"CyberGuard: Comprehensive Threat Detection & Monitoring System"

A Report submitted

In partial fulfilment for the Degree of

MASTER OF SCIENCE
IN
DIGITAL FORENSICS & INFORMATION SECURITY

Submitted By

PRAJAPATI HIMIL HARSHADKUMAR
(012200300003011)

Under the Supervision of

DR. LAVI TYAGI
(Associate Professor)

Submitted to

SCHOOL OF CYBER SECURITY & DIGITAL FORENSICS,
NATIONAL FORENSIC SCIENCES UNIVERSITY
GANDHINAGAR – 382009, GUJARAT, INDIA.
JULY, 2024

DECLARATION

I **"Prajapati Himil Harshadkumar"** having Enrollment Number **"012200300003011"** hereby declare

a. The work contained in the dissertation report entitled **"A Comprehensive to Detecting and Monitoring of Threats"** is being submitted in partial fulfilment for the award of the degree of **"M.Sc. Digital Forensics and Information Security"** to **School of Cyber Security** is an authentic record of my own work done under the supervision of **"Dr. Dinesh Singh "**.

b. The work has not been submitted to any other Institute/ School / University for any degree or diploma.

c. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the School.

d. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the dissertation and giving their details in the references.

e. Whenever I have quoted written materials from other sources and due credit is given to the sources by citing them.

f. From the plagiarism test, it is found that the similarity index of whole dissertation within 25% and single paper is less than 10 % as per the university guidelines.

Date:
Place: Institute of Forensic Science
National Forensic Sciences University
Gandhinagar

Signature of Student

CERTIFICATE

This is to certify that the work contained in the dissertation entitled **"A Comprehensive Approach to Detecting and Monitoring of Threats"**, submitted by **Prajapati Himil Harshadkumar (Enroll. No.: 012200300003011)** in partial fulfilment of the requirement for the award of the degree of **M.Sc. Digital Forensics and Information Security** to the **National Forensic Sciences University**, **Gandhinagar, Gujarat** is a record of bonafide work carried out by him/her under my direct supervision and guidance.

Date:
Place: Gandhinagar

Signature & Date
Supervisor(s)

ACKNOWLEDGEMENTS

It is a great pleasure to present this report on the project "**A Comprehensive Approach to Detecting and Monitoring of Threats**" undertaken by me as part of my Academic Project.

I am thankful to Dr. Lavi Tyagi for his continuous guidance and valuable suggestions provided by him at every stage of my project. Thank you for providing the necessary assistance and for their constant encouragement.

I would also thank the Institution, and all the classmates of the Department of Digital Forensics, Institute of Forensic Science, National Forensic Sciences University, Gandhinagar for their special attention and suggestions towards the project work.

With Sincere Regards,
Prajapati Himil Harshadkumar
MSc. Digital Forensics
and Information Security

ABSTRACT

In today's interconnected digital landscape, the need for robust threat detection and monitoring mechanisms has never been greater. Effective threat detection and monitoring are critical components of any robust cybersecurity strategy, enabling organizations to identify, assess, and mitigate potential risks proactively.

This project aims to introduce a comprehensive approach to identifying and overseeing potential threats in various digital environments. Leveraging open-source technology, we propose a multifaceted strategy that integrates advanced detection algorithms, real-time monitoring systems, and proactive threat intelligence analysis.

Small and Medium-sized organizations often face unique challenges, such as limited IT budgets and resources, which can make it difficult to implement and maintain effective cybersecurity measures. My Project addresses these challenges by providing scalable and affordable solution that do not compromise on security.

In addition, our plan reduces the requirement for deep cybersecurity knowledge by utilizing automated procedures and user-friendly interfaces. This allows companies with smaller IT teams to handle threats and respond to them efficiently. By integrating proactive threat intelligence, such companies can minimize their exposure and increase their overall resilience in the digital age by getting ahead of emerging threats.

Keywords: Threat Detection, Open-Source Technology, Real-Time Monitoring, Potential Risk, Scalable and affordable solutions, Detection Algorithm, Small and Medium Size Organization.

LIST OF ABBREVIATIONS

Abbreviation	Description
SIEM	Security Information and Event Management
CIA	Confidentiality, Integrity & Availability
SOC	Security Operation Center
SOAR	Security Orchestration Automation and Response
VT	Virus Total
XDR	Extended Detection and Response
FIM	File Integrity Monitoring
USB	Universal Serial Bus

LIST OF FIGURES

TABLE OF CONTENTS

1. Introduction

1.1 Purpose

In the digital world of today, cybersecurity is crucial. My project's goal is to provide a complete threat detection and monitoring system that will assist small and medium-sized businesses in defending themselves against cyber threats. Open-source technology is being used in my project to detect, evaluate, and reduce possible dangers in a variety of digital settings. This indicates that my project can lessen exposure and assist organisations in staying ahead of developing hazards.

My project provides a simple, cost-effective solution for small and medium-sized businesses. Because it does away with the requirement for in-depth cybersecurity knowledge, IT teams of various sizes can use it. My project enables companies to react swiftly and efficiently to new threats by utilising automated processes and user-friendly interfaces. The overall goal of my research is to guarantee the security and resilience of vital systems and data by offering a strong framework for spotting and countering any threats.

1.2 Objective

- ✓ Enhance IT Security using Open-Source Technologies, prioritizing Linux and Windows Environments.
- ✓ Strengthen malware detection capabilities to identify and mitigate potential threats, safeguarding systems and data integrity.
- ✓ Conduct comprehensive vulnerability scanning to identify security weaknesses and prioritize remediation efforts.
- ✓ Install File Integrity Monitoring (FIM) systems to keep an eye out for illegal modifications to important system files and protect the integrity of the system.

1.3 Literature review

Comparative analysis of IBM Qradar and Wazuh for security information and event management.
(Dario Suskalo, Zlatan Moric, Jasmin Redzepagic and Damir Regvart [December 2023])

- ➢ The aim was to evaluate their performance in addressing security challenges, especially after updates or in community-driven environments. Qradar excels in comprehensive security features and scalability, suitable for large enterprises, while Wazuh provides a cost-effective open-source option, recommended for smaller companies with limited budgets.

Intrusion Detection using Open-Source Tools. - (Jack TIMOFTE [January – 2008])

- In this paper, they have discussed the benefits and challenges of open-source intrusion detection systems (IDS) like OS-SEC, Prelude, and Snort, and explore their relationship with commercial support. They examines how organizations, both large and small, can effectively leverage open-source and commercial IDS solutions to enhance their network security posture.

A Review of Wazuh Capabilities for Detecting Attacks Based on Log Analysis- (Stefan Stanković, Slavko Gajin, and Ranko Petrović [June -2022])

- The aim of the paper is to demonstrate Wazuh's effectiveness in detecting various attack. The paper suggests integrating Wazuh with Suricata, a Network Intrusion Detection System capable of generating JSON logs, for enhanced security insights.

Analysis of attacks and prevention methods in cybersecurity - (Prof. Francesco Gringoli [October, 2022])

- The paper aims to evaluate Wazuh's effectiveness as an open-source tool for IT infrastructure protection, with a focus on threat detection, vulnerability management, and prevention.

Information And Security Event Management System - (Voulgaris Ioanni, Prof. Konstantinos Labrinoudakis)

- This Paper focuses on proposing, analyzing, and evaluating cybersecurity solutions based on the Elastic Stack (ELK), which is widely used as an enterprise-grade logging suite and search engine. It discusses the importance of efficient log analysis, the impact of GDPR compliance within the ELK environment, and the role of Elasticsearch as a search engine.

A Survey on Network Security Monitoring: Tools and Functionalities - (Z. S. Younus, M. Alanezi)

- This paper explores the growing prevalence of cybersecurity breaches and their impact on network infrastructure. It highlights various protective measures such as antivirus software, firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS) are employed to secure network devices.

Security Infrastructures and intrusion Systems - (Matthias Mildenberger, Artur Wachtel)

- In this Paper how we keep computer systems safe from hackers, especially in big and powerful setups like high-performance computing. It says that while tools like intrusion detection systems (IDS) are important for catching hackers, they're just one part of a bigger plan for keeping systems safe.

SIEMA: Bringing Advanced Analytics to Legacy Security Information and Event Management - (Pejman Najafi, Feng Cheng, Christoph Meinel)

- This paper addresses the limitations of current SIEM systems in performing advanced analytics and utilizing modern data mining, machine learning, and graph mining approaches. To overcome these limitations, the concept of Security Information/Event Management and Analytics (SIEMA) is introduced, emphasizing the need for next-generation SIEM systems with enhanced analytical capabilities.

Open-Source Security Tools for Small Businesses: A Review - (M. A. Khan, S. M. Shams, and M. R. Islam (2020))

- This review article discusses the role of open-source security tools in defending against cyber-attacks, specifically for small businesses. The authors highlight the benefits of open-source tools, including their cost-effectiveness, flexibility, and community-driven development. They review various open-source security tools, including firewalls, intrusion detection systems, and vulnerability scanners, and discuss their limitations and challenges.

The Role of Artificial Intelligence in Open-Source Cybersecurity - (S. K. Goyal, S. K. Singh, and A. K. Singh (2019))

- This review article explores the intersection of artificial intelligence (AI) and open-source cybersecurity, highlighting the potential benefits and challenges of combining these technologies. The authors discuss how AI can enhance open-source security tools by improving threat detection, incident response, and vulnerability identification.

2. Problem statement

Small and medium-sized organizations are increasingly vulnerable to cyber threats, despite the importance of cybersecurity in today's digital world. These organizations often lack the resources, expertise, and budget to implement and maintain robust cybersecurity measures, making them easy targets for attackers. As a result, they face significant risks, including:

- **Data breaches:**

Sensitive data is stolen or compromised, leading to financial losses, reputational damage, and regulatory non-compliance.

- **System downtime:**

Cyber-attacks can cause system failures, disrupting business operations and resulting in lost productivity and revenue.

- **Increased costs:**

The cost of responding to and recovering from a cyber attack can be substantial, diverting resources away from core business activities.

- **Lack of visibility**

Small and medium-sized organizations often lack real-time visibility into their digital environments, making it difficult to detect and respond to emerging threats.

3. How to Solve this Issue

Our research tackles the cybersecurity issues that Small and Medium-Sized Businesses encounter in the connected digital environment of today. We offer a complete and scalable solution that combines proactive threat intelligence analysis, real-time monitoring systems, and sophisticated detection algorithms by utilising open-source technology. Organisations are able to proactively detect, evaluate, and reduce possible risks thanks to this holistic approach. Effective cybersecurity is no longer limited by restricted IT budgets and resources due to our solution, which is especially made to be reasonably priced. Furthermore, the automated processes and user-friendly interfaces significantly reduce the need for in-depth cybersecurity expertise. This makes it possible for businesses with smaller IT teams to effectively manage and react to risks.

4. What is SOC

An organization's centralised entity in charge of tracking, identifying, evaluating, and handling cybersecurity issues is called a Security Operations Centre (SOC). It acts as the focal point for combining people, procedures, and technology to manage and enhance an organization's security posture. Real-time detection of possible security threats and vulnerabilities, along with prompt and efficient incident response, are the main objectives of a SOC.

SOC technology refers to a broad class of platforms and devices designed to provide comprehensive security protection. By combining and analysing security data from several IT infrastructure components, Security Information and Event Management (SIEM) systems are crucial elements that offer real-time insights and alarms. Network traffic monitoring and anomaly detection depend on intrusion prevention systems (IPS) and intrusion detection systems (IDS). IPS can also be used to stop possible threats before they happen. Workstations and servers are examples of endpoints, and tools for endpoint detection and response, or EDR, are made to monitor and defend against potential threats

5. SIEM Technology

5.1 What is SIEM

Systems for Security Information and Event Management, or SIEMs, are made to track and examine security-related data from a variety of sources, giving an all-encompassing picture of the security posture of an enterprise. A SIEM system's architecture normally includes of the following parts.

5.2 Component of SIEM

- **Event Collection:** SIEM systems collect log data from various sources, including:
  - Network devices (e.g., firewalls, routers, switches)
  - Servers, workstations, and applications
  - Intrusion detection and prevention systems (IDPS)
  - Network-based intrusion detection systems (NIDS)
  - Endpoint security systems (e.g., antivirus software)
- **Log Processing**: SIEM systems process log data by:
  - **Log aggregation:** aggregating log data from multiple sources
  - **Log parsing**: extracting relevant information from log data
  - **Log correlation**: identifying patterns and relationships between events
- **Analytics and Alerting**: SIEM systems use analytics and machine learning algorithms to:
  - Identify anomalies in log data that may indicate security threats
  - Trigger alerts for specific security events or incidents
  - Provide real-time visibility into security threats and incidents
- **Alert Management:** SIEM systems provide a centralized alert management console for incident responders to:
  - Triage and prioritize alerts
  - Investigate and respond to incidents
  - Collaborate with other teams and stakeholders
- **Reporting and Visualization**: SIEM systems provide reports and visualizations to help analysts:
  - Gain insights into security threats and incidents
  - Track compliance metrics
  - Identify trends and patterns in security-related data
- **Incident Response:** SIEM systems provide incident response capabilities, including:
  - Pre-defined incident response playbooks
  - Automation of repetitive tasks and workflows
  - Integration with other security tools and systems

5.3 SIEM ARCHITECTURE

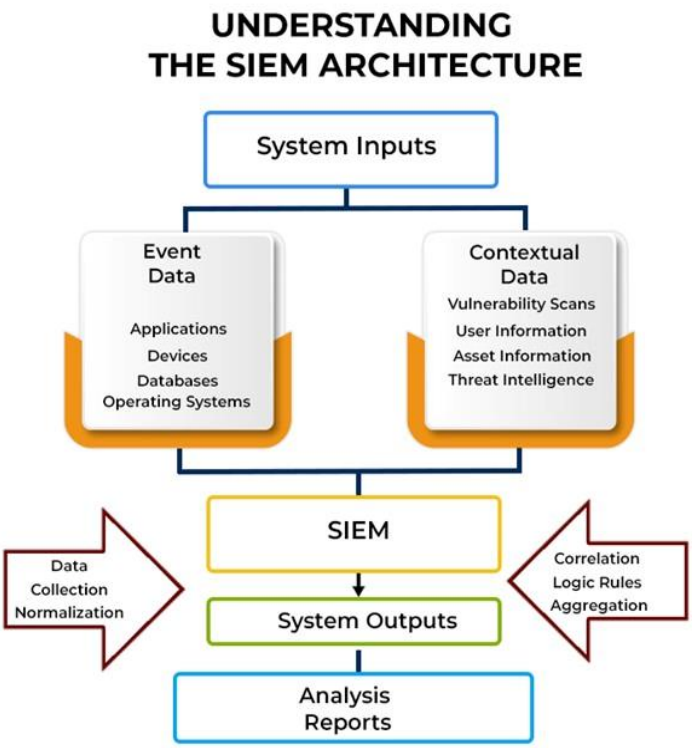


Figure 1 SIEM Architecture

- The logs are collected from various endpoints to central logging mechanism where they are stored in a common format.
- The correlation engine is used to define the rules and filters for correlating the event from various log sources.
- Management portal provides a GUI Interface to the user to manager the SIEM operations.

6. What is EDR?

Endpoint Detection and Response (EDR) is a cybersecurity technology that monitors and analyzes endpoint devices, such as laptops, desktops, and servers, to detect and respond to advanced threats in real-time. EDR solutions provide visibility into endpoint activity, allowing organizations to detect and respond to threats that might evade traditional security controls.

6.1 How EDR Works?

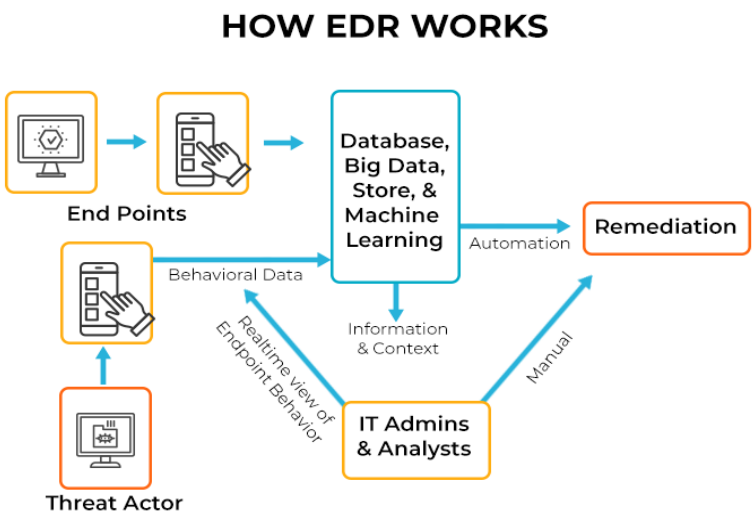


Figure 2 EDR WORKS

EDR systems utilize agents installed on local devices, which continuously monitor and collect data from their respective device environments. This data is sent to a centralized hub for processing and analysis, frequently making use of cutting-edge technology like machine learning and artificial intelligence (ML). The resulting statistical models enable real-time analysis of incoming endpoint data, allowing for timely identification of potential threats. EDR solutions employ several threat detection techniques to identify potential security breaches.

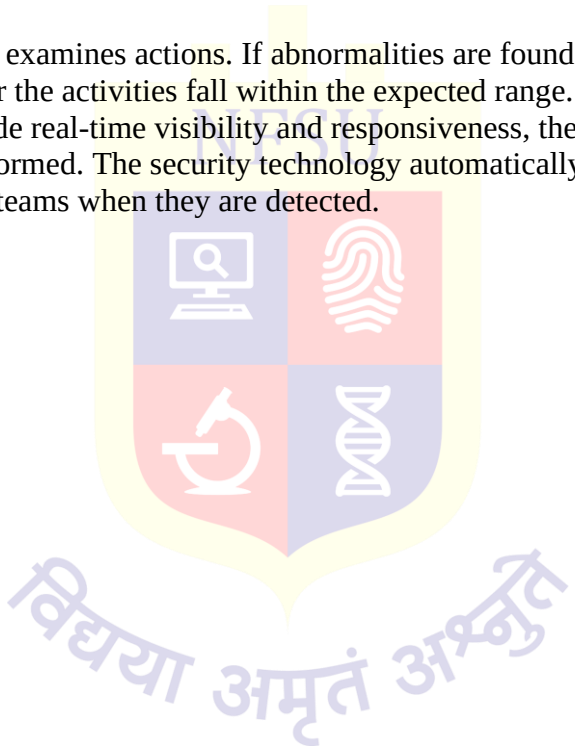
One method is signature analysis, where network traffic is compared to a database of known malware signatures to identify matches. Another technique is behavioral analysis, which monitors the normal behavior of an endpoint and flags unusual activity that may indicate a threat, even if all traffic signatures appear valid.

Additionally, EDR solutions use sandbox analysis, wherein potentially dangerous files are run in a safe setting so that their behaviour can be seen without running the risk of endpoint

damage. This allows for a more comprehensive understanding of the file's behavior and potential impact.

Finally, some EDR solutions employ whitelist/blacklist matching, where endpoint activities are compared to a predetermined list of approved and blocked IP addresses to determine whether network traffic should be allowed or denied.
EDR security's three main responsibilities

1. Data collection:- Communications, process executions, and user logins are all gathered from endpoint devices.
2. Data Recording:- This feature instantly logs all information pertaining to security occurrences.
3. Detection:- EDR examines actions. If abnormalities are found in the network, it will ascertain whether the activities fall within the expected range.
   In order to provide real-time visibility and responsiveness, these three duties are consistently performed. The security technology automatically reacts to threats and notifies security teams when they are detected.



7. What is XDR.

XDR (Extended Detection and Response) is a type of cybersecurity solution that combines traditional endpoint detection and response capabilities with advanced threat hunting and incident response techniques. It's designed to provide a comprehensive and proactive approach to detecting, analyzing, and responding to threats in real-time.

