HAMK
Häme University
of Applied Sciences

Practical Applications of Wazuh in On-premises Environments

Bachelor's thesis

Degree Programme in Computer Applications

Spring 2024

Hafiz Javid

HAMK
Häme University
of Applied Sciences

Degree Programme in Computer Applications		Abstract
Author	Hafiz Javid	Year 2024
Subject	Practical applications of Wazuh in On-premises Environments	
Supervisors	Deepak Kc	

This thesis demonstrates the capability of Wazuh, an open-source cybersecurity tool, in deployment as the Security Information and Event Management (SIEM) and Extended Detection and Response (XDR) platform within on-premises environments. The main objective of this research was to examine how Wazuh can identify and mitigate vulnerabilities while automating threat detection and response operations. The main research questions aimed to determine whether the deployment of Wazuh was indeed one of the most effective cybersecurity defenses and overall capabilities as both SIEM and XDR solutions.

The research incorporated a practical approach, integrating a broad theoretical review to determine the development and implementation phases. This thesis overviewed the basic understanding of fundamental concepts of cybersecurity, CIA triad. The experiment continued by turning into a hands-on session that built a virtual lab model of Ubuntu, Kali Linux, and Windows OS that used under Wazuh monitoring and management. Numerous types of data collection used throughout the process: performing simulated attack scenarios with Atomic Red Team, using VirusTotal to improve threat detection and applying the FIM technology. Moreover, port scanning and brute force attack from Kali Linux to Ubuntu shown the capabilities of Wazuh in detecting and mitigating real-world threats.

The execution of the study depicts that Wazuh is a powerful tool in cybersecurity management, portraying remarkable features such as swift real time threat detection and incident response. The joint of Wazuh and VirusTotal integration substantially amplified the capacity to discover and block threats. On the other hand, Atomic Red Team simulations provided significant evidence of correct performance of Wazuh to real cyber-attack approaches. Therefore, use of Wazuh can help businesses to enhance the defensive strategy. Moreover, everything will be carried further as Wazuh might be researched to locate in the cloud and on containers to ensure that the cybersecurity world sphere will not cease to grow.

Keywords Wazuh, VirusTotal, File Integrity Monitoring, SIEM / XDR, Cybersecurity
Pages 43 pages and appendices 1 page

Glossary

SIEM	Security Information and Event Management
XDR	Extended Detection and Response
CVE	Common Vulnerabilities and Exposures
FIM	File Integrity Monitoring
VM	Virtual Machine
CIA	Confidentiality Integrity and Availability
NIST	National Institute of Standards and Technology
GDPR	General Data Protection Regulation
PCI DSS	Payment Card Industry Data Security Standard
HIPAA	Health Insurance Portability and Accountability Act
CIS	Center for Internet Security
CLI	Command line Interface

Content

Figures

Code Snippets

Appendices

1 Introduction

In the digital world, cybersecurity stands for technological integrity and resilience in operations. While cyber-attacks getting more advanced, the digital world needs to be protected in a highly secure manner. This thesis focuses on how Wazuh, an open-source tool, can protect on-premises environments from digital attacks.

This thesis is intended for the academic community, cybersecurity experts, and enterprises looking to improve the security infrastructure. The thesis goes into the operational mechanics of Wazuh, focusing on vulnerability detection and mitigation capabilities. The research purposely excludes comparisons with other cybersecurity solutions to maintain focus on features of Wazuh.

The practical portion of this thesis will include the configuration and integration of Wazuh with VirusTotal and the File Integrity Monitoring (FIM) techniques. The goal is to create a fortified security architecture that is consistent with the CIS Benchmark for an ideal security posture. The end-user's need for a real-time, automated threat detection and response system will be addressed, thereby ensuring that the proposed solution is both robust and meets all the relevant industry demands. In the practical implementation section, there would be configuration and installation of the Wazuh-agents for monitoring and controlling on-premises infrastructure. Besides this, the integration of additional tools and strict inspection of the existing security standards will be examined to upgrade Wazuh’s effectiveness in mitigating the identified threats.

The fundamental aim of the practical section is to create and personalize a proper security solution designed to match the typical needs and problems of on-premises lab environment. In fact, focusing on rectifying the challenges clearly points to the ways of reinforcing the cyber defence system and ensuring security of valuable assets from new threats.

The research questions to drive the thesis are as follows:

- What is the role of Wazuh in identifying and mitigating vulnerabilities in the machine?
- What role do XDR and SIEM play in automating threat detection and response workflows orchestrated by Wazuh in on-premises environments?

On this research, these questions provide a base of the theory, which facilitate a study of an efficacy of this product and ability to embed in the flow of other frameworks to increase the level of security of the on-premises lab environments.

2 Security Threats and Solutions for on-Premises Labs

This chapter explores the basic frameworks and theoretical foundations required to understand the deployment and success of Wazuh, an open-source cybersecurity tool. The section delves into cybersecurity principles, architecture of Wazuh and features. The ideas and models presented here not only contextualize Wazuh's function in the cybersecurity ecosystem, but also express how theoretical structures influence the practical applications covered in following chapters. The chapter aims to execute the exploration to provide a general framework for the cybersecurity professional to comprehend the challenges and techniques in preventing the modern information systems.

2.1 Cybersecurity Principles and Challenges

According to the Jason (2014), Information security is defined as the process of safeguarding information from interference, tampering, theft and destruction by unauthorized individuals and agents. Physical and logical security are included in protecting structural as well as the infrastructural property and people, computer equipment, software, information, and individuals respectively. Information security is focused on the protection of data and information ownership and preventing loss in potential threats or breaches in relation to information assurance to counter different types of attacks. Appropriate measures of information security that are put in place and effectively managed help organizations to protect the information that should not be exposed to the public.

The CIA triad, as described by Mpekoa, is the core component in the information security architecture, focusing on fundamental role in achieving security. To create a comprehensive cybersecurity framework that meets different security needs across various domains, this architecture and NIST framework are essential, as shown in the Figure 1.

Confidentiality: This guarantees that data and information systems are only accessed by authorized users. Breach results in illegal disclosure of sensitive data (CIS, n.d.). Confidentiality in information security affords data protection to a degree wherein only authorized data viewers can access it. This aspect is important in ensuring that personal information and other secure items are not compromised. There is a general view of how it is possible to have the aspect of confidentiality harmed: either through hacking, leakage or by accident. There are many solutions that can be implemented to ensure business information confidentiality which are data encryption, data access controls and limitation of data distribution to specific personnel with the proper authorization only. For example, during the process of withdrawing money from a particular ATM, the system only allows to the account details and making transactions of the user after entering the right PIN code (Jason, 2014, p.6).

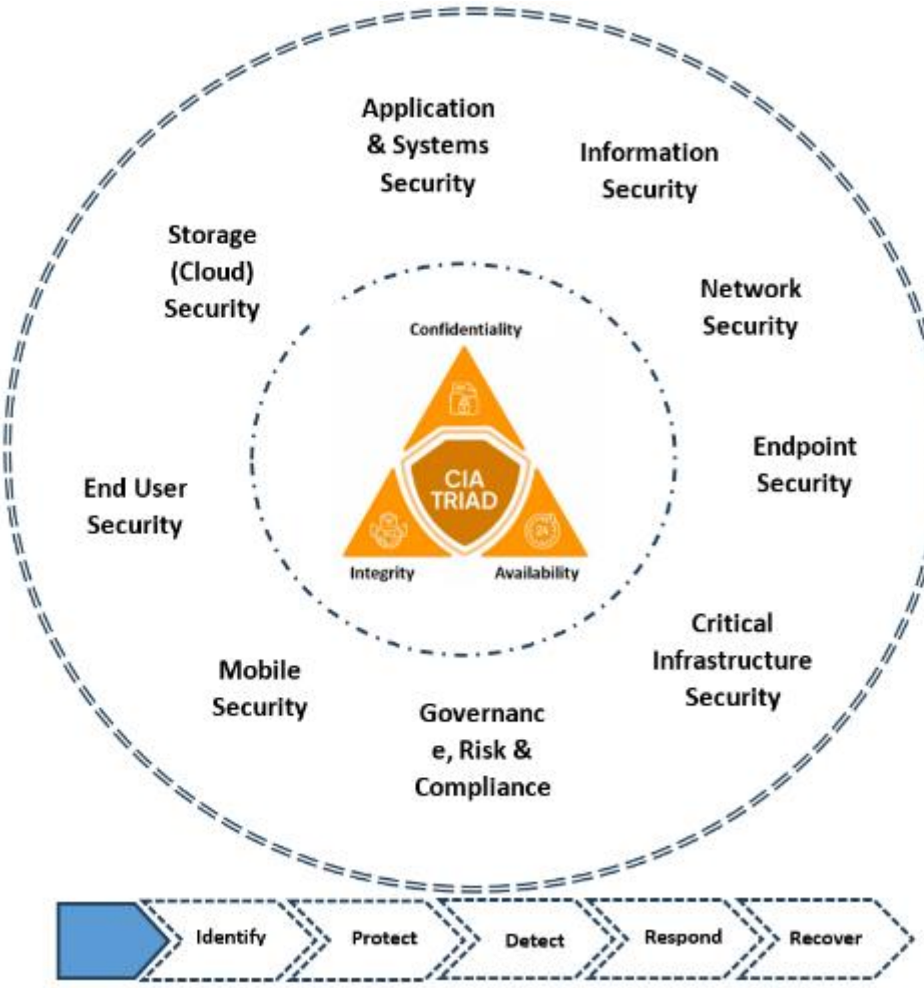
Integrity: Ensures that data is complete and has not been altered without authorization. The destruction of data involves unlawful alterations that transform the data's meaning (CIS, n.d.). Integrity means that data cannot be modified by unauthorized persons or by systems that are not specifically designed to do so. This means ensuring that no unauthorized person, device, or program can write, modify, or delete the data contained in a system. To reduce risks of data corruption, mechanisms such as checksums, hash functions and strict access controls are utilized. As an example, integrity is observed through privileges that don’t allow unauthorized individuals to modify information in a database. Data integrity is very crucial in making decisions or take some actions because corrupted data might lead to wrong activities (Jason, 2014, p.6).

Availability: This maintains that information systems are available for usage as necessary. DDoS attacks frequently cause the network disruptions (CIS, n.d.). This refers to data and systems availability at the right time without easily preventable delays or disruptions. This aspect is very important in ensuring that organization runs efficiently because loss of data could have severe consequences. A sample of the availability threats include hardware and software failures, cyber threats such as denial of service (DoS) attacks, and even disasters. Availability can be achieved through dual processors, backup copy creation, and disaster contingency planning. Important services should be highly available so that users can gain access to relevant resources at any given time (Jason, 2014, p.7).

The CIA triad, as described by Jason (2014, p.7) is one of the most basic concepts and models that are used to address security concerns. This work establishes that each component in the triad has specialized responsibilities in the general security of information systems. Confidentiality protection involved the restricted release of information or data to a certain people, while integrity means that the details or data received are accurate. Availability refers to the situation when systems, materials, and information are available when needed. By combining them, one gets a complete framework of information security where different aspects are considered to meet certain organization’s needs in countering risks.

In the Figure 1, CIA triad and the foundational principles are explained. The triad is the central elements of the various areas such as storage security, Information Security, Network security and mobile security etc. Moreover, the triad outlines the important steps to manage the cyber security like identity, protect, detect, respond, and recover. Overall, this diagram depicts the visual synopsis of the cybersecurity principles and connects the nature of security.

Figure 1 CIA triad (Mpekoa, n.d.)



2.2 Overview of CVE

Common Vulnerabilities and Exposures (CVE) is a massive database of shared security issues with software and hardware. Each listed problem is assigned a unique number to understand for everyone from developers of high technologies to academics and security specialists to discuss these problems and search for solutions. These CVE numbers and the severity level associated with them help businesses determine which issues should be prioritized (Balbix, n.d.).

CVE was established in 1999 and is maintained with current information from the MITRE Corporation. CISA funding comes from the US government, namely the Department of Homeland security and the Cybersecurity & Infrastructure Security Agency. A significant advantage of the CVE list is that this is open to the public and completely free of charge (CVE, n.d.).

Each issue on the CVE list is assigned an ID number by one of about 100 authorized organizations. There are even companies working in the technology industry, academic institutions, cybersecurity firms, and, of course, MITRE. The numbers follow a format: CVE designation is based on the format CVE-[Year]-[Number], where the year shows when the problem was brought to attention, and the number is provided as a serial by the governing body. For example, CVE-2019-0708 is a threat to Microsoft with the name originating from a flaw in the scripting engine for the Remote Desktop Protocol (RDP). This issue is one of the most significant ones that users face, known as BlueKeep (Balbix, n.d.).

CVE allows organizations to create a benchmark that will aid in determining the roles of the security tools needed by an organization and capabilities in protecting the organization. When a specific vulnerability has a CVE ID, organizations can aggregate a lot of information from different sources to prioritize the action plan. Security advisories can use CVE details to define risk patterns and identify specific hacks. For open-source projects, CVEs offer advantages by helping in sharing information, encouraging interaction and fostering trust amongst security communities. This practice closely follows current secure coding standards, improves vendor communication, utilizes current security tools, and increases the security of software (Tetrate, n.d.).

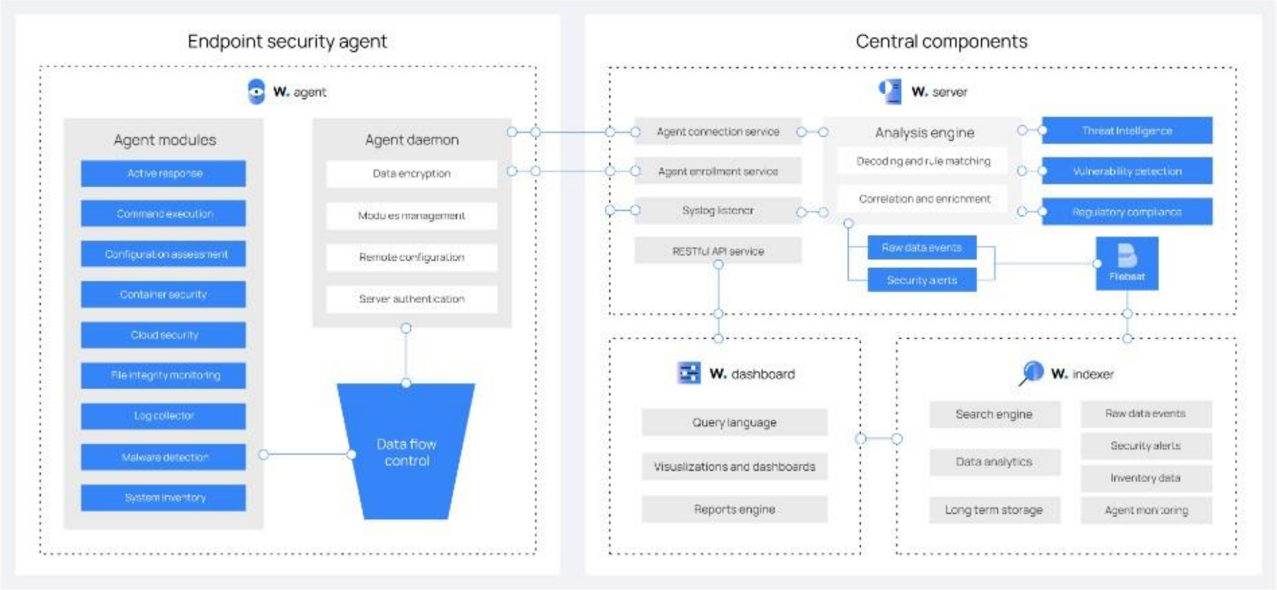
2.3 Features and Capabilities of Wazuh

Wazuh is an open-source security monitoring tool that detects threats, ensures compliance, and responds to incidents. Wazuh provides robust solutions in security and monitoring using the capabilities such as Security Information and Event Management and Extended Detection and Response (Wazuh, n.d).

There are security flaws, known as vulnerabilities, that exist in computer systems. These flaws can be exploited by attackers to gain unauthorized access. Once exploited, attackers can install malware, steal data, or execute harmful actions. To prevent this, businesses need security measures that quickly identify vulnerabilities before attackers can use them. Detecting and addressing vulnerabilities promptly strengthens a network's overall security (Wazuh, n.d).
The architecture of Wazuh includes several key components. The details of the core components are as follows: Wazuh can parse, analyse, and correlate log data to discover potential security events. This is called log analysis. Secondly, there is monitoring and reporting of changes to fundamental system and application files as a means of detecting potential security incidents. The process is called File Integrity Monitoring (FIM). Wazuh examines systems' exposure to known vulnerabilities in the vulnerability detection component. Wazuh agents are the machines either in cloud or on-premises that are monitored. Agents send all the data in the form of events to the Wazuh manager (Thakkar, 2023).

Figure 2 shows that Wazuh is a fully featured security solution consisting of core components such as Wazuh Server, Wazuh dashboard, Wazuh Indexer, and Wazuh agents. Wazuh indexer is a scalable engine which is capable of indexing and storing logs and alerts generated by the Wazuh server thanks to advanced SIEM capabilities. Wazuh server is responsible for managing agents to configure and manage agents remotely, analysing data gathered from agents, as well as utilizing threat intelligence to detect possible compromises. Wazuh dashboard provides for data mining, analysing, exploring, and managing Wazuh configuration while monitoring the status. In addition, the Wazuh agent, a multi-platform component that is represented in the form of an application installed on desktop systems, offers prevention, detection, and response capacities as shown in the Figure 2.

Figure 2 Architecture of Wazuh (Wazuh, n.d.)



3 Role of FIM in Security

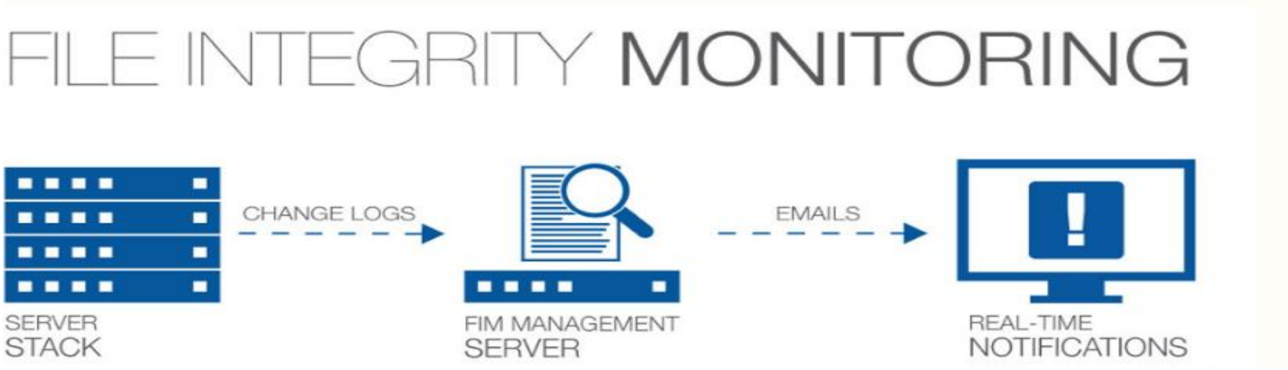
This chapter discusses the analysis of File Integrity Monitoring in the sphere of cybersecurity. The discussion begins with elaborating that FIM is used to discover security threats, to address regulations and to secure IT systems. The benefits of using real-time and customized monitoring are explained. Furthermore, FIM is discussed within the context of incident response as well as compliance regimes, which illustrates how FIM supports the protection of digital assets.

