Beykent Üniversitesi Fen ve Mühendislik Bilimleri Dergisi
Beykent University Journal of Science and Engineering

*Research Article*

BUJSE, 2026, 19 (1), 32-54
DOI: 10.20854/bujse.1918683
e-ISSN: 1307-3826
https://dergipark.org.tr/tr/pub/bujse.1918683

# Open-Source Automation in Cyber Incident Response Processes: A Low-Code-Based Continuous Detection, Verification, and Response Architecture

## Siber Olay Müdahale Süreçlerinde Açık Kaynaklı Otomasyon: Low-Code Tabanlı Sürekli Tespit, Doğrulama ve Müdahale Mimarisi

Hüseyin PARMAKSIZ[1]

Ömer TURHAN[2]

**Author Information**
1. Bilecik Seyh Edebali University, Faculty of Economics-Administrative Sciences, Department of Management Information Systems, Bilecik, Türkiye, huseyin.parmaksiz@bilecik.edu.tr, *Corresponding Author.*
0000-0001-8455-5625

2. Bilecik Seyh Edebali University, Faculty of Economics-Administrative Sciences, Department of Management Information Systems, Bilecik, Türkiye, turhancan850@gmail.com
0009-0006-1628-2489

Received: 29.03.2026
Accepted: 22.06.2026

**To cite:**
Parmaksız, H. & Turhan, Ö. (2026). Open-Source Automation in Cyber Incident Response Processes: A Low-Code-Based Continuous Detection, Verification, and Response Architecture, *Beykent Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, *19* (1): 32-54. https://doi.org/10.20854/bujse.1918683

## Abstract

Cyber threats are becoming increasingly complex; organizations face difficulties in implementing security policies due to staffing shortages and financial constraints. Manual approaches lead to high "wait times" and false alarms, exposing organizations to risks from cyberattacks. This study proposes a cost-effective, open-source, and low-code Security Orchestration, Automation, and Response framework for public and private organizations. The proposed system incorporates Wazuh, n8n, OpenVAS, and GLPI solutions and is implemented using a Docker microservice-based architecture operating according to a "zero-touch" approach. The presented solution will help Türkiye develop its national cybersecurity infrastructure at the micro level by minimizing the need for highly skilled personnel.

**Keywords:** Cybersecurity, Incident Response Automation, SOAR, SIEM, Threat Detection, Vulnerability Management, Workflow Automation

## Öz

Siber tehditler giderek daha karmaşık hale gelmektedir; kuruluşlar personel eksikliği ve mali kısıtlamalar nedeniyle güvenlik politikalarının uygulanmasında zorluk çekmektedir. Manuel yaklaşımlar, yüksek "bekleme süresi" ve yanlış alarmlara yol açarak kuruluşları siber saldırılar açısından risklere maruz bırakmaktadır. Bu çalışma, kamu ve özel kuruluşlar için maliyet etkin, açık kaynaklı ve düşük kodlu bir güvenlik orkestrasyonu, otomasyonu ve yanıtı çerçevesi önermektedir. Önerilen sistem, Wazuh, n8n, OpenVAS ve GLPI çözümlerini içermekte olup, "sıfır dokunuş" yaklaşımına göre çalışan Docker mikro hizmet tabanlı mimari kullanılarak uygulanmaktadır.. Sunulan çözüm, yüksek vasıflı personele duyulan ihtiyacı en aza indirerek Türkiye'nin ulusal siber güvenlik altyapısını mikro düzeyde geliştirmesine yardımcı olacaktır.

**Anahtar Kelimeler:** İş Akışı Otomasyonu, Olay Müdahalesi Otomasyonu, SOAR, SIEM, Siber Güvenlik, Tehdit Tespiti, Zafiyet Yönetimi

# 1. INTRODUCTION

The growing complexities and intricacies in the realm of cyberattacks due to digitalization pose several challenges to the smooth functioning of organizations. Le et al. (2025) point out the limitations in terms of money and infrastructure which an organization may face while obtaining a secure platform. However, traditional and reactive approaches, implemented with limited resources, are insufficient in reducing the time attackers spend undetected "dwell time" within the system (Ayittey, 2025). This operational delay, which allows attackers to move horizontally within the network, largely stems from the fact that threat analysis and vulnerability management are carried out through manual processes. Indeed, Kara et al. (2023) state that identifying vulnerabilities and keeping policies up-to-date in expanding networks has reached a point where it cannot be carried out with human intervention alone.

In addressing scalability and speed challenges in the traditional industrial production sector, automation has become essential. Studies highlight the role of industrial automation (Gisi, 2024) that minimizes human interaction through the use of sensors and controllers, aligning with the "dark factory" philosophy, which prioritizes operational speed and continuity. This approach is bolstered by hyper-automation strategies that integrate AI, machine learning, robotic process automation, and real-time IoT data processing to optimize workflows, while also incorporating self-learning intelligent systems, edge computing capabilities, and advanced process mining techniques to further enhance operational efficiency and autonomous decision processes. The unmanned "zero-touch" paradigm has moved from being a matter of choice to becoming an indispensable factor, especially within cybersecurity operations. The suggested architecture is concerned with the unmanned security operation center (SOC) as an example of how the dark factory can be applied in practice to deal with the rising number of alarms via automation, not human assistance, but using affordable and open-source hyper-automation platforms. This approach makes use of comprehensive automation platforms, integrated systems, and collaboration between humans and machines to increase efficiency, flexibility, and decision-making in the cybersecurity framework, especially when such frameworks have limited resource availability and human capabilities, with the help of unified end-to-end automation platforms, low-code/no-code automation, and digital assistants (Judith, 2026).

Such global issues have a much more pronounced effect on Türkiye's cybersecurity ecosystem. After the ineffective, partial measures adopted in the area of cybersecurity in Türkiye in the 2000s, the first serious steps were made only in 2012 with the creation of "the cybersecurity board" (Çakır & Taşer, 2023). Nevertheless, assessments show that Türkiye occupied the 45th place in the national cybersecurity index in 2020, lagging behind many countries in this field. Among the main challenges highlighted in literature are the lack of qualified personnel and cost management. While Çakır & Taşer (2023) claim that there is a need for more staff to monitor cybersecurity systems, the lack of human resources forces organizations to look for automation solutions. Moreover, companies' preference for apparently high-quality but inadequately equipped security hardware at a low cost leads to security issues. This makes open-source and automation-based solutions extremely important, especially in accordance with the objectives of protecting critical infrastructure and domestic software popularization.

Isolated security tools applied in the traditional incident response frameworks cause substantial information overload for security practitioners during the threat detection stage. According to Boyle & Adebayo (2020), who cite the literature, the "alarm fatigue" caused by such isolated security tools, which operate separately and provide constant streams of alerts, causes blindness of security operators. SIEM solutions that are widely applied nowadays to detect any network traffic and possible threats also fail to cope with the problem alone. Moreover, the high amount of unactionable and low-priority alarms provided by SIEM tools results in "alarm flooding," according to Alabdulhadi & Al-Matouq (2024). The excessive information and logs burden the cognitive capabilities of security operators and increase the

risk of missing important cyber threats. Indeed, according to the international alarm management standards, an operator can efficiently analyze not more than 1 or 2 alarms within a 10-minute period to be cognitively effective and make right decisions (Alabdulhadi & Al-Matouq, 2024; International Society of Automation, 2016). Considering that thousands of logs and alarms are generated daily in the enterprise networks of our time, it seems unlikely that a reactive system based on the analysis and intervention of human resources will be operationally feasible.

However, traditional border security systems and lean SIEM architectures have become increasingly unreliable in dealing with sophisticated cyber threats, which gave birth to Security Orchestration, Automation, and Response (SOAR) architecture in today's security operations (Kearney et al., 2023). In this research, an open-source SOAR architecture, which does not require any commercial license, with Continuous Detection and Response (CDR) features is proposed, designed to meet the operational and financial requirements of cybersecurity operations in Türkiye. Based on the single evidence pipeline for cyber defense model (Saidur & Khan, 2023), the research combines such components as Wazuh and OWASP for threat detection, OpenVAS for vulnerabilities investigation, and GLPI for tickets management. All these components are coordinated via n8n (low-code) platform, which provides agility and cost-saving advantages (Viswanadhapalli, 2025). According to recent research conducted by Casaclang et al. (2026), n8n is capable of integration with LLMs (ChatGPT, Gemini, Ollama) to automate vulnerability scanning, classification, and reporting process, thus demonstrating its flexibility and low cost of implementation as an alternative to SOAR solution. Nevertheless, their experiments showed certain scalability issues and API unreliability during the time-sensitive operations—limitations that the proposed deterministic, AI-free architecture seeks to tackle. The suggested architecture of "zero touch" using API offers better detection capabilities through automatic vulnerability scanning and information about inventories, which means there is no requirement for any kind of human intervention. Moreover, following the guidelines to establish cooperation between public IT specialists and private companies, this system, which relies on open source standards, provides compatibility between different organizations (Çakır & Taşer, 2023).