7.1 How XDR works:

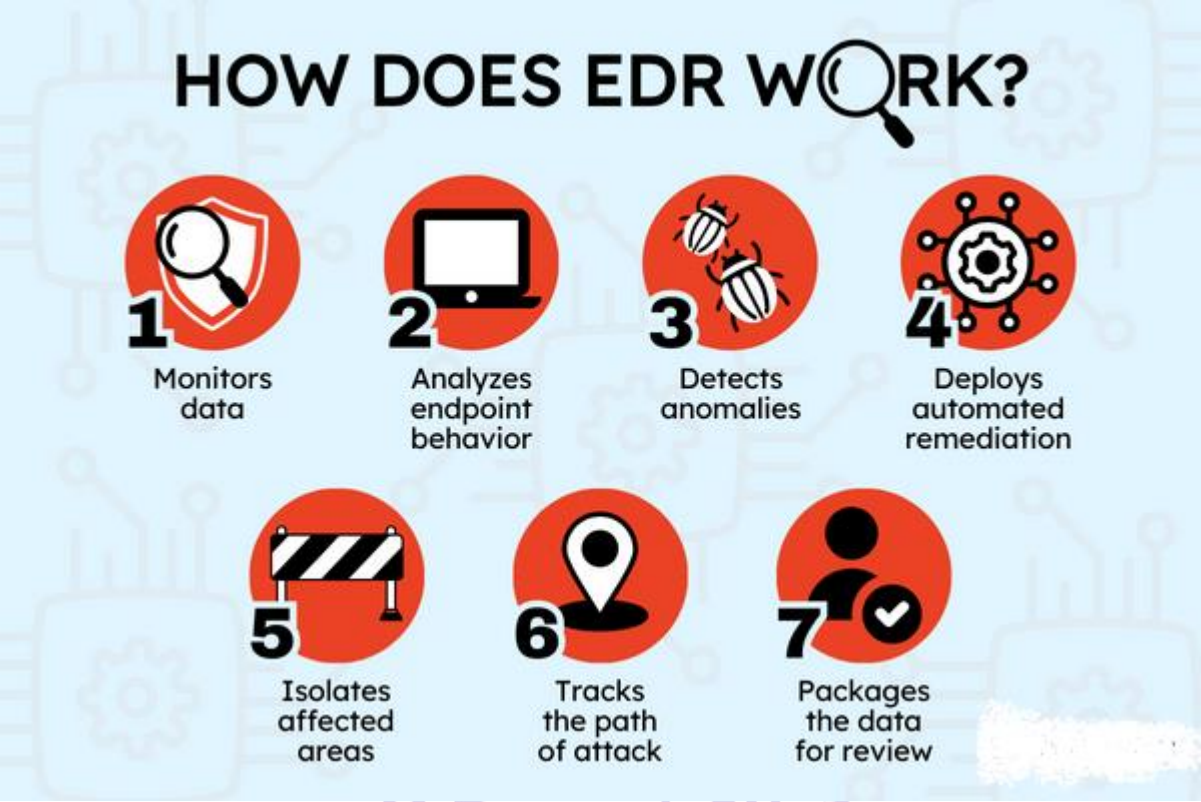


Figure 3 EDR Work

7.2 Components of XDR:

- **Endpoint Detection**: This component monitors and collects data from endpoints, such as servers, laptops, and mobile devices, to detect potential threats.
- **Advanced Threat Hunting**: This component uses machine learning algorithms and threat intelligence to analyze the collected data and identify potential threats.
- **Incident Response**: This component automates the response to detected threats, including containment, eradication, and remediation.

8. What is Wazuh?

Wazuh is an open-source security monitoring and threat detection platform that provides real-time monitoring, threat detection, and incident response capabilities for endpoints, networks, and clouds. It is designed to provide advanced threat detection and incident response capabilities to help organizations detect and respond to security threats in real-time.

8.1 Features of Wazuh

- **Real-time log analysis:** Wazuh collects logs from various sources, including operating systems, applications, and network devices, and analyzes them in real-time to detect potential security threats.
- **Anomaly detection:** Wazuh uses machine learning algorithms to identify abnormal behaviour that may indicate a security threat.
- **Threat intelligence:** Wazuh integrates with external threat intelligence feeds to provide real-time information on known threats and vulnerabilities.
- **Incident response:** Wazuh provides incident response capabilities, including automated response actions and customizable playbooks for responding to detected threats.
- **Compliance reporting:** Wazuh provides reporting capabilities that help organizations meet compliance requirements for security auditing and logging.

8.2 Component of Wazuh

- **Indexer:** Wazuh is an analytics and full-text search engine that is scalable. It retains and indexes alerts produced by the Wazuh server. This component allows for efficient querying and analysis of large amounts of data.
- **Server:** Wazuh is responsible for analysing, putting through decoders and algorithms, and using threat intelligence to find known indicators of compromise (IOCs) in the data received from agents. By adding more servers to the cluster, a single server can be horizontally expanded to process data from hundreds or thousands of agents. The server also manages agents, allowing for remote configuration and upgrades.
- **Dashboard:** Wazuh provides an online platform for analysing and visualising data. Pre-built dashboards are available for a range of use cases, such as regulatory compliance, cloud infrastructure monitoring, file integrity monitoring, threat hunting, vulnerability identification, and configuration review. In addition, Wazuh setup and status are managed via the dashboard.
- **Agents:** Endpoints including laptops, desktop computers, servers, cloud instances, and virtual machines all have the agent installed on them. They can operate on a number of operating systems, including Linux, Windows, macOS, and others, and offer threat prevention, detection, and response capabilities.

8.3 Wazuh Architecture

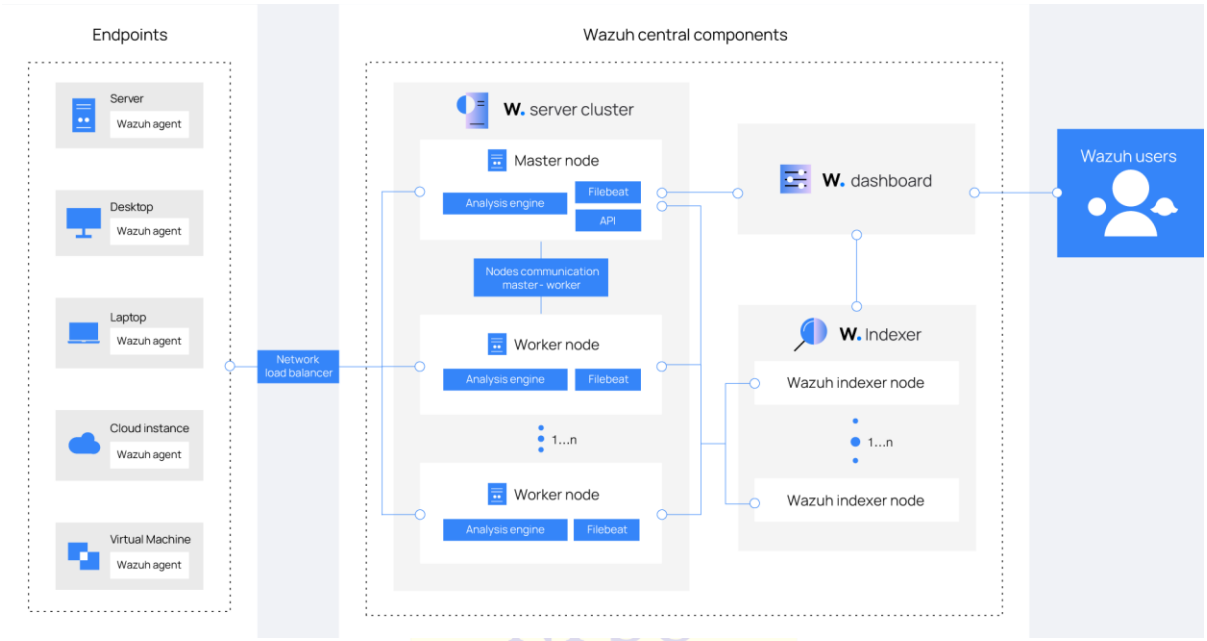


Figure 4 Wazuh Architecture

Wazuh's architecture is based on agents that communicate security information to a central server while operating on endpoints under observation. Agents can also be set up to gather information from agentless devices, such as routers, switches, firewalls, and access points, which can send logs via their API, Syslog, or SSH.
After decoding and analysing the incoming data, the Wazuh indexer cluster receives the results from the central server for storage. This cluster is a collection of nodes that collaborate to read and write indices. A single-node cluster is sufficient for small-scale deployments; multi-node clusters are useful for bigger deployments or those that need high availability.

8.4 Case Management in SOC:

In a Security Operations Center (SOC), Case Management refers to the process of managing and tracking incidents, security breaches, or other security-related events from detection to resolution. It involves coordinating the efforts of security analysts, incident responders, and other stakeholders to contain, eradicate, and remediate threats.

8.4.1 Components of Case Management in SOC:

- Incident Identification: Identifying potential security incidents or breaches through monitoring and alerting systems.
- Initial Response: Initial investigation and analysis to confirm the severity and scope of the incident.
- Containment: Isolating the affected systems or networks to prevent further damage.
- Eradication: Removing the threat or malware from the affected systems.
- Remediation: Restoring the affected systems or networks to a secure state.
- Post-Incident Activities: Reviewing the incident, identifying root causes, and implementing remediation plans to prevent similar incidents in the future.

8.4.2 Benefits of Case Management in SOC

- Improved Incident Response Time: Faster response times reduce the impact of incidents and minimize damage.
- Enhanced Collaboration: Case management fosters collaboration among security teams, improving communication and decision-making.
- Increased Efficiency: Streamlined processes reduce the time spent on incident handling and allow for more efficient use of resources.
- Better Root Cause Analysis: In-depth analysis helps identify vulnerabilities and prevents similar incidents from occurring in the future.
- Compliance: Case management ensures compliance with regulatory requirements and industry standards.

8.5 TheHive: A Case Management Tool for Incident Response

TheHive is an open-source, web-based incident response platform designed to streamline case management, threat hunting, and threat intelligence management. It provides a centralized hub for security teams to manage and track incidents from detection to resolution.

8.5.1 How it Works:

- Incident Detection: Receive alerts from various sources (SIEM, IDS/IPS, threat feeds) or manually create incidents.
- Case Creation: Create a new case for each incident, including classification, prioritization, and assignment.
- Threat Hunting: Investigate incidents using a structured threat hunting framework.
- Threat Intelligence Management: Collect, analyze, and visualize threat intelligence from various sources.
- Reporting: Generate reports on incident response activities, threat intelligence, and alert management.

8.6 Threat Intelligence in SOC:

In a Security Operations Center (SOC), Threat Intelligence (TI) plays a crucial role in enhancing incident response, threat hunting, and vulnerability management. TI is used to gather, analyze, and share information about potential threats to an organization's people, assets, data, and infrastructure.

8.6.1 How TI is used in SOC:

- Incident Response: TI informs incident response efforts by providing context on the threat actor's motivations, tactics, and techniques.
- Threat Hunting: TI enables threat hunting teams to proactively identify and disrupt potential threats.
- Vulnerability Management: TI identifies vulnerabilities in software and systems to prioritize patching efforts.
- Security Orchestration: TI integrates into security orchestration platforms to automate incident response and threat hunting workflows.

- Threat Detection: TI helps detect unknown threats by analyzing indicators of compromise (IOCs) and identifying patterns.

8.7 MISP (Malware Information Sharing Platform):

MISP is an open-source threat intelligence platform that enables information sharing and collaboration among security teams. It provides a centralized hub for collecting, storing, and sharing threat intelligence from various sources.

Key Features:

- Threat Intelligence Management: Collects, analyzes, and shares threat intelligence from various sources.
- Indicator-based Intelligence: Focused on IOCs, such as IP addresses, domains, hashes, and malware samples.
- Collaboration: Enables sharing of threat intelligence among security teams through a web-based interface.
- Data Analysis: Offers data analysis tools for identifying patterns and trends in threat intelligence.

8.8 Cortex:

Cortex is a cloud-based threat intelligence platform that provides real-time threat detection and analysis. It aggregates data from various sources to deliver actionable threat intelligence to security teams.

Key Features:

1. Real-time Threat Detection: Identifies threats in real-time using machine learning algorithms.
2. Threat Intelligence Integration: Integrates with various security tools and platforms.
3. Visualization: Provides visual representations of threat intelligence to facilitate understanding.
4. Alerting: Sends alerts to security teams based on identified threats.

8.9 Virus Total

Virus Total is a free online service that analyzes files and URLs for malware and other types of threats. It aggregates data from various antivirus engines to provide a comprehensive view of file or URL reputation.

9.Environment Setup, Tools & Technology

We will go over the technology and tool configuration, as well as the environment setup, for the SOC platform integration project in this chapter. Wazuh for monitoring and detection, TheHive for case management, Cortex and MISP for threat intelligence, and Ubuntu for endpoint operating system integration are all integrated in this project

9.1 Setup of the Environment:

Wazuh

Wazuh is used for vulnerability management, intrusion detection, log analysis, and security monitoring.

Configuration:

- Install Wazuh Server: To install the Wazuh server on a dedicated computer or virtual machine, adhere to the official Wazuh installation instructions.
- Install Wazuh Agent: To gather logs and security events, install the Wazuh agent on the Ubuntu endpoints.

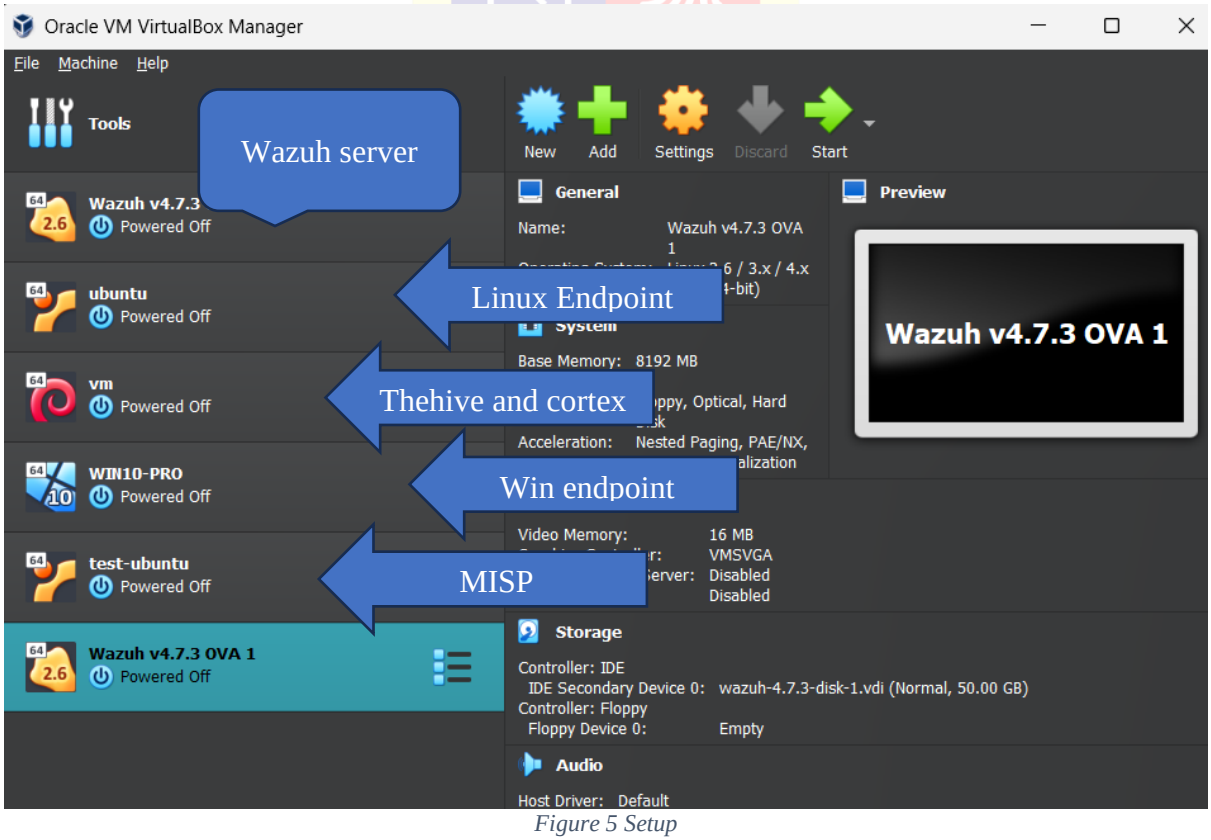


Figure 5 Setup

9.2 Technology & Tools:

9.2.1 Wazuh

An open-source security monitoring platform that manages compliance across endpoints, offers visibility, and detects threats. Log analysis, vulnerability, intrusion, and compliance monitoring are important features.

9.2.2 TheHive

- An open-source platform for security incident response created to assist CSIRTs and SOCs in effectively managing and looking into security occurrences.
- Integration with threat intelligence platforms, case management, teamwork, and incident tracking are some of the key features.

9.2.3 Cortex

- Description: An open-source programme for automating security incident response and observable analysis.
- Important features include several responders and analyzers, automated analysis, and connection with MISP and TheHive.

9.2.4 Platform for Sharing Malware Information (MISP):

An open-source platform for threat intelligence that facilitates the exchange, archiving, and correlation of threat data and indicators of compromise (IOCs).
Sharing of threat intelligence, correlation of threat data, and interaction with security technologies are some of the key features.

9.2.5 Ubuntu:

This project uses a well-known open-source Linux distribution for its endpoints. Stability, security, and broad community support are important characteristics.

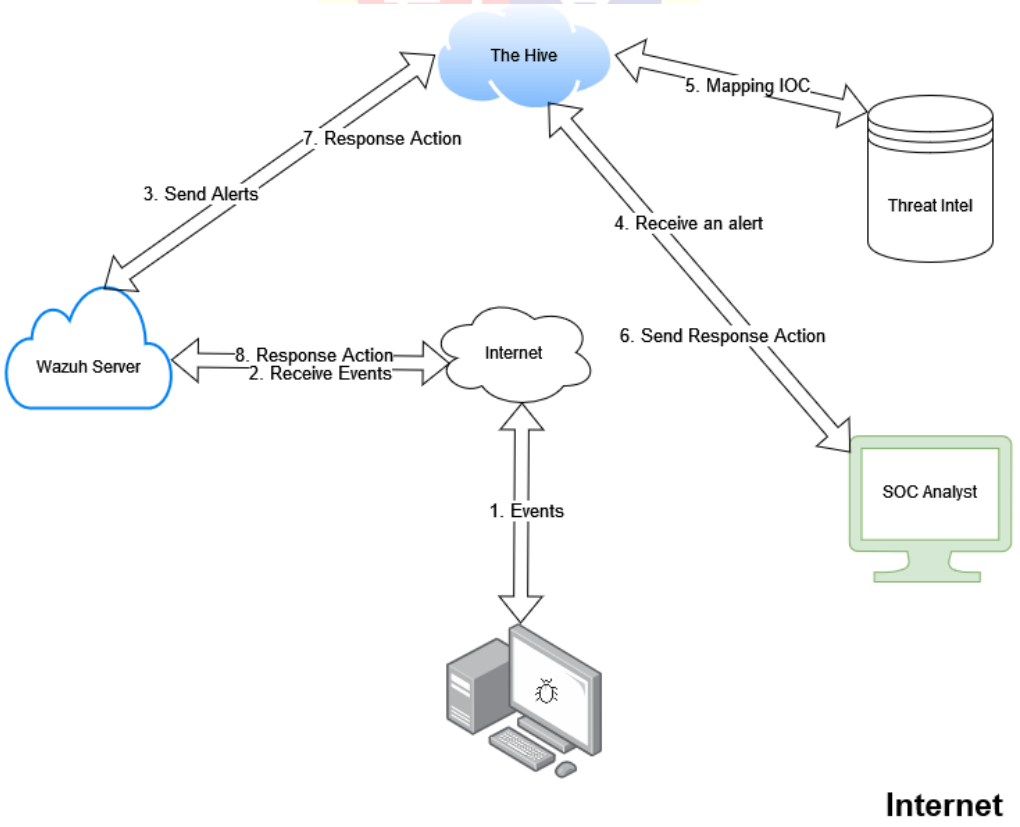


Figure 6 General Setup Flow

10. UseCase for opensource security

In order to safeguard their digital assets, businesses are increasingly using open source security solutions due to the complexity and sophistication of cyber threats. These systems provide a flexible, economical, and cooperative security strategy. This use case demonstrates how a company can improve its cybersecurity posture by utilising open source security solutions to establish a strong Security Operations Centre (SOC).

10.1 Detecting known bad actors with Wazuh and AbuseIPDB

A project called AbuseIPDB assists webmasters, security analysts, and systems administrators in identifying and reporting IP addresses linked to different types of harmful assaults. It offers an API to look for malicious behaviour on an IP address and report it.

Wazuh offers the integrator tool for integrating with third-party software. Integrations are carried out by using scripts to link the Wazuh manager with the software products' APIs.

• Setting up the integrator tool in preparation for a unique integration.
• Writing a Python script to handle alerts and compare the IP address in the source log with AbuseIPDB API checks.
• Formulating regulations according to the Confidence of Abuse scale.
• Identify when a person tries to log in using a non-private IP address; inquire of abuseipdb if the IP address is harmful; and get an answer from AbuseIPDB

Use case:-

We specifically examine the use case of producing alerts with extra AbuseIPDB data when public IP addresses authenticate SSH against a network endpoint.
We add rules to our Wazuh server in the /var/ossec/etc/rules/local_rules.xml file based on this situation, so that they will activate in the event that an SSH authentication attempt from a public IP is successful or unsuccessful.

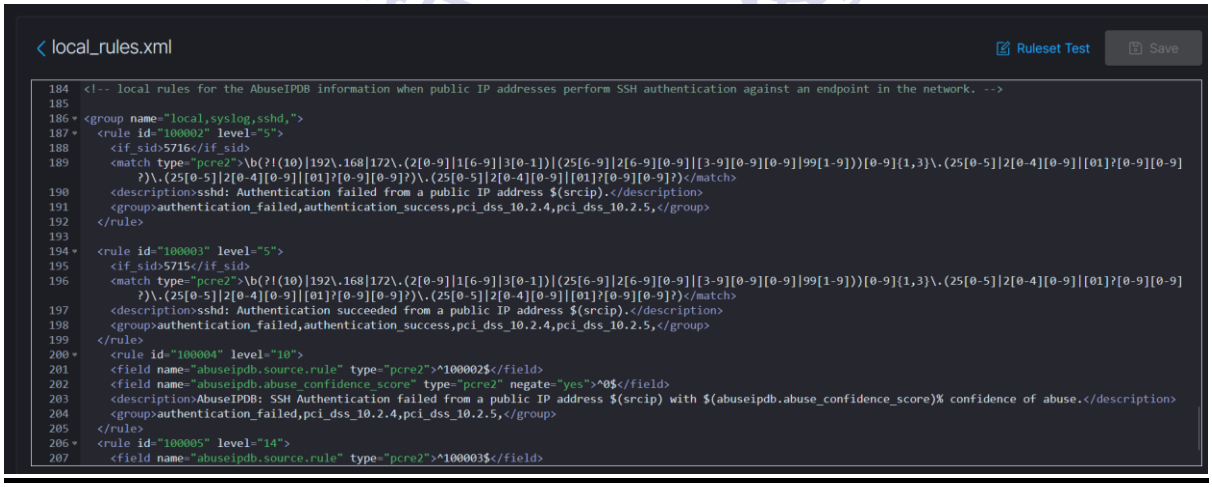


Figure 7 Wazuh Rules

Following an alert triggered by rule IDs 100002 or 100003, the Wazuh integration script requests information about the IP address in the alert from the AbuseIPDB Check IP API

endpoint. This is then applied to a rule that was developed using the Confidence of Abuse score.