3.1 Importance of FIM in Security

File Integrity Monitoring (FIM) is an integral part of the cybersecurity measures. FIM is responsible for monitoring, event logging, and identifying the modifications to allocated files and directories in the systems. This ability is essential to detect the potential security breaches, ensuring regulatory compliance, and to protect the general reliability of the IT environment (Fazzino, n.d.).

According to Fazzino, FIM is a critical security process that identify the changes in the files. FIM works by comparing current files to a known trusted version and identifying events that could indicate a security threat or a violation of rules. Figure 3 shows the clear representation the process of File Integrity Monitoring. In the process of File Integrity Monitoring, the logs are being received in the server stack when changes monitors in the FIM management server. After the analysis, the notifications may send to the relevant person via email in a real-time.

Figure 3 FIM in Security (Fazzino, n.d.)



3.2 Real-time and Custom Monitoring

Real-time monitoring is a technique that determines the current state of queues and channels within the queue manager which provides an accurate information in real time (IBM MQ 9.2, 2024). Wazuh employs real-time monitoring for File Integrity Monitoring. This means that FIM can detect changes, deletions, or other modifications to files immediately. This immediate detection is significant as this allows the early detection of security issues and reducing the risk of damage. FIM feature is useful for monitoring important directories and track changes instantly. Wazuh has capabilities to get notified immediately in the dashboard in real time whenever file in the specific directory changes (Wazuh, n.d.).

Wazuh enables the users to adapt the tracking by allowing to choose which files to monitor. With this functionality, users get the chance to focus on the most critical files while looking through the others, resulting the more effective monitoring in the system. Every detected change is logged, along with all the details such as date, time, and even the responsible. This data is crucial for the future investigations into security events (Biri, 2024).

3.3 Compliance and Integration with Incident Response Systems

Incident response involves the set of actions taken by the organizations to respond to the detected threats within infrastructure of organizations. These steps taken by the organizations help to analyse the impact of the cyber-attacks in the digital assets and the business operations. Security team takes leverages from the Wazuh capabilities of incident response to analyse, detect, and respond effectively to the security incidents (Wazuh, n.d.).

Wazuh's FIM can assist with compliance as required by regulations like the PCI DSS and HIPAA, maintaining precise logs of file changes. These logs indicate that the users are keeping the timeliness and uniqueness of records as expected. Wazuh offers an actionable FIM that is combined with alerting and response capability. If the change poses a great threat, the system will immediately send the notification to the emergency department. Thus, the system can trigger either direct actions to resolve the issue or send message to the team to take a suitable action (Baykara, 2020).

4 Application of Cybersecurity principles

The cybersecurity principles in on-premises environments are being implemented using the SIEM solution and XDR technology. The review of these technologies using the existing knowledge are being analysed in this chapter.

4.1 Security Threats and Challenges in On-Premises Labs

Statistics from the reputable cybersecurity companies often mention the vulnerabilities in software that do not updates or use outdated operating systems. These flaws generally allow hackers to take over laboratory systems and obtain unauthorized access to private data (NIST, 2013).

A NIST review found that problems with deployments, configurations, network devices, servers, and lab instruments might lead to security problems and breaches. If the machines are not configured properly, the attackers possibly get into restricted data (NIST, 2020).

An eminent cyber threat is the insider threat, the case where authorized personnel or employees of a company or organization commit a malicious activity. The stealing of sensitive data or disturbance in operation could happen as a result. Hence, the dispatch of countermeasures and early intrusion should be the main interest (NIST, 2016).

The investigation of cybersecurity crimes repeatedly involves social engineering attacks as propagated through phishing emails, fake phone calls, and social media manipulation in the researchers. One of the core tasks is to prepare awareness of security (Holt et al., n.d.).

Since the integrations of third-party keys and software in the laboratories have increased, that are vulnerable to the attacks of supply chain. Such activities can induce unintended code within these components which opens lab systems to malicious agents. (Cichonski et al., 2012).

4.2 CIS Benchmarks

CIS Benchmarks are guidelines for configuration of systems securely. The Center for Internet Security (CIS) is an entity committed to security of private and governmental organizations from cyber risks. This organization builds code of information security best practices (CIS) benchmarks for protecting IT systems and data against cyber-attacks. Through CIS-CAT scan systems are identified and a report is produced that compares the system parameters with the CIS benchmarks (Casares, 2019).

Creating CIS benchmarks is a two-stage process. Initially, professionals meet to create benchmarks and discuss the different versions from different experts. Subsequently, feedback from the global internet community is gathered and compiled to make necessary changes to make sure the benchmarking is efficient. CIS benchmarks provide the two levels for security settings. The first level of recommendations includes configurations that are relatively simple to perform and do not significantly impose on the operating system. Level 2 settings are for more secure conditions that may slightly decrease the usability of possible features caused by tightening of conditions (Briasmitatms, 2024).

4.3 SIEM and XDR for Security in Lab Environment

Two important technologies are Security Information and Event Management (SIEM) and Extended Detection and Response (XDR). By combining, analysing, and responding to data from many sources, SIEM systems offer a complete picture of an organization's security. This platform provides better visibility, finds threats automatically, and coordinates responses across all IT levels (2022 Gartner® Critical Capabilities for Security Information and Event Management, n.d.).

In the Figure 4, Wazuh offers flexibility with a wide range of tools like SIEM and XDR. This tool is designed to offer the security analysis with versatile features related to detect, prevent, and respond to threats as required.

Figure 4 Wazuh as SIEM and XDR solution (BleepingComputer, n.d.)

Unified XDR and SIEM protection across several platforms are provided by Wazuh, a free and open-source security solution. Across virtualized, on-premises, cloud-based, and containerized systems, Wazuh safeguards workloads to give businesses a reliable cybersecurity solution (BleepingComputer, n.d.).

4.3.1 SIEM Technologies

SIEM has evolved from simple systems for log attendance to comprehensive security management solutions. The SIEM solution gathers and transforms data from numerous sources to allow for unified security monitoring in real-time. The core functions include collecting data from various sources, then connecting and correlating the related events, providing alerts for security concerns, and later, the submission of reports for compliance purposes (Brandao & Nunes, 2021).

Typical SIEM systems have evolved and powerful features but often face challenges to withstand the volumes of data flow in today’s IT settings. The systems often have trouble integrating various data sources, processing data in real time efficiently, and filtering out false alerts. Moreover, SIEMs depend heavily on predetermined rules to identify the threats, making them vulnerable against advanced potentially new types of attacks (Brandao & Nunes, 2021).

The SIEM security system in Wazuh is a very powerful solution to control and make the infrastructure more secure. This combines features such as logging security events and analysing those logs to identify threats or detect the suspicious incidents. This solution scans the entire system to identify known vulnerabilities and conducts risk triage and prioritizes fixing those vulnerabilities. For the security configuration assessment, Wazuh checks systems against industry standards (such as CIS) to ensure that those could meet regulatory compliance and minimize the system vulnerability. The tool like Wazuh supports data collection, storage, and analysis of security needs (Wazuh, n.d.).

In the future, SIEM systems will increase the level of artificial intelligence (AI) and machine learning (ML) integration to fortify the platform. This will improve the ability to identify emerging threats and respond to incidents accordingly. These advancements will overcome the existing shortcomings by combining events more effectively and minimizing the need for human involvement in detecting and responding to threats (Brandao & Nunes, 2021).

4.3.2 XDR Technologies

XDR (Extended Detection and Response) is an advanced solution providing full protection for IT infrastructure. This technology covers networks, clouds, applications, and now even the endpoints.

XDR allows the combination of several tools into one system and thus reduces the workforce needed to manage the various security systems. This increases the visibility and analysis capabilities, and improve the response rates, resulting in comprehensive protection and faster incident threat handling (George et al., 2023).

XDR systems enable the automated association of collected data across all security layers, leading to a unified view of threats. This is the core reason for better knowledge in finding and countering the threat by the incorporation of the latest technology such as machine learning and analysis involving complex and scattered data sets. Finally, XDR performs marvellously as the XDR combines alerts into one display and shows truly smart intelligence that cuts down on complexity and brings up efficiency (George et al., 2023).

XDR provides improved monitoring comparatively to SIEM (Security Information and Event Management) solutions. Though SIEM only focuses on data collection and event correlation, XDR provides the capacity for proactive threat detection and response, such as automatic response and native support for cloud and hybrid environments. This complete approach empowers XDR to spot threats sooner and respond more efficiently, resulting in minimizing the impact of threats (George et al., 2023).

4.4 Leveraging Wazuh in Lab Settings

The open-source platform of Wazuh that covers both SIEM and XDR components is the security product that will increase the security of digital assets. Wazuh in research settings, which focus on both data privacy and security, stands as a major player in ensuring cybersecurity protection. Due to the extensive multi-layered approach, which makes the Wazuh very effective in detecting any security threat in laboratory environments (Wazuh, n.d.).

One of the crucial capabilities provided by Wazuh is the powerful log management system. Wazuh collects, monitors, and keeps logs from several spots in the dashboard. This feature is critical, experts can use tool for surveillance and comprehending the lab's information technology operations. This is very effective in instant analysis and historical data study. According to logs that come from various devices including scientific instruments plugged into laboratory network, there might be a deviation from existing rules and guidelines or the appearance of specific problems that can be traced in such sources (Arora, 2021).

Wazuh enhances the lab security by real-time detecting threats. Rules must be defined to analyse events and identify known threats. Wazuh also uses complex algorithms to spot strange things that might be new threats based on how things usually are. This is especially helpful in labs because new cyber threats can pop up quickly due to the valuable data and ideas contain in these environments (Wazuh, n.d.).

Wazuh ensures that security teams respond quickly once a threat identified. The tool can automatically take certain actions such as common threats, isolating the affected systems or blocking harmful IP addresses. Wazuh provides an active response feature also lets teams create custom response plans that suit the lab policies and procedures as well. Wazuh minimizes system downtime and reduces the impact of security breaches by tailoring responses. Laboratories can improve security by using Wazuh. This tool detects and stop threats in real-time and Wazuh also works well with other systems in the lab (Arora,2021).

4.5 Assessing the suitability of Wazuh in Cybersecurity

Wazuh's File Integrity Monitoring (FIM) module is an affordable and effective tool tracking and discovering unauthorized edits of files, folders, and Windows' registry. FIM is a real-time module and the other features of Wazuh join to protect systems and help to meet PCI DSS GDPR and NIST 800-53 performing the detailed logs and reports (Sadiq, 2024).

Maltiverse is a great tool that collects and examines in-depth information on known malicious entities such as domains, IP addresses and various digital artifacts. These data could be used to identify the vulnerabilities of a company and determine which threats are more crucial to tackle. Maltiverse enhances detection capabilities, providing security teams with a more comprehensive view of threats and allowing them to respond more effectively with the combination of Wazuh platform. This cooperation ensures that the companies prepared to defend them against cyber-attacks and to prevent data and systems’ corruption or misuse (Noman, 2023).

Wazuh monitors the status of infections with VirusTotal, which is a platform that integrates the VirusTotal, an anti-virus engine with online scanners' information in one place. This integration gives Wazuh the access to VirusTotal's immense database of files and ensures that files could be automatically scanned. This is performed through the combination of FIM that verifies the integrity of files and reduces the probability of malicious activity detection, which is a major factor in making the system more efficient (Wazuh, n.d).

Anomaly detection is the process of identifying patterns in a system which categorically differ from the expected behaviour. The platform applies a vast-spectrum approach to surveillance abnormal patterns that suggest potential intruders (Casares, 2019).

Data of the operating system and application logs are collected, so secured transport of integrated information is ensured to a central manager for rule-based analysis and storage. Wazuh rules reporting includes applications or system problems, the misconfiguration of systems, cybersecurity attacks, security policy violations, and a lot of other operational and security related issues that occur (Casares, 2019).

4.6 Integration and Customization

Wazuh uses Maltiverse's threat intelligence to enhance its threat detection capabilities. This combination adds alerts with more detailed facts on malicious domains and IP addresses, providing deeper context for security decisions (Wazuh, n.d).

Wazuh integrates with VirusTotal to analyse files and hashes, thus enhancing its threat detection along with the advanced scanning engine of VirusTotal. Such integration improves the Wazuh's File Integrity Monitoring with detailed malware insight (Nanty,2023).

Code Snippet 1 shows a piece of XML configuration code for interacting with VirusTotal. The name of the integration is "VirusTotal", supplies a placeholder for the API key, assigns the group "syscheck", and specifies that alerts will be in JSON format. To activate the integration, replace with the personal "API_KEY". An API key is a set of alphanumeric string that is used to communicate and exchange data between two software modules (AWS, n.d.). This is possible to integrate the different services in the Wazuh to get the real-time event in the dashboard. The configurations and customizations are different with each service, but the idea is same.

Code Snippet 1 Integration of VirusTotal (Nanty, 2023)

```
<integration>
  <name>virustotal</name>
  <api_key>API_KEY</api_key>
  <group>syscheck</group>
  <alert_format>json</alert_format>
</integration>
```

Slack is connected to Wazuh which makes this easy to communicate. The solely purpose of this combination is sending immediate alerts and security reports to certain Slack channels, helping security helps get quick responses. This integration allows synchronizing activities and making decisions as well as real-time tracking of monitoring dashboards through security issue events without monitoring all the time. Wazuh combines with Shuffled for the automated and incident response. Automated responses triggered by Wazuh alerts improve the operational nature by lowering the need for manual action and usual security choices (Wazuh, n.d.).

5 Research Methodology

This chapter describes the research methodology used in this thesis to serve an assessment of cybersecurity practices in an on-premises lab environment. Using various tools such as Wazuh, FIM, VirusTotal, Nmap, hydra, and the Atomic Red Team, the study bridges the gap between theoretical cyber security concepts and practical implementations The research methodology consists of literature review and the methodology for applying the cybersecurity tools in a simulated environment to test the effectiveness and performance.

5.1 Research Techniques and Approach

The thesis combined practical and research techniques to assess cybersecurity measures in on-premises lab environments. Wazuh employed other technologies like File Integrity Monitoring (FIM) and VirusTotal during this thesis study. This approach linked theoretical cybersecurity concepts to real-world applications. This approach not only evaluated the performance of the integrated technologies but also examined the theoretical basis, strengthening the understanding of modern cybersecurity solutions.

During the literature review section of the thesis, resources such as Google Scholar, HAMK Finna, and Wazuh Official documentation were used for the accomplishment. Major keywords during the search were open-source XDR and SIEM tools and technologies, Vulnerabilities detector, CVE, File Integrity Monitoring and VirusTotal. Latest journal articles were reviewed to complete the literature review from the above-mentioned platforms.

The thesis is functional because this study directly used cybersecurity tools like Wazuh, SIEM, and XDR. In a lab that mimics real-world settings, these tools were carefully planned, put into action, and tested. The practical side is shown by putting these techniques into place and connecting them, which shows how the tools can be used to improve security and deal with incidents.

5.2 Tools and Technologies Used

Wazuh is the core tool being used in this research and there were several other tools such as PowerShell on windows agent, Linux Terminal on ubuntu agents, and Nmap rand hydra on Kali Linux. Pwsh in the ubuntu was also used to simulate the attacks through Atomic Red Team, with VirusTotal and File Integrity Monitoring (FIM) in the Wazuh dashboard. VirusTotal is the tool integrated to strengthen threat detection in real-time. It provided the extra layer of security that empowered the findings from Wazuh. Virtual machines were being used to simulate a realistic network environment, enabling controlled testing environments. These VMs host different operating systems with customized configurations.

Four Wazuh agents were deployed in the dashboard whose operating systems were Ubuntu 18.04 LTS, Ubuntu 22.04 LTS, Kali Linux Kali 2023.4, and Windows 11 Education Edition, to examine and simulate real security threats. The Ubuntu and Kali machines were in Virtual Box, while the Windows 11 machines was the base operating system. There were additional tools to perform SSH logins, port scanning, and brute force attack tasks. These tasks were executed on Kali Linux to attack other agents and analyse the response of Wazuh. Nmap and hydra were used to perform port scanning of target machines and unauthorized access to other Ubuntu machines, respectively.

Each agent sends data about File Integrity Monitoring, suspicious file downloads, port scanning, SSH logins, and simulated attack in a controlled environment using MITRE ATTA&CK framework technique. The ID of the MITRE ATT&CK technique used is T1003.008. This ID represents the path of credentials of the user in the Linux machines. In this study, Atomic red Team was used to perform the attacking tests on the Wazuh agents, and real-time responses and events sent to the Wazuh Manager (Server).

6 Practical Implementation and Evaluation of Cybersecurity Measures

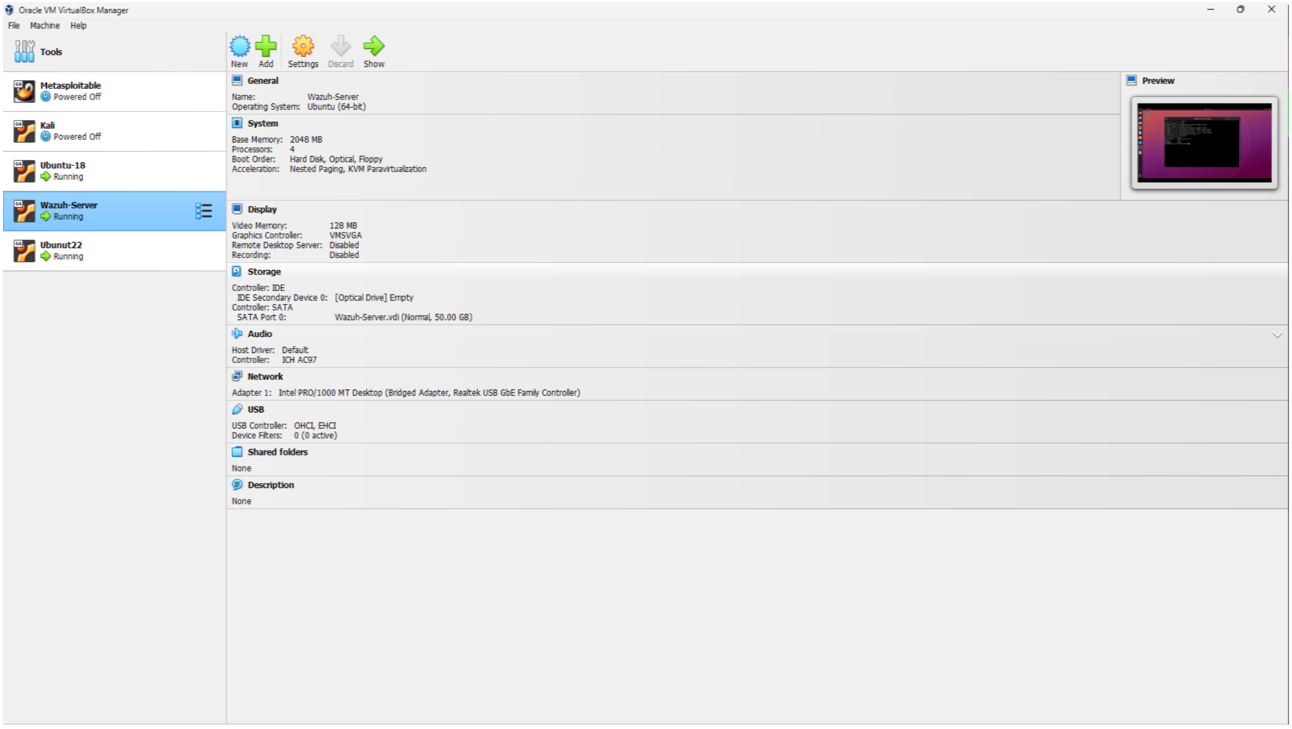
This chapter investigates vulnerability scan, setting up the File Integrity Monitoring (FIM), integration of VirusTotal database, and simulating attacks in a lab environment using Atomic Red Team.

