule-based* mechanisms that support statistical inference and tolerance to uncertainty.However, most

conventional Intrusion Detection System (IDS) implementations still rely heavily on static rule configurations, manual analysis, or machine learning-based approaches that require substantial computational resources and complex training processes. While machine learning techniques can improve detection accuracy, their deployment often demands large datasets, continuous model updates, and specialized infrastructure [4].

Base on research with server security using the honeypot concept was conducted to secure virtual server devices during communication, during interactions between client and server devices, and through the implementation of special networks, such as tunneling, to create secure network connections. This network is monitored using the honeypot's working principle, which is conceptually similar to a log-based authentication system when an attempted cybersecurity attack occurs [5]. The forthcoming Research is planned to be undertaken and focuses on building an SSH Server service for devices and securing it using the port-knocking concept. Port knocking is the rule-based approach for filtering access to network services that will be monitored [6]. Following our research into server device monitoring, we have taken the initiative to implement Zabbix on the server. This development represents our dedication to improving monitoring processes and enhancing overall performance for all stakeholders involved. Using the Zabbix platform as a server device monitoring solution demonstrates innovation in technology selection. Zabbix is known for its effective monitoring and proactive notification features, which enable faster problem response and better server performance. The design of the monitoring infrastructure is also a key innovation, as good design ensures the monitoring system's efficiency, reliability, and scalability [7].

Given the enormous volume of security-related data that can be exploited, effective mechanisms are required to manage, store, and prioritize this information in a coherent, structured manner to enable further analysis. In particular, security events must be aggregated and processed within a very short time frame, as timely responses are critical to mitigating the impact of cyberattacks. These requirements make low-latency data processing a fundamental aspect of modern security monitoring systems [8]. Based on these considerations, this Research proposes an event-driven intrusion detection and response approach using the n8n workflow automation platform. The proposed system processes incoming security events in real time through HTTP-based webhooks, applies rule-based classification to distinguish normal and anomalous activities, records events into structured log files, and automatically dispatches alert notifications.

The primary objective of this research is the development of a lightweight and easily reproducible workflow-based security automation system designed for network environments. This proposed system combines detection, logging, notification, and security response simulation into a single centralized workflow. As a result, it offers a practical alternative to traditional, complex Intrusion Detection Systems (IDS).

2. METHOD

This method uses an event-driven architecture with n8n as the workflow orchestration engine. Any suspicious network activity is sent to the system as an event via an HTTP webhook [9].

2.1 System Architecture

This Research focuses on the use of n8n architecture as a workflow. [10] In building an IDS alarm concept with a workflow to detect attempted SSH brute-force attacks [11] and search for server account credentials. This Research focuses on the use of n8n architecture as a workflow [12].

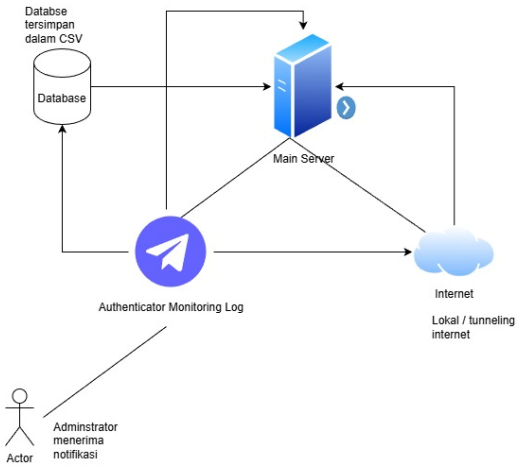


Figure 1. System Architecture Workflow

In the existing architecture, an administrator receives real-time authentication using a Telegram bot. The monitoring system supports real-time connections to public APIs implemented within the system architecture. [13].

2.2 Flowchart Pipeline

In Figure 2 below, it will be explained briefly how the pipeline system architecture is built using the n8n base.