The integration block with the following text needs to be added to the Wazuh manager configuration file, ossec.conf, in order to construct a custom integration:

```
<ossec_config>
  <!-- this for the abuesipabd API-->
<integration>
  <name>custom-abuseipdb.py</name>
  <hook_url>https://api.abuseipdb.com/api/v2/check</hook_url>
  <api_key>8a71ed63980a93e9d3c7151b4127bfed99d3ea239afe71718e1d9ef7b9ba92e55ba54b1755f3334c</api_key>
  <rule_id>100002,100003</rule_id>
  <alert_format>json</alert_format>
</integration>
```

Figure 8 Wazuh Config Rules

Writing the integration script

We then proceed to build a file named custom-abuseipdb.py in /var/ossec/integrations/ on the Wazuh server. It is significant to remember that:

The function that asks the IP address abuse information is call request_abuseipdb_info().

```
def request_abuseipdb_info(alert, apikey):
    alert_output = {}
    # If there is no source ip address present in the alert. Exit.
    if not "srcip" in alert["data"]:

      return(0)

    # Request info using AbuseIPDB API
    data = query_api(alert["data"]["srcip"], apikey)

    # Create alert
    alert_output["abuseipdb"] = {}
    alert_output["integration"] = "custom-abuseipdb"
    alert_output["abuseipdb"]["found"] = 0
    alert_output["abuseipdb"]["source"] = {}
    alert_output["abuseipdb"]["source"]["alert_id"] = alert["id"]
    alert_output["abuseipdb"]["source"]["rule"] = alert["rule"]["id"]
    alert_output["abuseipdb"]["source"]["description"] = alert["rule"]["description"]
    alert_output["abuseipdb"]["source"]["full_log"] = alert["full_log"]
    alert_output["abuseipdb"]["source"]["srcip"] = alert["data"]["srcip"]
    srcip = alert["data"]["srcip"]
    # Check if AbuseIPDB has any info about the srcip
    if in_database(data, srcip):
      alert_output["abuseipdb"]["found"] = 1

    # Info about the IP found in AbuseIPDB
    if alert_output["abuseipdb"]["found"] == 1:
        abuse_confidence_score, country_code, usage_type, isp, domain, total_reports,
```

Figure 9 custom-abuseipdb.py

This script extracts the source IP after reading the alerts JSON file. Next, to find out the reputation of the IP address that started the integration script, a request is sent to the AbuseIPDB API.

Creating rules with AbuseIPDB information

Information about alarms can be improved by using the AbuseIPDB response. As an illustration, we can notify you of a public IP address that successfully authenticated via SSH and has a non-zero abuse confidence score. To accomplish this, we can make the following custom rules in /var/ossec/etc/rules/local_rules.xml. To activate them, we must restart the manager:

```
200    <rule id="100004" level="10">
201      <field name="abuseipdb.source.rule" type="pcre2">^100002$</field>
202      <field name="abuseipdb.abuse_confidence_score" type="pcre2" negate="yes">^0$</field>
203      <description>AbuseIPDB: SSH Authentication failed from a public IP address $(srcip) with $(abuseipdb.abuse_confidence_score)% confidence of abuse.</description>
204      <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
205    </rule>
206    <rule id="100005" level="14">
207      <field name="abuseipdb.source.rule" type="pcre2">^100003$</field>
208      <field name="abuseipdb.abuse_confidence_score" type="pcre2" negate="yes">^0$</field>
209      <description>AbuseIPDB: SSH Authentication succeeded from a public IP address $(srcip) with $(abuseipdb.abuse_confidence_score)% confidence of abuse.</description>
210      <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
211    </rule>
212  </group>
213
214
215
216
```

Figure 10 Wazuh Local Rules

These rules can be triggered in a test via log injection on an endpoint enrolled to the Wazuh manager.

```
238
239   <!-- abuseipdb -->
240     <localfile>
241       <log_format>syslog</log_format>
242       <location>/var/log/test.log</location>
243     </localfile>
244
```

Figure 11 Endpoint Log Integration

OUTPUT DASHBOARDS

Figure 12 Dashboard

Figure 13 Dashboard

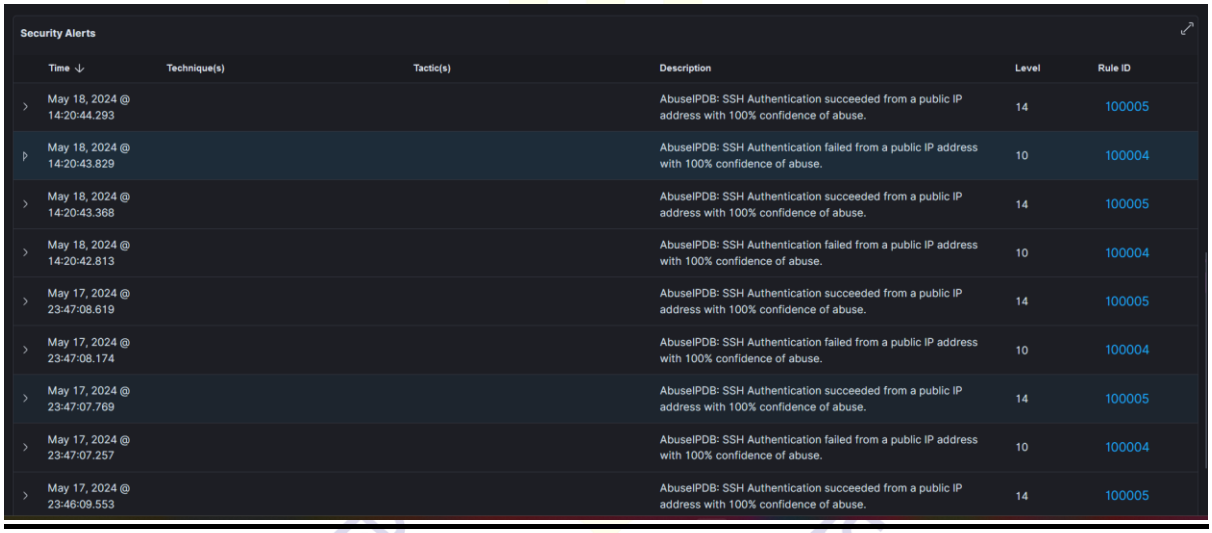


Figure 14 Logs

Figure 15 Active-response on ABUSEIPDB

```
root@ubuntu:/var/ossec/active-response/bin# tail -f /var/ossec/logs/active-responses.log
Saturday 18 May 2024 02:44:25 PM IST active-response/bin/restart.sh agent
Saturday 18 May 2024 02:44:49 PM IST active-response/bin/restart.sh agent
Saturday 18 May 2024 02:47:03 PM IST /var/ossec/active-response/bin/abuseipdb-firewall.sh Source IP 37.206.193.131 Added to Blacklist
Saturday 18 May 2024 02:47:35 PM IST /var/ossec/active-response/bin/abuseipdb-firewall.sh Source IP 37.206.193.131 Added to Blacklist
Saturday 18 May 2024 02:47:36 PM IST /var/ossec/active-response/bin/abuseipdb-firewall.sh Source IP 37.206.193.131 Added to Blacklist
Saturday 18 May 2024 02:47:37 PM IST /var/ossec/active-response/bin/abuseipdb-firewall.sh Source IP 37.206.193.131 Added to Blacklist
Saturday 18 May 2024 03:01:16 PM IST /var/ossec/active-response/bin/abuseipdb-firewall.sh Source IP 37.206.193.131 Added to Blacklist
Saturday 18 May 2024 03:01:47 PM IST /var/ossec/active-response/bin/abuseipdb-firewall.sh Source IP 37.206.193.131 Added to Blacklist
Saturday 18 May 2024 03:02:26 PM IST /var/ossec/active-response/bin/abuseipdb-firewall.sh Source IP 37.206.193.131 Added to Blacklist
Saturday 18 May 2024 03:02:27 PM IST /var/ossec/active-response/bin/abuseipdb-firewall.sh Source IP 37.206.193.131 Added to Blacklist
S
```

Figure 16 endpoint activity

```
root@ubuntu:/home/ubuntu/Desktop# iptables -vnL
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       all  --  *      *       37.206.193.131       0.0.0.0/0
    0     0 DROP       all  --  *      *       37.206.193.131       0.0.0.0/0
    0     0 DROP       all  --  *      *       37.206.193.131       0.0.0.0/0
    0     0 DROP       all  --  *      *       37.206.193.131       0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       all  --  *      *       37.206.193.131       0.0.0.0/0
    0     0 DROP       all  --  *      *       37.206.193.131       0.0.0.0/0
    0     0 DROP       all  --  *      *       37.206.193.131       0.0.0.0/0
    0     0 DROP       all  --  *      *       37.206.193.131       0.0.0.0/0
```

Figure 17 Endpoint Firewall iptables

```
Dec 10 01:03:02 host sshd[1234]: Accepted none for root from 64.62.197.132 port 1066 ssh2
Dec 10 01:02:02 host sshd[1234]: Failed none for root from 37.206.193.131 port 1066 ssh2
Dec 10 01:03:02 host sshd[1234]: Accepted none for root from 64.62.197.132 port 1066 ssh2
Dec 10 01:02:02 host sshd[1234]: Failed none for root from 37.206.193.131 port 1066 ssh2
Dec 10 01:03:02 host sshd[1234]: Accepted none for root from 64.62.197.132 port 1066 ssh2
```

Figure 18 Raw Logs

```
root@ubuntu:/var/ossec/active-response/bin# cat abuseipdb-firewall.sh
#!/bin/bash

LOCAL=`dirname $0`;
cd $LOCAL
cd ../

PWD=`pwd`

read -r INPUT_JSON
SRCIP=$(echo $INPUT_JSON | jq -r .parameters.alert.data.abuseipdb.source.srcip)
COMMAND=$(echo $INPUT_JSON | jq -r .command)
LOG_FILE="${PWD}/../logs/active-responses.log"

#------------------------ Analyze command -------------------------#
# Blocking Source IP
if [ ${COMMAND} = "add" ]
then
/sbin/iptables -I INPUT -s ${SRCIP} -j DROP
/sbin/iptables -I FORWARD -s ${SRCIP} -j DROP
echo "`date` /var/ossec/$0 Source IP $SRCIP Added to Blacklist" >> ${LOG_FILE}
else
/sbin/iptables -D INPUT -s ${SRCIP} -j DROP
/sbin/iptables -D FORWARD -s ${SRCIP} -j DROP
echo "`date` /var/ossec/$0 Source IP $SRCIP Removed from Blacklist" >> ${LOG_FILE}
fi

exit 0;
root@ubuntu:/var/ossec/active-response/bin# S
```

Figure 19 Firewall Drop Scripte

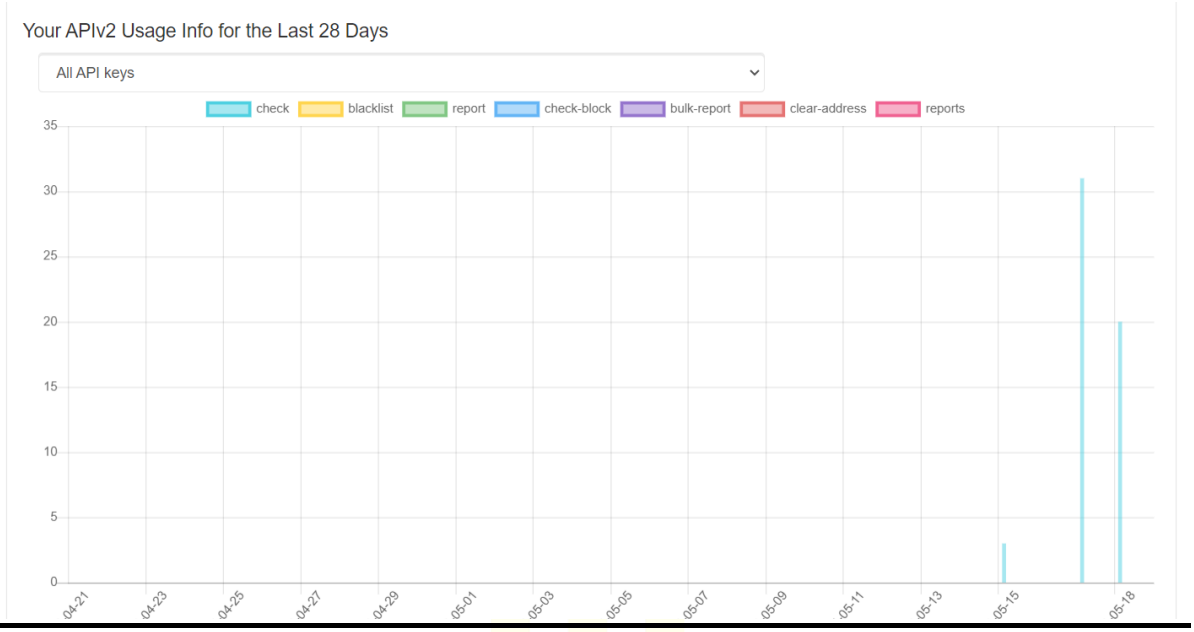


Figure 20 AbuseIP Web Api Calls Logs

10.2 Use Case: Blocking SSH brute-force attack with active response

Objective:

To automatically block IP addresses that perform SSH brute force attacks after 5 failed attempts using Wazuh Active Response.

Overview:

Brute force attacks over SSH involve an attacker attempting a number of username-password combinations in an attempt to enter a system. We can identify these attacks and block the offending IP addresses after a certain number of unsuccessful attempts by combining Wazuh with Active Response.

Results:

- Successfully detected and blocked IP addresses after 5 failed SSH login attempts.
- Enhanced security by automating the response to SSH brute force attacks.

```
296
297 <active-response>
298   <command>firewall-drop</command>
299   <location>local</location>
300   <agent_id>002</agent_id>
301   <rules_id>5710,5762,5760,5503</rules_id>
302   <timeout>100</timeout>
303 </active-response>
```

Figure 21 Script to drop

Conclusion: The integration of Wazuh with Active Response and firewall rules provides an effective solution for mitigating SSH brute force attacks, improving overall system security.

OUTPUT

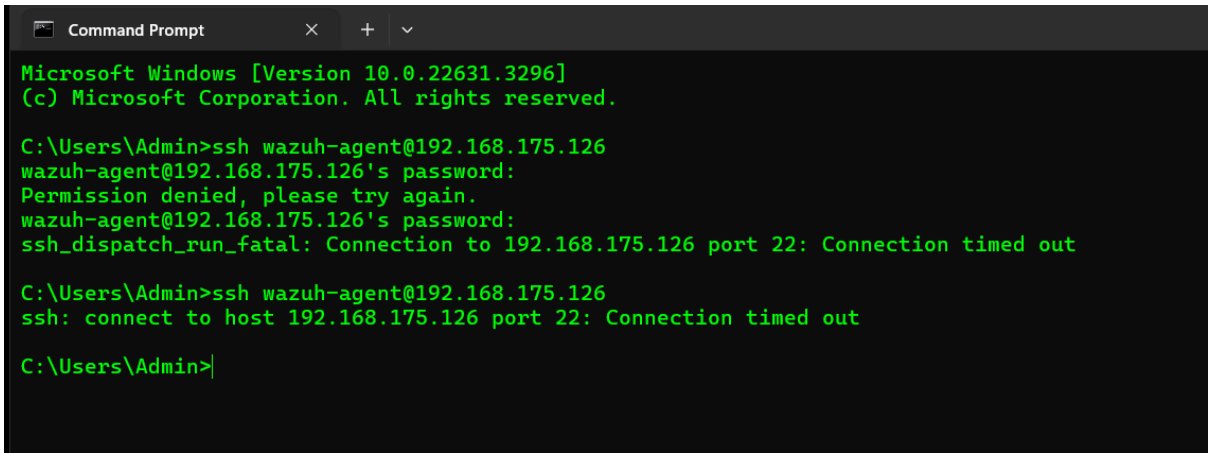


Figure 22 ssh connection packet drop automatically to stop brute force attack

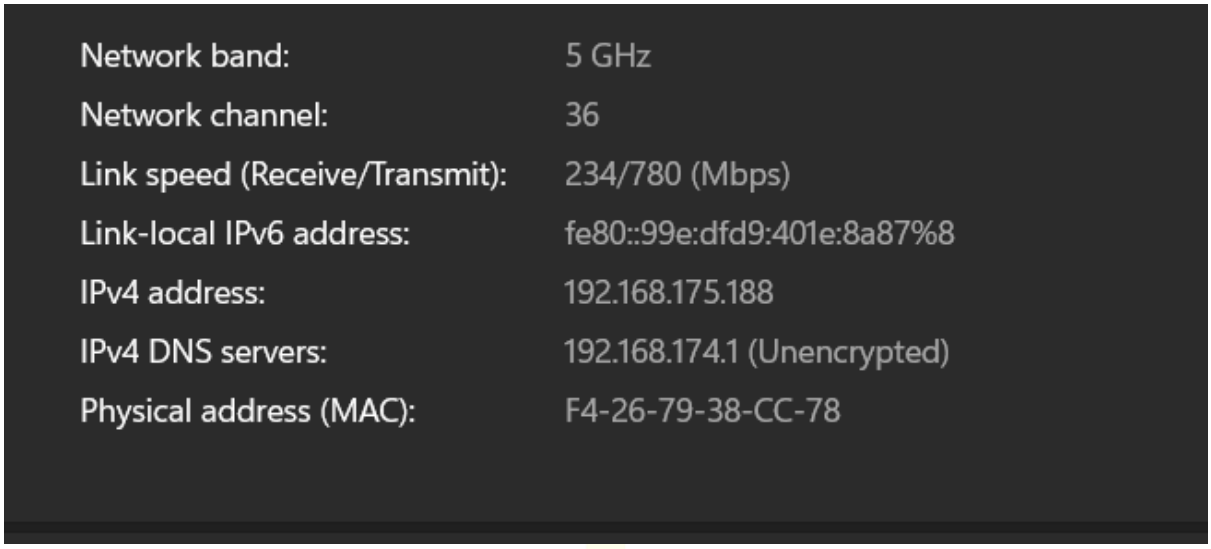


Figure 23 cross chek the ip

DASHBOARD

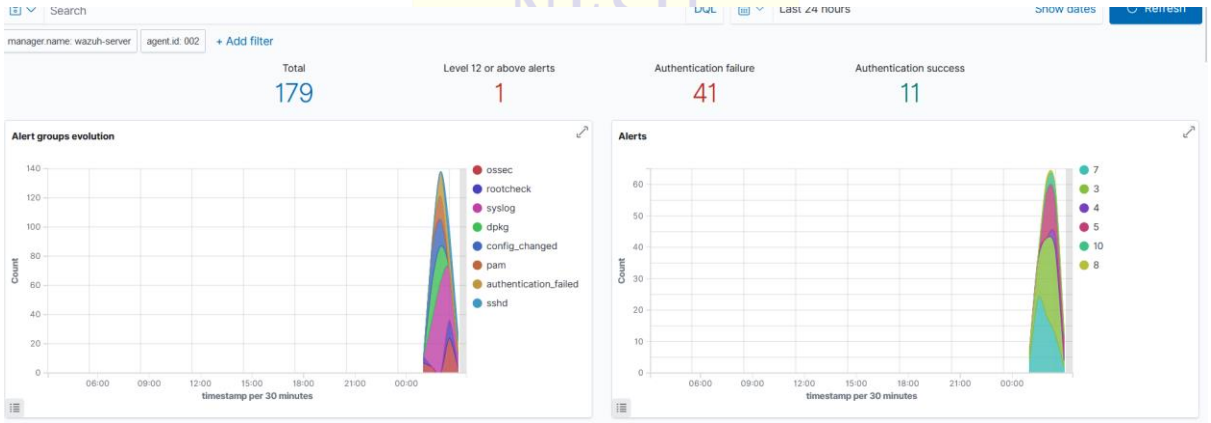


Figure 24 Dashboard

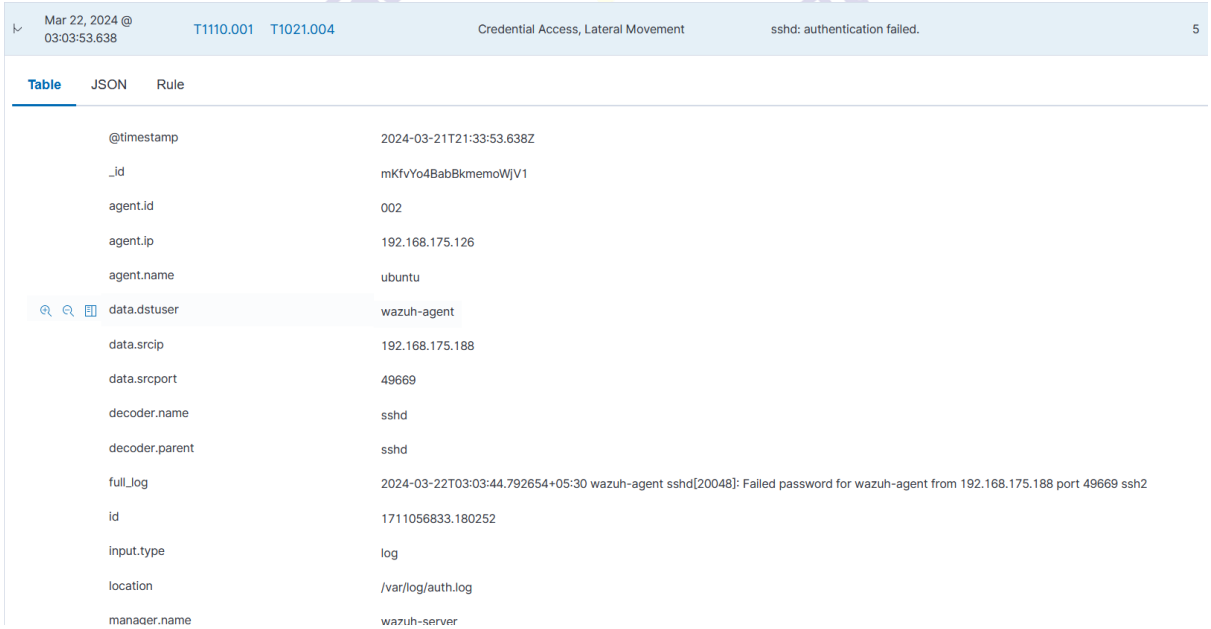


Figure 25 Details log

10.3 Automating Malware Detection and Remediation with VirusTotal

Wazuh connects to external APIs and warning systems like VirusTotal using the integrator module.
In this use example, a directory is monitored for modifications using the Wazuh File Integrity Monitoring (FIM) module, and the files within the directory are scanned using the VirusTotal API. Next, set up Wazuh to delete files that VirusTotal deems harmful and launch an active response script. We evaluate both Windows and Ubuntu endpoints for this use case.