6.1 Lab Environment for Wazuh Deployment and Testing

The thesis focused on producing a simulated virtual environment that covers all the necessary components needed to replicate real-world cases, with which the capabilities of Wazuh as SIEM and XDR solutions could be evaluated. The setup consisted of four machines that replicated the real-world scenarios and allowed for a fully managed and controlled environment to analyse comprehensive security.

Figure 5 shows the virtual environment for the Wazuh deployment and testing. In the environment, there were three Ubuntu machines, and one Kali Linux machine used. One of them served as the Wazuh server (22.04 LTS) while the other three machines played the role of Wazuh agents as shown in Figure 5. These machines ran on Ubuntu 18.04 LTS, Ubuntu 22.04 LTS, and Kali Linux 2023.4 while the fourth agent was the base operating system (Windows 11 Education Edition). Various tests were performed with the deployed agents such as Vulnerability scans, brute force attack, FIM and VirusTotal.

Figure 5 Virtual Environment of Ubuntu machines.



Metasploitable
Powered Off

Kali
Powered Off

Ubuntu-18
Running

Wazuh-Server
Running

Ubunut22
Running

General

```
Name:              Wazuh-Server
Operating System:  Ubuntu (64-bit)
```

System

```
Base Memory:   2048 MB
Processors:    4
Boot Order:    Hard Disk, Optical, Floppy
Acceleration:  Nested Paging, KVM Paravirtualization
```

Display

```
Video Memory:            128 MB
Graphics Controller:     VMSVGA
Remote Desktop Server:   Disabled
Recording:               Disabled
```

Storage

```
Controller: IDE
  IDE Secondary Device 0:  [Optical Drive] Empty
Controller: SATA
  SATA Port 0:             Wazuh-Server.vdi (Normal, 50.00 GB)
```

Audio

```
Host Driver:  Default
Controller:   ICH AC97
```

Network

```
Adapter 1:  Intel PRO/1000 MT Desktop (Bridged Adapter, Realtek USB GbE Family Controller)
```

USB

```
USB Controller:  OHCI, EHCI
Device Filters:  0 (0 active)
```

Shared folders

None

Description

None

6.2 Installation of Wazuh

There are three components involved in the installation of Wazuh, as shown in Figure 2. These components are Wazuh indexer, Wazuh server / manager, and Wazuh dashboard. There are two ways to install Wazuh in the Linux machine mentioned in the official documentation of Wazuh. First approach involves installing each component and subsequent dependencies by using the assistant. Second approach is straightforward which was used to install Wazuh in Ubuntu 22.04 LTS. In this method, one command handled the installation and configuration of the indexer, server, and dashboard.

Code Snippet 2 shows the command to install Wazuh and the required dependencies. The -o flag was used at the end of the command to overwrite the previous installation of Wazuh; however, this flag is not needed for a fresh installation. The process started with the execution of command to install Wazuh 4.7.4. There previously installed Wazuh and the relevant dependencies were removed to make the environment clean. After cleaning the environment, the installation of Wazuh components commenced. The components installed were Wazuh indexer, Wazuh Server, and the Wazuh dashboard.

Code Snippet 2 Installation of Wazuh

```
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash
./wazuh-install.sh -a -o
```

On successful completion of installation, an output shows that the login credentials to access Wazuh dashboard. The dashboard can be accessed by typing https:// <Wazuh Manager Ip> in the browser to see the interface. The Ip address can be found out by typing the following command in the ubuntu terminal.

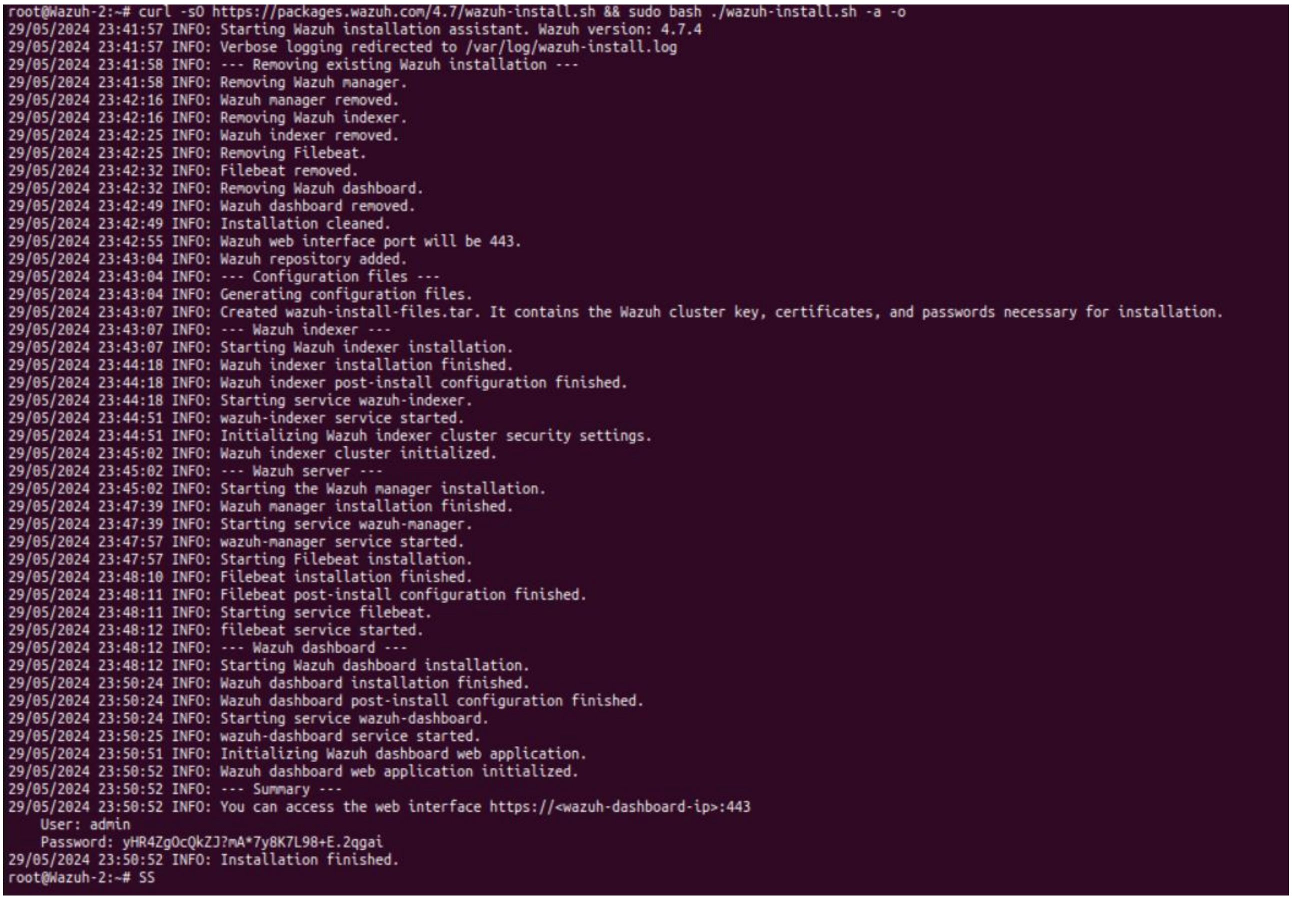
To find the IP address of the Linux machine, a simple command needs to be typed as shown in the Code Snippet 3. Since the Wazuh has been installed in Ubuntu machine, therefore, the IP address of the machine would be the IP of the Wazuh manager. If the machine is other than Linux operating system, then the command may vary.

Code Snippet 3 Finding the IP address.

```
ip a
```

Figure 6 shows the completion of the installation of Wazuh on Ubuntu 22.08 LTS. The command mentioned in the Code Snippet 2 tiggered the cleaning of the environment, the addition of the Wazuh repository, and the installation of required packages. The scripts started to install the three components one by one and displayed the relevant logs in the terminal. Upon completion of this process, there were credentials generated that must be saved for use to access the web interface of Wazuh. At this stage, the web interface of the Wazuh showed the zero agents in the dashboard. Agents need to be deployed to start analyzing the results and events shown in the dashboard from the Wazuh agents.

Figure 6 Completion of Wazuh installation.



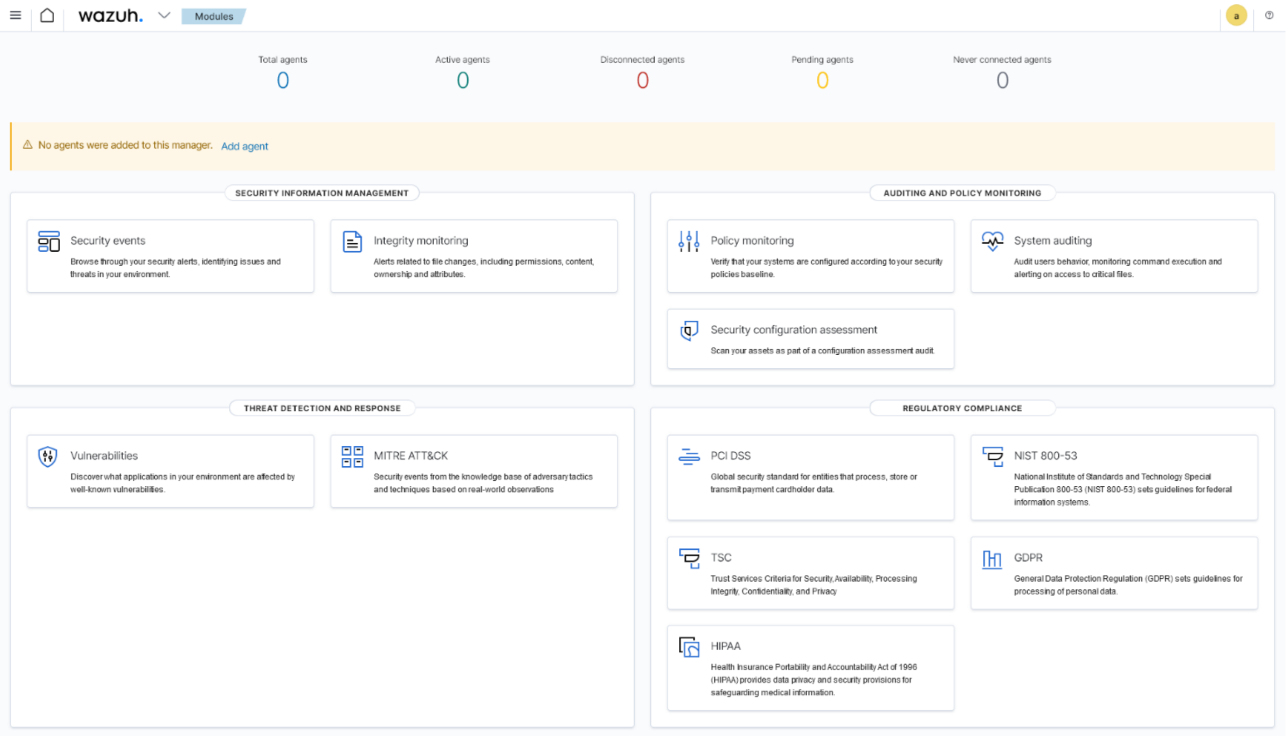
```
root@Wazuh-2:~# curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh -a -o
29/05/2024 23:41:57 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.4
29/05/2024 23:41:57 INFO: Verbose logging redirected to /var/log/wazuh-install.log
29/05/2024 23:41:58 INFO: --- Removing existing Wazuh installation ---
29/05/2024 23:41:58 INFO: Removing Wazuh manager.
29/05/2024 23:42:16 INFO: Wazuh manager removed.
29/05/2024 23:42:16 INFO: Removing Wazuh indexer.
29/05/2024 23:42:25 INFO: Wazuh indexer removed.
29/05/2024 23:42:25 INFO: Removing Filebeat.
29/05/2024 23:42:32 INFO: Filebeat removed.
29/05/2024 23:42:32 INFO: Removing Wazuh dashboard.
29/05/2024 23:42:49 INFO: Wazuh dashboard removed.
29/05/2024 23:42:49 INFO: Installation cleaned.
29/05/2024 23:42:55 INFO: Wazuh web interface port will be 443.
29/05/2024 23:43:04 INFO: Wazuh repository added.
29/05/2024 23:43:04 INFO: --- Configuration files ---
29/05/2024 23:43:04 INFO: Generating configuration files.
29/05/2024 23:43:07 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
29/05/2024 23:43:07 INFO: --- Wazuh indexer ---
29/05/2024 23:43:07 INFO: Starting Wazuh indexer installation.
29/05/2024 23:44:18 INFO: Wazuh indexer installation finished.
29/05/2024 23:44:18 INFO: Wazuh indexer post-install configuration finished.
29/05/2024 23:44:18 INFO: Starting service wazuh-indexer.
29/05/2024 23:44:51 INFO: wazuh-indexer service started.
29/05/2024 23:44:51 INFO: Initializing Wazuh indexer cluster security settings.
29/05/2024 23:45:02 INFO: Wazuh indexer cluster initialized.
29/05/2024 23:45:02 INFO: --- Wazuh server ---
29/05/2024 23:45:02 INFO: Starting the Wazuh manager installation.
29/05/2024 23:47:39 INFO: Wazuh manager installation finished.
29/05/2024 23:47:39 INFO: Starting service wazuh-manager.
29/05/2024 23:47:57 INFO: wazuh-manager service started.
29/05/2024 23:47:57 INFO: Starting Filebeat installation.
29/05/2024 23:48:10 INFO: Filebeat installation finished.
29/05/2024 23:48:11 INFO: Filebeat post-install configuration finished.
29/05/2024 23:48:11 INFO: Starting service filebeat.
29/05/2024 23:48:12 INFO: filebeat service started.
29/05/2024 23:48:12 INFO: --- Wazuh dashboard ---
29/05/2024 23:48:12 INFO: Starting Wazuh dashboard installation.
29/05/2024 23:50:24 INFO: Wazuh dashboard installation finished.
29/05/2024 23:50:24 INFO: Wazuh dashboard post-install configuration finished.
29/05/2024 23:50:24 INFO: Starting service wazuh-dashboard.
29/05/2024 23:50:25 INFO: wazuh-dashboard service started.
29/05/2024 23:50:51 INFO: Initializing Wazuh dashboard web application.
29/05/2024 23:50:52 INFO: Wazuh dashboard web application initialized.
29/05/2024 23:50:52 INFO: --- Summary ---
29/05/2024 23:50:52 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
    User: admin
    Password: yHR4ZgOcQkZJ?mA*7y8K7L98+E.2qgai
29/05/2024 23:50:52 INFO: Installation finished.
root@Wazuh-2:~# SS
```

6.3 Deploying Wazuh agents

After the installations of the Wazuh Server, no agents were displayed on the dashboard as no agent had been deployed yet, as shown in the Figure 7. This showed that the successful installation of Wazuh components. The next step was to start deploying new agents to start monitoring the security events on the Wazuh dashboard.

Figure 7 shows the fresh dashboard of Wazuh with no agents deployed. This interface was shown after the successful login with the provided credentials at the end of the process, as shown in the Figure 6. All the shown services were active by default in various sections of the dashboard such as SIEM, XDR, Regulatory Compliance, and Auditing and Policy monitoring. This interface was shown after the successful login with the provided credentials at the end of the process as shown in the Figure 7

Figure 7 Wazuh dashboard with no agents.



Total agents 0

Active agents 0

Disconnected agents 0

Pending agents 0

Never connected agents 0

⚠ No agents were added to this manager. Add agent

SECURITY INFORMATION MANAGEMENT

Security events
Browse through your security alerts, identifying issues and threats in your environment.

Integrity monitoring
Alerts related to file changes, including permissions, content, ownership and attributes.

AUDITING AND POLICY MONITORING

Policy monitoring
Verify that your systems are configured according to your security policies baseline.

System auditing
Audit users behavior, monitoring command execution and alerting on access to critical files.

Security configuration assessment
Scan your assets as part of a configuration assessment audit.

THREAT DETECTION AND RESPONSE

Vulnerabilities
Discover what applications in your environment are affected by well-known vulnerabilities.

MITRE ATT&CK
Security events from the knowledge base of adversary tactics and techniques based on real-world observations

REGULATORY COMPLIANCE

PCI DSS
Global security standard for entities that process, store or transmit payment cardholder data.

NIST 800-53
National Institute of Standards and Technology Special Publication 800-53 (NIST 800-53) sets guidelines for federal information systems.

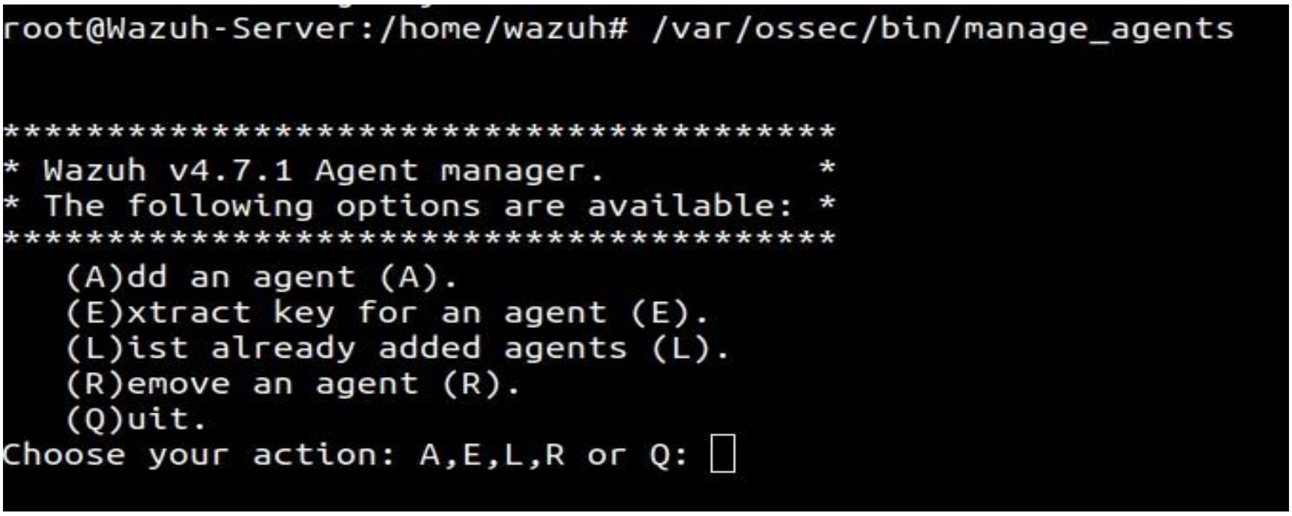
TSC
Trust Services Criteria for Security, Availability, Processing Integrity, Confidentiality, and Privacy

GDPR
General Data Protection Regulation (GDPR) sets guidelines for processing of personal data.

HIPAA
Health Insurance Portability and Accountability Act of 1996 (HIPAA) provides data privacy and security provisions for safeguarding medical information.

Two methods are employed to deploy agents to the Wazuh dashboard. One method is through the web interface on the dashboard and the other is through CLI on the server terminal, as shown in the Figure 8.

Figure 8 To deploy Wazuh agent through CLI.