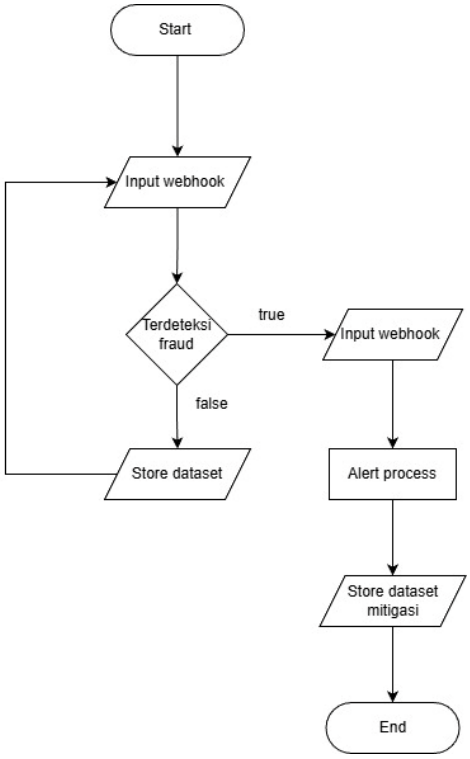


Figure 2. System Architecture Flowchart

2.3 Logging And Notification

Furthermore, it is essential to ensure that the configuration of devices, sensors, and connections is thoughtfully designed and implemented to fulfill network monitoring requirements. Each event is recorded in a CSV file as a structured log dataset. In the case of an anomaly, the system automatically sends a notification via Telegram, providing an early warning to network administrators [14]. Telegram aims to establish a more dynamic, streamlined communication experience, empowering users to actively engage rather than remain passive. By interacting directly, they can obtain responses that are more tailored, precise, and relevant to their individual needs.[15].

2.4 Response Scenario

This Research consists of various scenarios that apply existing attack to server devices. These attacks are used to test the extent to which the program can operate and to alert the system to the device under real-time conditions in the existing alarm system.

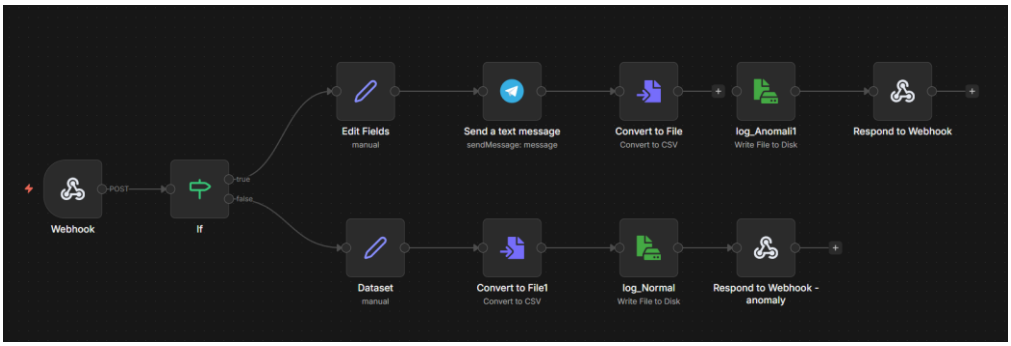


Figure 3. Architecture workflow n8n

3. RESULT AND DISCUSSION

In the scenarios, the server is granted access with port 22 authentication as the main baseline in this test. The server will be forced to be attacked in an end-to-end attack simulation workflow without firewall filtering, as a test and to implement the existing attack results, to prove that the workflow is available and well implemented in monitoring.

3.1 Response Scenario Normal

In this test, the test was conducted by applying a changed condition scenario in which the intruder is in a normal condition, with an attempt experiment <3. Thus, the results made the scenario a normal condition.

Table 1. first Attempt

ip	attempts	service	timestamp
1.1.1.1	3	ssh	2026-01-17T05:31:23.814-05:00

The results from table 1 indicate that the server successfully monitored and captured the credential logs needed for analysis, with the interpretation indicating an attempt to log in to the main server 192.168.100.x, which is integrated with the n8n workflow, using the IP 1.1.1.1.

Table 2. Capture Analyst Normal Attempt

Measurement	Captured	Displayed	Marked
Packets	233	233 (100.0%)	—
Time span (s)	12.291	12.291	—
Average pps	19.0	19.0	—
Average packet size (B)	256	256	—
Bytes	59672	59672 (100.0%)	0
Average bytes/s	4854	4854	—
Average bits/s	38 k	38 k	—

3.2 Response Scenario Anomaly

Next scenario intruder attempts to brute-force SSH on the SSH server, leveraging the command server's ability to detect attacks and conduct real-time network monitoring for the administrator. The server will then use the Telegram API to report to the administrator's cellphone, making the alarm more effective and less dependent on the server's monitoring log dataset.