In this use scenario, Wazuh's authentication to the VirusTotal API requires a VirusTotal API key.

Configuration for Ubuntu endpoint

Follow these steps to set up your environment to test the Ubuntu endpoint use case. These instructions also apply to other Linux distributions.

Ubuntu endpoint

To set up Wazuh to track changes in the Ubuntu endpoint's /home/Downloads directory very instantly, follow these steps. Along with installing the required packages, these procedures also construct the active reaction script that deletes harmful files.

• Look for the block in the /var/ossec/etc/ossec.conf Wazuh agent configuration file. Verify that is not set to yes. By doing this, the Wazuh FIM can keep an eye out for directory changes.
• Create a new entry in the block to set up a directory for near-real-time monitoring. You are now keeping an eye on the /home/Downloads directory:

```
102   <!-- File integrity monitoring -->
103
104   <syscheck>
105         <directories realtime="yes" check_all="yes" check_sum="yes" check_owner="yes"
    check_attrs="yes" check_mtime="yes" check-inode="yes" report_change="yes" whodata="yes">/home/
    ubuntu/Downloads</directories>
106         <directories realtime="yes" check_all="yes" check_sum="yes" check_owner="yes"
    check_attrs="yes" check_mtime="yes" check-inode="yes" report_change="yes" whodata="yes">/home/
    ubuntut/Documents</directories>
107         <nodiff>/home/ubuntu/Documents/private.txt</nodiff>
108   </syscheck>
109
110   <syscheck>
111     <disabled>no</disabled>
112
113     <!-- Frequency that syscheck is executed default every 12 hours -->
114     <frequency>43200</frequency>
115
```

Figure 26 Ubuntu Endpoint Configuration

Wazuh server

To enable the VirusTotal integration and set up an alert for changes in the endpoint directory, follow these steps on the Wazuh server. Additionally, these actions activate and enable the current reaction script in the event that a suspicious file is found.

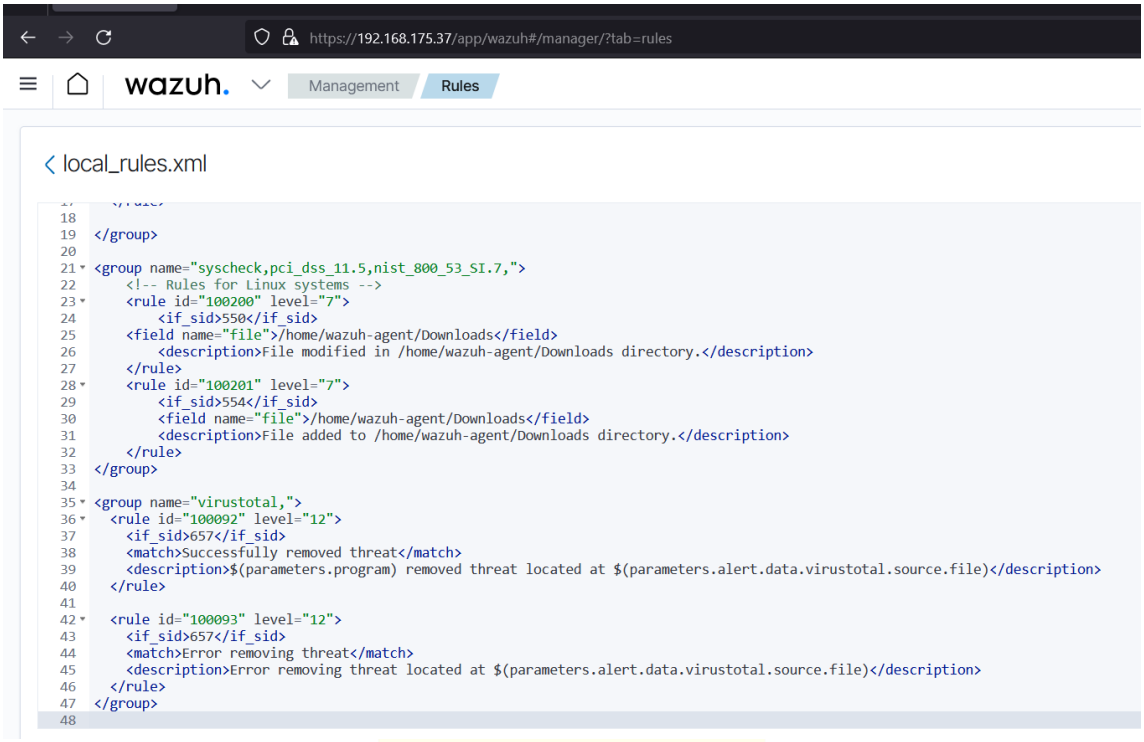


Figure 27 Wazuh Local Rules

```
<ossec_config>
  <integration>
    <name>virustotal</name>
    <api_key>d9a756b3a18225aa04ae33b9537ccd42cee75f3c92f34f9d0b402704d4ab530e</api_key> <!-- Replace with your VirusTotal API key -->
    <rule_id>100200,100201</rule_id>
    <alert_format>json</alert_format>
  </integration>
</ossec_config>

<ossec_config>
  <command>
    <name>remove-threat</name>
    <executable>remove-threat.sh</executable>
    <timeout_allowed>no</timeout_allowed>
  </command>

  <active-response>
    <disabled>no</disabled>
    <command>remove-threat</command>
    <location>local</location>
    <rules_id>87105</rules_id>
  </active-response>
</ossec_config>
```

Figure 28 wazuh config ruuleset

Attack emulation

• On the Ubuntu endpoint, download an EICAR test file to the /home/Download directory:

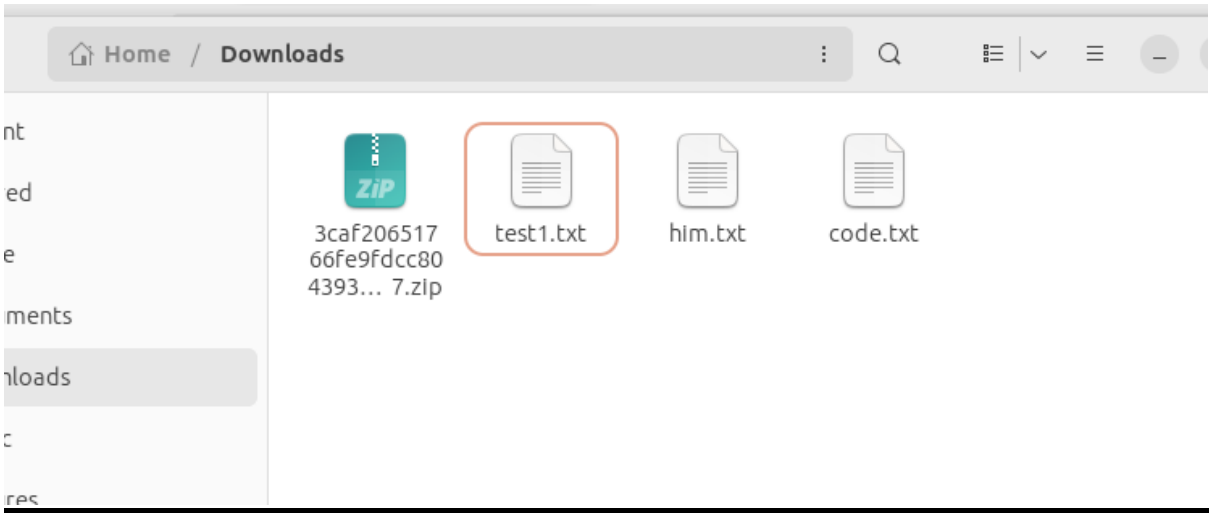


Figure 29 not malicious file

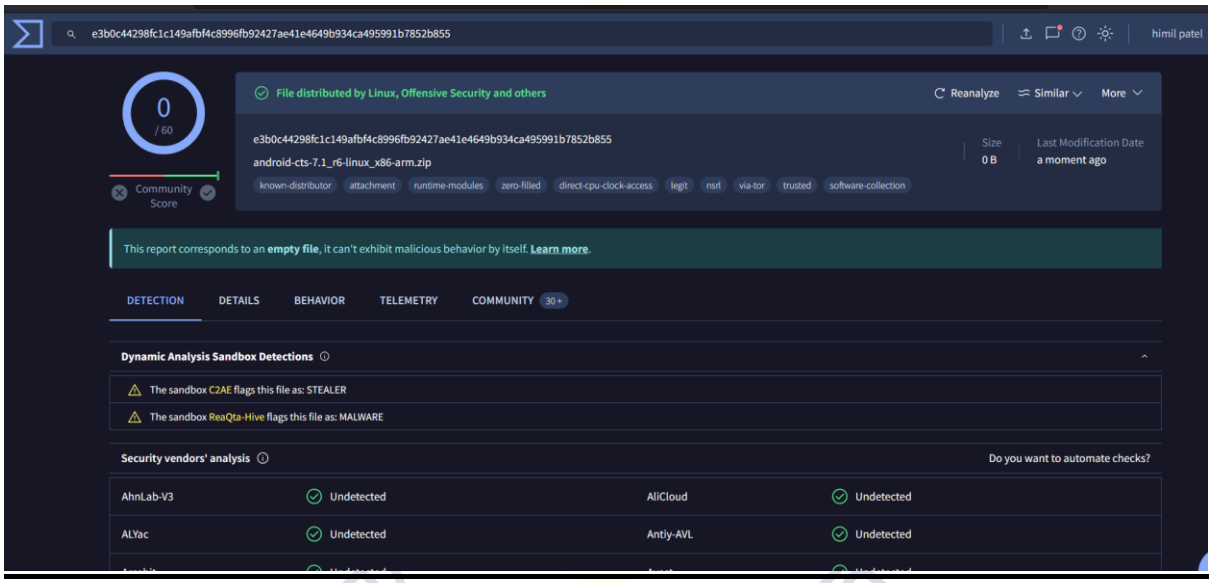


Figure 30 Not malicious file

Visualize the alerts

The Wazuh dashboard allows you to view the alert data. Go to the Threat Hunting module and use the search bar's filters to query the alerts in order to accomplish this.
The Linux system has rule.id: 553,100092,87105,100201

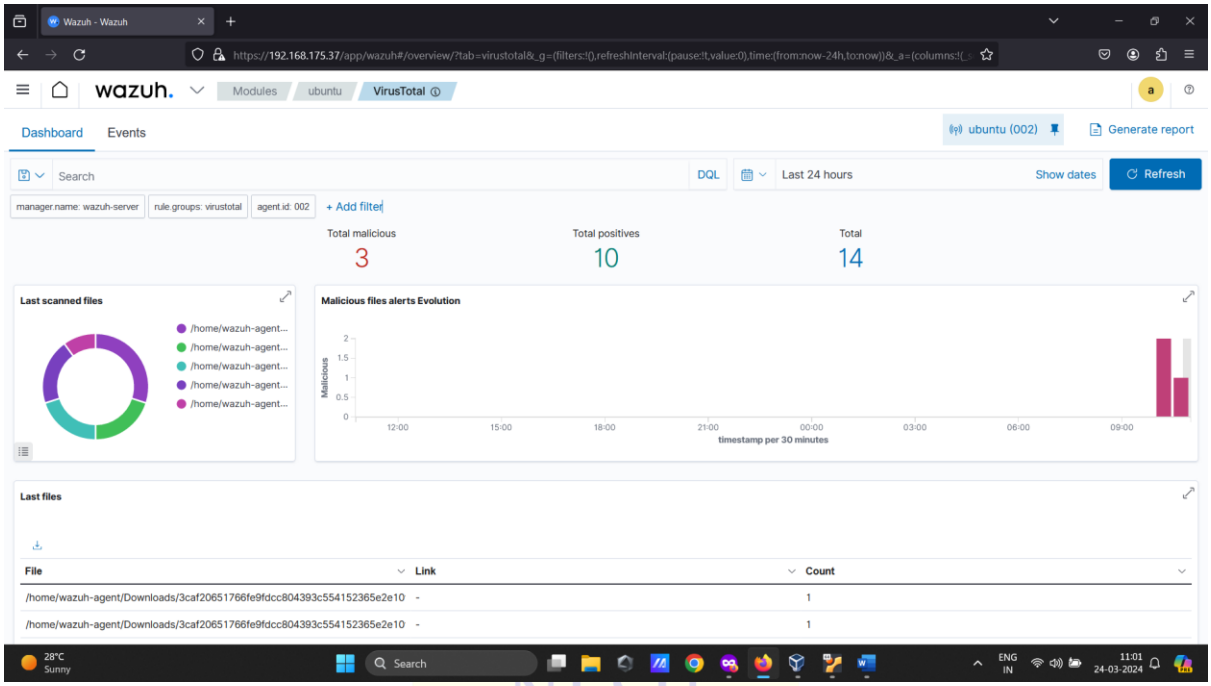


Figure 31 VT Dashboard

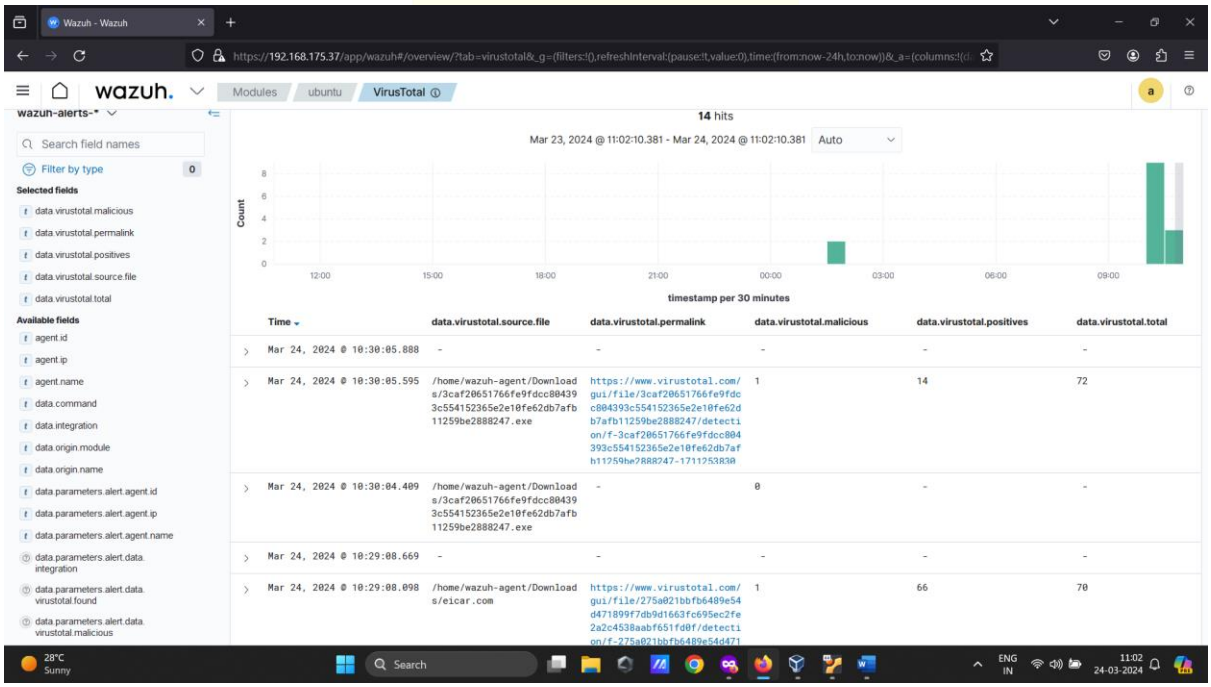


Figure 32 VT logs Dashboard

We can able to see the alert details in the event section and if we can do the cross check so that just click on the link provided by the Wazuh it will redirect to main VT page.

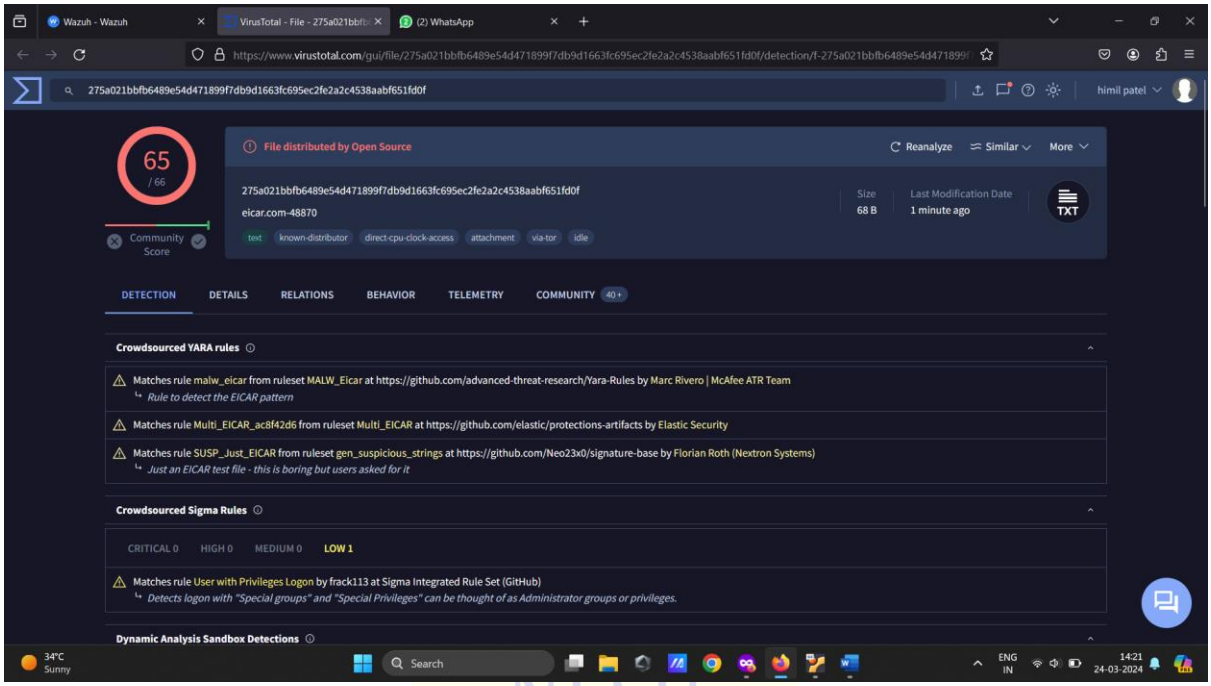


Figure 33 Working link of VT from Dashboard

10.4 Real-time USB Drive Anomaly Detection

On a Linux endpoint, keeping an eye on USB drives is crucial to preserving system security and integrity. USB drives can serve as a potential entry point for malware and unauthorized data access. By monitoring these drives, administrators can detect and prevent the introduction of malicious software or unauthorized data transfers.

Additionally, monitoring *USB drives* helps in ensuring compliance with security policies and regulations, safeguarding sensitive data, and mitigating the risk of data breaches. It's a proactive measure that enhances overall system security and protects against potential threats posed by external storage devices.

Out of the box, Wazuh provides a rule to monitor USB devices in Linux endpoints. However, the logs are limited and do not contain enough information about the event, as seen in the image below. Using `udev` rules, we can get richer logs about USB events.

Configuration

In this section, we configure the Ubuntu endpoint to utilize the `udev` utility to generate enhanced logs when a USB drive is connected to the endpoint. We also configure custom rules and a CDB list on the Wazuh server, enabling it to trigger an alert when an unauthorized USB device is connected to the Ubuntu endpoint.