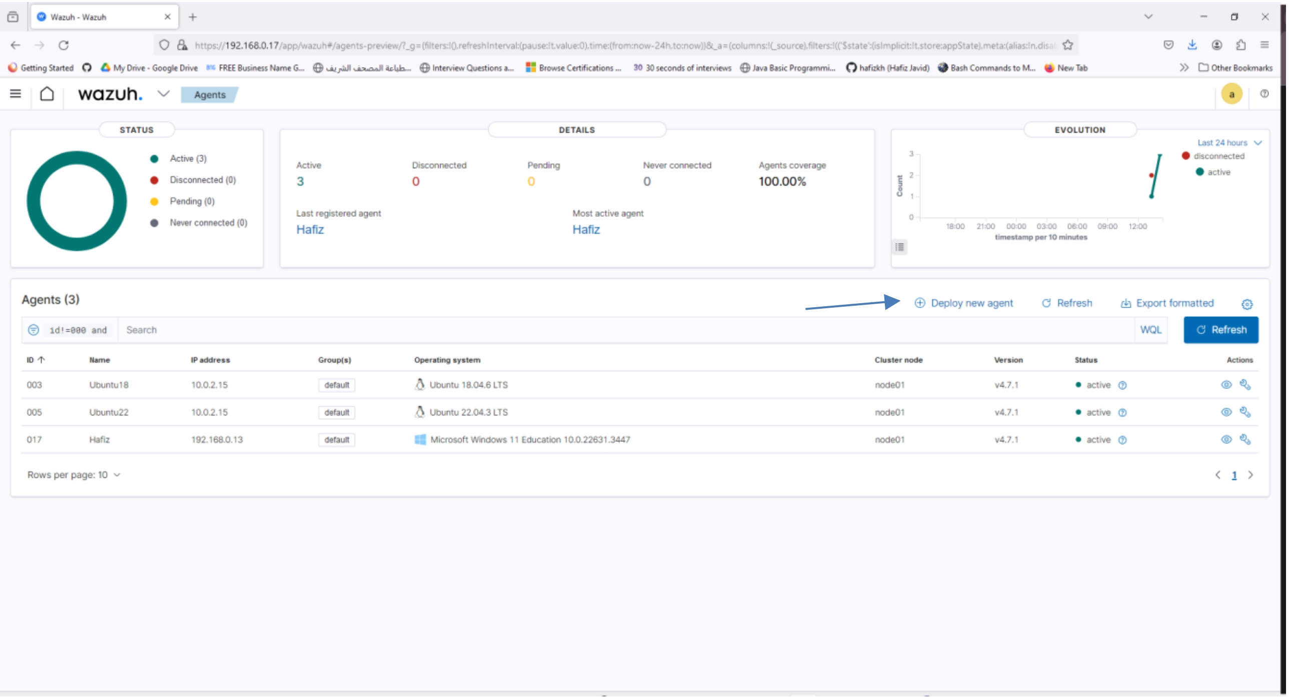


```
root@Wazuh-Server:/home/wazuh# /var/ossec/bin/manage_agents


****************************************
* Wazuh v4.7.1 Agent manager.          *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: 
```

To start with the deploying agent to demonstrate the method from the web interface on the Wazuh dashboard. There is the tab “Deploy new agent” on the dashboard as shown in the Figure 9.

Figure 9 shows the interface of Wazuh agents from Wazuh dashboard. In this figure, there are three deployed agents in which two of them are Ubuntu machines and one of them is Windows 11. The figure shows the status of all the agents, indicating whether active or disconnected, along with other details such as IDs, names, IP addresses, and the operating systems.

Figure 9 Interface of Wazuh agents.



Active	Disconnected	Pending	Never connected	Agents coverage
3	0	0	0	100.00%

Last registered agent: Hafiz

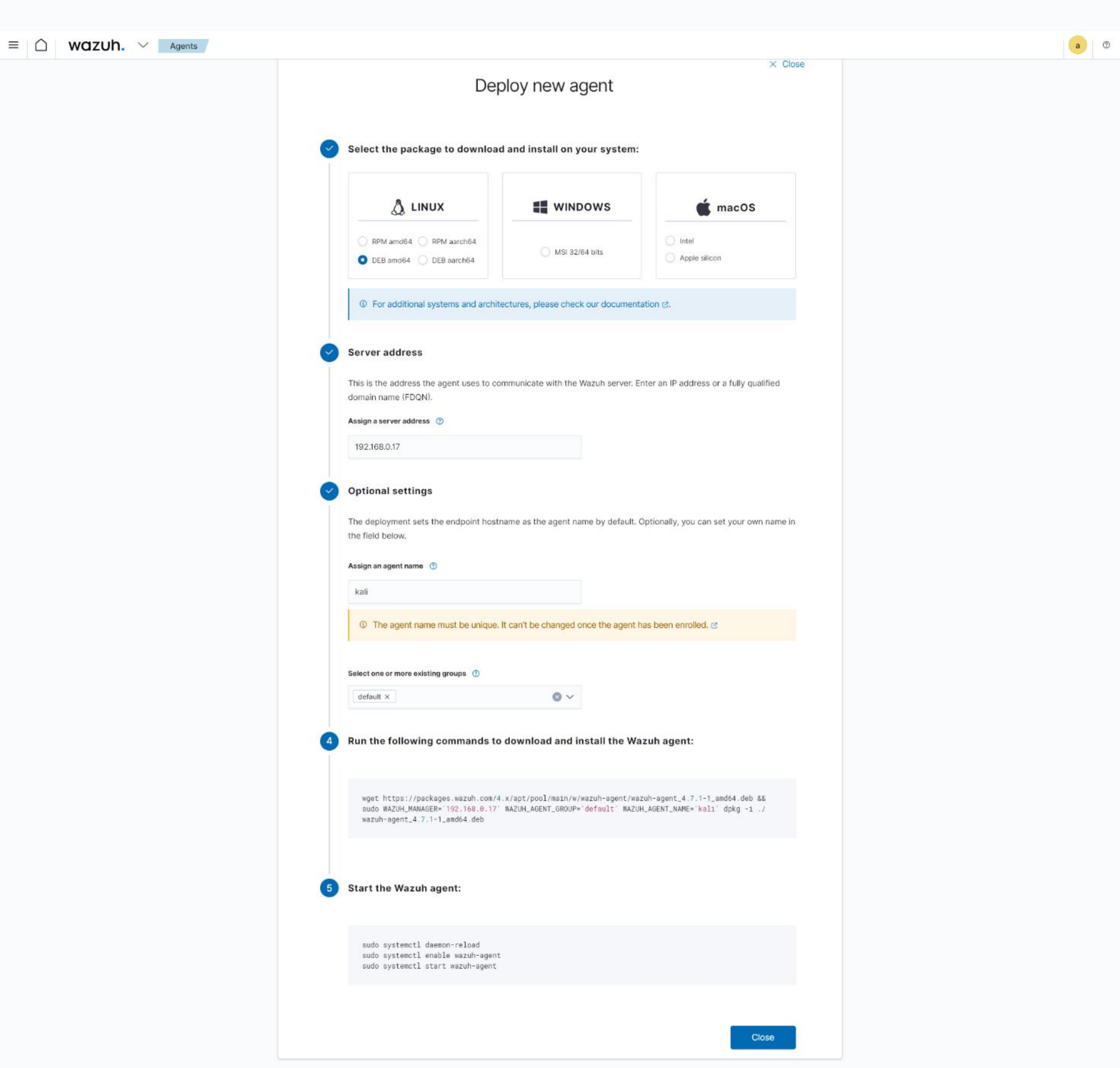
Most active agent: Hafiz

Agents (3)

ID ↑	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
003	Ubuntu18	10.0.2.15	default	Ubuntu 18.04.6 LTS	node01	v4.7.1	● active	
005	Ubuntu22	10.0.2.15	default	Ubuntu 22.04.3 LTS	node01	v4.7.1	● active	
017	Hafiz	192.168.0.13	default	Microsoft Windows 11 Education 10.0.22631.3447	node01	v4.7.1	● active	

There are required details that need to be filled out to complete the deployment process, as shown in Figure 10. In this figure, there are three options available in terms of operating systems. In this demonstration, Linux was selected as the Kali Linux machine was intended to be deployed on the Wazuh dashboard. All the required details were filled out such as the selection of operating system, the IP of the Wazuh manager, and the selection of existing groups. In this example, the default group selected. The command to download and install the Wazuh agent was ready to be executed in the terminal of Kali machine. To apply changes in the Wazuh agent, the enabling and starting the Wazuh agents is required by the following commands mentioned in Figure 10.

Figure 10 Details of newly deployed agents.



Deploy new agent

Select the package to download and install on your system:

LINUX
○ RPM amd64 ○ RPM aarch64
● DEB amd64 ○ DEB aarch64

WINDOWS
○ MSI 32/64 bits

macOS
○ Intel
○ Apple silicon

ⓘ For additional systems and architectures, please check our documentation.

Server address

This is the address the agent uses to communicate with the Wazuh server. Enter an IP address or a fully qualified domain name (FDQN).

Assign a server address: 192.168.0.17

Optional settings

The deployment sets the endpoint hostname as the agent name by default. Optionally, you can set your own name in the field below.

Assign an agent name: kali

ⓘ The agent name must be unique. It can't be changed once the agent has been enrolled.

Select one or more existing groups: default ×

4 Run the following commands to download and install the Wazuh agent:

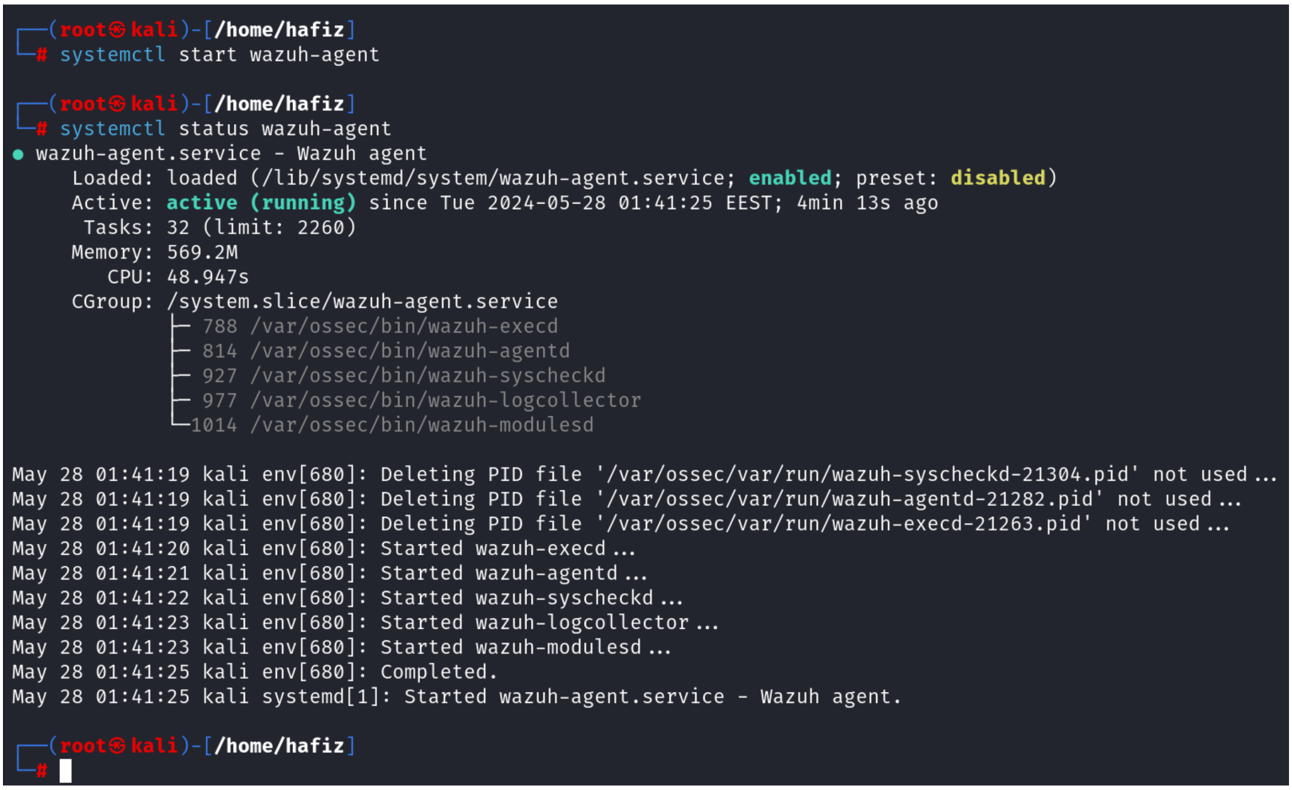
```
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.1-1_amd64.deb &&
sudo WAZUH_MANAGER='192.168.0.17' WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='kali' dpkg -i ./
wazuh-agent_4.7.1-1_amd64.deb
```

5 Start the Wazuh agent:

```
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
```

After executing the command mentioned in Figure 10 to download and install the Wazuh agent on the Kali machine, the status shows “active (running)” in the terminal, as shown in Figure 11. Once the status of the agent is running, there is need to examine the configuration file of the Wazuh agent. The path of the configuration file is “/var/ossec/etc/ossec.conf”. In the configuration file, the IP of the Wazuh manager must be the correct. If there is any variance in the IP, that must be the corrected to ensure the alerts and events are send to the Wazuh dashboard. Root privileges are required to run all the commands to install Wazuh and Wazuh agents.

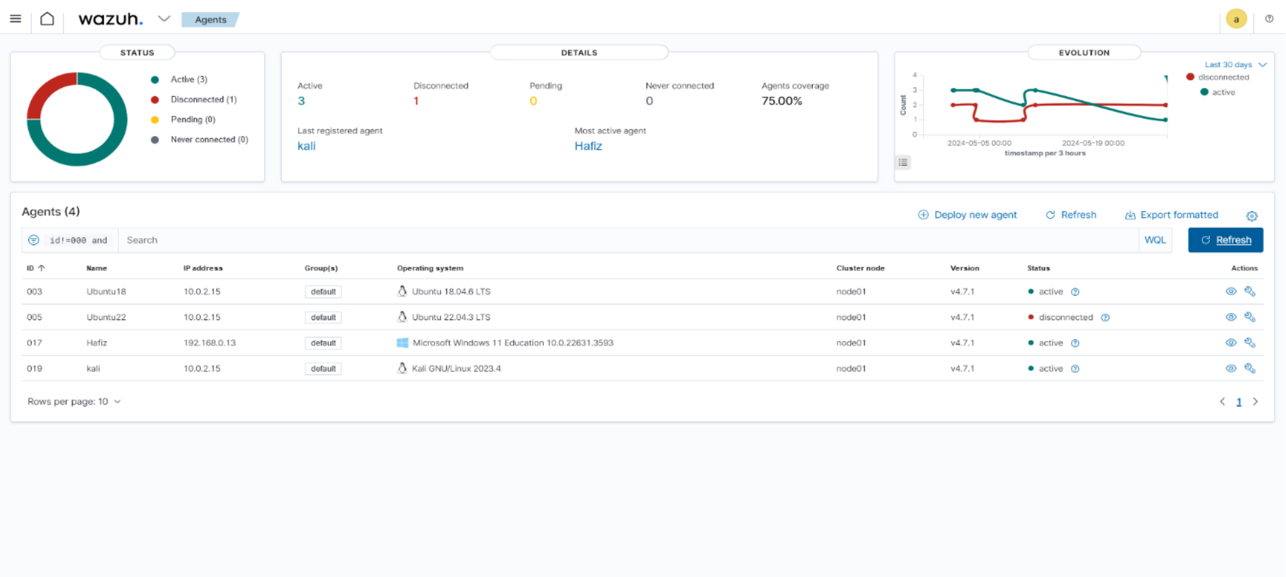
Figure 11 Status of the Wazuh agent.



```
┌──(root㉿kali)-[/home/hafiz]
└─# systemctl start wazuh-agent

┌──(root㉿kali)-[/home/hafiz]
└─# systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
     Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; preset: disabled)
     Active: active (running) since Tue 2024-05-28 01:41:25 EEST; 4min 13s ago
      Tasks: 32 (limit: 2260)
     Memory: 569.2M
        CPU: 48.947s
     CGroup: /system.slice/wazuh-agent.service
             ├─ 788 /var/ossec/bin/wazuh-execd
             ├─ 814 /var/ossec/bin/wazuh-agentd
             ├─ 927 /var/ossec/bin/wazuh-syscheckd
             ├─ 977 /var/ossec/bin/wazuh-logcollector
             └─1014 /var/ossec/bin/wazuh-modulesd

May 28 01:41:19 kali env[680]: Deleting PID file '/var/ossec/var/run/wazuh-syscheckd-21304.pid' not used...
May 28 01:41:19 kali env[680]: Deleting PID file '/var/ossec/var/run/wazuh-agentd-21282.pid' not used...
May 28 01:41:19 kali env[680]: Deleting PID file '/var/ossec/var/run/wazuh-execd-21263.pid' not used...
May 28 01:41:20 kali env[680]: Started wazuh-execd...
May 28 01:41:21 kali env[680]: Started wazuh-agentd...
May 28 01:41:22 kali env[680]: Started wazuh-syscheckd...
May 28 01:41:23 kali env[680]: Started wazuh-logcollector...
May 28 01:41:23 kali env[680]: Started wazuh-modulesd...
May 28 01:41:25 kali env[680]: Completed.
May 28 01:41:25 kali systemd[1]: Started wazuh-agent.service - Wazuh agent.

┌──(root㉿kali)-[/home/hafiz]
└─# █
```

There should be a new agent that appears in the dashboards after creating the new agent and pushing to the Wazuh dashboard, as shown in Figure 9. The fourth agent identified as ‘kali’ with other attributes including IP addresses, cluster, status, operating system, as depicted in Figure 12. This is a positive sign that the agent is deployed, and the security events are being sent to the Wazuh dashboard.

Figure 12 Fourth deployed agent.



Active 3 | Disconnected 1 | Pending 0 | Never connected 0 | Agents coverage 75.00%

Last registered agent: kali

Most active agent: Hafiz

Agents (4)

ID ↑	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
003	Ubuntu18	10.0.2.15	default	Ubuntu 18.04.6 LTS	node01	v4.7.1	● active	
005	Ubuntu22	10.0.2.15	default	Ubuntu 22.04.3 LTS	node01	v4.7.1	● disconnected	
017	Hafiz	192.168.0.13	default	Microsoft Windows 11 Education 10.0.22631.3593	node01	v4.7.1	● active	
019	kali	10.0.2.15	default	Kali GNU/Linux 2023.4	node01	v4.7.1	● active	

To restart Wazuh on the web interface, this is required to run the following commands on the terminal of the Wazuh Server, as described in the Code Snippet 4. These commands ensure that Wazuh indexer and the Wazuh manager started properly. The commands are given in Code Snippet 4.

Code Snippet 4 Restarting the Wazuh Indexer and Manager

```
systemctl daemon-reload

systemctl enable wazuh-indexer

systemctl start wazuh-indexer

sytemctl restart Wazuh-manager
```

As seen in Code Snippet 4, the manager configurations are reloaded and maintained up to date. The other commands launch the Wazuh indexer and restart the Wazuh manager instantly.

6.4 Vulnerability detection and Mitigation with Wazuh

Robust cybersecurity requires the ability to identify and fix vulnerabilities in machines. This section intends to examine Wazuh's vulnerability detection feature. The first part explains enabling the vulnerability detection in the configuration file of Wazuh server. The second part demonstrates the testing phase to ensure the effectiveness of these configurations and the security of the system. A comprehensive approach secures the environment against the major threats.

6.4.1 Vulnerability Detection Configuration

Enabling the Vulnerability Detector module and configuring the scan are necessary before conducting vulnerability scans in Wazuh. After the installation of Wazuh agent, Syscollector is activated by default, whereas the Vulnerability Detector module remains disabled on the Wazuh server.