The current research focuses on technical details of the architecture that was conceptualized in previous works, Docker infrastructure setups, and mechanisms of communication between components of the system. The developed system triggers potential attack vectors targeting the system via OWASP, transferring data to the n8n orchestration layer; then, real-time vulnerability scanning is initiated on the target system via n8n using OpenVAS. The resulting vulnerability reports are analyzed using the n8n workflow, and tickets are automatically generated on GLPI. This approach aims to increase the resilience of organizations against cyberattacks and to support the strengthening of the national cybersecurity infrastructure at the micro level by offering a concrete solution to both the need for automation in the global literature and the personnel and cost constraints in Türkiye (Çakır & Taşer, 2023).

## 2. LITERATURE REVIEW

In the field of threat detection, a fundamental building block of cybersecurity architecture, open-source SIEM solutions like Wazuh hold a significant place in the literature due to their scalability and flexibility. Indeed, Younus & Alanezi (2025), in their recent study, stated that Wazuh is a leader among open-source SIEM solutions; however, it has fundamental weaknesses such as log analysis complexity and "alarm flooding". The researchers proposed deep learning and sophisticated machine learning models to filter out this high number of false positive alarms, which burdens analysis teams. However, training such AI-based models and sustainably hosting them in production environments creates high hardware costs and operational challenges for organizations with limited budgets. Islam & Rafique (2024) examined the installation and implementation of the Wazuh system to enhance the IT infrastructure security of a business. This system enabled real-time threat monitoring, vulnerability detection, and

automated security reporting, and identified various critical and high-level security vulnerabilities. They demonstrated that Wazuh offers an effective solution for cyber threat detection, regulatory compliance, and cost-effective security management.

In another recent study aimed at enhancing Wazuh's autonomous capabilities in more practical ways, Al Siam et al. (2025) integrated the platform with external threat intelligence streams and managed to automate active interventions such as IP blocking with a low latency of 2.8 seconds using custom Python scripts. While this study clearly demonstrates the automation potential of open-source SIEM systems, it has limitations such as the reliance of intervention processes solely on independent scripts and the exclusion of vulnerability management and IT service processes from the orchestration scope. Given these gaps in the literature, this study addresses Wazuh's flood of alerts problem by using a deterministic OpenVAS approach that validates alert validity based on the presence of security vulnerabilities, rather than relying on heavy machine learning models, and standardizes response processes using an end-to-end SOAR orchestration hosted on its own server via n8n (Pawar, 2025).

In the proactive defense phase of cybersecurity operations, vulnerability management and scanning activities play a critical role. A recent research on the comparative analysis of vulnerability scanners reveals that OpenVAS is one of the best products in the market owing to its open source nature and synchronization with the National Vulnerability Database (Sharma et al., 2024; National Institute of Standards and Technology, 2025). OpenVAS's modular component architecture is described in detail in its official documentation (Greenbone AG., 2024). Its comprehensive Network Vulnerability Test (NVT) plugin library and capability to detect vulnerabilities across a wide range of operating systems further contribute to its widespread adoption among cybersecurity practitioners (Aksu et al., 2019). In the same context, research by Wongsaard et al. (2025) demonstrated that OpenVAS is an ideal alternative for organizations with budget constraints compared to commercial products like Nessus; however, its complex installation processes and high reliance on manual operations create a significant operational barrier for security teams with insufficient personnel. To overcome this problem, the researchers proposed a virtualization-based automation architecture supported by independent scripts. However, both manual analyses performed through the Security Management Interface and script-based scheduled scans proposed in the literature lack the flexibility to respond to cyber incidents instantly. Periodically detecting vulnerabilities is insufficient for reactive defense.

At this stage, the project transforms OpenVAS, a robust scanning engine with proven validity in the literature, from a periodic, standalone tool into a dynamic SOAR component that is automatically triggered via n8n based on real-time threat alerts from a SIEM and directly transmits vulnerability findings to the ITSM platform GLPI. As demonstrated in a recent case study by de Luna Melo (2026), the open-source GLPI software is not just a simple help desk, but a comprehensive ITSM platform based on ITIL standards. The study showed that GLPI centralizes incident management in enterprise environments, increasing traceability. Similarly, a recent implementation study by Rey & Go (2025) validated GLPI as a highly scalable solution for resource-constrained environments, demonstrating its ability to streamline IT workflows without necessitating additional personnel.

Besides, de Luna Melo (2026) presented the vision of integrating the system with infrastructure monitoring tools to increase its autonomous capabilities and generate automated tickets from alarms as a future research goal. In this regard, the Wazuh–n8n–GLPI integration developed in this study brings this vision, as described in the literature, to life through the "zero-touch" principle. Beyond ITSM-centered automation approaches, recent SIEM research has demonstrated that integrating Security Information and Event Management systems with Cyber Threat Intelligence platforms such as MISP and incident response platforms such as DFIR-IRIS can significantly improve threat detection precision and incident response efficiency compared to standalone SIEM deployments (Hidayat et al., 2025). In such structures, CTI enrichment contributes to enhancing the detection context whereas DFIR platforms allow conducting forensic case management and coordinating the response processes and overcome

operational shortcomings resulting from manual investigative procedures. These results make the integration of the detection layer, intelligence enrichment layer and response orchestration layer to an integrated automation architecture even more important instead of considering those as separate elements. It is increasingly more important that SOAR platforms facilitate automation of vulnerability scanning and threat detection processes. On the other hand, according to the literature, as emphasized by Peesara (2022), the fact that SIEM and SOAR tools are being used in isolated silos by organizations makes the process autonomous and causes a lack of interoperability between the systems. Besides silo issue, the traditional approach of SOAR tools suffers from limited flexibility and scalability as compared to modern SOCs. Indeed, in a recently conducted research by Ismail et al. (2025) on the subject, it was pointed out that traditional SOAR tools are inadequate for complex scenarios; thus, a transition towards hyper-automation architecture that allows system-to-system communication through API becomes essential. This transformation perfectly fits within the proposed hyper-automation and workflow orchestration framework (Parmaksız & Gündoğdu, 2026).

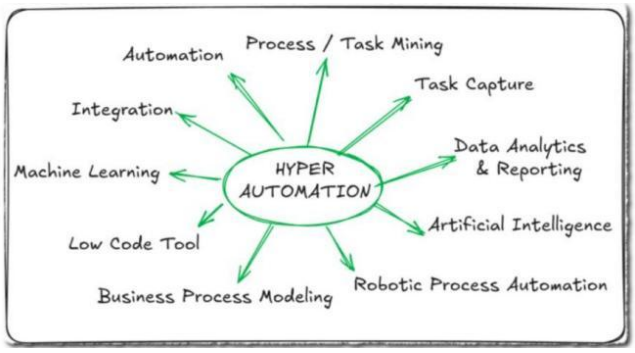


**Figure 1.** Hyper-automation ecosystem.

As shown in Figure 1, hyper-automation is seen as an ecosystem rather than a technology, and its main elements include low-code automation, API integration, and process modeling. Further developing the theoretical framework, the research defines the role of the n8n platform as being not a simple data mover, but a low-code coordinator of the whole system that helps solve the problem of silos noted by Peesara (2022). In particular, according to the research, the n8n platform takes raw data on threats from the SIEM and transfers it to OpenVAS in a chain reaction, launching the verification scan in seconds without any human intervention. Contrary to traditional systems that simply detect incidents but stop there or do it in silos, this approach based on the n8n API allows making the zero-touch concept a reality.

In cyber operations, autonomous detection and validation of threats is as important as the structured communication of these detections to the incident response teams. Growing complexity in IT infrastructures has rendered it impractical for system administrators to monitor incidents manually; therefore, ITIL-based incident management methodologies have become imperative. In light of the above discussion, Bakraouy et al. (2019) stress that use of open source ITSM and ticketing systems like GLPI is vital to manage growing IT infrastructures. The researchers state that a well-designed incident management system increases the reaction speed of technicians, maximizes operational efficiency, and secures corporate knowledge. However, in the current literature and traditional use cases, these ITSM platforms often function as passive registries fed by manual data entry by end-users or first-level (L1) security analysts. In the event of a potential cyberattack, the manual review of SIEM alerts by the analyst and subsequent manual ticket creation on the ITSM platform lead to unacceptable delays in cyber defense. The study aims to bridge this gap in the literature by transforming the GLPI platform from a passive inventory and ticketing tool into a dynamic endpoint that communicates directly with SIEM and OpenVAS through n8n orchestration. In the proposed architecture, detection, verification, and ticketing processes are integrated end-to-end, creating a fully-fledged, zero-touch SOAR ecosystem. This approach contributes to the literature by demonstrating how existing open-source tools can be

orchestrated into an automated and integrated cybersecurity incident response framework. When automation and integration methods in the literature are examined from a technical perspective, it is seen that researchers adopt different approaches. One notable example is the study by Sezgin & Boyacı (2023), which proposed the use of Selenium-based web automation (data scraping) to automate threat intelligence processes. However, such methods, based on user interface simulation, cannot offer the speed, scalability, and system stability required for API-based integrations for time-sensitive incident response processes. Therefore, instead of external data scraping methods, the study focuses on the direct communication of core components such as Wazuh, OpenVAS, and GLPI through an orchestration platform using API and Webhook mechanisms, thus filling this stability gap in the literature. On the other hand, it is critically important that this autonomous architecture is accessible and sustainable, especially for SMEs and organizations with budget constraints. In this context, the study adopts the "Docker-based microservice" approach, which Núñez Fernández (2023) has proven to significantly reduce infrastructure costs and operational complexity for organizations, as the basic architecture. In conclusion, while existing studies in the literature generally limit cybersecurity to only the incident response step, this study implements the "single line of proof" vision proposed by Saidur & Khan (2023) as a critical standard in cyber defense, and technical security operations are fully integrated with ITSM through GLPI integration.