Ubuntu endpoint

By default, Linux devices are equipped with the udev utility. Device nodes in the /dev directory are dynamically managed by the Linux kernel's udev device manager. It is in charge of identifying and setting up devices when they are plugged into and unplugged from the system. The udev tool facilitates communication between applications and hardware elements.
We create a `udev` rule that detects when a USB device is connected to the monitored endpoint. The rule triggers a script that writes details about the event to a log file, which the Wazuh agent reads and forwards to the Wazuh server for analysis.

1. Make a file in the /var/ossec/bin/ directory called usb_detect.sh:
2. In the /var/ossec/bin/usb_detect.sh file, add the following script:

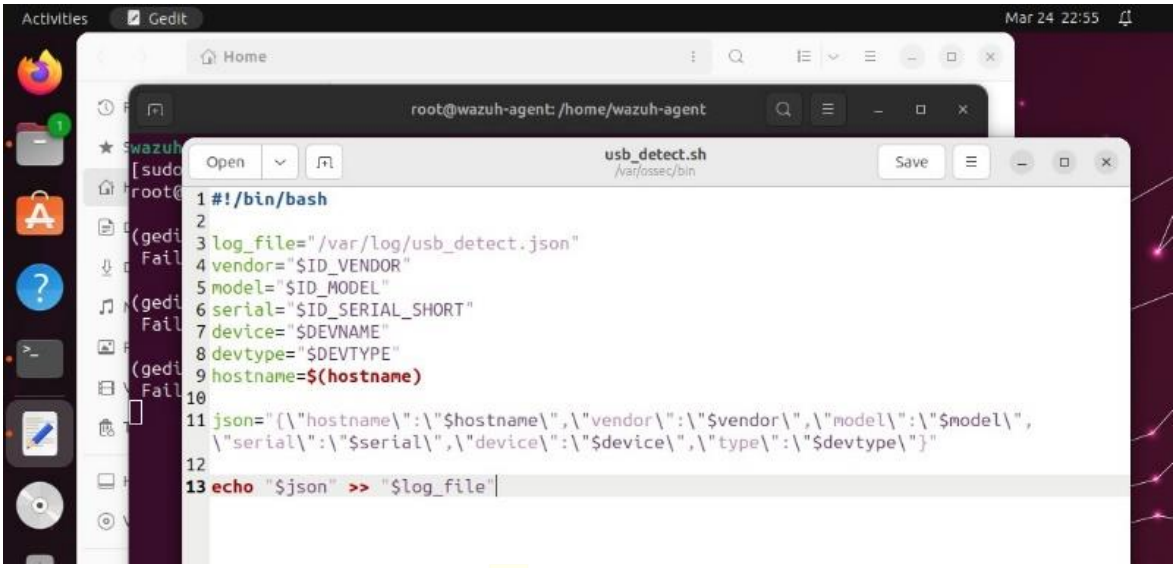


Figure 34 usb detection script

This script allows the `udev` utility to collect detailed information about the USB device that has been attached to the endpoint. It also writes the log to the `/var/log/usb_detect.json` JSON file, which can easily be ingested by Wazuh.

3. Modify the file permission to prevent unauthorised users from running the script:
4. To gather the logs from the /var/log/usb_detect.json file, append the following settings to the Wazuh agent's /var/ossec/etc/ossec.conf file:

```
217   </localfile>
218
219 </ossec_config>
220
221 <ossec_config>
222   <!-- Logcollector for udev USB detected Logs -->
223   <localfile>
224     <log_format>json</log_format>
225     <location>/var/log/usb_detect.json</location>
226   </localfile>
227 </ossec_config>
```

Figure 35 agent configuration

5. Restart the Wazuh agent to apply the changes:

Wazuh server

We generate a CDB list of approved USB devices on the Wazuh server. Furthermore, we design a custom rule to send out a notification whenever an unauthorised USB device is linked to the endpoint under observation.

Detecting USB drives

To find out when a USB device is connected to a monitored endpoint, create the following rule:

1. To the file /var/ossec/etc/rules/local_rules.xml, add the following rule:

```
48
49
50  <!-- Rule for USB monitoring in Linux-->
51  <group name="Linux, usb,">
52    <rule id="111010" level="7">
53      <field name="serial">\w+</field>
54      <field name="type">usb_device</field>
55      <description>A PNP device $(vendor) $(model) was connected to $(hostname).</description>
56    </rule>
57
58    <rule id="111011" level="8">
59      <if_sid>111010</if_sid>
60      <list field="serial" lookup="not_match_key">etc/lists/usb-drives</list>
61    <description>Unauthorized PNP device $(vendor) $(model) was connected to $(hostname).</description>
62    </rule>
63  </group>
```

Figure 36 wazuh local rules

OUTPUT

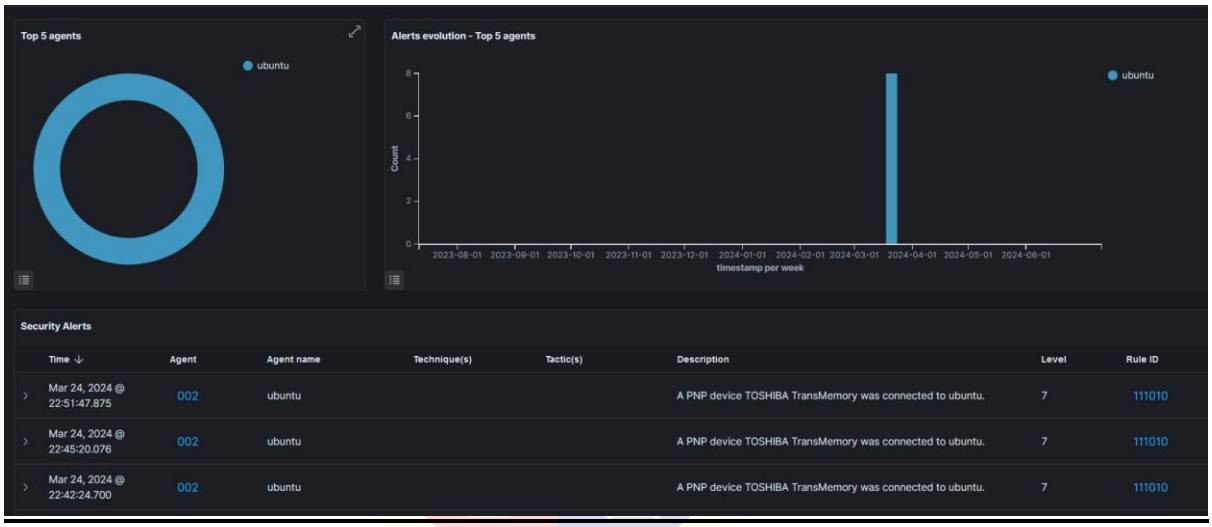


Figure 37 Dashboard

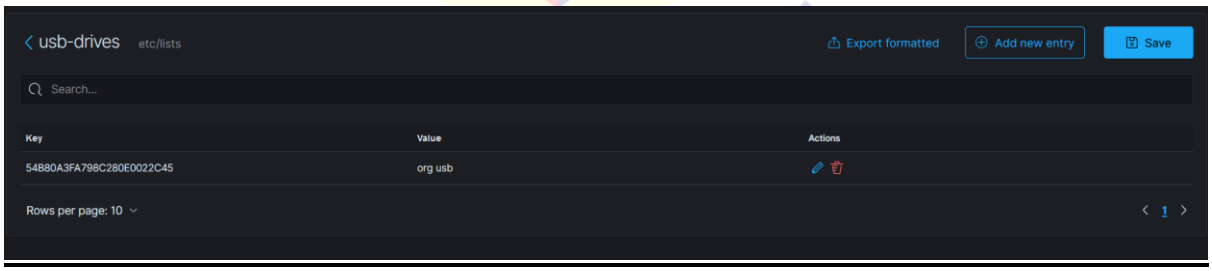


Figure 38 CDB List to authorise the usb

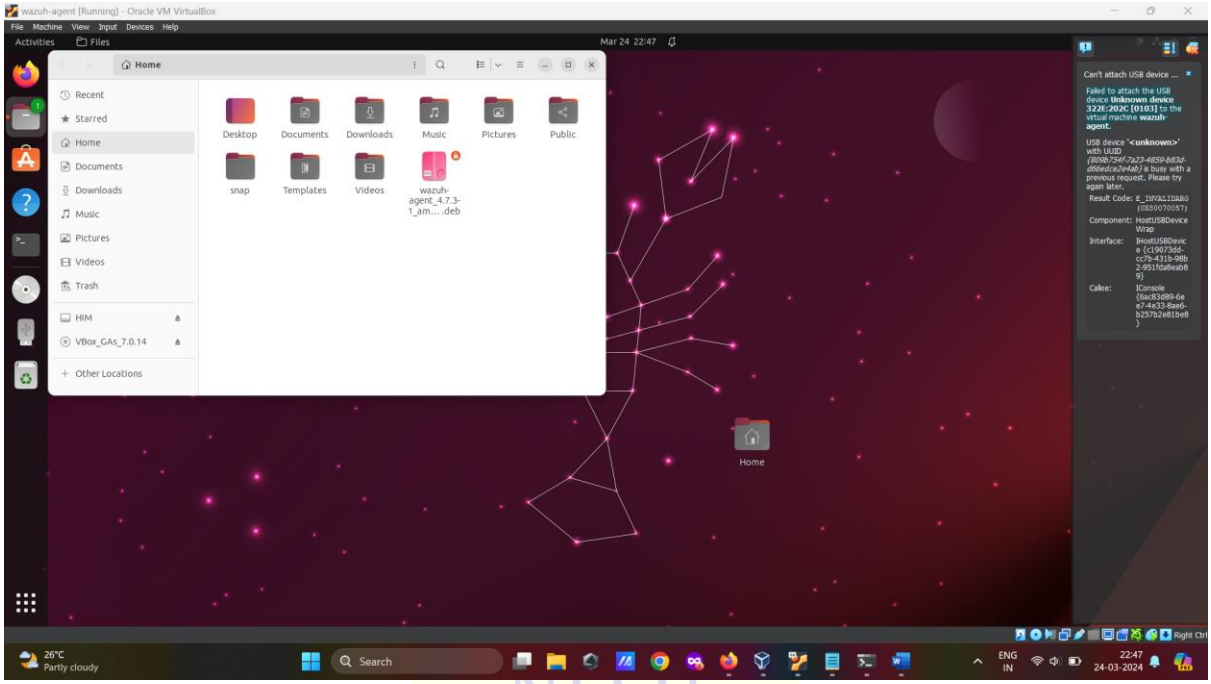


Figure 39 Attach the usb

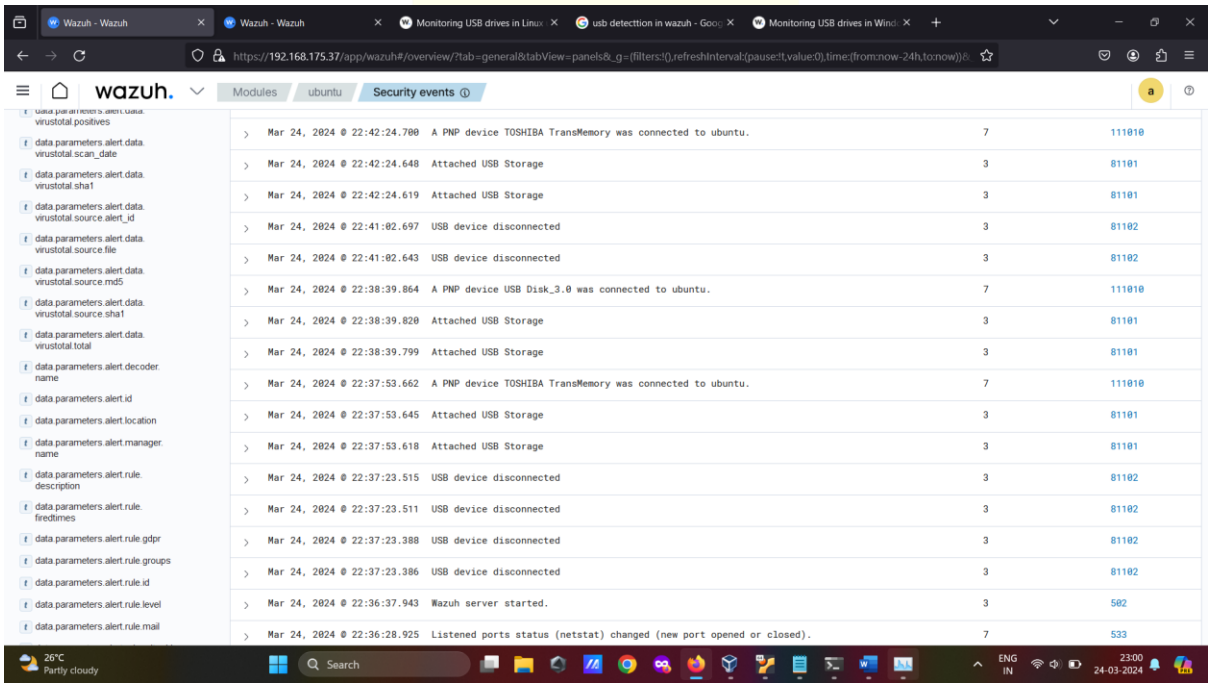


Figure 40 Logs of usb attach and detached

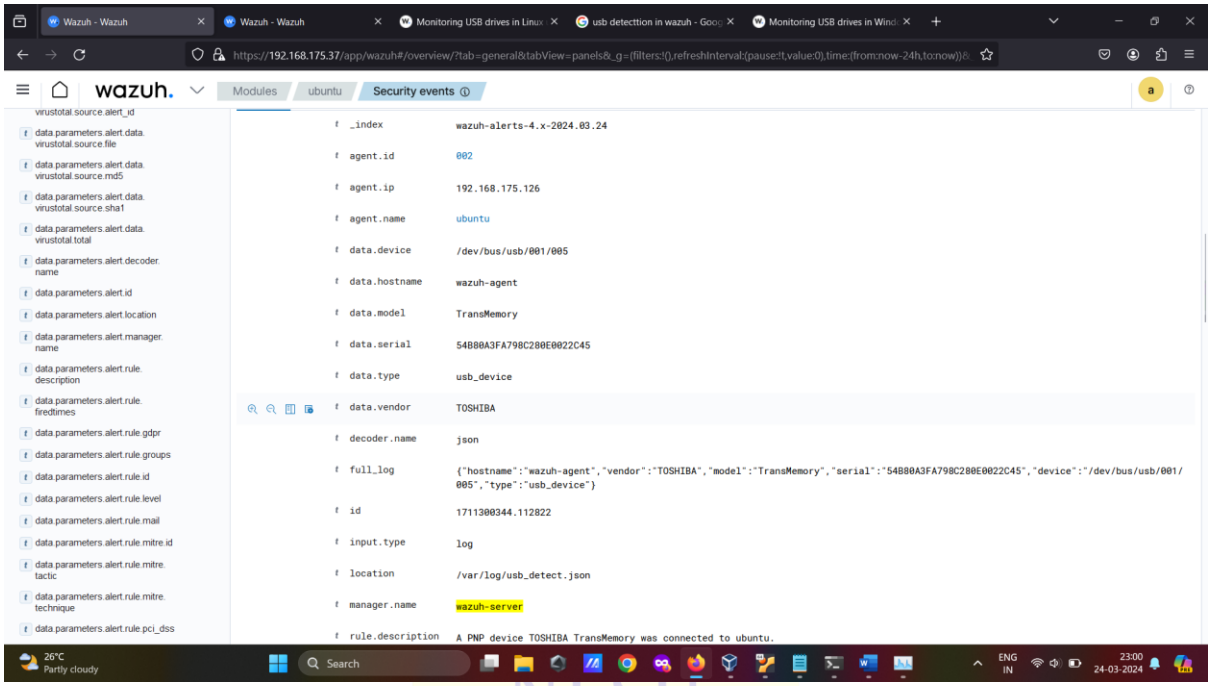


Figure 41 Details Logs

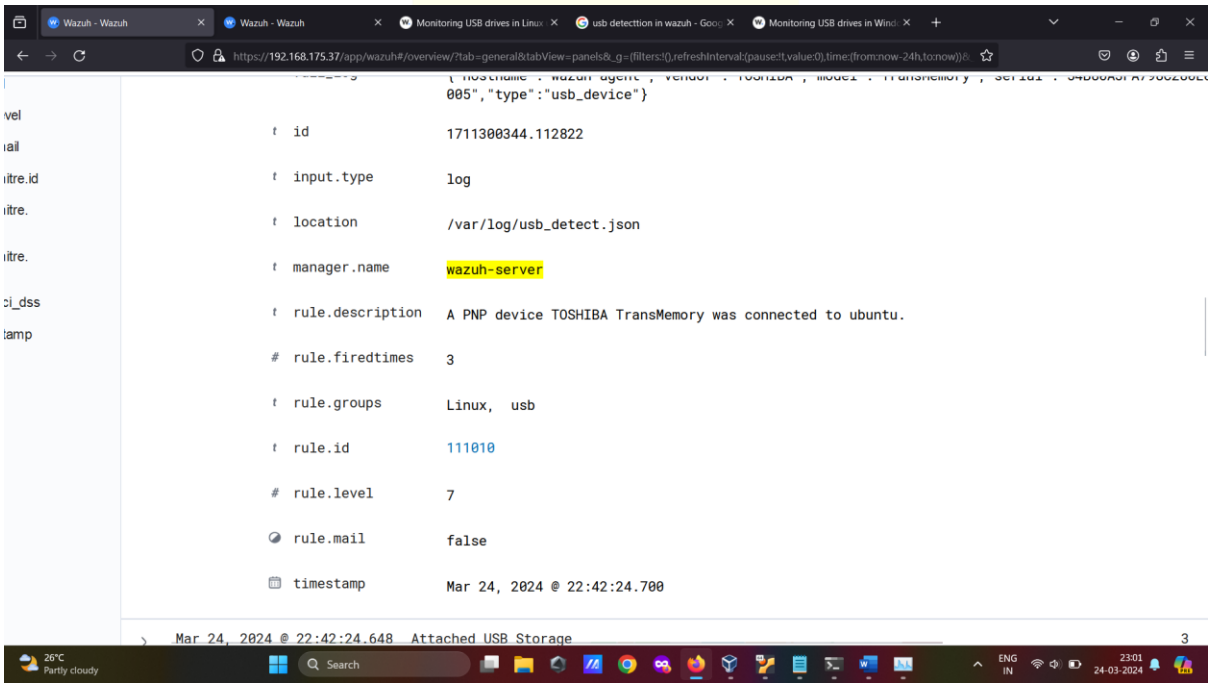


Figure 42 Details Logs

10.5 Restricting Unauthorized Software Usage

Scope:

This report outlines the implementation of an automated software blocking policy using PowerShell 7.0, Wazuh, and Sysmon. The focus is on enhancing security measures within the organization by identifying and blocking unauthorized software installations and executions.

Objective:

The primary objective is to establish a robust, automated mechanism to detect and block unauthorized software on organizational systems. This is achieved by leveraging the capabilities of PowerShell 7.0 for scripting, Wazuh for monitoring and alerting, and Sysmon for detailed event logging.

Purpose:

This policy's goal is to stop malicious or unauthorised software from causing security breaches. Our goal is to lower the risk of malware infections, data breaches, and other security issues by automating the detection and blocking process.

Importance:

Enforcing an automated software blocking strategy is essential to preserving organisational systems' security and integrity. Unauthorised software can cause serious data loss or theft, present vulnerabilities, and interfere with operations. An efficient policy makes sure that only authorised software is run, improving the organization's overall security posture.