The following block of settings was added in each of agents’ configuration file. The path of the configuration file is /var/ossec/etc/ossec.conf. Code Snippet 5 shows the snippet of XML configuration that was added in the ossec.conf file under the syscollector module. The configuration shows that data collection takes place every hour and the operating system’s information is set to gather and applied hotfixes and the module is currently enabled.

Code Snippet 5 Vulnerability detector block of Settings.

```
<wodle name="syscollector">
   <disabled>no</disabled>
   <interval>1h</interval>
   <os>yes</os>
   <packages>yes</packages>
   <hotfixes>yes</hotfixes>
</wodle>
```

Secondly, there is another vulnerability detection configuration block that needs to be configured in the Wazuh Server. The configuration block contains the information of every possible operating system available like Ubuntu, Debian, RedHat, Amazon Linux, and Windows.

Code Snippet 6 is an XML block of code to set up vulnerability detection system which checks the vulnerabilities in various operating systems regularly. The general settings trigger the detector to start a scan with the interval of 5 minutes and the full scan of the machines take place in every 6 hours. This scan initiates the process when the system starts. The repeated scans help in identifying and mitigating the security issues.

In the XML block of code, there is the configuration of each operating system such as Canonical for Ubuntu, Debian, RedHat, Amazon Linux, and Windows. In every 1 hour, system checks for new vulnerabilities and this system supports every provider of operating system. Code Snippet 6 shows Ubuntu, Debian, and RedHat but in the actual snippet, it shows the NVD provider to obtain comprehensive vulnerability data. This setup maintains the high level of security across various operating systems.

After successful completion of configuration in both Wazuh Server and Wazuh agent, Server must be restarted to apply the changes by the following command in the Wazuh-server terminal as shown in Code Snippet 4.

Code Snippet 6 Configuration of Vulnerability detector for Wazuh Server

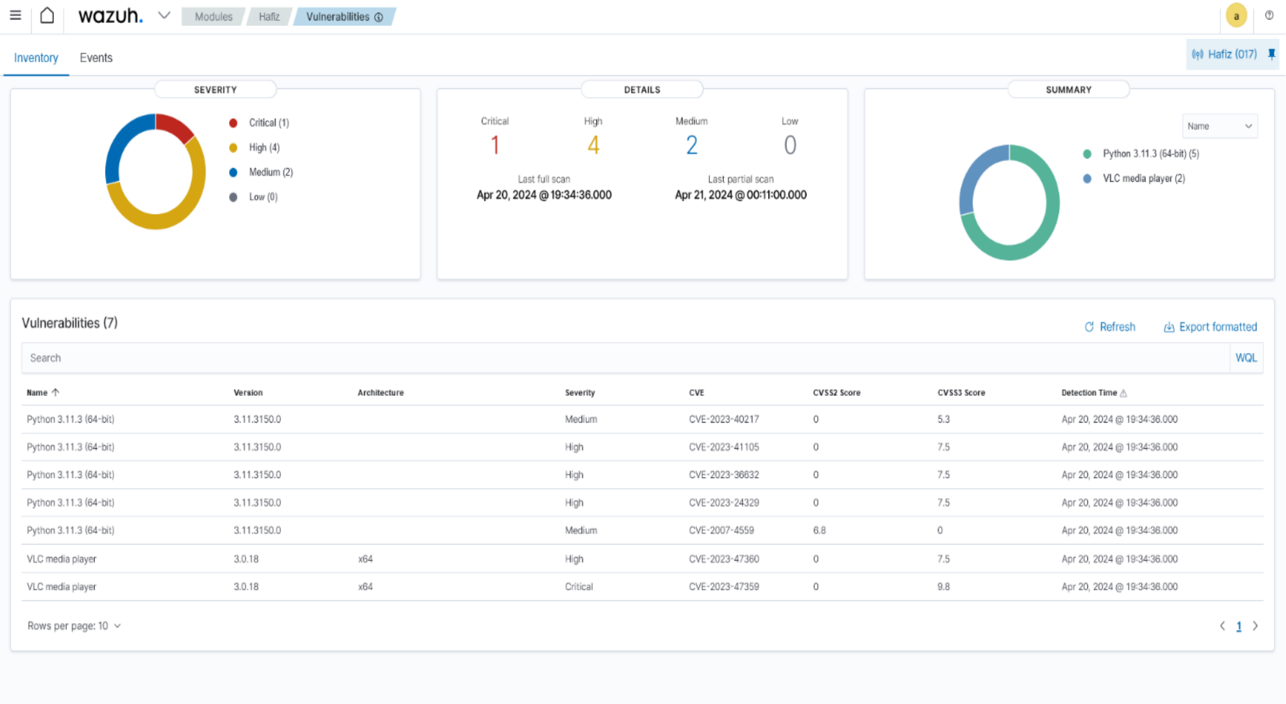
```
<vulnerability-detector>
   <enabled>yes</enabled>
   <interval>5m</interval>
   <min_full_scan_interval>6h</min_full_scan_interval>
   <run_on_start>yes</run_on_start>

   <!-- Ubuntu OS vulnerabilities -->
   <provider name="canonical">
      <enabled>yes</enabled>
      <os>trusty</os>
      <os>xenial</os>
      <os>bionic</os>
      <os>focal</os>
      <os>jammy</os>
      <update_interval>1h</update_interval>
   </provider>

   <!-- Debian OS vulnerabilities -->
   <provider name="debian">
      <enabled>yes</enabled>
      <os>buster</os>
      <os>bullseye</os>
      <os>bookworm</os>
      <update_interval>1h</update_interval>
   </provider>

   <!-- RedHat OS vulnerabilities -->
   <provider name="redhat">
      <enabled>yes</enabled>
      <os>5</os>
      <os>6</os>
      <os>7</os>
      <os>8</os>
      <os>9</os>
      <update_interval>1h</update_interval>
```

6.4.2 Testing Vulnerability Detection and Mitigation

The configured Wazuh manager conducted a thorough scan of the Windows agents within the network. Full and partial scans were the two types of scans used. The full scan is a comprehensive analysis that is set to run on a regular interval of every 6 hours and verifies that all system configurations and applications are free of known vulnerabilities. To maintain an up-to-date view of the system's security posture, the partial scan acts as a frequent check and focuses on programs or configurations changed since the last full scan.

The full scan was carried out as shown in Figure 13, on April 20, 2024, at 19:34:36, searching the Windows agents for known vulnerabilities. Several vulnerabilities were found thanks to full scanning process, and the vulnerabilities were categorized into critical, high, medium, and low categories based on severity. Particularly, a critical flaw in VLC Media Player was found, indicating a serious security threat that needed to be fixed.

Figure 13 Vulnerability scan of Wazuh agent.



Critical	High	Medium	Low
1	4	2	0

Last full scan
Apr 20, 2024 @ 19:34:36.000

Last partial scan
Apr 21, 2024 @ 00:11:00.000

Vulnerabilities (7)

Name ↑	Version	Architecture	Severity	CVE	CVSS2 Score	CVSS3 Score	Detection Time ⚠
Python 3.11.3 (64-bit)	3.11.3150.0		Medium	CVE-2023-40217	0	5.3	Apr 20, 2024 @ 19:34:36.000
Python 3.11.3 (64-bit)	3.11.3150.0		High	CVE-2023-41105	0	7.5	Apr 20, 2024 @ 19:34:36.000
Python 3.11.3 (64-bit)	3.11.3150.0		High	CVE-2023-36632	0	7.5	Apr 20, 2024 @ 19:34:36.000
Python 3.11.3 (64-bit)	3.11.3150.0		High	CVE-2023-24329	0	7.5	Apr 20, 2024 @ 19:34:36.000
Python 3.11.3 (64-bit)	3.11.3150.0		Medium	CVE-2007-4559	6.8	0	Apr 20, 2024 @ 19:34:36.000
VLC media player	3.0.18	x64	High	CVE-2023-47360	0	7.5	Apr 20, 2024 @ 19:34:36.000
VLC media player	3.0.18	x64	Critical	CVE-2023-47359	0	9.8	Apr 20, 2024 @ 19:34:36.000

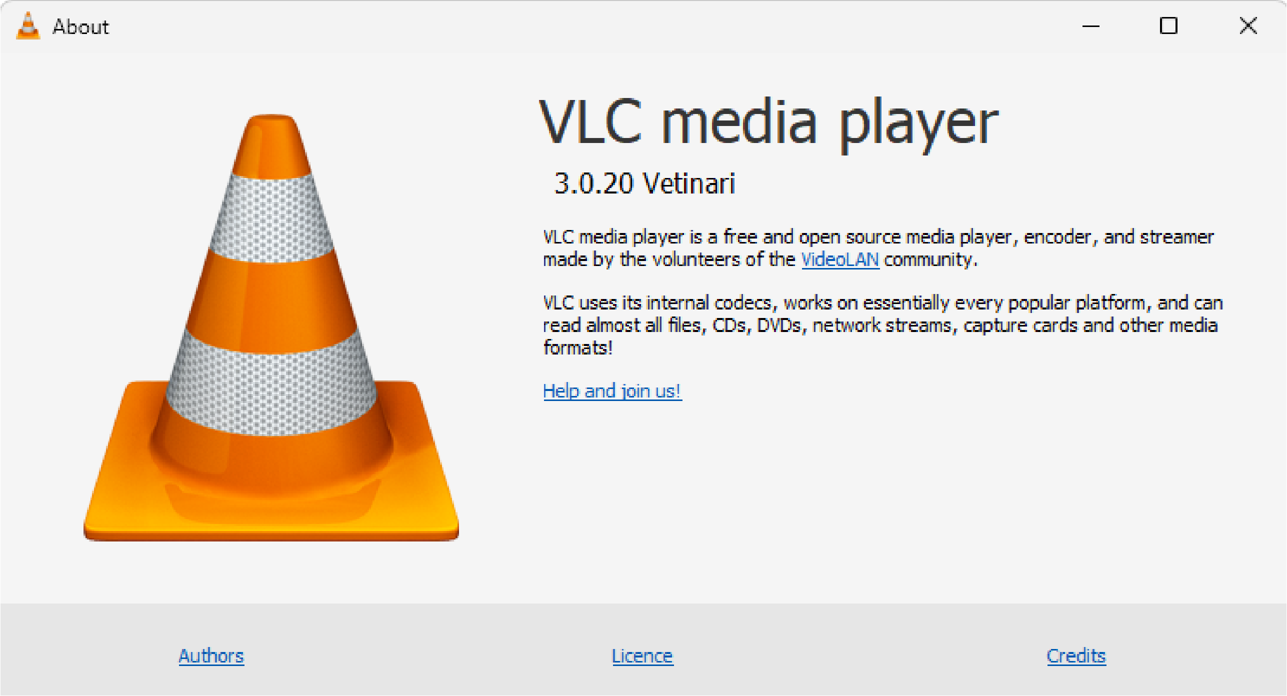
The vulnerabilities found in VLC media player and Python were classified as critical and high respectively. Typically, critical vulnerabilities system suggest that system can compromise with minimal human intervention. In Figure 13, detected vulnerabilities were addressed into different categories. The details were mentioned in the CVE (Common Vulnerabilities and Exposures), providing access to thorough details, such as possible effects and corrective actions.

The primary takeaway from CVE-2023-47359 advisory is that two vulnerabilities in the VLC media player concerning the MMS over HTTP protocol have been addressed. These weaknesses include a heap buffer overflow issue (CVE-2023-47359) and an integer underflow issue (CVE-2023-47360). The users are encouraged to update the VLC software to the recent version which is version 3.0.20 to address these risks. The updated version of the VLC player is shown in Figure 14.

This vulnerability has been given the identifier CVE-2023-24329 and is related to the urllib module. This is because the parse component of Python versions prior to 3.11.4 permits attackers to evade blocklisted URLs by employing those that start with a blank. Upgrading to Python 3.11.4 or later versions resolves this problem. Therefore, the version of Python was updated to 3.12.3 to mitigate the issue in terms of vulnerability, as shown in Figure 15.

Figure 14 shows the updated version of VLC media player from 3.0.18 to 3.0.20 that addressed in the description of CVE-2023-47359. This helped mitigate the critical vulnerability in the Windows agent.

Figure 14 VLC player version.



VLC media player

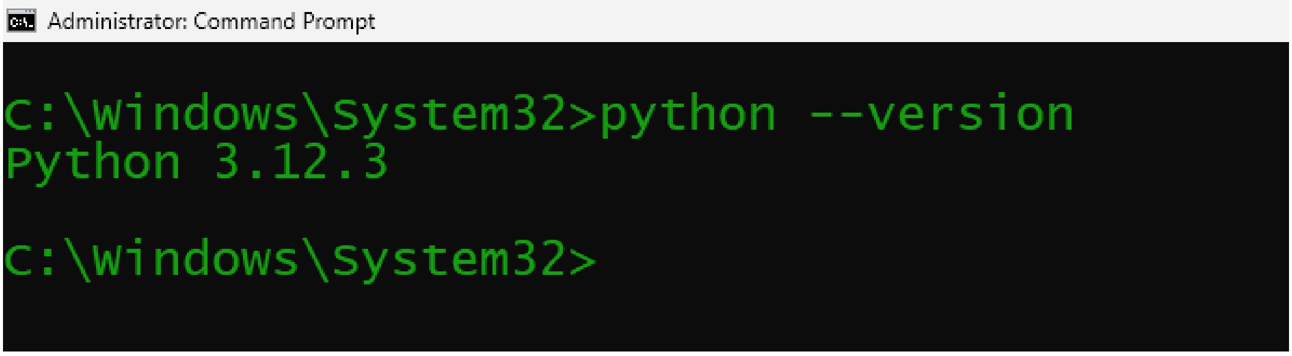
3.0.20 Vetinari

VLC media player is a free and open source media player, encoder, and streamer made by the volunteers of the VideoLAN community.

VLC uses its internal codecs, works on essentially every popular platform, and can read almost all files, CDs, DVDs, network streams, capture cards and other media formats!

Figure 15 shows the latest version of Python, i.e. 3.12.3, that mitigated the vulnerability and showed the results in the Wazuh dashboard as shown in Figure 16 after the subsequent scan. This mitigation was addressed by CVE-2023-24329.

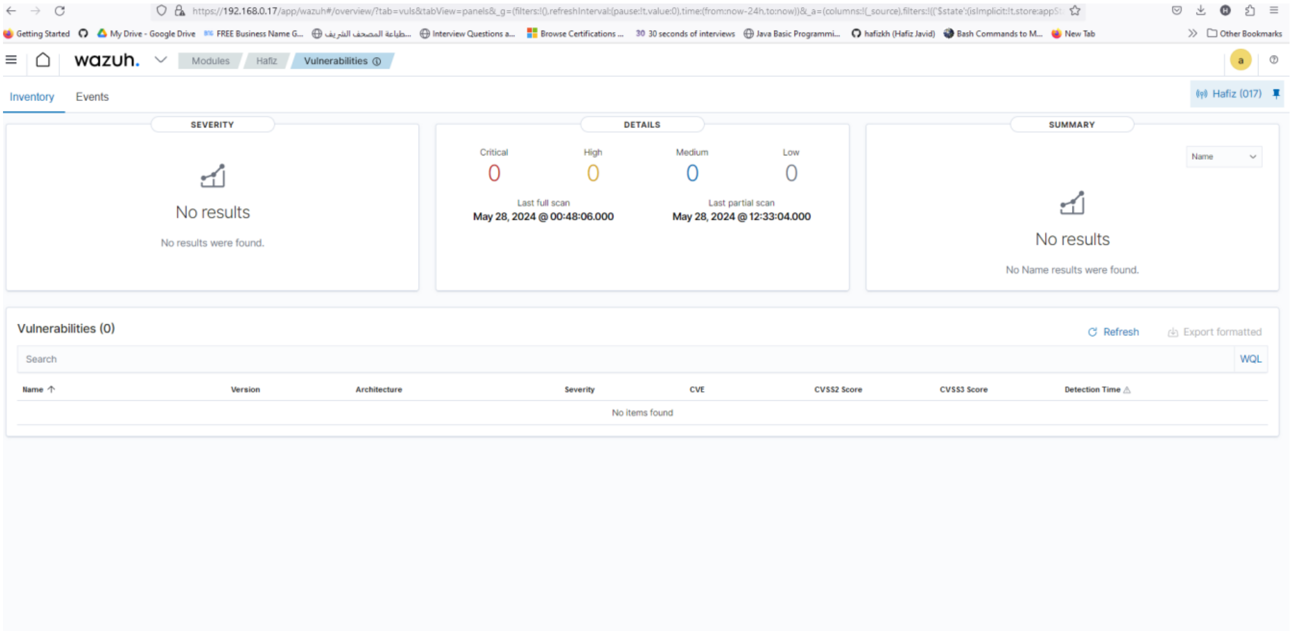
Figure 15 Python Environment Version



```
C:\Windows\System32>python --version
Python 3.12.3

C:\Windows\System32>
```

After an initial scan by Wazuh, a critical security flaw in VLC Media Player was found on the Windows machine agent. The vulnerability was critical that needed to be fixed right away. So, VLC Media Player was updated to the latest version according to instruction of CVE-2023-47359. Moreover, the Python Environment was updated to version 3.12.3 to address CVE-2023-24329. The next full scan took place by the tool on May 28, 2024, at 00:48:06 as shown in Figure 16 and that can be noted that the system showed zero vulnerability. When the vulnerable software was updated, the security of Windows machine improved after the second subsequent scan. Notably, the critical vulnerability with VLC Media Player and Python vulnerabilities were no longer listed as shown in Figure 16.

Figure 16 Resolved Vulnerabilities from Wazuh agent.



Critical	High	Medium	Low
0	0	0	0

Last full scan
May 28, 2024 @ 00:48:06.000

Last partial scan
May 28, 2024 @ 12:33:04.000

Vulnerabilities (0)

Name ↑	Version	Architecture	Severity	CVE	CVSS2 Score	CVSS3 Score	Detection Time

No items found

The vulnerability scans showed that the current state of the system is more secure after updating the VLC Media Player and the Python environment. Features of Wazuh such as scanning and monitoring helped in identifying and mitigating the security vulnerabilities, thus enhancing the security of the deployed agents.

6.5 Evaluation of Wazuh’s SIEM and XDR Capabilities

In today's ever-changing cybersecurity landscape, this is crucial to quickly identify, assess, and respond to security risks. This section delves into the capabilities of Wazuh as a Security Information and Event Management (SIEM) and Extended Detection and Response (XDR) solution, especially the File Integrity Monitoring (FIM) module and VirusTotal integration. The assessment considered how Wazuh uses these tools to enhance security in a lab environment, making sure that both known and new threats can be handled. The evaluation happened in two phases: Firstly, the focus was on FIM configurations and effectiveness to keep track of important system files in real time. Secondly, integration of VirusTotal database contributed to threat detection.