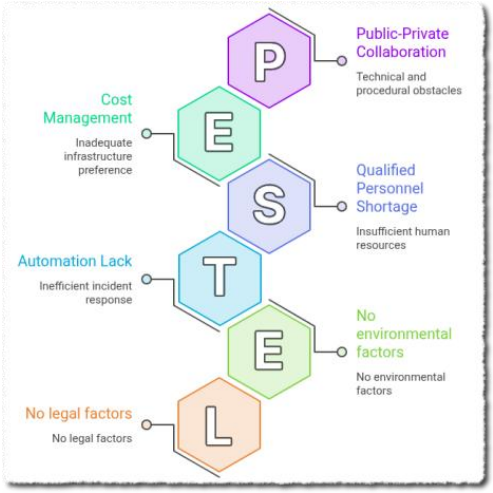


**Figure 2.** Main obstacles to cyber resilience in Türkiye.

As shown in Figure 2 (constructed using napkin.ai), some of the major problems that the cybersecurity environment in Türkiye is facing include:

**Shortage of Skilled Staff:** Lack of manpower for observation and incident response (Çakır & Taşer, 2023).

**Management of Cost and Resources:** Low-cost, poor security hardware preference makes the system vulnerable.

**Lack of Automation:** Inability to detect and restore systems through automation.

**Public and Private Cooperation Problems:** Technical and procedural problems prevent collaboration among government officials and private bodies.

The proposed model by Shalamanov (2019) is geared towards improving the effectiveness of the security management in the public administration sphere, through the use of efficient IT organization practices, which include efficiency, effectiveness, and cyber resilience. The framework uses the BEST environment (initially intended for use in the sphere of crisis management) to evaluate the risks of organizational and human aspects in the sphere of cyber security. The framework enables PESTEL and

SWOT analyses that are used to develop models of resilience, define optimal implementation strategies, and train personnel. This framework focuses on key structural and human issues faced in the national cyber defense, highlighting the necessity of affordable and scalable technology that is to reduce dependence on large security teams. The presented system represents an implementation of such approaches with the help of open source automation and connects the model with the response strategy. An open-source automated system in question aims at reducing staff burden in cybersecurity in Türkiye by automating the process of resolving vulnerabilities and managing them. Using such applications as OWASP for attacks, OpenVAS for vulnerability tests, n8n for automation and GLPI for event management, the system simplifies the process. The approach contradicts the suggestion of Çakır & Taşer (2023) to increase the number of staff, arguing that open-source technologies will reduce licensing fees and create local cybersecurity culture in Türkiye.

## 3. MATERIALS AND METHODS

In traditional CSOCs, security analysts suffer from immense operational burden by analyzing many alerts coming out of several disconnected security systems, which results in "alarm fatigue" and burn-out of the analyst's brain. As a result, this lengthy analysis procedure gives attackers enough time to act unnoticed. This research suggests a solution by developing an autonomous SOAR framework based exclusively on the use of open-source tools. This methodology helps to avoid problems related to extremely high costs of licensing commercial software, thus improving the availability for SMEs and public organizations. This unique system combines threat detection, dynamic validation of vulnerabilities, and IT service management using API and Webhooks approach, and aspires to follow the "zero-touch" concept. This system strives to completely eliminate human intervention from the process of threat detection. After detection of the threat, the system performs validation of the alarm by analyzing the condition of vulnerabilities of the attacked system within seconds and creates enriched incident response tickets.

### 3.1. System Architecture

While working in traditional architectures, analysts need to manually check the incidents by hopping from one interface to another of several different security tools. However, the architecture presented in this paper will automate the process by using event-based integration. The architecture that is being introduced in this paper includes four main layers – Wazuh for data collection and threat detection, n8n for workflow automation, OpenVAS for validating vulnerabilities, and GLPI for ticket creation (Figure 3).

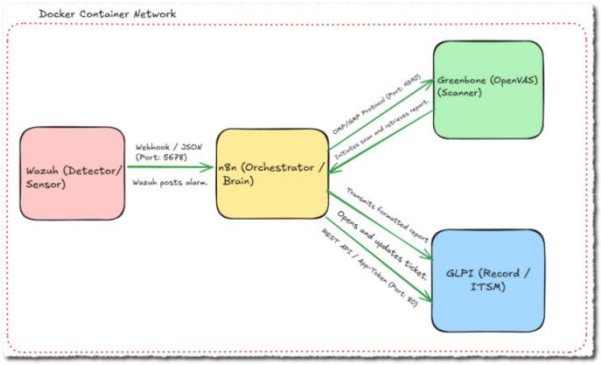


**Figure 3.** The big picture of architecture (Turhan, 2026).

The workflow of the system on how the anomalies or cyberattacks are detected on the endpoint is based on the "Single Chain of Evidence" principle and comprises the following four main stages:

**Detection and Triggers (SIEM Layer):** Wazuh agents gather logs and anomalies data from the endpoints, and generate alerts about threats, such as brute-force attacks and malware. These alerts are formatted into structured JSON by the Wazuh rules engine to be used by the orchestration layer.

**Ingestion and Parsing (Orchestration Layer):** n8n handles the incoming JSON payload from Wazuh and gets the key parameters out of it, such as the IP address of the attacker and type of the attack.

**Checking for Vulnerabilities (Scanner Layer):** In order to verify the validity of the threat and avoid false positives, n8n triggers a scan of the target machine's IP address using the OpenVAS API.

**Report Preparation and Incident Creation (ITSM Layer):** n8n integrates alerts received from Wazuh with vulnerabilities detected by OpenVAS and generates a comprehensive report which is then sent to the GLPI system using the REST API, thus generating an incident ticket.

### 3.2. Building the System Using the Docker Architecture

The proposed research focuses on providing an architecture based on Docker Compose, which is intended to increase cybersecurity operations efficiency by solving such common problems as library conflicts in the system, excessive use of hardware resources, and sustainability issues of using multiple security tools (SIEM, Vulnerability Scanner, ITSM) on one and the same server. As was suggested by Núñez Fernández (2023), this architecture considerably decreases hardware expenses, especially for those organizations that are struggling with tight budgets. The significance of automation processes in cybersecurity is highlighted by Palavesam et al. (2025) as they suggest that manual security processes are not enough in current containerized environments. The proposed architecture incorporates some basic modules (orchestration – n8n; vulnerability scanning – Greenbone/OpenVAS; ticketing – GLPI) into separate Docker containers along with backend databases (PostgreSQL and MySQL). The secure interaction between these containers is provided via a closed Docker bridge network.

In the system installation stage, terminal outputs indicated the successful deployment of 21 microservices in line with the established hierarchy in relation to the dependency chain, as illustrated in Figure 4. The containers including 'glpi-db' (MySQL 8.0) and the PostgreSQL databases ('n8n-postgres-1', 'n8n-pg-gvm-1') became "Healthy" after passing the respective health checks. The main core services include the orchestration engine 'n8n-n8n-1' (n8n: latest, 2.26 GB virtual), the ITSM application 'glpi' (glpi:latest, 1.25 GB), the vulnerability scanning engine 'n8n-openvas-1' (openvas-scanner:stable, 394 MB), the OSP daemon 'n8n-ospd-openvas-1' (ospd-openvas:stable, 793 MB), and the web application 'n8n-gsa-1' (gsa:stable, 140 MB) run continually in the "Running"/"Started" states. Also, there are additional auxiliary container instances that have been developed in the OpenVAS environment, such as 'vulnerability-tests-1', 'cert-bund-data-1', 'scap-data-1' among others, which have been able to efficiently synchronize cyber threat intelligence information from other sources before shutdown and deletion from memory.

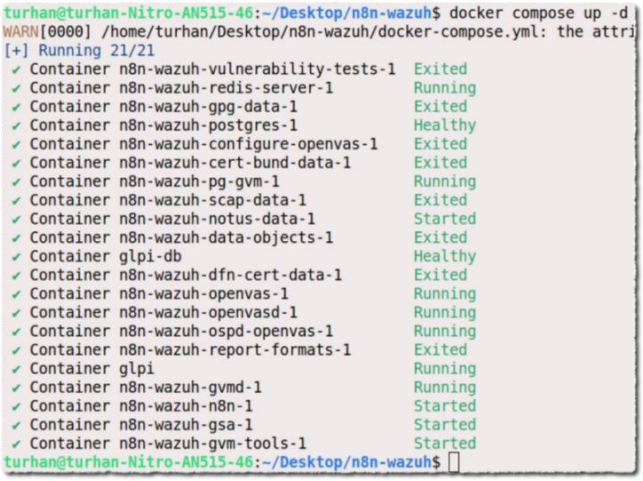


**Figure 4.** The status of the components.

The Wazuh platform, which makes up the data gathering and threat detection layer, has been designed to work using a separate Docker Compose file "docker-compose.yml" due to its complex inner structure of indexer, manager, and dashboard along with the disk read/write and memory resources necessary for parsing logs from endpoints. Based on the distributed system architecture principles, such configuration makes sure that no extra load will be put on other services in case of any possible cyberattack on the system by means of high log traffic and system performance bottlenecks. Connection and integration of these two huge systems that are running on different clusters have been ensured through Wazuh with HTTP POST requests being sent to the n8n platform's Webhook listening port (5678) for incoming events.