Explanation: How it Working

We can install sysinternal on the end point so that with the help of the sysmon we can get the more pro-active log details based on that we can set the rules in siem tools for detection

example like event:1 is used for process creation means any application is started or open by the user based on that event 1 we can check the company name of that application is allowed on the cbd list or not base on that allied application are run and this checking is done by the siem

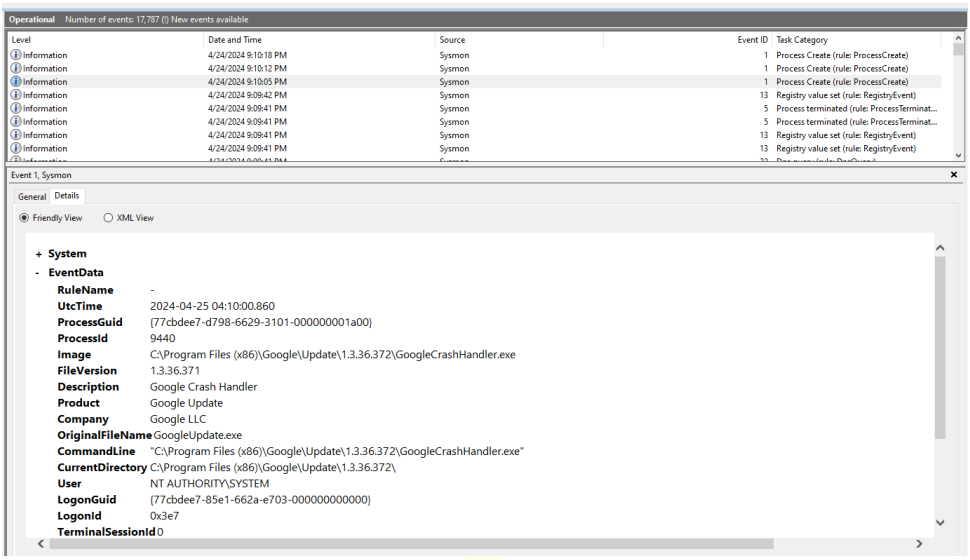


Figure 43 Details of log Dictation

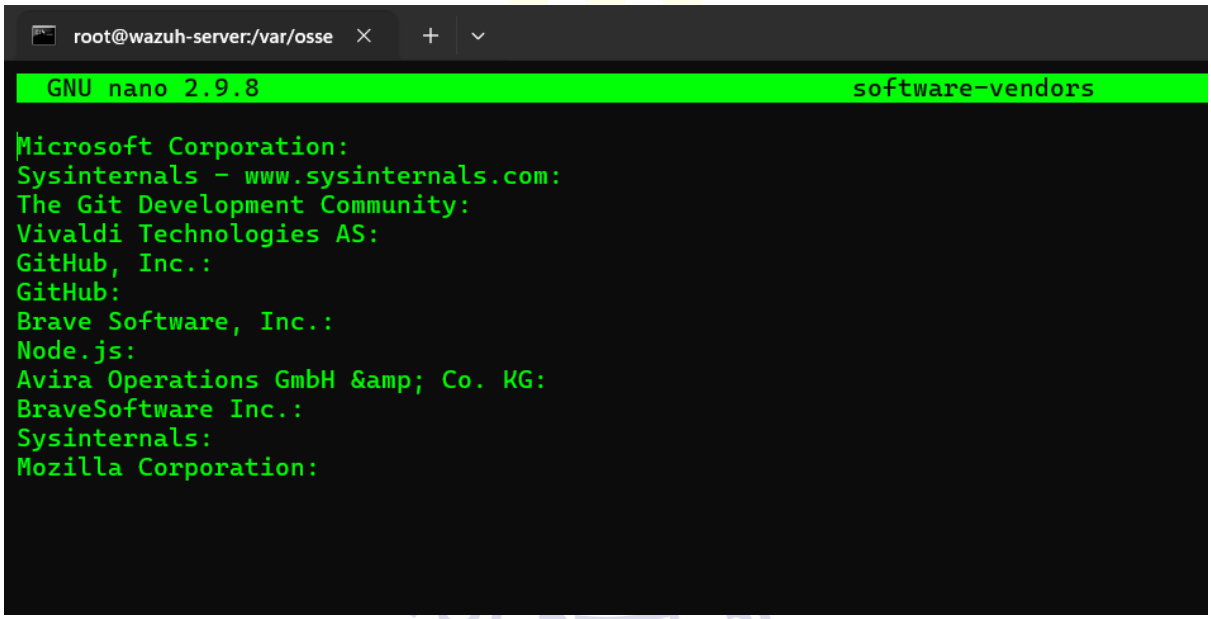


Figure 44 This is the CDB list where we can list the allowed software company name Whitelist

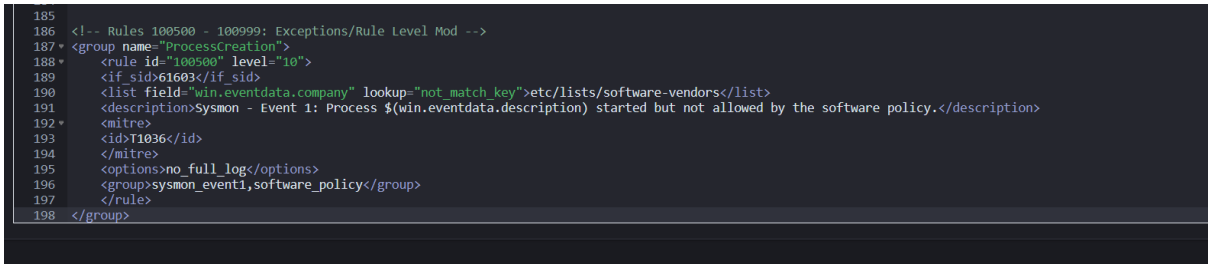


Figure 45 Rule set of the detection

```
<!--
<active-response>
  software-policy
</active-response>
-->

<command>
  <name>pssuspend</name>
  <executable>pssuspend.cmd</executable>
  <timeout_allowed>no</timeout_allowed>
</command>

<active-response>
 <disabled>no</disabled>
 <level>10</level>
 <command>pssuspend</command>
 <location>local</location>
 <rules_group>software_policy</rules_group>
</active-response>
```

Figure 46 Make active response for the blocking application

pssuspend.cmd

```
:: Simple script to run Sysinternals PSSuspend.
:: The script executes a powershell script and appends output.
@ECHO OFF
ECHO.

pwsh.exe -executionpolicy ByPass -File "c:\Program Files\Sysinternals\pssuspend.ps1"

:Exit
```

Figure 47 This is the pssuspend.cmd put this into the active-response folder at the agent side

pssuspend.ps1 - Notepad

File Edit Format View Help

```
# Read the Alert that triggered the Active Response in manager and convert to Array
$INPUT_JSON = Read-Host
$INPUT_ARRAY = $INPUT_JSON | ConvertFrom-Json
$INPUT_ARRAY = $INPUT_ARRAY | ConvertFrom-Json
$ErrorActionPreference = "SilentlyContinue"
#Switch For Rule Group From Alert
$switch_condition = ($INPUT_ARRAY."parameters"."alert"."rule"."groups"[1]).ToString()
#Create Notification shown in User's context.
$notification = '{
$msecs=3000
$Text=""An application was suspended due to the software policies in place""
$Title=""Application Suspended""
Add-Type -AssemblyName System.Windows.Forms
$global:balloon = New-Object System.Windows.Forms.NotifyIcon
$path = (Get-Process -id $pid).Path
$balloon.Icon = [System.Drawing.Icon]::ExtractAssociatedIcon($path)
$balloon.BalloonTipIcon = [System.Windows.Forms.ToolTipIcon]::Warning
$balloon.BalloonTipText = "$Text"
$balloon.BalloonTipTitle = "$Title"
$balloon.Visible = $true
$balloon.ShowBalloonTip($msecs)
}'
switch -Exact ($switch_condition){
"software_policy"
    {
#Extract Process ID and File Path from Alert
       $process_id_alert = $INPUT_ARRAY."parameters"."alert"."data"."win"."eventdata"."processId"
       $process_file_alert = $INPUT_ARRAY."parameters"."alert"."data"."win"."eventdata"."image"
       $process_file_alert = $process_file_alert -replace "\\\\", "\"
#Get-Process by Process ID and extract process full path
       $running_process_name = (Get-Process -Id $process_id_alert -FileVersionInfo).Filename
#Execute PSSuspend if match with alert
       if ($running_process_name -eq $process_file_alert) {
# Get User's Session ID, used for notification popup
        $user_session_id=(Get-Process -PID $process_id_alert).SessionID
# Execute Notification in user's context.
        c:\"Program Files"\Sysinternals\psexec64.exe /nobanner /accepteula -i $user_session_id pwsh.exe -executionpolicy bypass -WindowStyle Hidden -Command "& $notification"
# Suspend Process, sleep and then kill it.
        c:\"Program Files"\Sysinternals\pssuspend64.exe /nobanner /accepteula $process_id_alert
        Start-Sleep -s 3
        c:\"Program Files"\Sysinternals\pskill64.exe /accepteula $process_id_alert
       }
    break;
    }
}
```

Figure 48 This is the real script that can block the software at the agent side

›	Apr 22, 2024 @ 09:28:05.645	Sysmon - Event 1: Process creation.	5	101101	Microsoft Corporation
›	Apr 22, 2024 @ 09:28:05.413	Service startup type was changed	3	61104	-
›	Apr 22, 2024 @ 09:28:05.411	New Windows Service Created to start from windows root path. Suspicious event as the binary may have been dropped using Windows Admin Shares.	12	92650	-
›	Apr 22, 2024 @ 09:28:04.639	Sysmon - Event 1: Process creation.	5	101101	Sysinternals
›	Apr 22, 2024 @ 09:28:04.626	Sysmon - Event 13: RegistryEvent (Value Set).	5	101113	-
›	Apr 22, 2024 @ 09:28:04.607	Sysmon - Event 13: RegistryEvent (Value Set).	5	101113	-
›	Apr 22, 2024 @ 09:28:04.597	Executable dropped in Windows root folder	6	92217	-
›	Apr 22, 2024 @ 09:28:04.593	Sysmon - Event 13: RegistryEvent (Value Set).	5	101113	-
›	Apr 22, 2024 @ 09:28:04.580	Sysmon - Event 1: Process creation.	5	101101	Sysinternals - www.sysinternals.com
›	Apr 22, 2024 @ 09:28:04.579	Sysmon - Event 22: DNS Query.	5	101100	-
›	Apr 22, 2024 @ 09:28:04.570	Sysmon - Event 1: Process creation.	5	101101	Microsoft Corporation
›	Apr 22, 2024 @ 09:28:04.570	Scripting file created under Windows Temp or User folder	6	92200	-
›	Apr 22, 2024 @ 09:28:04.564	Sysmon - Event 1: Process creation.	5	101101	Microsoft Corporation
›	Apr 22, 2024 @ 09:28:03.524	Sysmon - Event 1: Process Google Chrome started but not allowed by the software policy.	10	100500	Google LLC
›	Apr 22, 2024 @ 09:28:03.520	Sysmon - Event 1: Process Google Chrome started but not allowed by the software policy.	10	100500	Google LLC

Figure 49 Event dashboard logs if google open and microsoft allowed app open

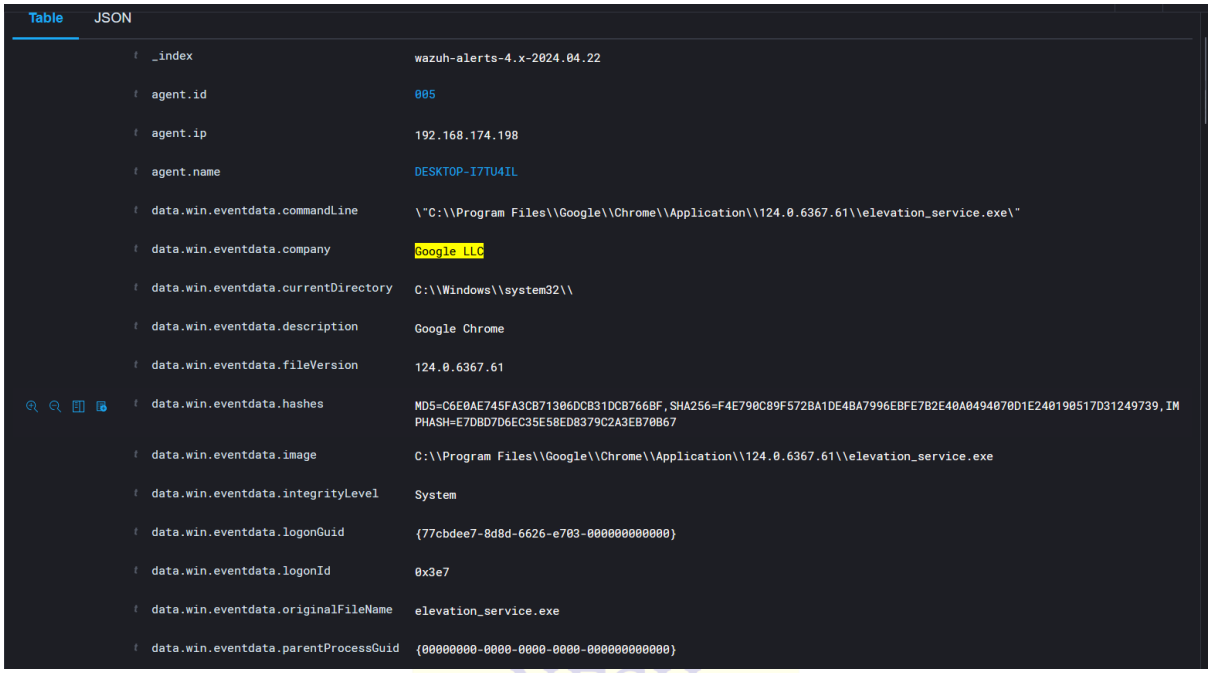


Figure 50 Log Details

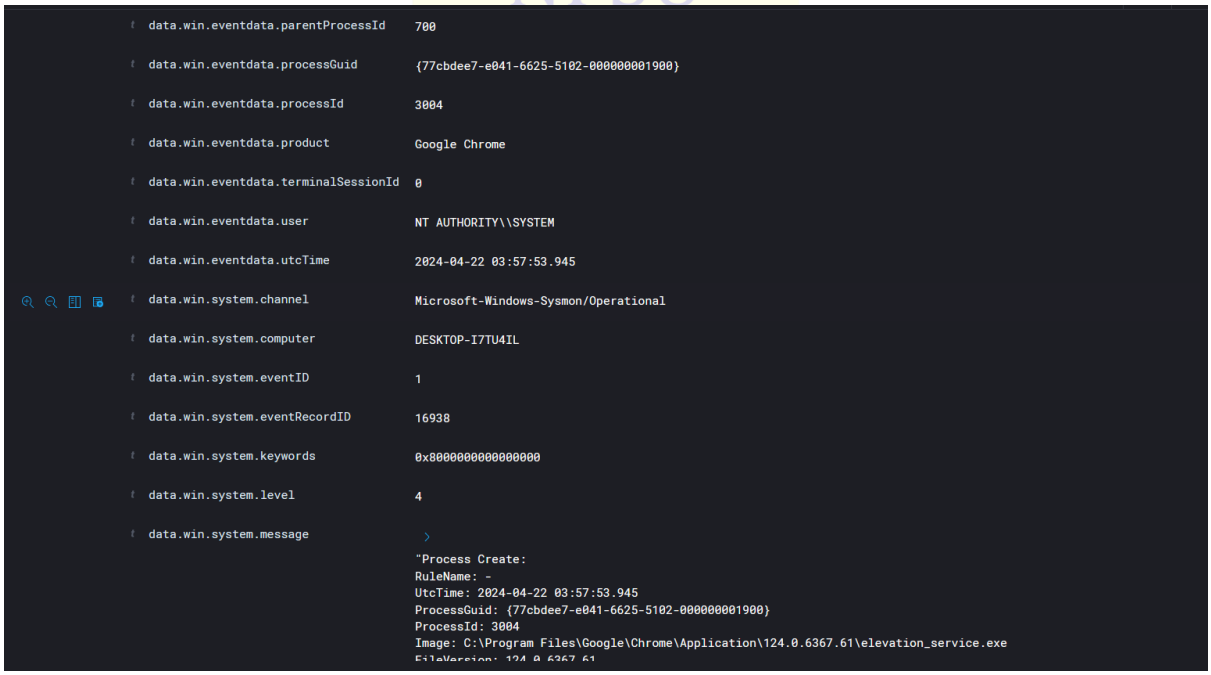


Figure 51 Google chrome app block

t	manager.name	wazuh-server
t	rule.description	Sysmon - Event 1: Process Google Chrome started but not allowed by the software policy.
#	rule.firedtimes	20
t	rule.groups	ProcessCreationsysmon_event1, software_policy
t	rule.id	100500
#	rule.level	10
	rule.mail	false
t	rule.mitre.id	T1036
t	rule.mitre.tactic	Defense Evasion
t	rule.mitre.technique	Masquerading
	timestamp	Apr 22, 2024 @ 09:28:03.524

Figure 52 Blocking policy run successfully

Apr 22, 2024 @ 09:27:41.501	C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\powershell.exe created a new scripting file under Windows Temp or User data folder	9	92201	-
Apr 22, 2024 @ 09:27:41.485	Sysmon - Event 1: Process creation.	5	101101	Microsoft Corporation
Apr 22, 2024 @ 09:27:41.470	Sysmon - Event 1: Process creation.	5	101101	Microsoft Corporation
Apr 22, 2024 @ 09:27:41.459	Sysmon - Event 1: Process creation.	5	101101	Microsoft Corporation
Apr 22, 2024 @ 09:27:41.441	C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\PowerShell.exe created a new scripting file under Windows Temp or User data folder	9	92201	-
Apr 22, 2024 @ 09:27:27.147	Sysmon - Event 22: DNS Query.	5	101100	-
Apr 22, 2024 @ 09:27:08.816	Software protection service scheduled successfully.	3	60642	-

Figure 53 Here we can see the software protection service scheduled successfully

There the output of the end point user, what if they are try to open the google chrome what is show to there screen after 1 sec the bullon pop-up came and stop the software

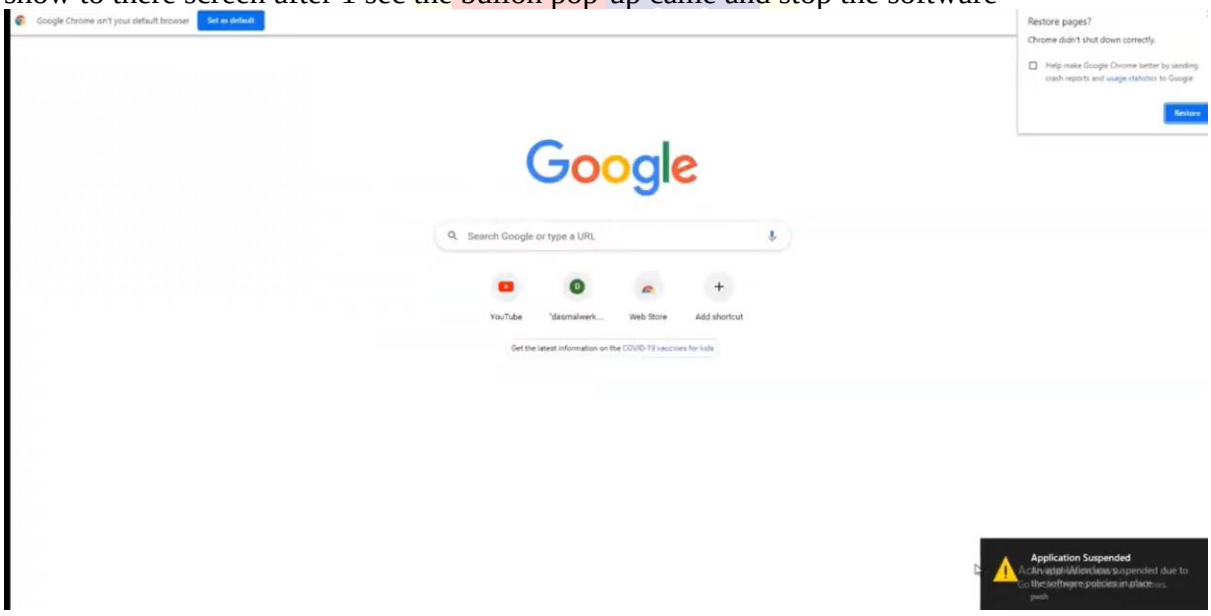


Figure 54 output of the end point user

Results and Benefits:

- **Enhanced Security:** Unauthorized software is promptly detected and blocked, reducing the risk of security incidents.
- **Real-time Response:** Integration with Wazuh and Sysmon ensures that actions are taken in real-time, preventing unauthorized software from running.

10.6Automated File Quarantine Using YARA Rules

Scope:
This study describes how to use YARA rules to construct a system that can identify and quarantine dangerous files that are uploaded to a web server. The system's goal is to improve the web server's security by locating and separating potentially dangerous files before they have a chance to do any harm.

Goal:
The main goal is to develop an automated system that uses YARA rules to check uploaded files to the web server for harmful content. A questionable file will be quarantined if it is found in order to stop it from running or propagating.

This project's goal is to safeguard the web server against malicious files that can cause data loss, disruptions in service, or security lapses. Through the analysis of file contents utilising YARA rules, the system can

This project's goal is to safeguard the web server against malicious files that can cause data loss, disruptions in service, or security lapses. The system can detect known patterns of harmful behaviour and quickly quarantine suspect files by utilising YARA rules to analyse file contents.

Importance:

Attackers frequently attempt to upload malicious files to web servers in order to take advantage of security holes. To ensure the web server's security and integrity, as well as the protection of the company's data and services, a strong detection and quarantine system must be put in place.

End users can upload files to many web apps. Modern web apps are convenient and frequently demand this feature, but in order to detect and prevent malicious file uploads, security teams must have the right security measures in place. An attack known as "malicious file uploading" is uploading files to a server or computer in a way that leaves a backdoor code unlocked, allowing the attacker to access the system later.

Protections like these can be implemented by developers:

- Limiting the types of files that are allowed;
- Verifying file types;
- Eliminating embedded risks;
- Authenticating users;
- Setting a maximum file length or size and more.

While these defense mechanisms are highly recommended, it is crucial for systems to scan uploaded files for malware and take action when malicious files are detected.

This post covers how we can take advantage of YARA, Wazuh, and Wazuh's Active Response capabilities to scan uploaded files to our web server with Yara rules and quarantine folders that have been flagged as malicious within seconds.

What is YARA?

YARA is a tool designed to assist malware researchers in identifying and categorising malware samples, among other things. Using textual or binary patterns, YARA allows you to generate descriptions of malware families—or anything else you wish to define. Every rule, also known as a description, is composed of a set of strings and a boolean expression that establishes its logic.