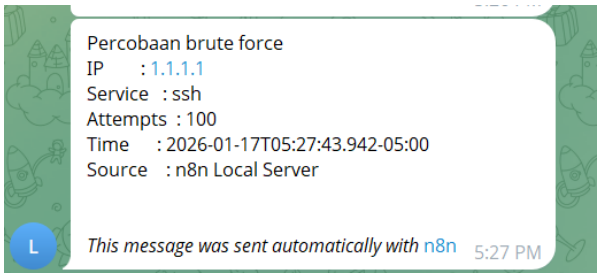


Figure 4. Telegram Alaram Report

In figure 4 it is explained that the intruder tried to make a brute force attempt with 100 attempts with the same IP against the normal attack trial, this was done to test the performance with curl even though this could be implemented into a situation with the concept of a zero day attack which would result in multiple notifications because the attempt given was only for 1 message.

Table 3. Capture Analyst Anomaly Attempt

Measurement	Captured	Displayed	Marked
Packets	233	233 (100.0%)	—
Time span (s)	12.291	12.291	—
Average pps	19	19	—
Average packet size (B)	256	256	—
Bytes	59672	59672 (100.0%)	0
Average bytes/s	4854	4854	—
Average bits/s	38 k	38 k	—

Later in this advanced scenario, researchers tried to implement high-level stress on the server to test the performance required to log and monitor the system.

3.3 High Stress Scenario

In this pressure section, the servers will be the main target of an attacker executing a brute-force attack. The test was carefully structured with incremental features at each stage to simulate real-world attack patterns, ensuring a thorough evaluation. The test was divided into several workload levels to measure the responsiveness and resilience of the n8n system and the logging mechanisms implemented.

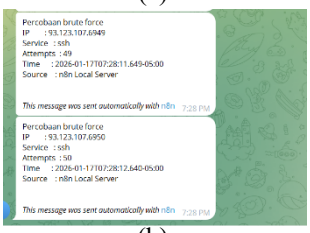
Table 4. Pressure Test Level

Stress Level	Total Attempts (Requests)	Scenario Description
Low	10 – 50	Tests the basic functionality of the webhook and validates the initial workflow logic.
Medium	100 – 250	Simulates a small-scale brute-force attack to observe data processing speed and system responsiveness.
High	300 – 500	Evaluates system load performance when handling simultaneous data surges and concurrent executions.
Extreme	500 – 1000	A high-stress scenario designed to identify the server's breaking point and n8n's capacity (within Docker) to manage massive execution queues.

3.3.1 Low Interaction Test

In this Research, the administrator received a notification that a brute-forcebrute-force attack attempt had occurred against 22 SSH services on the server, whichas shown in Figure 5 below.

(a)



(b)

Figure 5. Low Attack Interaction Test

In this Research, the intruder successfully attempted an SSH brute-force attack on port 22. The port was attempted with the IP address 93.123.107.69 (50), which is an iteration of the program, which is a minor program. Data capture was also performed simultaneously; the results are shown in Table 5.

Table 5. Low Pressure Test Level

Measurement	Captured	Displayed	Marked
Packets	18,179	18,179 (100.0%)	—
Time span (s)	62.524	62.524	—
Average pps	290.8	290.8	—
Average packet size (B)	736	736	—
Bytes	13,387,597	13,387,597 (100.0%)	0
Average bytes/s	214 k	214 k	—
Average bits/s	1712 k	1712 k	—

3.3.2 Medium Interaction Test

Next, the Research will continue at a medium level with a brute-force attempt of 200 against the same service and port, but with a different intruder IP, namely 220.123.107.x. In this Research, the behavior of the IP changes at the host end, making it difficult to track, yet it still maintains the same pattern.

Figure 6. Medium Attack Interaction Test

The results of this stress level still use the incremental feature, but the intruder is becoming more intelligent, changing IP blocks from static to dynamic when attempting brute-force attacks against the same service. This is evidenced by the automated log-monitoring experiment, which generated 250 real-time alert notifications of attempts with a time delay from the workflow automation server.

Table 6. Medium Pressure Test Level

Measurement	Captured	Displayed	Marked
Packets	57,693	57,693 (100.0%)	—
Time span (s)	270.818	270.818	—
Average pps	213	213	—
Average packet size (B)	674	674	—
Bytes	38,879,391	38,879,391 (100.0%)	0
Average bytes/s	143 k	143 k	—
Average bits/s	1,148 k	1,148 k	—

Table 6 shows that bytes are quite significant when the stress level is increased to medium; this also affects the required database.

3.3.3 High Interaction Test

The final stress measurement was the total number of attempts the intruder made to access the SSH server on the server device. The server experienced a very high traffic spike, as evidenced by the quality of service (quality of service) applied by Wireshark in capturing data, as seen in Table 7.