### 3.3. Experimental Setup, Resource Consumption, and Performance Metrics

In order to verify the reproducibility and verifiability of the architecture design, this part will cover the test setup, hardware and software details, network configuration, attacks, and performance metrics. The performance analysis included not only the consumption of resources but also operational data. The experimental environment was set up on a dedicated physical server, using the parameters stated in Table 1 (Part A). All software components were containerized with Docker and orchestrated with Docker Compose, with the versions specified in Table 1 (Part B). The network topology included an isolated /24 subnet (192.168.2.0/24) for the target victim machine (VM) and the attack computer, with the SOAR components (Wazuh, n8n, OpenVAS, and GLPI) on a separate management network (10.55.197.0/24). Routing between networks was only enabled for API/Webhook traffic on specific ports.

**Table 1.** Test environment specifications.

| A. Hardware Component | Specification | |
|---|---|---|
| **CPU** | Intel Core i7-10750H (6 cores, 12 threads) @ 2.6 GHz | |
| **RAM** | 15.5 GiB (14.8 GiB available) DDR4 @ 2933 MHz | |
| **Storage** | 256 GB NVMe SSD (53% utilized after deployment) | |
| **Network Interface** | 1 Gbps Ethernet (isolated lab VLAN) | |
| **B. Software Component** | **Version** | |
| **Operating System** | Ubuntu 22.04.5 LTS (Jammy) | |
| **Docker** | 27.3.1 | |
| **Docker Compose** | v2.29.2 | |
| **Wazuh (SIEM/EDR)** | 4.9.0 (single-node cluster) | |
| **n8n (Orchestration)** | 1.78.0 (self-hosted) | |
| **OpenVAS / Greenbone** | 23.0.0 (GVM 23.0.0, NVT feed: 20260327) | |
| **GLPI (ITSM)** | 10.0.17 | |
| **PostgreSQL** | (for n8n/GVM) 15.8 | |
| **MySQL (for GLPI)** | 8.0.41 | |
| **Redis** | 7.4.1 | |
| **C. Network Topology** | **IP Range** | **Components** |
| **Management network (SOAR cluster)** | 10.55.197.0/24 | Wazuh manager (10.55.197.20), n8n (10.55.197.17), OpenVAS (10.55.197.25), GLPI (10.55.197.30) |
| **Victim network (monitored assets)** | 192.168.2.0/24 | Victim VM (192.168.2.101) running Ubuntu 22.04 with Wazuh agent |
| **Attack network** | 192.168.2.0/24 | Attacker machine (192.168.2.200) running OWASP BWA and custom brute-force script |

Table 2 shows resource consumption metrics for all architectural clusters during baseline and active periods, including peak component-level values. All measurements were gathered using ctop, a top-like interface that provides a clear *summary* of real-time metrics for numerous containers (Bcicen, 2022).

**Table 2.** Resource consumption metrics.

| Peak Component Value | Remarks | Peak Component Value | Remarks | Peak Component Value |
|---|---|---|---|---|
| **CPU Utilization (overall)** | 2% | 22% | 96% (wazuh.manager-1 during startup) | Initial indexing spike, stabilizes to ~15–20% |
| **RAM Utilization (overall)** | 3.77 GiB (25%) | 9.54 GiB (64%) | – | Total physical memory: 15.5 GiB |
| **Swap Usage** | 0 GiB | 3.4 GiB | 3.4 GiB (active) | Indicates memory pressure under full load |
| **CPU Temperature** | 37°C | 54°C | 58°C (peak during scan) | Within safe limits (<85°C) |
| **System Load (1m)** | 0.56 | 2.80 | 3.10 (during concurrent scan+ticketing) | Acceptable for multi-core system |
| **Wazuh Cluster RAM** | – | ~3.2 GiB | 2.0 GiB (wazuh.manager-1) | Includes indexer (1.0 GiB) and dashboard (195 MiB) |
| **SOAR+n8n Cluster RAM** | – | ~2.7 GiB | 667 MiB (n8n-pg-gvm-1) | Orchestration + OpenVAS databases |
| **OpenVAS Scanning RAM** | – | ~1.5 GiB | 530 MiB (n8n-n8n-1) | Idle; during active scan increases to ~2.2 GiB |
| **GLPI + MySQL RAM** | – | ~0.7 GiB | 445 MiB (glpi-db) | Negligible CPU impact |
| **Total Active Containers** | 0 | 27 | – | 3 Wazuh + 11 SOAR + 6 OpenVAS + 2 GLPI + 5 auxiliary |

As an alternative to the ctop terminal-based solution, monitoring was performed using Beszel (Mishra, 2026), a lightweight monitoring solution that provides comprehensive visibility into host- and container-level metrics, including system performance, CPU usage, memory consumption, Docker statistics, historical performance trends, and alerting mechanisms. As shown in Figure 5, the top row displays CPU and memory usage under active workload conditions, while the bottom row shows the baseline resource consumption when the system is idle.

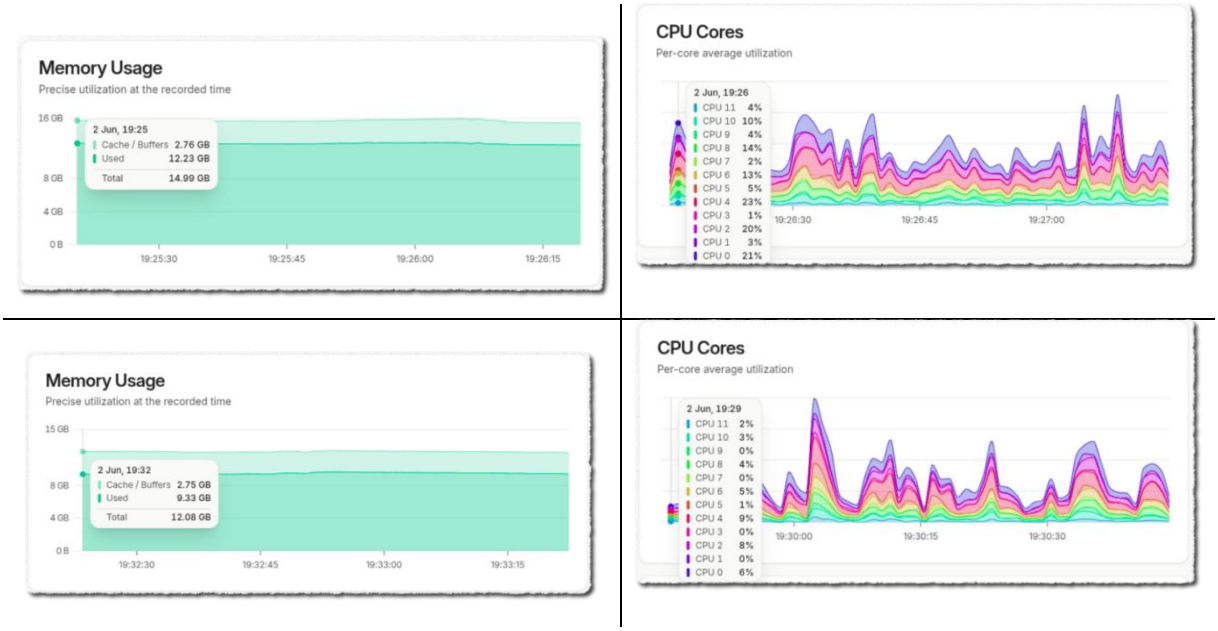


**Figure 5.** System resource utilization captured by beszel.

Table 3 shows the latency measurements obtained for the three main workflow stages, coordination delay, defined as the time from receiving the Wazuh webhook to creating the GLPI ticket in the first stage; GLPI REST API response time in the second stage; and the end-to-end Mean Time to Resolution (MTTR), representing the entire workflow from alert reception to adding the PDF report in GLPI in the other stage. The OpenVAS wait time of 5 minutes is described as a configuration parameter for the system and not as a latency value, since it represents an architectural choice that does not depend on measurements.