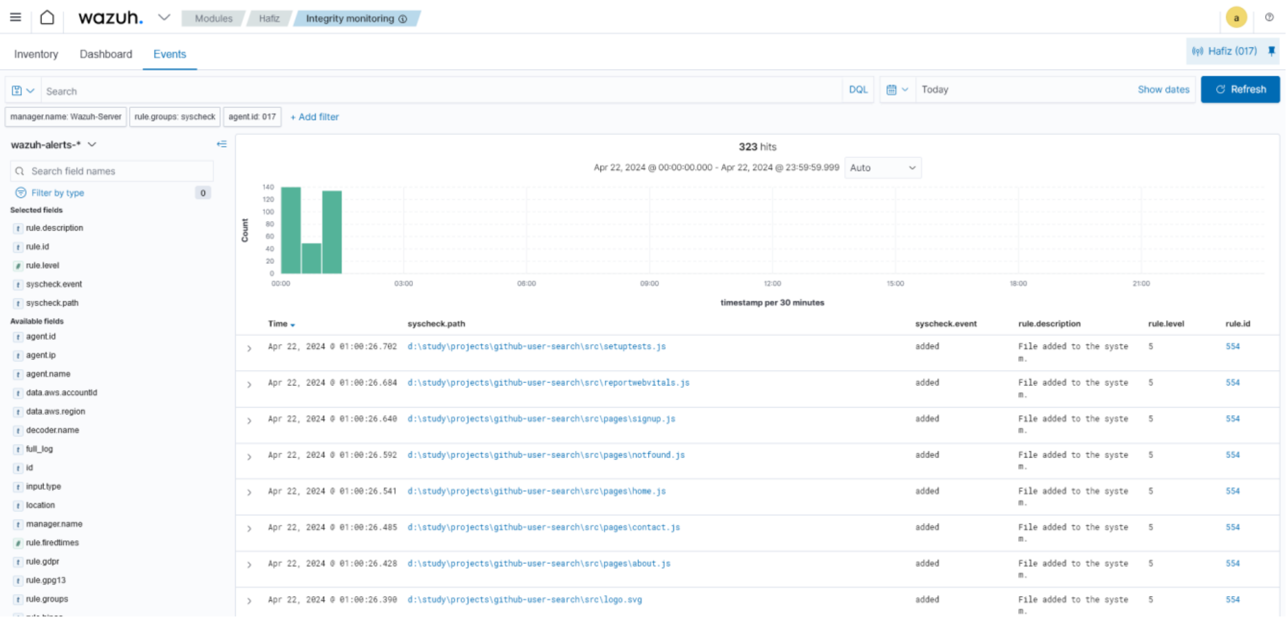
6.5.1 File Integrity Monitoring

Configuration of File Integrity Monitoring (FIM) is essential to improve the protection of sensitive directories in real-time. FIM was configured to monitor the allocated file and directories in the configuration file (ossec.conf) in the Wazuh agent. In this way, any changes in those files and directories were flagged and reported the events in the Wazuh dashboard in real-time. The FIM module continuously monitors the designated directory, ensuring integrity.

There were certain tests conducted after the configurations in the Wazuh agent of FIM to check the reliability of the module. Few changes were made in the allocated directories to mimic the real changes and events about these changes were sent to the dashboard along with all the required details in the form of Json format. This can be seen in the events in Figure 17, showing what happened in real time and indicating possible action that need to take place.

The screenshot in Figure 17 glimpses File Integrity Monitoring (FIM) in the Wazuh dashboard. A timeline of detected events was demonstrated. Each entry shows the details of specific information about the time, type, and affected file when a change was identified in the directory. This real-time logging improves security by enabling the immediate detection and response to possible threats.

Figure 17 File Integrity Monitoring



manager.name: Wazuh-Server | rule.groups: syscheck | agent.id: 017

Time	syscheck.path	syscheck.event	rule.description	rule.level	rule.id
Apr 22, 2024 @ 01:00:26.702	d:\study\projects\github-user-search\src\setuptests.js	added	File added to the system.	5	554
Apr 22, 2024 @ 01:00:26.684	d:\study\projects\github-user-search\src\reportwebvitals.js	added	File added to the system.	5	554
Apr 22, 2024 @ 01:00:26.640	d:\study\projects\github-user-search\src\pages\signup.js	added	File added to the system.	5	554
Apr 22, 2024 @ 01:00:26.592	d:\study\projects\github-user-search\src\pages\notfound.js	added	File added to the system.	5	554
Apr 22, 2024 @ 01:00:26.541	d:\study\projects\github-user-search\src\pages\home.js	added	File added to the system.	5	554
Apr 22, 2024 @ 01:00:26.485	d:\study\projects\github-user-search\src\pages\contact.js	added	File added to the system.	5	554
Apr 22, 2024 @ 01:00:26.428	d:\study\projects\github-user-search\src\pages\about.js	added	File added to the system.	5	554
Apr 22, 2024 @ 01:00:26.390	d:\study\projects\github-user-search\src\logo.svg	added	File added to the system.	5	554

6.5.2 Integration of VirusTotal

VirusTotal integrated with Wazuh to enhance security. The key for the VirusTotal API is used to connect the data hub of VirusTotal. Then a few corresponding steps such as updating the Wazuh server's configuration file (ossec.conf) and script were written to remove the threat in a separate file. The Wazuh utilizes VirusTotal massive database in the checking of files' reputations in this means. In this process, VirusTotal looks at the file history and domain reputation to establish credibility and integrity.

Besides the API key, a script was written for the Wazuh agent to automate the threat detection process. The script removes the malware threat if detected. This prevents the spread and detects the suspicious file; therefore, immediate action is taken to remove the suspicious file. The script is significant for responding to external threats with instant action. Since this was integrated into Wazuh, local rules in the agent, were modified to notify the script automatically when such alerts were generated by VirusTotal.

The file downloaded onto Ubuntu 18.04 LTS and Ubuntu 22.04 LTS machines (Wazuh agent) was marked as suspicious on VirusTotal. The tool quickly recognized the file as malicious and then made sure to remove from the machine. All this information was automatically documented, and, in the next step, the VirusTotal module in the Wazuh dashboard provided the details about the event. The events were generated in the VirusTotal events module as well as in the FIM module because a file was used in testing.

The file that was used for the testing of VirusTotal in the conducted study is from EICAR's website, which is a reputed organization that provides test files for antivirus and other security software applications. The file can be downloaded using Code Snippet 7.

In the current exercise, the file downloaded from the EICAR secured source was named "suspicious-file.exe" and uploaded to the Wazuh agent to mimic the behaviour of a threat that does not pose any threat to the machine. In the testing environment, one can validate the measures of security and the effectiveness of analysis for threats and risks.

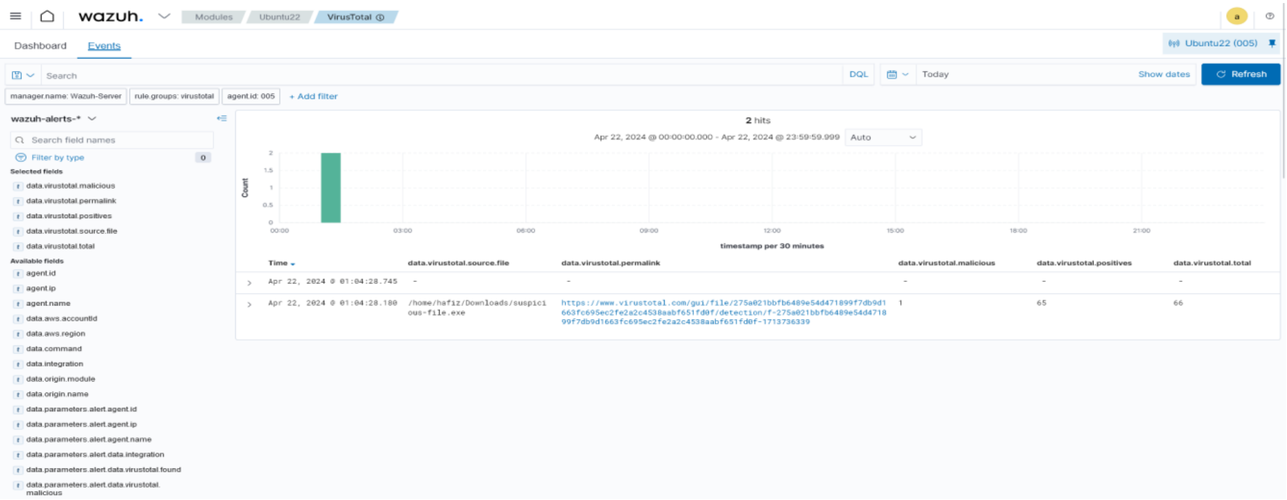
Code Snippet 7 Source of Suspicious file.

```
sudo curl -Lo /home/hafiz/Downloads/suspicious-file.exe
https://secure.eicar.org/eicar.com
```

The purpose of introducing this file to the lab environment is to assess the configuration of the Wazuh platform regarding these activities as well as how Wazuh handles these activities. The efficiency of the security system and the reaction time are evaluated using the file with the VirusTotal. This results from the selection of this file from EICAR, which makes the testing environment safe, hence, providing accurate outcomes without the interference of real malware.

The screenshot in Figure 18 from the Wazuh dashboard demonstrated the VirusTotal connection. Filtered events were displayed from VirusTotal, disclosing which files were identified as potentially harmful. Additionally, the log confirmed that the file was handled. This shows the power of the integrated system to identify threats and respond automatically.

Figure 18 VirusTotal Integration in Wazuh



manager.name: Wazuh-Server | rule.groups: virustotal | agent.id: 005

Time	data.virustotal.source.file	data.virustotal.permalink	data.virustotal.malicious	data.virustotal.positives	data.virustotal.total
Apr 22, 2024 @ 01:04:28.745	-	-	-	-	-
Apr 22, 2024 @ 01:04:28.180	/home/hafiz/Downloads/suspicious-file.exe	https://www.virustotal.com/gui/file/275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f/detection/f-275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f-1713736339	1	65	66

6.6 Simulated Attack with Atomic Red Team

Atomic Red Team is a PowerShell based tool used to execute elementary tests for security teams to analyze the network. These tests are small, have as little interdependence as possible and are organized in a structure that is well suited for use in test automation frameworks. The tool leverages the MITRE ATT&CK framework that categorizes different techniques. Atomic Red Team provides several tests known as “atomics”. These atomics are used to simulate the environment and evaluate the defense of the network.

Atomic Red Team is a comprehensive platform that contains a huge list of attack types to model real-life cyber threats. One of the techniques used is Credential Dumping (T1003), which intends to mimic the actual extraction of password hashes from the system, while another, known as Phishing (T1566), intends to imitate the actual delivery of a payload through a URL in an email. Other available attacks include Win32/PowerShell command-line shell and scripting language (T1059.001), SSH Brute Force (T1110.001), an attempt to guess password strength, and Port Scanning (T1046), a search on the target host for open ports and services. Also, impersonation of injecting a malicious code into running process is also performed by Process Injection (T1055) technique, and Data Exfiltration (T1020) conducts a simulation of extracting data from a network.

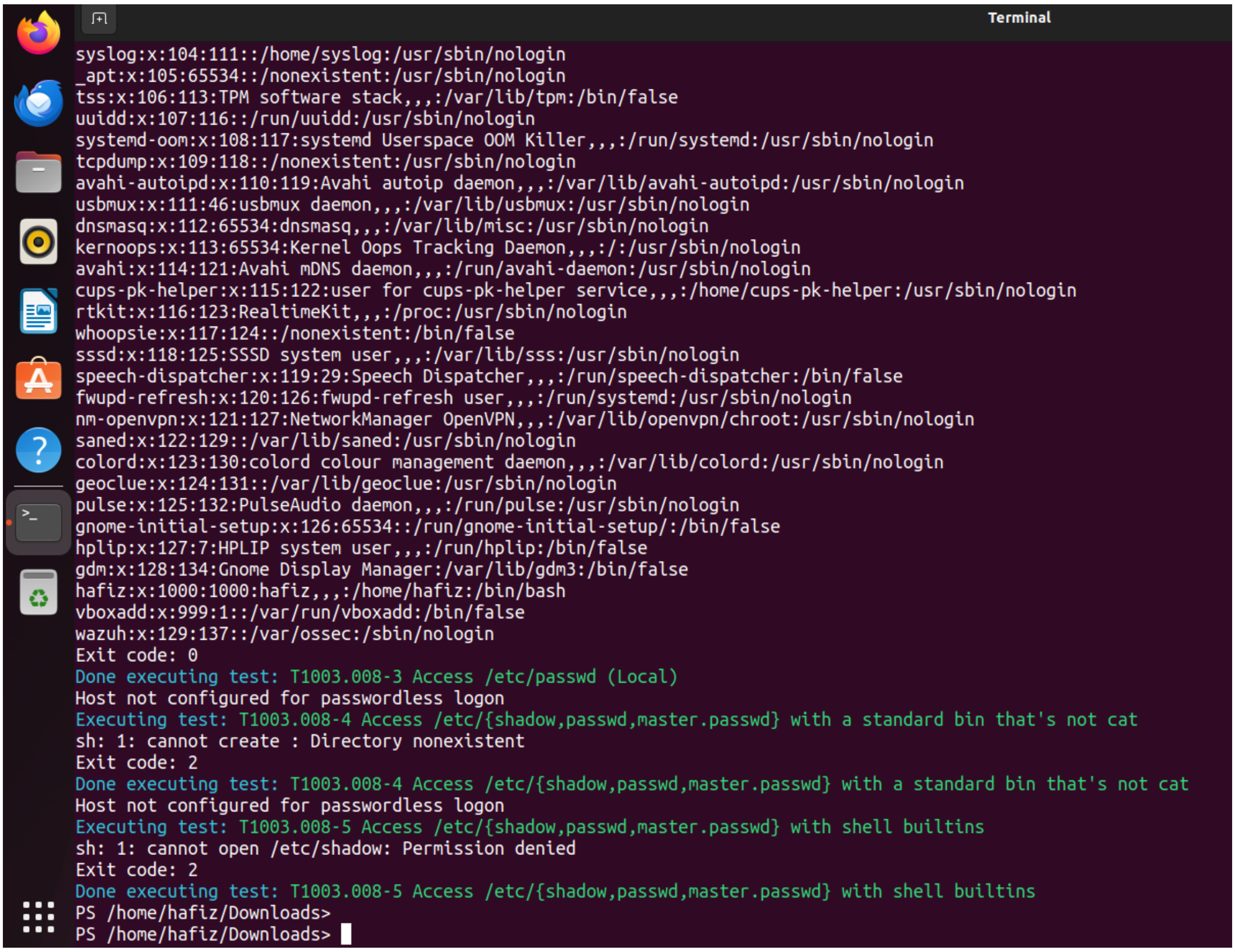
6.6.1 OS Credentials Dumping Atomics

To test the reliability of cybersecurity protections, a virtual environment was created in the Ubuntu 22.04 LTS. The framework of MITRE ATT&CK was mimicked by Atomic Red Team tool. The simulated attack was carried out to crack the credentials of the Linux user using the MITRE ATT&CK technique ID (T1003.008), focusing on sensitive authentication files (/etc/passwd and /etc/shadow).

A simulation was conducted to try to retrieve and possibly steal data from system files that hold user login credentials in a Unix-based operating system. The script executed several commands to determine how well the system would respond when given unauthorized access to the target machine. The results showed that some commands were not permitted, which proved that the settings worked well as the contents of /etc/shadow (a file with encrypted user password information) were not able to access. This kind of simulation is highly effective for testing how well a system is protected from cyber threats.

Figure 19 shows an attack using the Atomic Red Team framework with the MITRE ATT&CK ID T1003.008. A few attempts were made to access the /etc/passwd and /etc/shadow files in Unix to manage user accounts and authentication. However, the "Permission denied" messages show that the simulation was not successful to access these files, proves that the security of the system effectively blocked the unauthorized access to the credentials.

Figure 19 Simulated attack by Atomic Red Team with MITRE ATT&CK framework.



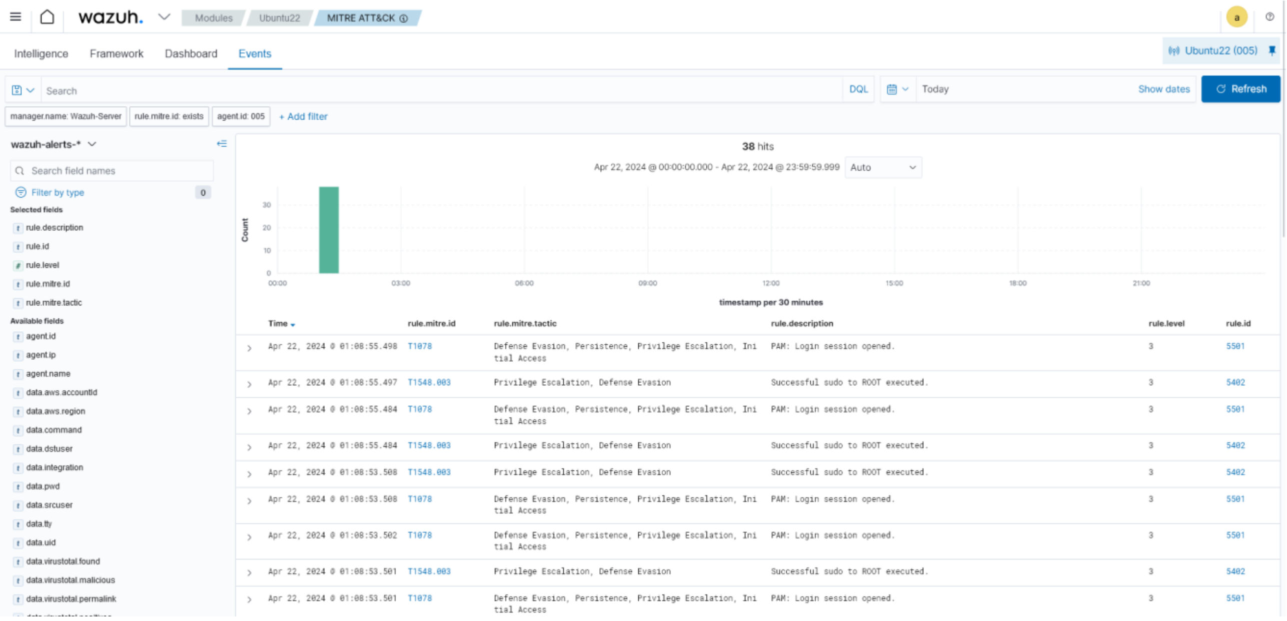
```
syslog:x:104:111::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:113:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:116::/run/uuidd:/usr/sbin/nologin
systemd-oom:x:108:117:systemd Userspace OOM Killer,,,:/run/systemd:/usr/sbin/nologin
tcpdump:x:109:118::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
avahi:x:114:121:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
cups-pk-helper:x:115:122:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
rtkit:x:116:123:RealtimeKit,,,:/proc:/usr/sbin/nologin
whoopsie:x:117:124::/nonexistent:/bin/false
sssd:x:118:125:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
speech-dispatcher:x:119:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
fwupd-refresh:x:120:126:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
nm-openvpn:x:121:127:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
saned:x:122:129::/var/lib/saned:/usr/sbin/nologin
colord:x:123:130:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:124:131::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:125:132:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:126:65534::/run/gnome-initial-setup/:/bin/false
hplip:x:127:7:HPLIP system user,,,:/run/hplip:/bin/false
gdm:x:128:134:Gnome Display Manager:/var/lib/gdm3:/bin/false
hafiz:x:1000:1000:hafiz,,,:/home/hafiz:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
wazuh:x:129:137::/var/ossec:/sbin/nologin
Exit code: 0
Done executing test: T1003.008-3 Access /etc/passwd (Local)
Host not configured for passwordless logon
Executing test: T1003.008-4 Access /etc/{shadow,passwd,master.passwd} with a standard bin that's not cat
sh: 1: cannot create : Directory nonexistent
Exit code: 2
Done executing test: T1003.008-4 Access /etc/{shadow,passwd,master.passwd} with a standard bin that's not cat
Host not configured for passwordless logon
Executing test: T1003.008-5 Access /etc/{shadow,passwd,master.passwd} with shell builtins
sh: 1: cannot open /etc/shadow: Permission denied
Exit code: 2
Done executing test: T1003.008-5 Access /etc/{shadow,passwd,master.passwd} with shell builtins
PS /home/hafiz/Downloads>
PS /home/hafiz/Downloads>
```

The fact that Wazuh security platform detected simulated attack patterns and has been the right tool for this activity is a sign of excellent performance. Suitable alerting related to these activities is very important. The gathered logs provided a detailed record of the incident, event analysis and fortifying the security measures against future attacks.