Table 7. High Pressure Test Level

Measurement	Captured	Displayed	Marked
Packets	96,997	96997 (100.0%)	—
Time span, s	540.61	540.61	—
Average pps	179.4	179.4	—
Average packet size, B	789	789	—
Bytes	76,530,805	76530805 (100.0%)	0
Average bytes/s	141 k	141 k	—
Average bits/s	1132 k	1132 k	—

A total of 500 attacks were launched against the server, using credentials that changed according to the server's dynamic IP address. It is shown in Figure 7. Regarding the SSH server's log attempt, 500 brute-force attacks were detected.

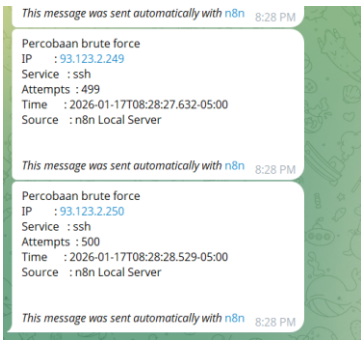


Figure 8. Log Attempt High Test

In this log, the system reads the attack log,, which is integrated with the system using the n8n workflow as an event-driven approach forevent-driven approach to detect anomalous patterns in an autonomous system.

3.3.4 Different of platform's

In the comparison of brute-force attack platforms, the results obtained include the use of bitstream in the attempted experiments; the valuation results show a gap, with mid-low having almost the same results as the existing matrix level. Aligns with that, B comparative carried out,obtained that makese

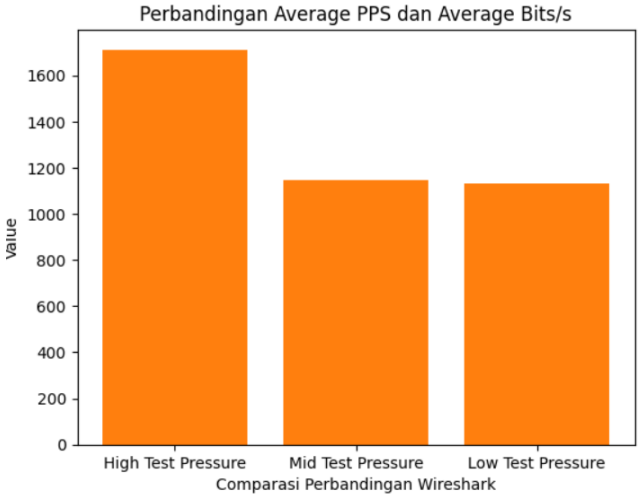


Figure 9. Comparison of Average Bit's

4. CONCLUSION

The experimental results show that the proposed SSH attack monitoring workflow operates effectively across normal, anomalous, and high-stress scenarios. In the normal condition (≤3 login attempts), the system recorded only 233 packets, with an average rate of 19 pps and 38 kbps, indicating minimal network overhead and no false anomaly detection. When brute-force behavior was introduced, the system successfully detected repeated attempts and generated real-time alerts, while maintaining stable capture performance, proving its reliability in identifying anomalous SSH access patterns.

Under high-stress conditions, the system demonstrated strong resilience as attack intensity increased. At low, medium, and high stress levels, traffic volumes rose significantly from 18,179 packets (1,712 kbps) to 57,693 packets (1,148 kbps), and finally to 96,997 packets (1,132 kbps), with total transmitted data reaching 76.5 MB in the highest scenario. Despite this escalation, all 500 brute-force attempts were successfully logged and processed through the event-driven n8n workflow. These results confirm that the proposed system can handle increasing attack loads while maintaining accurate detection and real-time monitoring performance.

ACKNOWLEDGEMENTS

I express my gratitude to Allah SWT. Thanks to His grace, we have reached this stage. I also thank my colleague, Mr. Rivaldi, S.Kom, for his assistance in this Research, especially for contributing Research ideas by applying N8N in the field of cybersecurity in real-world case studies. I hope this Research can help the community, especially our beloved students who are now at SMK Telkom 2 Medan.