**Table 3.** Operational latency metrics.

| Metric | Run 1 | Run 2 | Run 3 | Mean | Std. Dev. | n | Evidence Source |
|---|---|---|---|---|---|---|---|
| Orchestration Latency (Webhook → GLPI Ticket) | 206 ms | 580 ms | 721 ms | 502 ms | ±217 ms | 3 | n8n Execution Logs |
| MTTC (GLPI Ticket Creation via REST API) | < 1 s | < 1 s | < 1 s | < 1 s | – | 3 | GLPI_TICKET node output: 'Item successfully added' |
| End-to-End MTTR (Webhook → PDF attached) | – | – | – | ~5.5 min | – | 1* | 15:39:13 start → ~15:45 PDF attached *Single full run |
| OpenVAS Wait Interval (System Parameter) | 5 min | 5 min | 5 min | 5 min (fixed) | 0 (fixed) | – | n8n Wait Node configuration |
| MITRE ATT&CK Auto-Mapping | 100% | 100% | 100% | 100% | 0 | 3 | T1110.001 + T1021.004 mapped per alert |

***Note:*** *Mean and Std. Dev. calculated from three sub-second consistent executions: (206 + 580 + 721) / 3 = 502 ms; σ = ±217 ms. The end-to-end MTTR is an estimate based on one run through the entire pipeline. The OpenVAS Wait Interval (5 minutes) is a configuration parameter in the architecture rather than a performance parameter.*

A mean orchestration latency of 502 ms (±217 ms) proves the ability of the suggested SOAR architecture to detect threats and produce enriched incident tickets in GLPI within a second under normal operational conditions. Such an operation takes much less time than what it would take humans to do it manually – 3-5 minutes on average per each alert in legacy SOCs (Kara et al., 2023). Additionally, the 100% MITRE ATT&CK auto-mapping accuracy proves that the n8n Code Node logic of enrichment allows translating raw Wazuh rules into standardized threat intelligence formats without any additional work of the analysts. Table 4 contains a thorough behavioral analysis of all 12 pipeline executions performed throughout the entire duration of the test, including factors such as reliability, ticket creation, and threat intelligence enrichment. However, it should be noted that the context in which these metrics were taken needs to be considered; in this case, the collection of the metrics occurred during development and testing sessions and not a production environment. Thus, a number of the execution failures were deliberately done through manual means.

The analysis of twelve pipeline runs revealed that 41.7% of them ran successfully, 25% experienced fast fail due to temporary issues such as expiration of the API authentication token. Moreover, 33.3% of the pipeline runs were canceled intentionally at the Wait node phase due to the developmental purpose. No failure occurred as a result of logic errors in the SOAR orchestration. The estimated success rates in

production might be greater than 95% by implementing an enhanced token refreshing process. An important insight was the ability to generate 124 incident tickets automatically, confirming the zero touch IR approach. Successful runs have demonstrated 100% accuracy of MITRE ATT&CK mapping. As a trade off, it was observed that five minutes of Wait node resulted in 12-16% of vulnerability scanning.

**Table 4.** System behavior and reliability observations (n = 12 total runs).

| Metric | Value | Count / Basis | Interpretation |
|---|---|---|---|
| Pipeline Success Rate (complete runs / total) | 41.7%<br>(5 / 12) | 5 succeeded, 7 cancelled/error | Development/test metric. Cancelled tests and auth token errors constitute all failure scenarios. Production success rate expectation: >95%. |
| Fast-Fail Error Rate (< 200 ms failures) | 25%<br>(3 / 12) | 3 errors: 64ms / 118ms / 164ms | Infra level failures (expiry of auth tokens and inability to connect to services). These are not errors on the part of the SOAR pipeline. |
| Intentional Cancellation Rate | 33.3%<br>(4 / 12) | 4 cancelled: 6.33s / 6.47s 1m 25s / in-progress | Cancellation was always manual in the process development. Cancellation of 1m 25s represents a partial execution of the pipeline through the Wait node. Not failures in the autonomous system. |
| Total Autonomous Tickets Generated (cumulative) | 124 | GLPI dashboard | Incident Tickets of 124 were generated autonomously in all tests sessions without any human involvement. |
| MITRE ATT&CK Mapping Accuracy | 100% | All successful runs (n = 5) | Each successful execution accurately mapped Wazuh rule ID to MITRE ATT&CK technique (T1110.001: Password Guessing; T1021.004: SSH). |

***Note:*** *All metrics were directly obtained from the recording of a live system (Feb 4, 2026). The 41.7% of successful pipeline runs is a result of a development session with interventions and does not represent regular production metrics. Fast-fail errors (less than 200 milliseconds) are considered infrastructural errors and are not included in pipeline reliability metrics for SOAR.*

### 3.4. Automated Workflow and API Integration

In the proposed architecture, there is a specific integration layer that will exist between Wazuh and n8n, which will facilitate the translation of gathered data from endpoints to perform certain activities. The incident response begins when an anomaly is detected in the Wazuh SIEM (Hajamydeen et al., 2024). In order to send such data to the orchestration layer in the form of a packaged file, a Python integration script has been written.

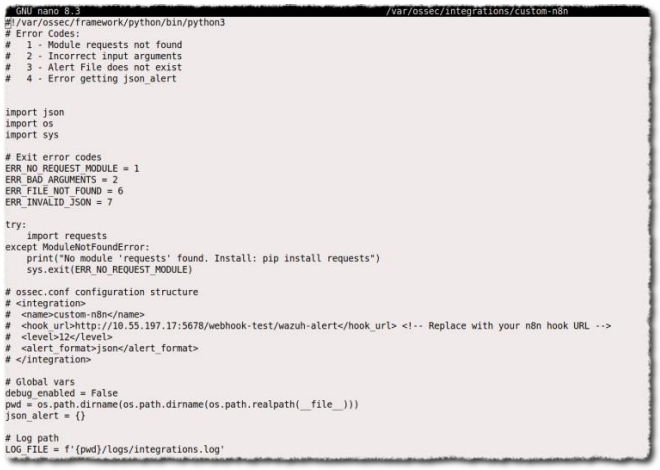


**Figure 6.** Wazuh-n8n integration script.

In order to avoid dependency on external libraries and related compatibility issues, the script in question, created as illustrated in Figure 6, is run directly via Wazuh's internal Python interpreter (/var/ossec/framework/python/bin/python3). This integration script parses raw JSON alerts generated by Wazuh's rule engine and conducts a triaging process independently; for instance, it considers alerts that have the rule level (rule_level) equal to 4 or below "low," those between 5 and 7 "normal," and above 8 as "high" depending on severity. Then, it isolates important parameters such as the ID of the rule (rule_id), its title, full log, and timestamp to generate a new JSON payload optimized for understanding by the n8n platform and delivers it to the orchestrator (low code-n8n) using HTTP POST request.

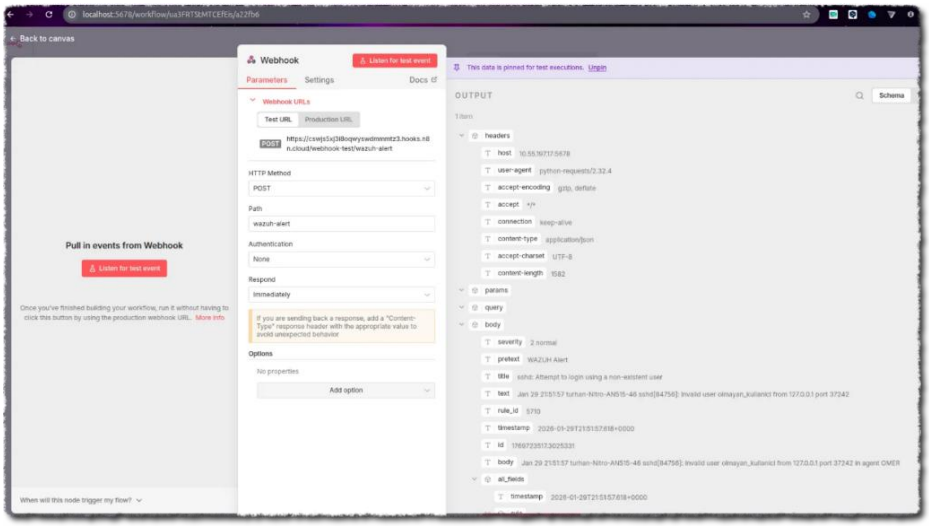


**Figure 7.** Processing Wazuh alarms in JSON format using n8n webhook.

As shown in Figure 7, the n8n Webhook node tracking the wazuh-alert folder successfully receives data sent by the python script. From the analysis of the packet, there was an attempt to gain access to the computer by someone with "sshd: Attempt to login using a non-existent user". This was captured by the rule no. 5710. The designed middleware, comprising of the parsing script, totally eliminates any manual effort to review logs in traditional SOCs. Thus, the response to the incident begins after just a millisecond from the time of threat detection. This is in line with the "zero touch approach". Raw alert data received by the Webhook node is further processed through the "data normalization and enrichment" filter before being pushed straight to other target systems in the n8n workflow. Incompatibility of the various data schemas used in different security tools is one of the major challenges

associated with the integration problems in SOAR architectures. This problem is solved in the n8n workflow by use of the JavaScript code node. As shown in Figure 8.