Figure 20 shows the outputs on the dashboard of Wazuh regarding the security of Ubuntu 22.04 LTS. The screenshot shows the various alerts that demonstrate the ability of Wazuh to detect and record various attack-related activities. These are categorized according to the MITRE ATT&CK framework. Some recorded events indicate the attempts to elevate privileges, execute commands, and start login sessions. Alerts like "Successful sudo to ROOT executed" and "Login session opened" are common signs of an attack, where attackers intend to gain access in the system.

Figure 20 shows the events section of the Wazuh dashboard. Security alerts logged from an Ubuntu system during an Atomic Red Team attack simulation. The alerts include events related to privilege escalation and evasion, such as "Successful sudo to ROOT executed" and "Login session opened." These logs help to see how the system's defenses performed against the mock attack, displaying the monitoring and alerting capabilities of Wazuh against simulated threats.

Figure 20 Events from Wazuh dashboard for MITRE ATT&CK



Time	rule.mitre.id	rule.mitre.tactic	rule.description	rule.level	rule.id
Apr 22, 2024 @ 01:08:55.498	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3	5501
Apr 22, 2024 @ 01:08:55.497	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3	5402
Apr 22, 2024 @ 01:08:55.484	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3	5501
Apr 22, 2024 @ 01:08:55.484	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3	5402
Apr 22, 2024 @ 01:08:53.508	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3	5402
Apr 22, 2024 @ 01:08:53.508	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3	5501
Apr 22, 2024 @ 01:08:53.502	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3	5501
Apr 22, 2024 @ 01:08:53.501	T1548.003	Privilege Escalation, Defense Evasion	Successful sudo to ROOT executed.	3	5402
Apr 22, 2024 @ 01:08:53.501	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	PAM: Login session opened.	3	5501

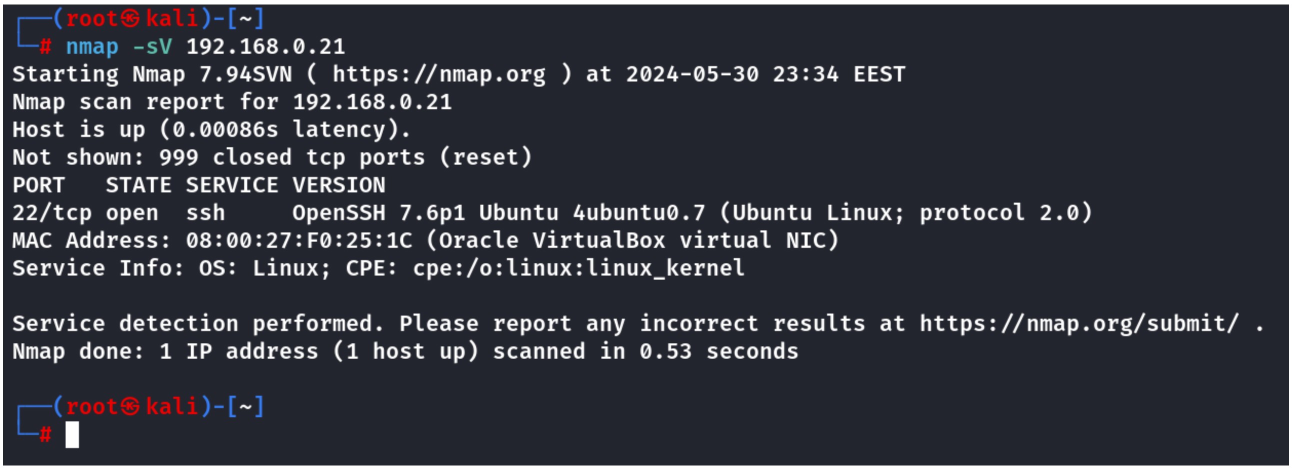
Atomic Red Team was first used in practicing port scanning and SSH login brute force attacks but a shift to using Kali Linux to perform these attacks were preferred because of the availability of more extensive toolkits coupled with flexibility within a virtual operating environment. For instance, the Mitre ATT&CK technique T1003.008 was used successfully from this platform. Kali Linux used for performing port scanning and SSH login brute force attacks where the presence of open ports, services on those ports, and login credentials were identified using Nmap and Hydra, respectively.

6.7 Port Scanning

The port scan performed from Kali Linux machine with IP address 192.168.0.20 and the target machine was another Wazuh agent of Ubuntu 18.04 LTS with IP address 192.168.0.21. Before scanning the port, certain configurations in the Virtual Box needed to be made, such as setting the Network Adapter to Bridged, allowing the machines could have two ways communicate with each other. A ping check from Kali to Ubuntu was done before port scanning process. The scan showed that port 22/tcp is open, as shown in Figure 21. The results revealed that the MAC address and confirming that the operating system is Linux.

Figure 21 showed the result of port scanning of the Ubuntu machine performed form the Kali machine within the same network, but the adapter had to be Bridged to communicate with each other.

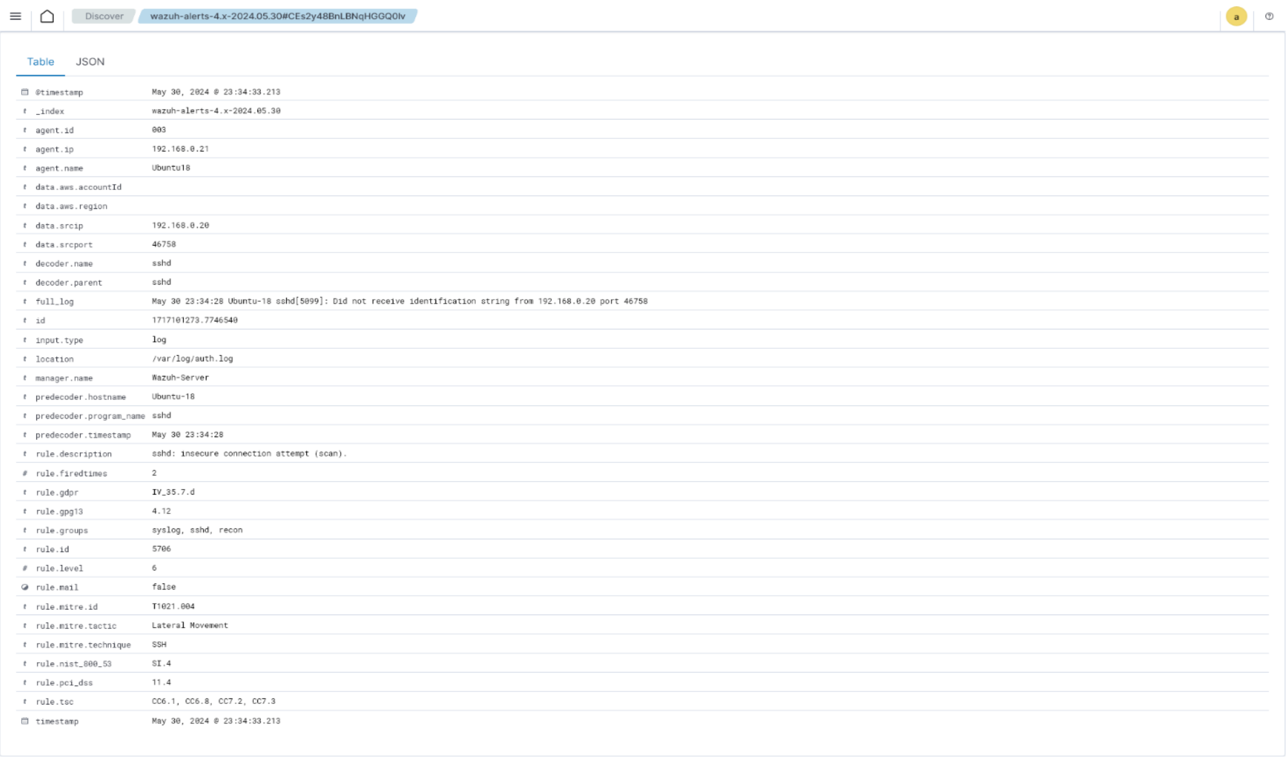
Figure 21 Port Scanning using Nmap.



```
┌──(root㉿kali)-[~]
└─# nmap -sV 192.168.0.21
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-30 23:34 EEST
Nmap scan report for 192.168.0.21
Host is up (0.00086s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
MAC Address: 08:00:27:F0:25:1C (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds

┌──(root㉿kali)-[~]
└─# 
```

Figure 22 shows the Wazuh alert that the scan was detected from the Kali Linux (192.168.0.20) to the agent IP (192.168.0.21). This scan targeted the SSH service on Ubuntu 18, and the alert highlights that an attempt failed to provide the identification string. This scan follows the tactics and techniques of MITRE ATT&CK with an ID of T1021.004 and this ID is associated with SSH lateral movement.

Figure 22 Wazuh response of Port Scan



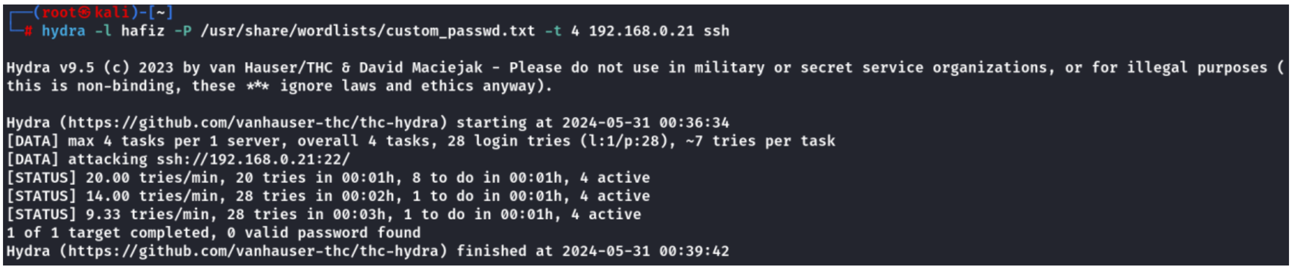
📅	@timestamp	May 30, 2024 @ 23:34:33.213
t	_index	wazuh-alerts-4.x-2024.05.30
t	agent.id	003
t	agent.ip	192.168.0.21
t	agent.name	Ubuntu18
t	data.aws.accountId	
t	data.aws.region	
t	data.srcip	192.168.0.20
t	data.srcport	46758
t	decoder.name	sshd
t	decoder.parent	sshd
t	full_log	May 30 23:34:28 Ubuntu-18 sshd[5099]: Did not receive identification string from 192.168.0.20 port 46758
t	id	1717101273.7746540
t	input.type	log
t	location	/var/log/auth.log
t	manager.name	Wazuh-Server
t	predecoder.hostname	Ubuntu-18
t	predecoder.program_name	sshd
t	predecoder.timestamp	May 30 23:34:28
t	rule.description	sshd: insecure connection attempt (scan).
#	rule.firedtimes	2
t	rule.gdpr	IV_35.7.d
t	rule.gpg13	4.12
t	rule.groups	syslog, sshd, recon
t	rule.id	5706
#	rule.level	6
◑	rule.mail	false
t	rule.mitre.id	T1021.004
t	rule.mitre.tactic	Lateral Movement
t	rule.mitre.technique	SSH
t	rule.nist_800_53	SI.4
t	rule.pci_dss	11.4
t	rule.tsc	CC6.1, CC6.8, CC7.2, CC7.3
📅	timestamp	May 30, 2024 @ 23:34:33.213

6.8 SSH Login

A brute force attack was performed from a Kali Linux machine (192.168.0.20) to an Ubuntu 18.04 LTS machine (192.168.0.21) to test the active response of Wazuh. This attack targeted the SSH login of the Ubuntu machine. A popular tool, Hydra was used to perform the attack, as shown in Figure 23.

In Figure 23, a popular tool, Hydra was used to attempt the unauthorized access to the target machine. During the execution of command mentioned in Figure 23, several password combinations were tested. Hydra was configured to perform four parallel tasks to maximize the SSH login attempts.

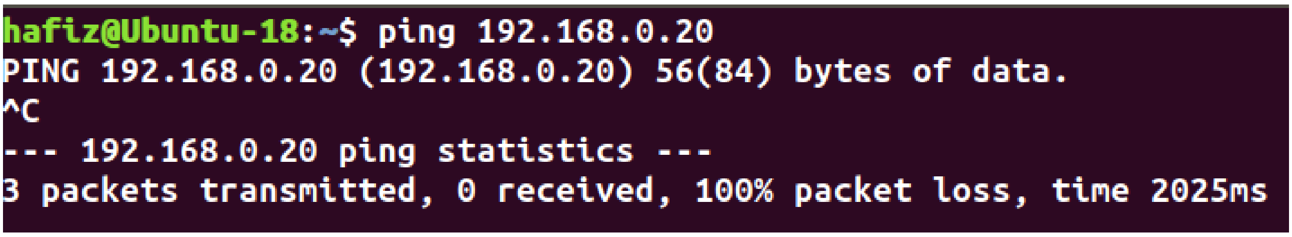
Figure 23 SSH login attempt.



```
┌──(root㉿kali)-[~]
└─# hydra -l hafiz -P /usr/share/wordlists/custom_passwd.txt -t 4 192.168.0.21 ssh

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (
this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-31 00:36:34
[DATA] max 4 tasks per 1 server, overall 4 tasks, 28 login tries (l:1/p:28), ~7 tries per task
[DATA] attacking ssh://192.168.0.21:22/
[STATUS] 20.00 tries/min, 20 tries in 00:01h, 8 to do in 00:01h, 4 active
[STATUS] 14.00 tries/min, 28 tries in 00:02h, 1 to do in 00:01h, 4 active
[STATUS] 9.33 tries/min, 28 tries in 00:03h, 1 to do in 00:01h, 4 active
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-31 00:39:42
```

Figure 24 shows the blocking of Kali’s IP. While the attack was ongoing and the Kali Linux machine was generating the suspicious activity, Wazuh, set up with the active response rule, detected the attack. This provided an additional layer of security because the Wazuh server was programmed to block the Kali machine’s IP address as soon as the attack was detected. This action helped in stopping any further login attempts, mitigated the brute attack efficiently.

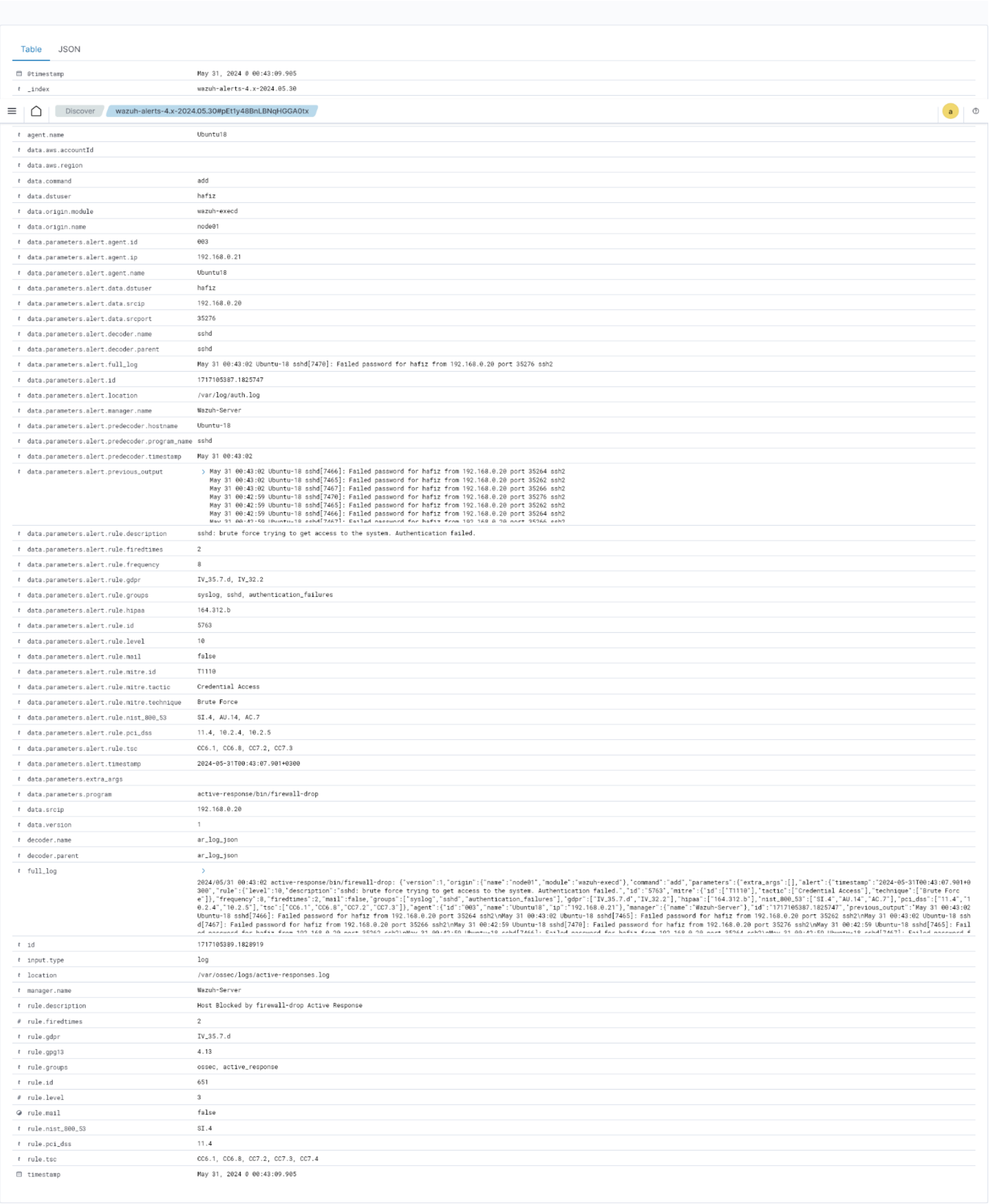
Figure 24 IP blocked of Kali Linux



```
hafiz@Ubuntu-18:~$ ping 192.168.0.20
PING 192.168.0.20 (192.168.0.20) 56(84) bytes of data.
^C
--- 192.168.0.20 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2025ms
```

Figure 25 shows that Wazuh responded actively to block the IP using the firewall-drop active response, as described in the rule. This incident can be seen in the security events as well as in the MITRE ATT&CK module because the technique and tactics used for this attack are from MITRE ATT&CK framework with the ID T1110.

Figure 25 Wazuh response of brute force.