REFERENCES

[1] A. G. Prawiyogi and A. S. Anwar, "Perkembangan Internet of Things (IoT) pada Sektor Energi: Sistematik Literatur Review," *J. MENTARI Manajemen, Pendidik. dan Teknol. Inf.*, vol. 1, no. 2, pp. 187–197, 2023, [Online]. Available: https://journal.pandawan.id/mentari/article/view/254%0Ahttps://journal.pandawan.id/mentari/article/download/254/251

[2] Farhatun Nisaul Ahadiyah, "Perkembangan Teknologi Infomasi Terhadap Peningkatan Bisnis Online," *INTERDISIPLIN J. Qual. Quant. Res.*, vol. 1, no. 1, pp. 41–49, 2023, doi: 10.61166/interdisiplin.v1i1.5.

[3] M. Irfan, M. Elvia, and S. Dania, "x," *Jursima*, vol. 11, no. 1, pp. 110–121, 2023.

[4] S. Neupane *et al.*, "Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities," *IEEE Access*, vol. 10, no. August, pp. 112392–112415, 2022, doi: 10.1109/ACCESS.2022.3216617.

[5] R. L. Nauval Alfarizi, T. M. Diansyah, "Simulasi Pengamanan Virtual Server Menggunakan Dionaea

Honeypot Dan Tunneling Sebagai Proses Pengamanan Komunikasi Data," *Snastikom*, vol. 9, no. 4, pp. 41–48, 2022.

[6] D. Desmira and R. Wiryadinata, "Rancang Bangun Keamanan Port Secure Shell (SSH) Menggunakan Metode Port Knockking," *INSANtek*, vol. 3, no. 1, pp. 1–5, 2022, doi: 10.31294/instk.v3i1.552.

[7] A. Rahma, F. Indriyani, and T. A. A. Sandi, "Perancangan Dan Implementasi Monitoring Perangkat Server Menggunakan Zabbix Pada PT. Rizki Tujuh Belas Kelola," *J. Insa. J. Inf. Syst. Manag. Innov.*, vol. 3, no. 2, pp. 85–95, 2023, doi: 10.31294/jinsan.v3i2.3009.

[8] S. Khriji, Y. Benbelgacem, R. Chéour, D. El Houssaini, and O. Kanoun, "Design and implementation of a cloud-based event-driven architecture for real-time data processing in wireless sensor networks," *J. Supercomput.*, vol. 78, no. 3, pp. 3374–3401, 2022, doi: 10.1007/s11227-021-03955-6.

[9] A. Kumar, "How to Connect AI Agents with n8n for End-to-End Automation," no. Iv, pp. 73–81, 2024.

[10] Nurhaliza and Suendri, "Utilizing GPT-4o Mini in Designing a WhatsApp Chatbot to Support the New Student Admission Process at Telkom University," *Journal.Ittelkom-Pwt.Ac.Id/Index.Php/Dinda*, vol. 5, no. 2, pp. 258–267, 2025.

[11] C. Pamungkas, P. Hendradi, D. Sasongko, and A. Ghifari, "Analysis of Brute Force Attacks Using National Institute Of Standards And Technology (NIST) Methods on Routers," *J. Informatics Inf. Syst. Softw. Eng. Appl.*, vol. 5, no. 2, pp. 115–125, 2023, doi: 10.20895/inista.v5i2.1039.

[12] J. A. Dharma and Rino, "Network Attack Detection Using Intrusion Detection System Utilizing Snort Based on Telegram," *bit-Tech*, vol. 6, no. 2, pp. 118–126, 2023, doi: 10.32877/bt.v6i2.943.

[13] Yusril Athallah dan Rizqi Agung, "VOL. VIII NO. 1 FEBRUARI 2022 JURNAL TEKNIK INFORMATIKA STMIK ANTAR BANGSA Rancang Bangun Prototype Monitoring Lampu Jalan Secara Otomatis Menggunakan Mikrokontroller ESP32 Dan Api Bot Telegram," vol. VIII, no. 1, pp. 12–19, 2022, [Online]. Available: http://awesomerockguy.blogspot.com/2015/10/tutorial-

[14] N. R. Fachrurrozi, A. A. Wirabudi, and S. A. Rozano, "Design of network monitoring system based on LibreNMS using Line Notify, Telegram, and Email notification," *Sinergi (Indonesia)*, vol. 27, no. 1, pp. 111–122, 2023, doi: 10.22441/sinergi.2023.1.013.

[15] A. khusnul Umam, E. Wijayanti, and A. A. Chamid, "Pengembangan Chatbot Pada Platform Telegram Sebagai Media Informasi Seputar Handphone," *bit-Tech*, vol. 8, no. 1, pp. 33–40, 2025, doi: 10.32877/bt.v8i1.2150.