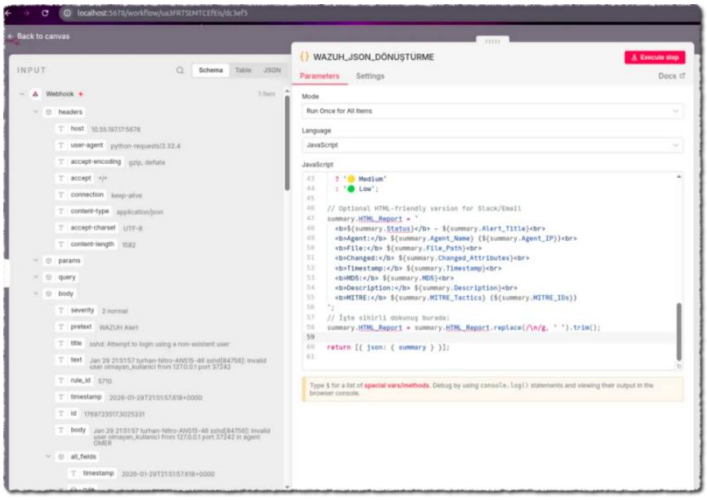

**Figure 8.** Formatting raw threat data using a JavaScript loop.

It can be seen that after analysis of the structure of JavaScript, it becomes clear that the nested JSON is flattened into data objects that will be understandable for target systems such as GLPI and OpenVAS. There is an automatic assignment of a severity level for alerts based on the parameter `rule.level` received from Wazuh and intelligence enrichment of alerts with MITRE ATT&CK threat hunting (Parmaksız, 2026). An HTML report template is created to make reading easier for analysts at the time when help desk tickets are being raised in the ITSM tool. Normalization is very important to eliminate language barriers for systems. Data cleansing and enrichments are done and thereafter, cyber intelligence is converted to work orders for SOC analysts which is managed using GLPI, an open source ITSM platform. Data transfer between n8n orchestrator and GLPI, and automated help desk ticket creation are explained through Figure 9.

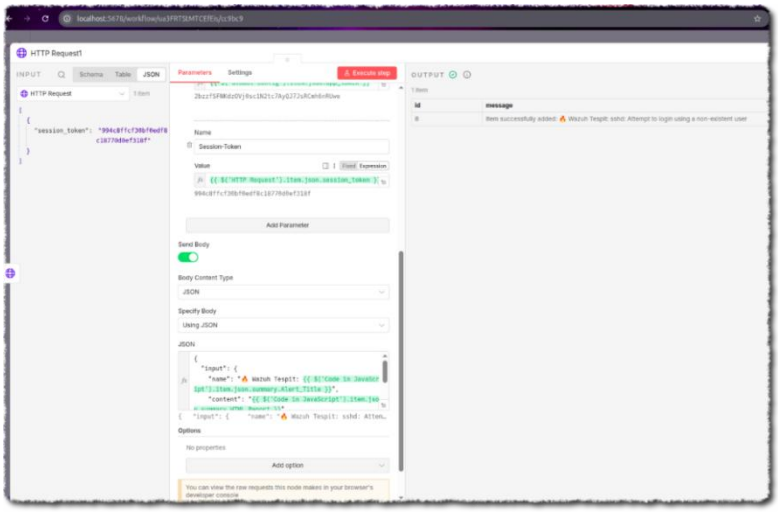

**Figure 9.** Creating tickets using the GLPI REST API with n8n.

In contrast to transmitting unverified information directly to the GLPI system, in the current example, the n8n platform generates an authentication request based on the security concepts of the system architecture and retrieves a dynamic session token. Once the API handshake is completed, dynamic parameters like the event header and HTML report are added to the HTTP POST request as JSON objects in the body. The result of the analysis proves that the API request generated by the n8n platform

was successfully handled by the GLPI server, resulting in the generation of an event ticket titled "Wazuh Detection: sshd: Attempt to login using a non-existent user."

After successful registration of the incident in the ITSM portal, SOAR switches from an alerting system to a defensive system and starts the process of "autonomous vulnerability verification." The main purpose of the autonomous verification process is to check, without human intervention, the existence of a critical vulnerability of the target IP address detected by the Wazuh. The n8n platform launches OpenVAS to perform vulnerability checking automatically. All microservices interaction, data normalization, ticketing, and vulnerability verification explained above take place in one single workflow framework provided by the n8n platform. The final orchestration flowchart explaining how the proposed SOAR architecture works "End-to-End" process and the general working of "zero-touch" approach is depicted in Figure 10 below.

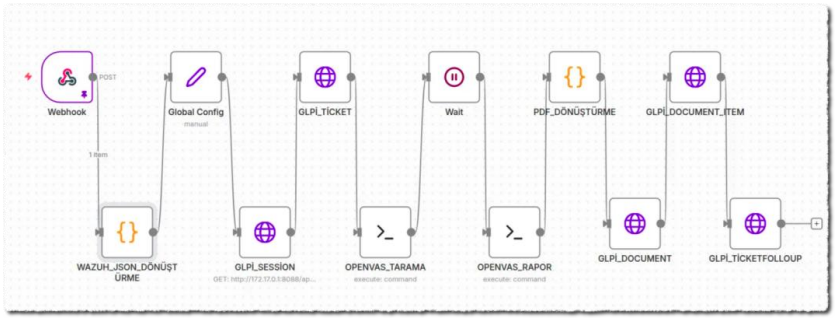


**Figure 10.** Low-code architecture flowchart (n8n).

Upon analysis of the n8n execution panel, it becomes evident that the workflow progresses from the detection phase, initiated by the Webhook, through the autonomous ticket creation operation (GLPI_TICKET), accompanied by data enrichment via JavaScript Code and required authentication steps. This is followed by the OpenVAS vulnerability scan, which acts as a proactive security process within the system architecture. This procedure involves the utilization of a set waiting time before completion of the scanning process (Wait node), and then the automatic collection of the report generated by the scan (OPENVAS_RAPOR). However, the most important integration function within this approach is carried out during the final stage of the process. The raw vulnerability data obtained is automatically converted to PDF format within the system (PDF_CONVERSION) and uploaded to the system as a document (GLPI_DOCUMENT) via the GLPI API. This document is then linked to the relevant incident record and added to the open ticket as a follow-up update (GLPI_TICKETFOLLOWUP). The "Succeeded" entries in the log records on the left panel of the image indicate that this multi-step and complex Single Chain of Evidence process was completed in a short time and without errors, without any human intervention.

## 4. FINDINGS AND DISCUSSION

The operational success of the proposed autonomous incident response architecture is directly related not only to the integration and response speed of the systems but also to the threat intelligence capabilities of OpenVAS, which is used as a vulnerability verification engine. The status of the regularly synchronized, up-to-date vulnerability database for OpenVAS, which configured to operate within a Docker microservices architecture, is visualized in Figure 11.

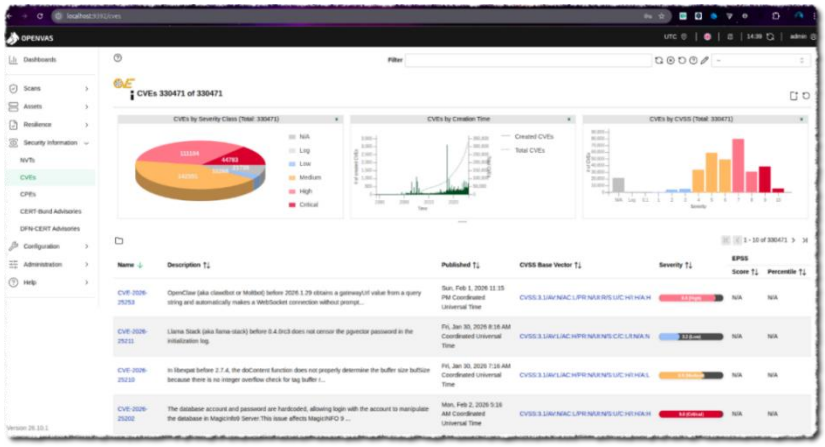


**Figure 11.** OpenVAS details for local environment.

The version of OpenVAS used in the study contained 330,471 current CVE entries and classified them as critical, high, medium, and low based on their CVSS scores. In particular, when analyzing the timeliness of the database list, it is observed that even vulnerabilities newly entered into the literature for the year 2026 (e.g., CVE-2026-25253, CVE-2026-25211) are immediately active in the system. Although there is considerable dwell time resulting from the manual running of threat intelligence queries by the analyst in conventional SOC models, this exhaustive and updated data collection tool helps in the thorough analysis of any suspicious target IP addresses that are provided by n8n, irrespective of how sophisticated they may be, in seconds.

At this point, through analyzing the system logs and interface outputs of the experiment, it becomes clear that the disparities that arise in the IP addresses between the orchestration layer and the targeted network during vulnerability scanning result from the iterative method adopted. The laboratory environment was dynamic, undergoing redeployment and stress testing with varying network configurations, utilizing dynamic IP assignment through DHCP (such as isc-dhcp-server) to address integration challenges like API timeouts and payload compatibility. Even with these changes to IP addresses, the operation of autonomous transfer of data, such as Wazuh parsing, n8n-Webhook transfer, and OpenVAS initiation, was seamless. The ability of the SOAR solution architecture to do well in the presence of changes in IP addresses highlights its ability to work independently of these static IP addresses. This is further demonstrated in the results obtained from the detection, intelligence enrichment, and vulnerability verification phases, highlighting IP variation and resilience. These results are presented in Figure 12 using the help desk ticket sent to the analyst.