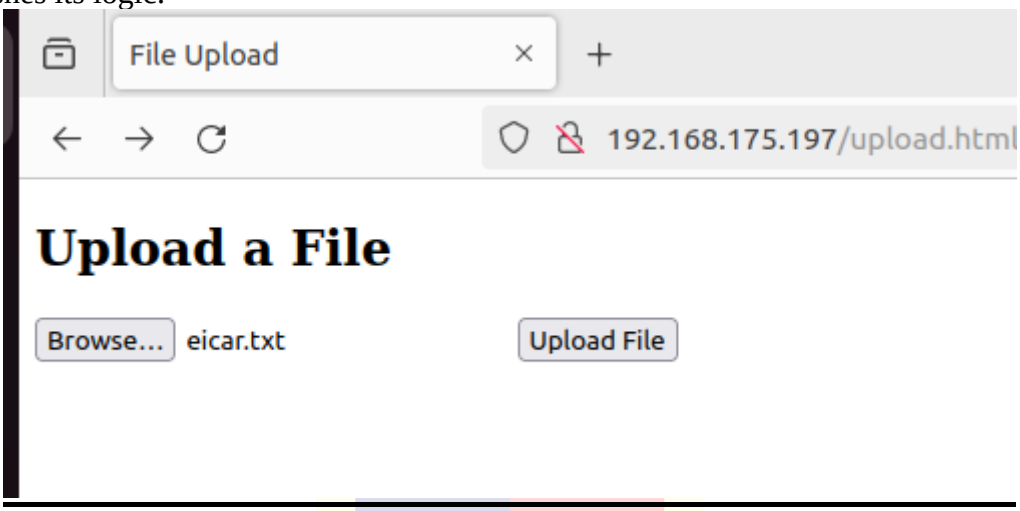


Figure 55 Baseic html page

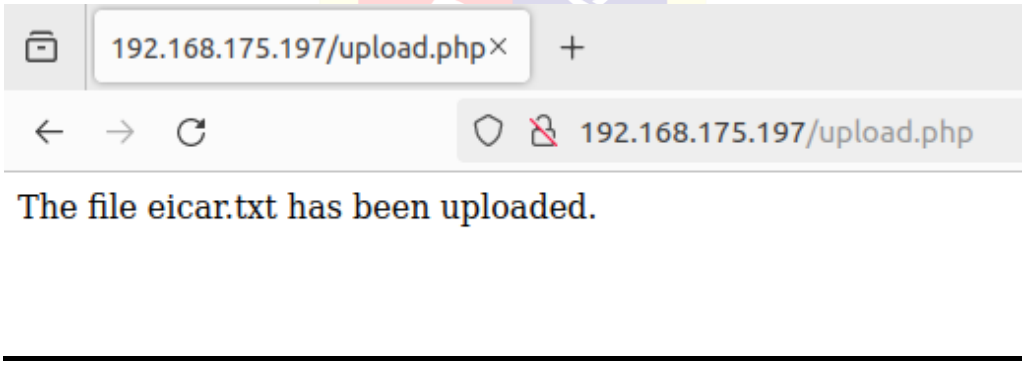


Figure 56 eicar file uploaded successfully

Output

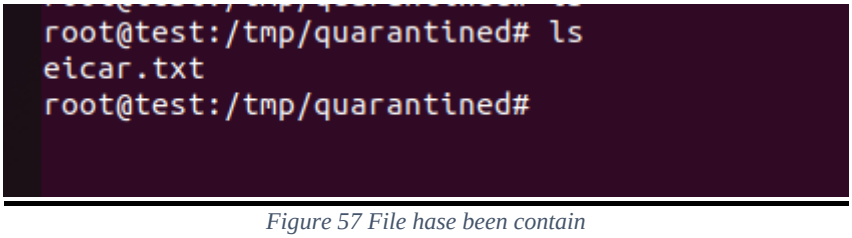


Figure 57 File hase been contain

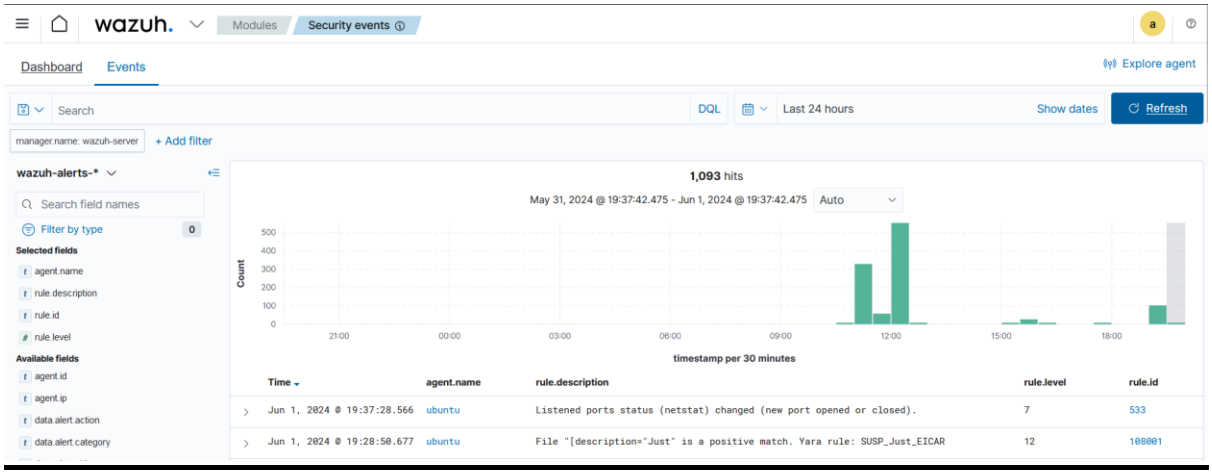


Figure 58 Here we got the all the logs

```
root@test:/var/www/html# ls -al
total 20
drwxr-xr-x 3 root     root     4096 Jun  1 19:25 .
drwxr-xr-x 3 root     root     4096 May 26 18:18 ..
drwxr-xr-x 2 www-data www-data 4096 Jun  1 19:28 upload
-rw-r--r-- 1 root     root      514 Jun  1 19:25 upload.html
-rw-r--r-- 1 root     root      479 Jun  1 19:25 upload.php
root@test:/var/www/html# S
```

Figure 59 webserver steup

```
<!--this for the yara rules-->
<directories realtime="yes">/tmp/yara/malware</directories>
<directories check_all="yes" realtime="yes">/var/www/html/upload</directories>
```

Figure 60 endpoint rules

```
root@test:/var/ossec/active-response/bin# ls -al
total 296
drwxr-x--- 2 root wazuh  4096 Jun  1 15:49 .
drwxr-x--- 3 root wazuh  4096 May  4 17:16 ..
-rwxr-x--- 1 root wazuh 21352 Apr 25 15:30 default-firewall-drop
-rwxr-x--- 1 root wazuh 19304 Apr 25 15:30 disable-account
-rwxr-x--- 1 root wazuh 19304 Apr 25 15:30 firewalld-drop
-rwxr-x--- 1 root wazuh 21352 Apr 25 15:30 firewall-drop
-rwxr-x--- 1 root wazuh 19328 Apr 25 15:30 host-deny
-rwxr-x--- 1 root wazuh 17360 Apr 25 15:30 ip-customblock
-rwxr-x--- 1 root wazuh 17992 Apr 25 15:30 ipfw
-rwxr-x--- 1 root wazuh 16840 Apr 25 15:30 kaspersky
-rwxr-x--- 1 root wazuh 14429 Apr 25 15:30 kaspersky.py
-rwxr-x--- 1 root wazuh 17744 Apr 25 15:30 npf
-rwxr-x--- 1 root wazuh 19312 Apr 25 15:30 pf
-rwxr-x--- 1 root wazuh   695 Apr 25 15:30 restart.sh
-rwxr-x--- 1 root wazuh 16360 Apr 25 15:30 restart-wazuh
-rwxr-x--- 1 root wazuh 17120 Apr 25 15:30 route-null
-rwxr-x--- 1 root wazuh 19272 Apr 25 15:30 wazuh-slack
-rwxr-x--- 1 root wazuh  1760 Jun  1 15:49 yara.sh
root@test:/var/ossec/active-response/bin# S
```

Figure 61 Yara script into the active response

```
#
#------------------------- Aadjust IFS to read files -------------------------#
SAVEIFS=$IFS
IFS=$(echo -en "\n\b")
# Static active response parameters
LOCAL=`dirname $0`
#------------------------- Folder where Yara rules (files) will be placed ------
git_repo_folder="/usr/local/signature-base"
yara_file_extenstions=( ".yar" )
yara_rules_list="/usr/local/signature-base/yara_rules_list.yar"

#------------------------- Main workflow -------------------------#

# Update Github Repo
cd $git_repo_folder
git pull https://github.com/Neo23x0/signature-base.git

# Remove .yar files not compatible with standard Yara package
rm $git_repo_folder/yara/generic_anomalies.yar $git_repo_folder/yara/general_clo

# Create File with rules to be compiled
if [ ! -f $yara_rules_list ]
then
    /usr/bin/touch $yara_rules_list
else rm $yara_rules_list
fi
for e in "${yara_file_extenstions[@]}"
do
  for f1 in $( find $git_repo_folder/yara -type f | grep -F $e ); do
    echo "include \"""$f1"\""" >> $yara_rules_list
  done
done
# Compile Yara Rules
/usr/share/yara/yara-4.2.3/yarac $yara_rules_list /usr/local/signature-base/yara
IFS=$SAVEIFS
```

Figure 62 File of yara.sh

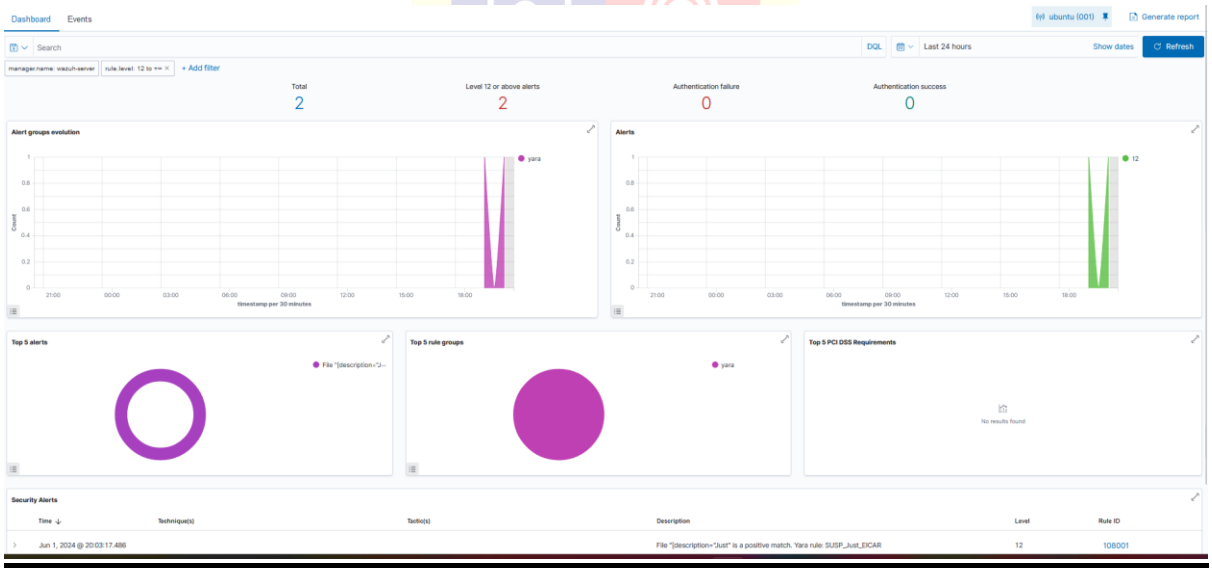


Figure 63 Dashboard

Outcomes and Advantages:

- ✓ Enhanced Security: By efficiently identifying and blocking malicious files, the system lowers the possibility of a web server hack.
- ✓ Automation: Reduces the need for human interaction by automating the file scanning and quarantining processes.
- ✓ Real-time Protection: Upon upload, suspicious files are instantly quarantined to offer real-time protection.

In conclusion, the web server's security is greatly increased by implementing a malicious file detection and quarantine system that makes use of YARA rules. Security breaches are avoided, and the web server's integrity is guaranteed, thanks to

the system's automatic identification and isolation of potentially dangerous files. Sustaining trust in web services and safeguarding organisational assets require a proactive strategy.

10.7 Streamlining Security Incident Response Through Telegram Integration and Real-Time Alerts

Objective

The goal is to create a real-time alarm system that uses Telegram to notify administrators when a password-breaking attack is detected on any system in the company via the telegram bot

Scope:

The system tracks network activity and login attempts in all systems used by the organisation, identifies any password-breaking assaults, and instantly notifies a specific Telegram channel with the IP address of the compromised device.

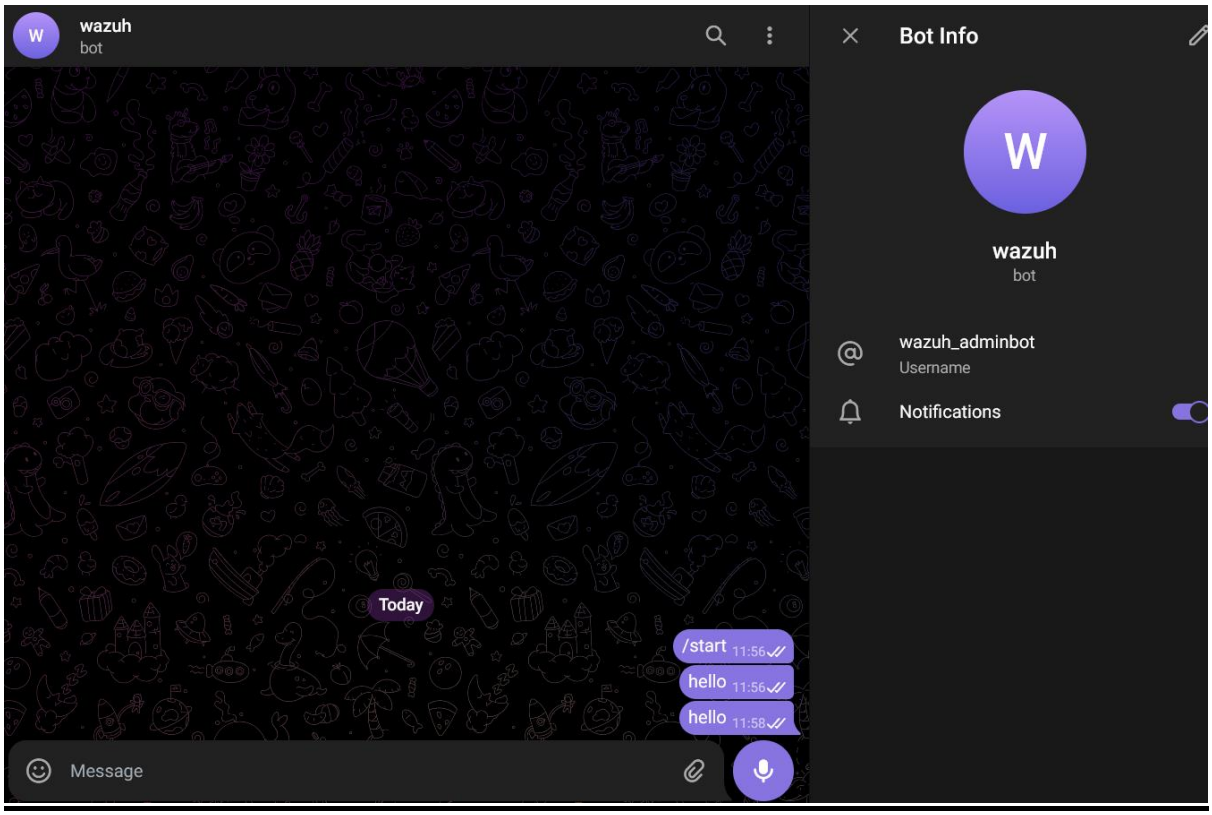


Figure 64 Bot starting

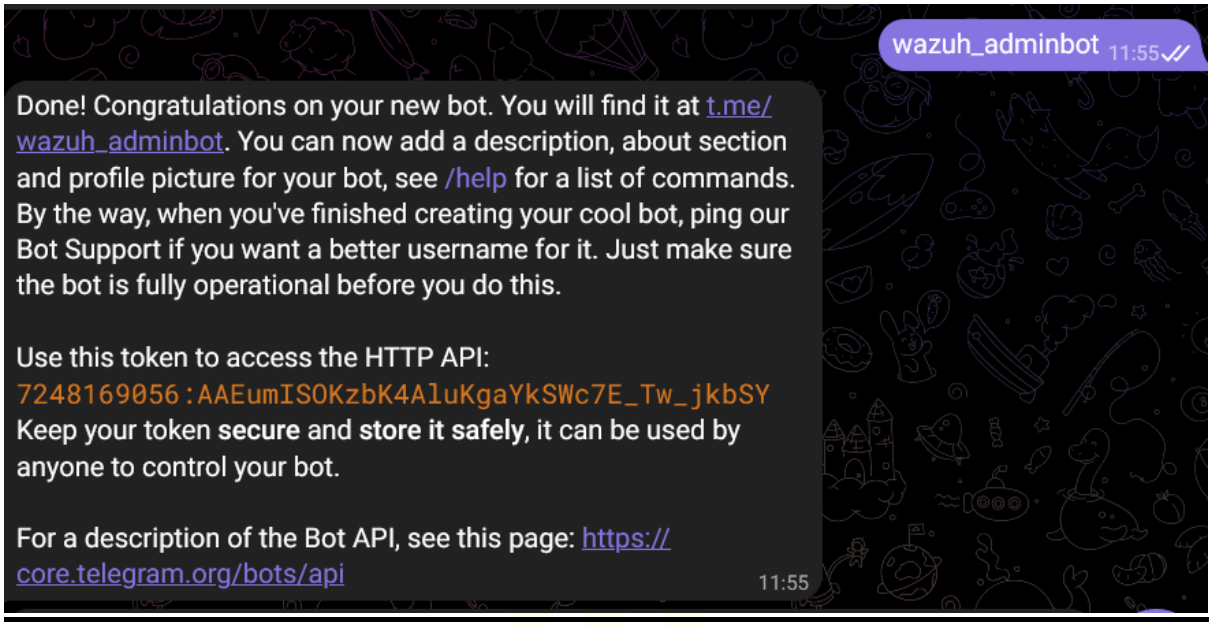


Figure 65 Telegram bot creation

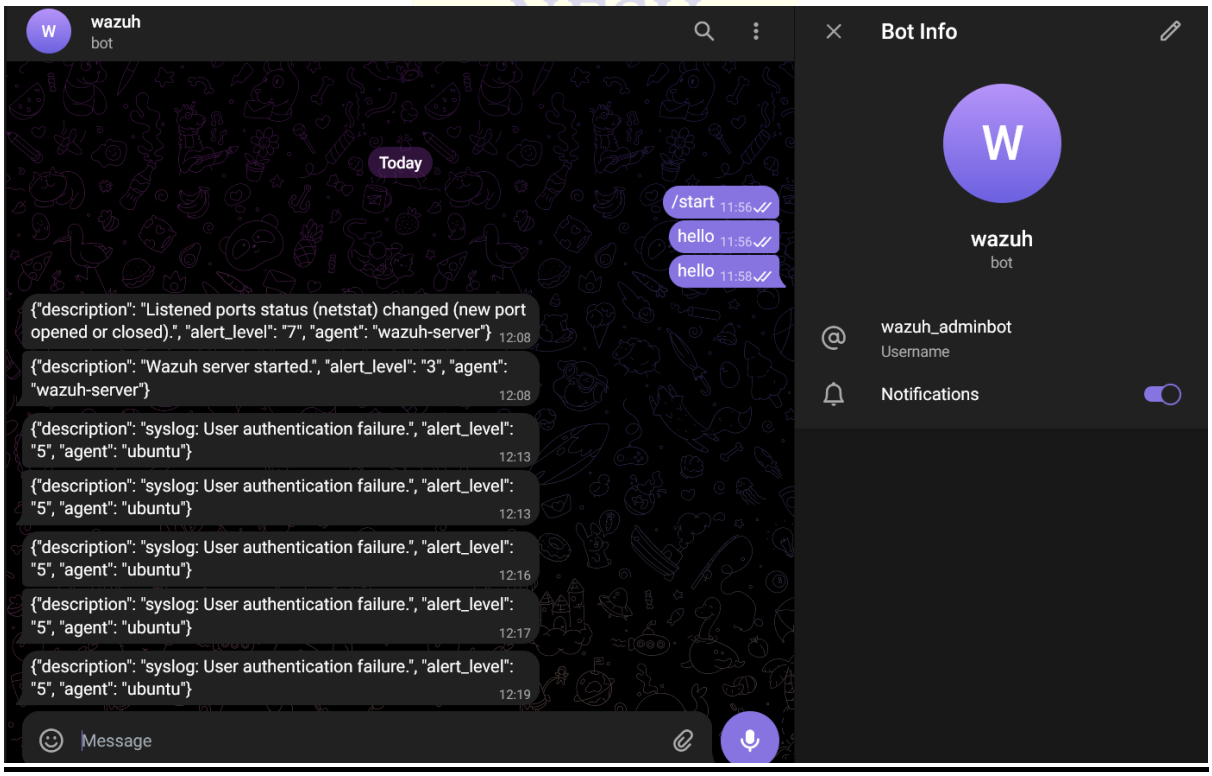


Figure 66 Message received from the Wazuh

```
<!-- Osquery integration -->

<integration>
  <name>custom-telegram</name>
  <rule_id>2501</rule_id>
  <level>5</level>
  <hook_url>https://api.telegram.org/bot7248169056:AAEumISOKzbK4AluKgaYkSWc7E_Tw_jkbSY/sendMessage</hook_url>
  <alert_format>json</alert_format>
</integration>
```

Figure 67 Wazuh rules for alert on telegram

Benefits:

- **Immediate Response:** Real-time alerts enable administrators to take swift action to mitigate potential threats.
- **Enhanced Security:** Continuous monitoring and prompt notifications help in quickly identifying and responding to password-breaking attacks.
- **Automation:** The system automates the detection and alerting process, ensuring consistent and reliable notifications.