@timestamp	May 31, 2024 @ 00:43:09.905
_index	wazuh-alerts-4.x-2024.05.30
agent.name	Ubuntu18
data.aws.accountId	
data.aws.region	
data.command	add
data.dstuser	hafiz
data.origin.module	wazuh-execd
data.origin.name	node01
data.parameters.alert.agent.id	003
data.parameters.alert.agent.ip	192.168.0.21
data.parameters.alert.agent.name	Ubuntu18
data.parameters.alert.data.dstuser	hafiz
data.parameters.alert.data.srcip	192.168.0.20
data.parameters.alert.data.srcport	35276
data.parameters.alert.decoder.name	sshd
data.parameters.alert.decoder.parent	sshd
data.parameters.alert.full_log	May 31 00:43:02 Ubuntu-18 sshd[7470]: Failed password for hafiz from 192.168.0.20 port 35276 ssh2
data.parameters.alert.id	1717105387.1825747
data.parameters.alert.location	/var/log/auth.log
data.parameters.alert.manager.name	Wazuh-Server
data.parameters.alert.predecoder.hostname	Ubuntu-18
data.parameters.alert.predecoder.program_name	sshd
data.parameters.alert.predecoder.timestamp	May 31 00:43:02
data.parameters.alert.previous_output	› May 31 00:43:02 Ubuntu-18 sshd[7466]: Failed password for hafiz from 192.168.0.20 port 35264 ssh2 May 31 00:43:02 Ubuntu-18 sshd[7465]: Failed password for hafiz from 192.168.0.20 port 35262 ssh2 May 31 00:43:02 Ubuntu-18 sshd[7467]: Failed password for hafiz from 192.168.0.20 port 35266 ssh2 May 31 00:42:59 Ubuntu-18 sshd[7470]: Failed password for hafiz from 192.168.0.20 port 35276 ssh2 May 31 00:42:59 Ubuntu-18 sshd[7465]: Failed password for hafiz from 192.168.0.20 port 35262 ssh2 May 31 00:42:59 Ubuntu-18 sshd[7466]: Failed password for hafiz from 192.168.0.20 port 35264 ssh2 May 31 00:42:59 Ubuntu-18 sshd[7467]: Failed password for hafiz from 192.168.0.20 port 35266 ssh2
data.parameters.alert.rule.description	sshd: brute force trying to get access to the system. Authentication failed.
data.parameters.alert.rule.firedtimes	2
data.parameters.alert.rule.frequency	8
data.parameters.alert.rule.gdpr	IV_35.7.d, IV_32.2
data.parameters.alert.rule.groups	syslog, sshd, authentication_failures
data.parameters.alert.rule.hipaa	164.312.b
data.parameters.alert.rule.id	5763
data.parameters.alert.rule.level	10
data.parameters.alert.rule.mail	false
data.parameters.alert.rule.mitre.id	T1110
data.parameters.alert.rule.mitre.tactic	Credential Access
data.parameters.alert.rule.mitre.technique	Brute Force
data.parameters.alert.rule.nist_800_53	SI.4, AU.14, AC.7
data.parameters.alert.rule.pci_dss	11.4, 10.2.4, 10.2.5
data.parameters.alert.rule.tsc	CC6.1, CC6.8, CC7.2, CC7.3
data.parameters.alert.timestamp	2024-05-31T00:43:07.901+0300
data.parameters.extra_args	
data.parameters.program	active-response/bin/firewall-drop
data.srcip	192.168.0.20
data.version	1
decoder.name	ar_log_json
decoder.parent	ar_log_json
full_log	› 2024/05/31 00:43:02 active-response/bin/firewall-drop: {"version":1,"origin":{"name":"node01","module":"wazuh-execd"},"command":"add","parameters":{"extra_args":[],"alert":{"timestamp":"2024-05-31T00:43:07.901+0300","rule":{"level":10,"description":"sshd: brute force trying to get access to the system. Authentication failed.","id":"5763","mitre":{"id":["T1110"],"tactic":["Credential Access"],"technique":["Brute Force"]},"frequency":8,"firedtimes":2,"mail":false,"groups":["syslog","sshd","authentication_failures"],"gdpr":["IV_35.7.d","IV_32.2"],"hipaa":["164.312.b"],"nist_800_53":["SI.4","AU.14","AC.7"],"pci_dss":["11.4","10.2.4","10.2.5"],"tsc":["CC6.1","CC6.8","CC7.2","CC7.3"]},"agent":{"id":"003","name":"Ubuntu18","ip":"192.168.0.21"},"manager":{"name":"Wazuh-Server"},"id":"1717105387.1825747","previous_output":"May 31 00:43:02 Ubuntu-18 sshd[7466]: Failed password for hafiz from 192.168.0.20 port 35264 ssh2\nMay 31 00:43:02 Ubuntu-18 sshd[7465]: Failed password for hafiz from 192.168.0.20 port 35262 ssh2\nMay 31 00:43:02 Ubuntu-18 sshd[7467]: Failed password for hafiz from 192.168.0.20 port 35266 ssh2\nMay 31 00:42:59 Ubuntu-18 sshd[7470]: Failed password for hafiz from 192.168.0.20 port 35276 ssh2\nMay 31 00:42:59 Ubuntu-18 sshd[7465]: Failed password for hafiz from 192.168.0.20 port 35262 ssh2\nMay 31 00:42:59 Ubuntu-18 sshd[7466]: Failed password for hafiz from 192.168.0.20 port 35264 ssh2\nMay 31 00:42:59 Ubuntu-18 sshd[7467]: Failed password f
id	1717105389.1828919
input.type	log
location	/var/ossec/logs/active-responses.log
manager.name	Wazuh-Server
rule.description	Host Blocked by firewall-drop Active Response
rule.firedtimes	2
rule.gdpr	IV_35.7.d
rule.gpg13	4.13
rule.groups	ossec, active_response
rule.id	651
rule.level	3
rule.mail	false
rule.nist_800_53	SI.4
rule.pci_dss	11.4
rule.tsc	CC6.1, CC6.8, CC7.2, CC7.3, CC7.4
timestamp	May 31, 2024 @ 00:43:09.905

The Wazuh dashboard reported this active response event, indicating that the host IP address of the Kali Linux machine was blocked using firewall-drop active response rule. This log entry indicates that indeed there was a brute force attack on the target Ubuntu machine, which was neutralized by the system, thus improving the security protection.

7 Results and Analysis

The current study focused on the information security appliance Wazuh, which combines SIEM and XDR capabilities within a controlled virtual on-premises lab environment. Wazuh demonstrated the abilities in identifying and mitigating the vulnerabilities in the system, track changes in real-time in the allocated directories and responding to threats during the practical implementations and testing.

Wazuh proved the efficiency in identifying the system vulnerabilities when there were full and partial scans conducted in different agents. A critical vulnerability in VLC Media Player was detected in the Windows machine, while high vulnerabilities were discovered in the Python program. CVEs were described about the remedies both programs to upgrade the versions to eliminate the risks of vulnerabilities. Wazuh succeeded in eliminating the discovered critical vulnerability, which was confirmed by the fact that subsequent scans revealed the absence from the system. Thus, Wazuh proved effective in addressing the security risks. The File Integrity Monitoring (FIM) worked perfectly by displaying changes to the selected directory while ensuring that other actions were not affected. This provided immediate alerts and showcased the capabilities of Wazuh in this regard.

The study also built a successful connection between Wazuh and VirusTotal by using a personal API, which further increased the detection and response capabilities against threats. The integration of VirusTotal and Wazuh successfully detected threats and neutralized risks after downloading the suspicious file. The practical experience with the MITRE ATT&CK framework and the Atomic Red team showed that Wazuh can efficiently observe and document such detailed threat patterns and effectively reflect the alerts on the Wazuh dashboard. Furthermore, port scanning and SSH brute force attacks were performed using Kali Linux. The results demonstrated that Wazuh can identify and mitigate threats instantly through active response configurations.

The results showed that Wazuh, as a SIEM and XDR system, with each module not merely contributing but truly being an asset protection layer against the digital threats. Vulnerabilities were efficiently mitigated by Wazuh’s incident response during the testing stage, as displayed during the practical phase. The research highlighted the applicability of Wazuh across the different operating systems including Ubuntu, Kali Linux, and Windows, in the process of identifying both authorized changes and potential security breaches.

Real-time evaluations highlighted the continuous threat mitigation of Wazuh because of VirusTotal integration that created a wider threat intelligence database. The simulated attack extensively illustrated active threat detection and provided tracing logs of attack by Wazuh. This in-depth logging then provided an opportunity to investigate the weaknesses of the machines as well as to adapt the security measure based on attack patterns.

The evaluation stated that Wazuh, as a SIEM and XDR solution, outperforms because of having real-time monitoring and speedy reaction to cyber threats. This feature makes the tool vital for modern security specialists who provide not only cloud, but also container security. Vulnerability scanner and the FIM had a great contribution to timely detected and response using Wazuh which effectively resolved identified vulnerabilities to enhance security of digital assets.

The thesis proved that Wazuh is the right tool for large-scale organizations, but there are also areas of improvement. Regular modifying and improvements should be taken to remain informed of the dynamic cybersecurity environment. The information obtained from the study will provide background for more investigation into the inner workings of cybersecurity tools which are used to protect digital products in the organizations.

In conclusion, Wazuh proved a highly suitable tool as a SIEM and XDR technologies in which each section played a vital role to secure the lab environment. Wazuh enabled the several areas for the researchers for improvement by providing actionable understandings into system’s vulnerabilities and response efficiency. The results emphasize the value of the discussed tool in detecting, alerting, and responding to threats, improving the cybersecurity strategy in an environment.

8 Summary

This study evaluated Wazuh to investigate the necessary research questions in a virtualized on-premises lab environment. The first question was about the possible roles that Wazuh could play in identifying and mitigating the vulnerabilities through a practical demonstration. The capabilities of Wazuh were tested by doing partial and complete vulnerability scans on the Wazuh agents, determining vulnerabilities categorized as critical, high, medium, and low. The competence of both File Integrity Monitoring (FIM) and Wazuh in detecting real-time changes in allocated directories, showed a very high effectiveness in the field of vulnerability management.

For the second question on the role of XDR (Extended detection and response) to automate threat detection and response operating workflows and SIEM (Security information and event management), the research showed the ability of Wazuh to be integrated with the operating module. Wazuh proved that the perfect correlation of SIEM and XDR functionalities in detecting, recording, and responding to various intrusion patterns. Wazuh utilized the other external threat information services including VirusTotal and simulated the realistic attacks utilizing Atomic Red Team. In addition, the port scanning and SSH brute force attacks were performed using Kali Linux; the results showed that Wazuh can identify and mitigate threats through active response configurations The results, displayed on dashboard of Wazuh, enhanced the advanced functionality of threat response, clearly visualizing that Wazuh is the essential tool for automating and orchestrating security workflows in on-premises environments.

Thanks to this study, a deep understanding of several fundamental issues of cybersecurity and functionality as a part of the Wazuh operational system was gained. While the requirement for continuous scanning as a part of vulnerability scanning and the immediate detection and addressing of risks through real-life action was proved by Wazuh. The accomplishment of FIM in Wazuh showed a firm way to track changes and preserve the integrity of files. This real time monitoring is highly important for detecting malicious alterations of data and maintaining the data integrity. Wazuh’s independent work with the other threat intelligence tools such as VirusTotal, helps maximize the threat detection and defense response. This versatility between security tools is prerequisite for any comprehensive defense approach. The Atomic Red Team utilized to simulate attacks provided useful information concerning the threat detection procedures of Wazuh and succeeded in revealing the capabilities of the tool to detect, analyze and respond to advanced threats. Ultimately, study results established Wazuh as an extremely secure tool for protecting cloud-based architecture and container environments. The characteristics include the ability to adapt to various environments and continuous improvement, make this tool even more secure in the future.

8.1 Future Opportunities

Looking forward, the role of Wazuh in cybersecurity is set to grow tremendously in the light of the increasing complexity of the IT environments and advanced techniques used by cyber attackers. The migration to the cloud and containers as security model means new challenges and possibilities for Wazuh. The use of cloud services and container technologies become more established, the ability of Wazuh to adjust as well as the strong security features will be crucial in the protection of such infrastructures. Future research and development efforts should be aimed at improving Wazuh in these respects to ensure fully protected in any environment. This may involve increasing scalability, integrating with additional third-party applications, and boosting the intelligence response systems and mechanisms.

Furthermore, the ever-changing threat environment requires that more regular updates and functionalities in Wazuh be introduced to the system to stay on top of possible new vulnerabilities. These findings especially highlight the necessity of versatile, open-source tools like Wazuh for cybersecurity positions in the future. In addition, Wazuh can keep up with the emerging security challenges and become a core element in securing digital assets in a rapidly connected world.

References

Mpekoa, N. (2024). An analysis of cybersecurity architectures. International Conference on Cyber Warfare and Security, 19(1), 200–207. https://doi.org/10.34190/iccws.19.1.2115

Wazuh. (n.d.). Components—Getting started with Wazuh · Wazuh documentation. Retrieved April 14, 2024, from https://documentation.Wazuh.com/current/getting-started/components/index.html

Fazzino, S. (n.d.). What is file integrity monitoring & why you need it | fintalk. Retrieved April 15, 2024, from https://www.jackhenry.com/fintalk/file-integrity-monitoring-what-it-is-and-why-you-need-it-in-your-cybersecurity-arsenal

NIST. (2013). Cybersecurity framework. NIST. https://www.nist.gov/cyberframework

NIST. (2020). Security and privacy controls for information systems and organizations (NIST Special Publication (SP) 800-53 Rev. 5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-53r5

NIST. (2016, September 22). Csrc presentation: Mitigating the insider threat - building a secure workforce | csrc. CSRC | NIST. https://csrc.nist.gov/presentations/2012/mitigating-the-insider-threat-building-a-secure

Holt, T., Sarkar, A., & Kumaraguru, P. (n.d.). Security challenges in scientific workflows: A survey. Computers & Security. 2020.

Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). Computer security incident handling guide (NIST Special Publication (SP) 800-61 Rev. 2). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-61r2

2022 gartner® critical capabilities for security information and event management. (n.d.). Splunk. Retrieved April 20, 2024, from https://www.splunk.com/en_us/form/gartner-critical-capabilities-siem.html

Brandao, R., & Nunes, A. (2021). Security Information and Event Management: A Comprehensive Review. International Journal of Information Security Science.

CIS (n.d.). Election security spotlight – cia triad. Retrieved May 29, 2024, from https://www.cisecurity.org/spotlight/ei-isac-cybersecurity-spotlight-cia-triad/

IBM. (n.d.). What is siem? | ibm. Retrieved April 20, 2024, from https://www.ibm.com/topics/siem

Fortinet. (n.d.). What is xdr? Extended detection and response security. Fortinet. Retrieved April 20, 2024, from https://www.fortinet.com/resources/cyberglossary/what-is-XDR

George, A. S., Sagayarajan, S., Baskar, T., & Hovan George, A. S. (2023). Extending Detection and Response: How MXDR Evolves Cybersecurity. Partners Universal International Innovation Journal. https://doi.org/DOI: 10.5281/zenodo.8284342.

Wazuh. (n.d.). Siem. Wazuh. Retrieved April 15, 2024, from https://Wazuh.com/platform/siem/

Arora, V. (2021, May 3). Wazuh: Security information and event management (Siem) for small and medium-sized enterprises. Medium. https://varularora.medium.com/Wazuh-security-information-and-event-management-siem-for-small-and-medium-sized-enterprises-b2cf1cc7ce0c

Wazuh. (n.d.). Proof of Concept guide · Wazuh documentation. Retrieved April 15, 2024, from https://documentation.Wazuh.com/current/proof-of-concept-guide/index.html

Sadiq, E. (2024, January 30). Enhancing data security with the Wazuh open source FIM. Wazuh. https://Wazuh.com/blog/enhancing-data-security-with-the-Wazuh-open-source-fim/

Noman, A. A. (2023, November 30). Empowering threat visibility with Wazuh and Maltiverse. Wazuh. https://Wazuh.com/blog/maltiverse-empowering-threat-visibility/

Wazuh. (n.d.). VirusTotal integration—Malware detection · Wazuh documentation. Retrieved April 16, 2024, from https://documentation.Wazuh.com/current/user-manual/capabilities/malware-detection/virus-total-integration.html

Casares, M. (2019, February 21). How can Wazuh help secure your environment? Wazuh. https://Wazuh.com/blog/how-can-Wazuh-help-secure-your-environment/

Nanty, S. (2023, November 9). Detecting and removing malware using virustotal integration on windows endpoints with Wazuh. Medium. https://medium.com/@nantysean/detecting-and-removing-malware-using-virustotal-integration-on-windows-endpoints-with-Wazuh-8e994bac1fd0

Biri, R. (2024, January 18). Wazuh—File integrity monitoring. Medium. https://medium.com/@RitajBiri/wazuh-file-integrity-monitoring-ec9d30764c30

Baykara, S. (2020, April 18). Pci dss and file integrity monitoring. PCI DSS GUIDE. https://pcidssguide.com/the-pci-dss-and-file-integrity-monitoring/

Wazuh. (n.d.). File integrity monitoring—Capabilities · Wazuh documentation. Retrieved May 16, 2024, from https://documentation.wazuh.com/current/user-manual/capabilities/file-integrity/index.html

Wazuh. (n.d.). Open-source incident response. Wazuh. Retrieved May 16, 2024, from https://wazuh.com/use-cases/incident-response/

Thakkar, U. (2023, May 15). Understanding wazuh architecture. Medium. https://urvesh-thakkar.medium.com/understanding-wazuh-architecture-d8928b7a392e

Wazuh. (n.d.). Integration—Local configuration (Ossec. Conf) · Wazuh documentation. Retrieved April 16, 2024, from https://documentation.Wazuh.com/current/user-manual/reference/ossec-conf/integration.html

Wazuh. (n.d.). Integration with third-party APIs—Wazuh server · Wazuh documentation. Retrieved April 16, 2024, from https://documentation.Wazuh.com/current/user-manual/manager/manual-integration.html

Installing invoke atomicredteam. (n.d.). GitHub. Retrieved April 20, 2024, from https://github.com/redcanaryco/invoke-atomicredteam/wiki/Installing-Invoke-AtomicRedTeam

Wazuh. (n.d.). Role of Wazuh in building a robust cybersecurity architecture. BleepingComputer. Retrieved April 26, 2024, from https://www.bleepingcomputer.com/news/security/role-of-Wazuh-in-building-a-robust-cybersecurity-architecture/

AWS. (n.d.). What is an api key? - Api keys and tokens explained - aws. Amazon Web Services, Inc. Retrieved May 16, 2024, from https://aws.amazon.com/what-is/api-key/

IBM MQ 9.2. (2024, January 31). https://www.ibm.com/docs/en/ibm-mq/9.2?topic=network-real-time-monitoring

Jason, A. (2014). The basics of information security—Understanding the Fundamentals of InfoSec in Theory and Practice (Second Edition). Syngress.

Briasmitatms. (2024, January 31). Center for internet security (Cis) benchmarks—Microsoft compliance. https://learn.microsoft.com/en-us/compliance/regulatory/offering-cis-benchmark

Balbix. (n.d.). What is a CVE? Retrieved May 29, 2024, from https://www.balbix.com/insights/what-is-a-cve/

CVE. (n.d.). Cve website. Retrieved May 29, 2024, from https://www.cve.org/About/Overview

Tetrate. (n.d.). What are the benefits of CVEs? Tetrate. Retrieved May 29, 2024, from https://tetrate.io/faq/what-are-the-benefits-of-cves/

Appendix 1: Material management plan

This thesis is based on practical and research-oriented techniques. In the thesis study, a diary is kept all the time in the local machine as well as OneDrive as a backed-up file. There has been regular back-up in the local machine to avoid any problem and used same file to write the whole thesis. As far as the technical information about the practical implementation concerns, there was not any company and third person involved to collect, analyse, or any survey results. Wazuh, an open-source tool was the core component during this study. In this analysis, all the testing and analysis results performed during this study are available in the Wazuh dashboard. In the dashboard, there were logs and events generated after the different tests performed. These logs and events data will be saved approximately one year after the certain tests performed. Hence, the results of these tests will be saved one year as the testing section was done in April 2024.