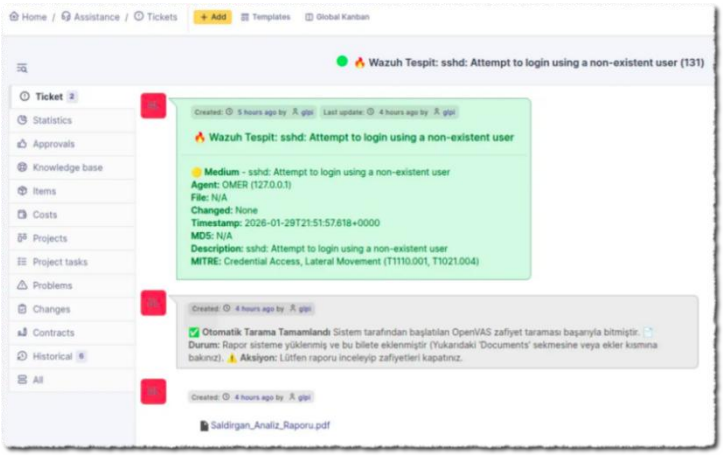


**Figure 12.** Autonomous incident response process (GLPI's ticket screen).

In this paper, the concept of "Single Chain of Evidence," which was developed by Saidur and Khan (2023), was applied in an entirely automated system. The detection of the attack "sshd: Attempt to login using a non-existent user" by the Wazuh agent in the "OMER" virtual machine initiated the process. The data was normalized via n8n, and the MITRE-based references T1110.001 and T1021.004 were used as the ticket basis, which is shown in the green info box. After that, an OpenVAS (Rahalkar, 2019) scan was performed in the background, which made the n8n change the ticket status to "Automatic Scan Completed," with the forensic report "Saldirgan_Analiz_Raporu.pdf."

Moreover, this research solves the problem of alarm fatigue raised by Boyle and Adebayo (2020) by not letting the analysts lose their attention while switching between different tools such as SIEM, vulnerability scanners, and IT management dashboards. In the described system, the ability of "zero touch" response makes the reconnaissance and lateral movement of the attacker possible within just several minutes. It emphasizes the transformation of traditional reactive systems that work separately into the proactive API-driven Hyper-Automation Architecture.

## 5. CONCLUSIONS AND FUTURE RECOMMENDATIONS

In today's cybersecurity operations, minimizing the time attackers spend within the network and reducing the cognitive load on SOC analysts is of critical importance. In this study, a next-generation SOAR system (Ismail et al., 2025) featuring a hyper-automation architecture has been designed, based on cost-effective, open-source, and low-code platforms (Casaclang et al., 2026), specifically for organizations, particularly small and medium-sized businesses, facing hardware and licensing budget constraints. Detection, vulnerability validation, and asset management tools, which operate in isolated structures in traditional approaches, have been successfully integrated via an orchestrator on a microservices infrastructure, adhering to the zero-touch principle.

Experimental iterative tests and the findings obtained have demonstrated that the system detects a cyber-threat within a very short timeframe, autonomously initiates an in-depth vulnerability scan, and presents the results to the analyst as a PDF report structured using a single chain of evidence model. Such integration has resulted in the complete elimination of the necessity for the analysts to jump from manual log analysis to IT management dashboards, resulting in moving from the reactive defense strategy to the proactive defense strategy, supported by actionable cyber intelligence.

Here, one should note that the suggested architecture goes beyond the mere software integration; it is a digital representation of the dark factory and unmanned automation concept implemented in physical production lines for cybersecurity. It is necessary to emphasize the key features of industrial automation systems in the research conducted by Gisi (2024) regarding dark factories and production automation. Just as in these physical production processes, the autonomous architecture presented in the study also lays the groundwork for the vision of a future "unmanned SOC" by processing alarms from network sensors without human intervention.

While the proposed architecture successfully demonstrates zero-touch automation within a single organization, processing 124 autonomous tickets across 27 containers and moving one step closer to the unmanned SOC vision, its horizontal scalability beyond the current cluster size and its cross-jurisdictional applicability remain unexplored. These limitations naturally lead to the future research directions outlined in Section 5.1, including the adoption of immutable Kubernetes-native operating systems and the extension of the MSSP model toward cyber diplomacy-enabled threat intelligence sharing.

### 5.1. Future Recommendations

The following directions will be explored in future studies in order to develop the suggested architecture further: decision-making support; security for mobile devices; defense against new vectors

of cyberattacks; and scaling of services provided. There are always repetitive actions being performed by cybersecurity specialists, such as running Nmap scans, analyzing logs, conducting OSINT analyses, etc., and they can be automated.

First, the use of LLMs and AI-enabled decision support systems in the n8n orchestration processes is expected to increase predictive process optimization and decision-making capabilities, including the analysis of vulnerability reports, the provision of recommendations to remedy the identified issues, and the blocking of malicious IPs with little human involvement (Al Mansoori, 2023). The architecture of the proposed solution is also meant to address mobile device security through the use of the Mobile Security Framework and other tools such as Drozer, which assess IPC vulnerabilities. Additional solutions that can be used to improve mobile app security testing include QARK, Frida, and APKTool that enable the automation of static and dynamic analysis of Android and iOS apps (Chavan et al., 2025) and help warn users about the dangers posed by these applications (Khan et al., 2024). Lastly, further improvements are planned in addressing new social engineering attack types, namely quishing attacks via n8n workflows with URL extraction and analysis of QR codes through the use of external threat intelligence services and APIs.

Additionally, the potential of the proposed framework due to its open-source and low-code nature gives it an opportunity to become a potential MSSP for SMEs. It allows reducing financial risks associated with data breaches, increasing the availability of cybersecurity solutions, and creating a sustainable local ecosystem of cybersecurity free from dependency on proprietary technologies. In terms of cyber diplomacy, the MSSP paradigm can help SMEs from allied countries collaborate with each other, share their threat intelligence, and coordinate incident responses without getting locked into any vendors. Further research should focus on how low-code SOAR can contribute to the establishment of mutual trust in bilateral cybersecurity arrangements in technologically underdeveloped areas. Other potential areas of research could look into the integration of threat intelligence feeds from OSINT into the proposed architecture in order to improve the regional cybersecurity monitoring and situational awareness. From an infrastructure perspective, the described SOAR architecture can gain additional horizontal scalability through the use of immutable and Kubernetes-specific operating systems like Talos Linux (Nathaniel & Syahputra, 2025), which reduces the configuration drift, the attack surface and enables quick node provisioning in large clusters. In turn, it makes it possible to deploy microservices such as n8n, OpenVAS, and GLPI on many endpoints without the need for maintaining systems manually.

## 6. REFERENCES

Aksu, M. U., Altuncu, E., & Bicakci, K. (2019). A first look at the usability of openvas vulnerability scanner. *In Workshop on Usable Security (USEC)* (pp. 1-7). https://doi.org/10.14722/usec.2019.23026

Al Mansoori, F. (2023). Layered Deep Neural Systems for Dynamic Multi-Agent Interaction, Adaptive Workflow Optimization, and Autonomous DecisionExecution in n8n. *ThinkTide Global Research Journal*, 4(4), 30-38.

Al Siam, A., Hassan, M. M., Masum, A. K. M., & Bhuiyan, T. (2025). Automating Malware Detection and Response via Real-Time Threat FeedIntegration with Wazuh SIEM. *In 2nd International Conference on Computing, Applications and Systems (COMPAS)* (pp. 1-6). IEEE. https://doi.org/10.1109/COMPAS67506.2025.11381876

Alabdulhadi, S., & Al-Matouq, A. (2024). Efficient and standardized alarm rationalization for cybersecurity monitoring. *IEEE Access*, 12, 166936-166944. https://doi.org/10.1109/ACCESS.2024.3492264

Ayittey, G. (2025). A security operations and analytics framework: Continuous detection and response [Doctoral dissertation, University of East London]. https://doi.org/10.15123/uel.8z91z

Bakraouy, Z., Abbass, W., Baïna, A., & Bellafkih, M. (2019). The IT Infrastructure's Industrialization and Mastering. *J. Commun.*, 14(10), 884-891. https://doi.org/10.12720/jcm.14.10.884-891

Bcicen, B. (2022). ctop: Top-like interface for container metrics. GitHub. Accessed April 11, 2026. https://github.com/bcicen/ctop

Boyle, E., & Adebayo, Y. (2020). Designing intelligent security orchestration, automation, and response (SOAR) systems with AI. *Revista Cintex*, 25(2), 14-27.

Casaclang, H., Ionescu, B., Kim, Y., & Jo, J. Y. (2026). Self-Hosted Workflow Automation for AI-Based Cybersecurity Operation. *Journal of The Colloquium for Information Systems Security Education*, 13(1), 21-21. https://doi.org/10.53735/cisse.v13i1.241

Chavan, N., Patel, H., & Shah, K. (2025). Comprehensive Approach to Android Penetration Testing: Techniques, Tools and Best Practices for SecuringMobile Applications. *In 12th International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 01-07). IEEE. http://doi.org/10.23919/INDIACom66777.2025.11115701

Çakır, H., & Taşer, M. (2023). Türkiye'de yapılan siber güvenlik faaliyetlerinin ve eğitim çalışmalarının değerlendirilmesi. *Gazi University Journal of Science Part C: Design and Technology*. https://doi.org/10.29109/gujsc.1165131

de Luna Melo, W. (2026). Implantação de GLPI para gestão de serviços de TI em instituição de ensino superior. *Revista Processus Multidisciplinar*, 7(13), e131582-e131582. https://n2t.net/ark:/44123/multi.v7i13.1582

Gisi, P. J. (2024). The Dark Factory and the future of manufacturing: A Guide to Operational Efficiency and Competitiveness. *Productivity Press*. https://doi.org/10.4324/9781032688152

Greenbone AG. (2024). Greenbone Community Edition: Background and architecture. Accessed March 27, 2026. https://greenbone.github.io/docs/latest/background.html

Hajamydeen, A. I., Hasni, M. D., & Abdullah, M. I. (2024). Integrating Wazuh for Efficient Real-Time Threat Monitoring and Vulnerability Assessment in a SOCEnvironment. In Utilizing Renewable Energy, Technology, and Education for Industry 5.0 (pp. 292-320). *IGI Global Scientific Publishing*. https://doi.org/10.4018/979-8-3693-2814-9.ch013

Hidayat, M. R. T., Widiyasono, N., & Gunawan, R. (2025). Optimasi deteksi malware pada SIEM Wazuh melalui integrasi cyber threat intelligence dengan MISPdan DFIR-IRIS. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(1). https://doi.org/10.23960/jitet.v13i1.5686

International Society of Automation. (2016). ANSI/ISA-18.2-2016: Management of alarm systems for the process industries. ISA. Accessed March 27, 2026. https://www.isa.org/products/ansi-isa-18-2-2016-management-of-alarm-systems-for

Islam, M. R., & Rafique, R. (2024). Wazuh SIEM for Cyber Security and Threat Mitigation in Apparel Industries. *International Journal of Engineering Materials and Manufacture*, 9(4), 136-144. https://doi.org/10.26776/ijemm.09.04.2024.02

Ismail, Kurnia, R., Brata, Z. A., Nelistiani, G. A., Heo, S., Kim, H., & Kim, H. (2025). Toward robust security orchestration and automated response in securityoperations centers with a hyper-automation approach using agentic artificial intelligence. *Information*, 16(5), 365. https://doi.org/10.3390/info16050365

Judith, D. (2026). Future Trends and Innovations in Hyper Automation. *In Implementing Hyper Automation: Strategies, Best Practices and Case Studies* (pp. 293-306). Bentham Science Publishers. https://doi.org/10.2174/9798898130001260101

Kara, Ş., Zengin, A., & Hızal, S. (2023). Ağ Sistemlerinin Güvenliği İçin Siber Saldırıların Ayrık Olaylı Sistem Tanımlama Tabanlı Modellenmesi ve Simülasyonu. *Mühendislik Bilimleri ve Araştırmaları Dergisi*, 5(2), 186-202. https://doi.org/10.46387/bjesr.1268038

Kearney, P., Abdelsamea, M. M., Schmoor, X., Shah, F., & Vickers, I. *Combating alert fatigue in the security operations centre. SSRN (2023)*. Available at SSRN 4633965. http://dx.doi.org/10.2139/ssrn.4633965

Khan, S. A., Adnan, M., Ali, A., Raza, A., Ali, A., Naqvi, S. Z. H., & Hussain, T. (2024). An android applications vulnerability analysis using MobSF. *In International Conference on Engineering & Computing Technologies (ICECT)*(pp. 1-7). IEEE. https://doi.org/10.1109/ICECT61618.2024.10581312

Le, T. D., Le Dinh, T., & Uwizeyemungu, S. (2025). A cybersecurity framework for enhancing Small and medium-sized enterprises (SMEs) security posture usinguser behaviour analytics. *Enterprise Information Systems*, 19(10), 2529282. https://doi.org/10.1080/17517575.2025.2529282

Mishra, A. K. (2026). I replaced Portainer, Grafana, and Prometheus with this stack. XDA Developers. Accessed April 11, 2026. https://www.xda-developers.com/i-replaced-portainer-grafana-and-prometheus-with-this-stack

National Institute of Standards and Technology. (2025). National Vulnerability Database (NVD) developers: Vulnerabilities. Accessed March 27, 2026. https://nvd.nist.gov/developers/vulnerabilities

Nathaniel, D. A., & Syahputra, M. E. (2025). A Comparative Analysis of Security Baselines Across Kubernetes Distributions. *In International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD)* (pp. 1-7). IEEE. https://doi.org/10.1109/ICoABCD67551.2025.11470958

Núñez Fernández, H. (2023). Despliegue de una plataforma de respuesta a incidentes de seguridad (SIRP) con software libre [Master's thesis/Bachelor's thesis]. http://hdl.handle.net/2183/33365

Palavesam, K. V., Arcot, S. V., Krishnamoorthy, M. V., & V, E. G. (2025). Building Automated Security Pipeline for Containerized Microservices. *Journal of Mathematics Mechanics and Computer Science.* https://doi.org/10.9734/jamcs/2025/v40i21969

Parmaksız, H. (2026). Vulnerability-Based Enrichment of Türkiye-Specific Cyber Threat Intelligence: Proactive Threat Hunting and Critical VulnerabilityAnalysis with TR-CERT and Exploit-DB Integration. *Güvenlik Stratejileri Dergisi*, 22(53), 155184. https://doi.org/10.17752/guvenlikstrtj.1792238

Parmaksız, H., & Gündoğdu, G. (2026). Bibliometric Analysis of Psychosocial Factors in Telehealth Subdisciplines. *Bilecik Şeyh Edebali Üniversitesi Sosyal Bilimler Dergisi*, 11(1), 42-58.

Pawar, M. (2025). A practical evaluation of self-hosted N8N for secure and scalable workflow automation. *International Journal of Scientific Research in Engineering and Management*, 9(6), 1-9.

Peesara, N. R. (2022). The Strategic Fusion of SIEM and SOAR in Modern SOCs. Available at SSRN 5664250. http://dx.doi.org/10.2139/ssrn.5664250

Rahalkar, S. (2019). Quick Start Guide to Penetration Testing. *Apress*. https://doi.org/10.1007/978-1-4842-4270-4

Rey, W. P., & Go, M. H. D. (2025). Scalable IT Management in Resource-Constrained Environments: A Case Study of GLPI Implementation in a Philippine Power Plant. *In 14th International Conference on Information Communication and Applications* (pp. 69-75). https://doi.org/10.1145/3743158.3783859

Saidur, M. J. I., & Khan, M. K. (2023). Automating NIST 800-53 Control Implementation: A Cross-Sector Review of Enterprise Security Toolkits. *ASRC Procedia: Global Perspectives in Science and Scholarship*, 3(1), 160-195. https://doi.org/10.63125/prkw8r07

Sezgin, A., & Boyacı, A. (2023). Açık kaynaklardan test otomasyon araçlarıyla siber tehdit istihbaratı çıkarılması. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 35(1), 283-290. https://doi.org/10.35234/fumbd.1217219

Shalamanov, V. (2019). Organising for IT effectiveness, efficiency and cyber resilience in the academic sector: National and regional dimensions. *Information & Security*, 42, 49-66.

Sharma, M., Desai, D., Arun, A. R., & Rajagopalan, N. (2024). Openvas vs the rest: unveiling the competitive edge in vulnerability scanners. *In 3rd International Conference for Innovation in Technology (INOCON)* (pp. 1-6). IEEE. https://doi.org/10.1109/INOCON60754.2024.10511864

Turhan, Ö. (2026). Siber olay müdahale süreçlerinde otomasyon yaklaşımı: Açık kaynaklı, low-code tabanlı sürekli tespit ve müdahale entegrasyonu. Medium. Accessed April 11, 2026. https://short-url.cc/1sx1K

Viswanadhapalli, V. (2025). The future of intelligent automation: how low-code/no-code platforms are transforming AI decisioning. *International Journal of Engineering and Computer Science*, 14(1), 26803-26825. https://doi.org/10.18535/ijecs/v13i11.4946

Wongsaard, P., Thaenkaew, P., & Suksomboon, K. (2025). Enabling Cost-Effective Vulnerability Scanning with OpenVAS: A Nested Virtualization Approach for Understaffed Security Teams. *In 6th Asia Service Sciences and Software Engineering Conference* (pp. 47-56). https://doi.org/10.1145/3775030.3775037

Younus, Z. S., & Alanezi, M. (2025). Proactive SIEM-Based Framework for Cyberattack Monitoring and Classification. *Baghdad Science Journal*, 22(9), 3114-3132. https://doi.org/10.21123/2411-7986.5067

## DECLARATIONS

Author Contributions and Conflict of Interest Statement: The 1st author contributed 60%, and the 2nd author contributed 40%. There is no conflict of interest.

Acknowledgments and Information Note: The article complies with national and international research and publication ethics. Ethics Committee approval was not required for the study.

Use of Artificial Intelligence: In this study, generative artificial intelligence tools such as figure representation (napkin.ai) were used. Grammarly and Quillbot were used for language and general checks. All responsibility rests with the author(s).

Data Availability: The data pertaining to this study can be provided by the author(s) upon request.

Similarity Rate: According to the originality report obtained from the iThenticate program, the similarity rate of the article is 2%.

The title of the research paper is “Siber Olay Müdahale Süreçlerinde Otomasyon Yaklaşımı: Açık Kaynaklı, Low-Code Tabanlı Sürekli Tespit ve Müdahale Entegrasyonu.” The paper was presented at the 24th Academic Computing Conference (AB 2026), held at Istanbul Beykent University on February 3-5, 2026. Furthermore, owing to its novelty, the paper was chosen to be reviewed in this journal.