Conclusion:

Putting in place a real-time alert system for password-breaking attacks greatly strengthens the organization's security stance. Administrators may respond swiftly to possible risks, lower the likelihood of successful attacks, and preserve the integrity of organisational systems by utilising Telegram's fast notifications feature.

10.8 Docker container security monitoring with Wazuh

Since its creation, Docker has gained popularity as a framework for deploying applications because of its advantages. It facilitates, for instance, the augmentation of applications' portability and operational robustness within organisations. An open-source tool called Docker is used to package apps into containers. Docker containers serve as runnable, lightweight, standalone instances of Docker images that separate the programme within from the operating system. Because of these advantages, a lot of businesses have embraced the technology that allows them to package software in standard units for shipping, deployment, and development in a timely manner.
The attack surface for organisations has grown due to the growing use of containerised software. Cyber threat actors now have another resource to aim their attacks against. For this reason, it is essential to keep an eye on containers at all times in order to fully see their execution environment and events.

• Track Docker events, including create, start, connect, mount, exec_start, detach, die, exec_create, exec_detach, and so on.
• Keep an eye on the use of CPU, memory, and network traffic in Docker containers.
• Recognise when memory and CPU use inside containers surpass predetermined limits.
• Keep an eye on the uptime and health of Docker containers.

Monitoring with Wazuh

Wazuh has the Docker listener and command monitoring modules that can be used to collect security and runtime events from Docker containers. The Docker listener module communicates with the Docker API to collect events related to Docker containers. The command monitoring module is used to monitor the output of specific commands and trigger alerts if they match a rule.

Configuring Ubuntu endpoints

1. Set up pip and Python:
install python3 with apt-get
2. To execute the containers, install Docker and the Python Docker Library:
curl -sSL | sh pip3 https://get.docker.com/ Put in place Docker 4.2.0.
3. Give the Wazuh agent permission to accept commands from the Wazuh server remotely. For security purposes, agents by default do not allow remote commands.
/var/ossec/etc/local_internal_options.conf => echo "logcollector.remote_commands=1"
4.To implement the aforementioned adjustments, restart the Wazuh agent:
wazuh-agent restart with systemctl

/var/ossec/etc/local_internal_options.conf => echo "logcollector.remote_commands=1"
4.To implement the aforementioned adjustments, restart the Wazuh agent:
wazuh-agent restart with systemctl

Server Setup

1. Form the container Wazuh agent group:
agent_groups -a -g container -q /var/ossec/bin
2. Use the following command to get each Wazuh agent's ID:
3. Assign the Wazuh agent that is hosting the Docker containers to the container group

(/var/ossec/bin/manage_agents -l). The group can have more than one agent assigned to it. By doing this, you can be confident that every agent in your environment running Docker containers gets the same configuration.
Substitute with the endpoint's agent ID that is hosting the Docker container.
Agent groups -a -i /var/ossec/bin -g holding -q
4. Modify the /var/ossec/etc/shared/container/agent.conf configuration file using the following parameters. In order to acquire Docker container information, this activates the Docker listener module and sets the commands to run on the monitored endpoint.

```
<agent_config>
  <!-- Configuration to enable Docker listener module. -->
  <wodle name="docker-listener">
    <interval>10m</interval>
    <attempts>5</attempts>
    <run_on_start>yes</run_on_start>
    <disabled>no</disabled>
  </wodle>

  <!-- Command to extract container resources information. -->
  <localfile>
    <log_format>command</log_format>
    <command>docker stats --format "{{.Container}} {{.Name}} {{.CPUPerc}} {{.MemUsag
    <alias>docker container stats</alias>
    <frequency>120</frequency>
    <out_format>$(timestamp) $(hostname) docker-container-resource: $(log)</out_form
  </localfile>

  <!-- Command to extract container health information. -->
  <localfile>
    <log_format>command</log_format>
    <command>docker ps --format "{{.Image}} {{.Names}} {{.Status}}"</command>
    <alias>docker container ps</alias>
    <frequency>120</frequency>
    <out_format>$(timestamp) $(hostname) docker-container-health: $(log)</out_format
```

Figure 68 server configuration

5. To decode the logs you received from the Wazuh agent, create a decoders file called docker_decoders.xml in the /var/ossec/etc/decoders/ directory and add the following decoders:

```
<!-- Decoder for container resources information. -->
<decoder name="docker-container-resource">
  <program_name>^docker-container-resource</program_name>
</decoder>

<decoder name="docker-container-resource-child">
  <parent>docker-container-resource</parent>
  <prematch>ossec: output: 'docker container stats':</prematch>
  <regex>(\S+) (\S+) (\S+) (\S+) / (\S+) (\S+) (\S+) / (\S+)</regex>
  <order>container_id, container_name, container_cpu_usage, container_memory_
</decoder>

<!-- Decoder for container health information. -->
<decoder name="docker-container-health">
  <program_name>^docker-container-health</program_name>
</decoder>

<decoder name="docker-container-health-child">
  <parent>docker-container-health</parent>
  <prematch>ossec: output: 'docker container ps':</prematch>
  <regex offset="after_prematch" type="pcre2">(\S+) (\S+) (.*?) \((.*?)\)</re
  <order>container_image, container_name, container_uptime, container_health_
</decoder>
```

Figure 69 wazuh decoder file

1. To alert the container information, create a rules file called docker_rules.xml in the /var/ossec/etc/rules/ directory and add the following rules:

```
<group name="container,">
  <!-- Rule for container resources information. -->
  <rule id="100100" level="5">
    <decoded_as>docker-container-resource</decoded_as>
    <description>Docker: Container $(container_name) Resources</description>
    <group>container_resource,</group>
  </rule>

  <!-- Rule to trigger when container CPU and memory usage are above 80%. -->
  <rule id="100101" level="12">
    <if_sid>100100</if_sid>
    <field name="container_cpu_usage" type="pcre2">^(0*[8-9]\d|0*[1-9]\d{2,})</fi
    <field name="container_memory_perc" type="pcre2">^(0*[8-9]\d|0*[1-9]\d{2,})</
    <description>Docker: Container $(container_name) CPU usage ($(container_cpu_u
    <group>container_resource,</group>
  </rule>

  <!-- Rule to trigger when container CPU usage is above 80%. -->
  <rule id="100102" level="12">
    <if_sid>100100</if_sid>
    <field name="container_cpu_usage" type="pcre2">^(0*[8-9]\d|0*[1-9]\d{2,})</fi
    <description>Docker: Container $(container_name) CPU usage ($(container_cpu_u
    <group>container_resource,</group>
  </rule>

  <!-- Rule to trigger when container memory usage is above 80%. -->
  <rule id="100103" level="12">
    <if_sid>100100</if_sid>
    <field name="container_memory_perc" type="pcre2">^(0*[8-9]\d|0*[1-9]\d{2,})</
    <description>Docker: Container $(container_name) memory usage ($(container_me
    <group>container_resource,</group>
  </rule>
```

Testing the configuration (attack command)

To check for excessive CPU and memory use, we use the stress-ng utility programme. Run this test on one of the containers—the nginx-container, for example.

• To install the stress-ng utility and access the container shell, run the following commands:

nginx-container /bin/bash docker exec -it apt update && apt install stress-ng -y

• When both CPU and memory utilisation surpass 80%, run the following command to cause a high-level alert. For three minutes, the command is active.

stress-ng -l 80 -vm -c 1 1--vm-bytes 500m - t 3m

• When memory use hits 80%, run the following command to set off a high-level alarm. For three minutes, the command is active.

-vm 1 --vm-bytes stress-ng 500m - t 3m

• When CPU utilisation reaches 80%, use the following command to initiate a high-level alert. For three minutes, the command is active.

stress-ng -t 3m -c 1 -l 80

• The existence of the configuration file /etc/nginx/nginx.conf is confirmed by the nginx-container health check. To set off a high-level alarm in the event that the container becomes unhealthy, delete the configuration file while in the container shell:

rm nginx.conf in /etc/nginx

```
root@326721ca82b8:/# stress-ng -c 1 -l 80 -vm 1 --vm-bytes 500m -t 3m
stress-ng: debug: [464] invoked with 'stress-ng -c 1 -l 80 -vm 1 --vm-bytes 500m -t 3m' by user 0 'root'
stress-ng: debug: [464] stress-ng 0.15.06
stress-ng: debug: [464] system: Linux 326721ca82b8 6.5.0-35-generic #35~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Tue May  7 09:00:52 UTC 2 x86_64
stress-ng: debug: [464] RAM total: 7.6G, RAM free: 3.5G, swap free: 2.0G
stress-ng: debug: [464] temporary file path: '.', filesystem type: overlayfs
stress-ng: debug: [464] 3 processors online, 3 processors configured
stress-ng: debug: [464] main: can't set oom_score_adj
stress-ng: info:  [464] setting to a 180 second (3 mins, 0.00 secs) run per stressor
stress-ng: info:  [464] dispatching hogs: 1 cpu, 1 vm
stress-ng: debug: [464] cache allocate: shared cache buffer size: 24576K
stress-ng: debug: [464] starting stressors
stress-ng: debug: [465] cpu: can't set oom_score_adj
stress-ng: debug: [465] cpu: started [465] (instance 0)
stress-ng: debug: [464] 2 stressors started
stress-ng: debug: [466] vm: can't set oom_score_adj
stress-ng: debug: [466] vm: started [466] (instance 0)
stress-ng: debug: [466] vm: using method 'all'
stress-ng: debug: [465] cpu: using method 'all'
stress-ng: info:  [465] stressor terminated with unexpected signal signal 4 'SIGILL'
stress-ng: debug: [464] process [465] terminated
stress-ng: info:  [467] stressor terminated with unexpected signal signal 4 'SIGILL'
stress-ng: debug: [466] vm: exited [466] (instance 0)
stress-ng: debug: [464] process [466] terminated
stress-ng: metrc: [464] stressor       bogo ops real time  usr time  sys time   bogo ops/s     bogo ops/s CPU used per       RSS Max
stress-ng: metrc: [464]                          (secs)    (secs)    (secs)   (real time) (usr+sys time) instance (%)         (KB)
stress-ng: metrc: [464] cpu                   2      0.00      0.00      0.00         0.00           0.00          0.00            0
stress-ng: metrc: [464] vm               633994     10.45      5.75      4.36     60642.02       62705.61         96.71       514148
stress-ng: fail:  [464] cpu instance 0 corrupted bogo-ops counter, 2 vs 0
```

Figure 70 make temp stress on the container

Alert visualization (custom Dashboard)

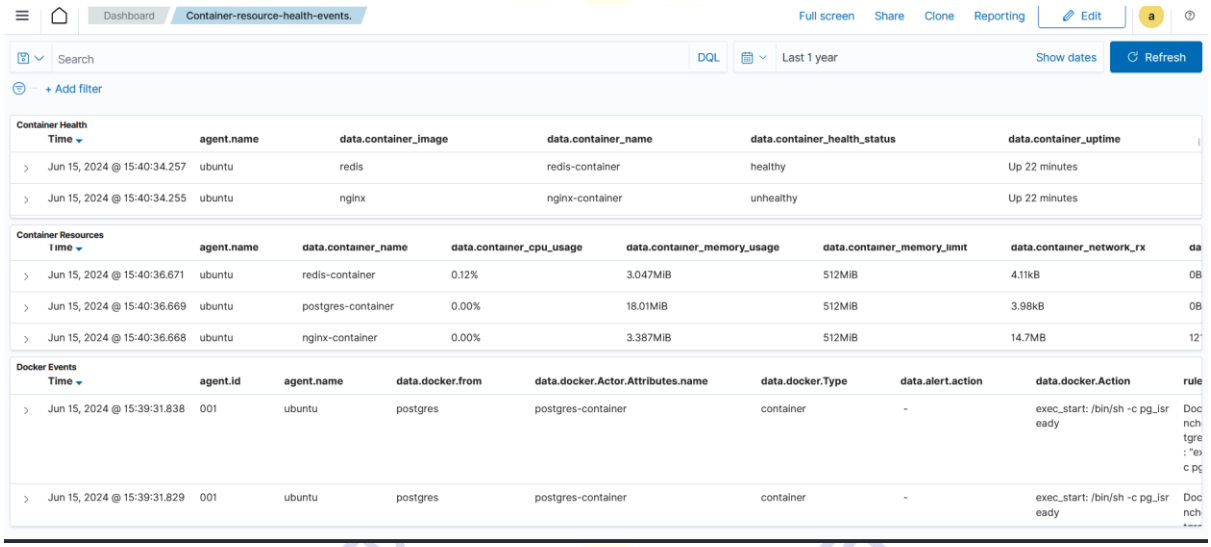


Figure 71 custom Dashboard

Jun 15, 2024 @ 16:22:49.161	Docker Container: Successful Web GET request on Container: docker/webserver from 172.17.0.1	5	100009	Discovery
Jun 15, 2024 @ 16:22:49.151	Docker Container: Successful Web GET request on Container: docker/webserver from 172.17.0.1	5	100009	Discovery
Jun 15, 2024 @ 16:22:47.155	Docker Container: Successful Web GET request on Container: docker/webserver from 172.17.0.1	5	100009	Discovery
Jun 15, 2024 @ 16:22:47.152	Docker Container: Successful Web GET request on Container: docker/webserver from 172.17.0.1	5	100009	Discovery
Jun 15, 2024 @ 16:22:45.203	Docker Container: Successful Web GET request on Container: docker/webserver from 172.17.0.1	5	100009	Discovery
Jun 15, 2024 @ 16:22:45.150	Docker Container: Successful Web GET request on Container: docker/webserver from 172.17.0.1	5	100009	Discovery

Figure 72 Monitoring Docker container logs

Jun 15, 2024 @ 16:23:45.875	Docker: Command launched in container postgres-container. Action: "exec_start: /bin/sh -c pg_isready"	3	87907	-
Jun 15, 2024 @ 16:23:45.867	Docker: Command launched in container redis-container. Action: "exec_start: redis-cli --raw incr ping"	3	87907	-
Jun 15, 2024 @ 16:23:25.510	Docker: Container nginx-container received the action: die	7	87924	-
Jun 15, 2024 @ 16:23:25.473	Docker: Network container_env_network disconnected	4	87929	-

Figure 73 Monitoring user interaction with Docker resources

```
root@test:/home/ubuntu/Desktop# docker run -d --name test-container httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
2cc3ae149d28: Pull complete
840d8df643b2: Pull complete
4f4fb700ef54: Pull complete
9d1465828338: Pull complete
4a16a983b278: Pull complete
9129890c4c50: Pull complete
Digest: sha256:10182d88d7fbc5161ae0f6f758cba7adc56d4aae2dc950e51d72c0cf68967cea
Status: Downloaded newer image for httpd:latest
949d141d98017ff7ca1b576d543ce01d878c0748e5660d07beaad328b8315fa5
root@test:/home/ubuntu/Desktop# docker pause test-container
test-container
root@test:/home/ubuntu/Desktop# docker unpause test-container
test-container
root@test:/home/ubuntu/Desktop# docker stop test-container
test-container
root@test:/home/ubuntu/Desktop# docker rm test-container
test-container
```

Figure 74 docker activity of push start

›	Jun 15, 2024 @ 19:17:56.860	ubuntu	Docker: Container test-container Resources	5	100100
›	Jun 15, 2024 @ 19:17:55.818	ubuntu	Docker: Container test-container is Paused	5	100105
›	Jun 15, 2024 @ 19:15:57.170	ubuntu	Docker: Container test-container Resources	5	100100
›	Jun 15, 2024 @ 19:15:55.658	ubuntu	Docker: Container test-container is Paused	5	100105

Figure 75 log of all docker activity

Conclusion

Maintaining a safe and effective environment is made possible by the high visibility of containers in Dockerized settings. In this manner, companies may minimise disruptions and promptly detect and address problems. Wazuh makes it simple to examine the health of containers, identify irregularities in their use of resources, and more.

10.9 Real-Time Vulnerability Management for Containers

Protecting containers and the infrastructure that supports them can be achieved by doing container vulnerability scans. Applications can run in isolated settings with containers, which preserve platform consistency. One of the most important proactive security practices is using container vulnerability scanning to identify and fix security risks within containers. Given the dynamic nature of software and the corresponding security threats, this is crucial in preventing bad actors from taking advantage of it.

A software security platform called Snyk assists developers and organisations in identifying and resolving vulnerabilities in their apps and open source dependencies. To fix container vulnerabilities, it features a Command-Line Interface (CLI). In order to find potential security problems, it ensures a rigorous evaluation of container images, libraries, and dependencies through the use of extensive container vulnerability scanning. Container security is strengthened in a variety of contexts by this proactive security approach.

Organisations can greatly reduce the risk of security breaches and improve the overall security posture of their containerised environments by implementing a strong strategy for container vulnerability assessment by combining the capabilities of the Snyk CLI and Wazuh.

Authenticating the Snyk CLI with your Snyk account

This section provides step by step instructions for authenticating the Snyk CLI with your Snyk account on the Ubuntu endpoint.

1. Login to your Snyk account.

2. Select the **Account settings** option located in the dropdown menu at the bottom of the left navigation menu on the Web UI.

3. Click the **click to show** box and copy your API token in the KEY box.

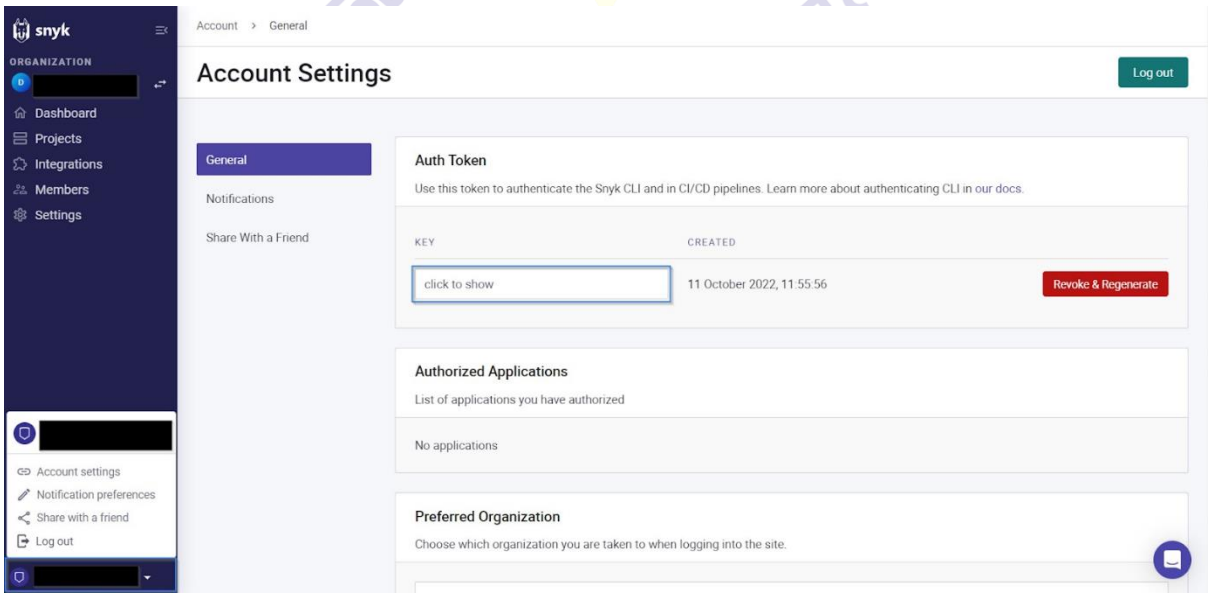


Figure 76 Synk Web

4. Run the following command to log in to your Snyk account and authenticate the CLI:

snyk auth <TOKEN>

Vulnerability scanning script

We configured a Bash script to look through the Ubuntu endpoint's containers. Furthermore, we set up the Wazuh command module to run the Snyk script on the agent host on a regular basis. To set up command monitoring on the container host, follow the instructions below.

1. Create and switch into a custom script directory `/var/ossec/custom-script/` using the following command:

sudo mkdir /var/ossec/custom-script/ && cd $_

2. Create a new file `snyk-vulns.sh`:

sudo touch ./snyk-vulns.sh

3. Copy and paste the code below to the `/var/ossec/custom-script/snyk-vulns.sh` file. This script lists all running containers and iterates through them to perform scans using the Snyk CLI binary. Subsequently, it writes the scan results to the `/var/ossec/logs/container_vulnerability_scan.log` file:

Snyk rules configuration

In order to record the outcome of the Snyk scan on the Ubuntu endpoint under observation, we establish rules in the Wazuh server.
1. Use the following command to create a rule file in the Wazuh server's /var/ossec/etc/rules/ directory:
nanoSnyk Vulnerability Rules /var/ossec/etc/rules.xml
2. Modify the /var/ossec/etc/rules/snyk-vulns_rules.xml file with the following custom rules:

```
<group name="containers, snyk-vulnerabilities,">


<!-- This rule detects the json output of snyk vulnerability scan.-->
  <rule id="111160" level="0">
    <decoded_as>json</decoded_as>
    <field name="packageName">\.+</field>
    <field name="cvssScore">\.+</field>
    <field name="dockerBaseImage">\.+</field>
    <description>Snyk: Alert</description>
  </rule>

<!-- This rule detects the critical severity vulnerability for the Snyk scan.-->
  <rule id="111161" level="14">
    <if_sid>111160</if_sid>
    <field name="severity">critical</field>
    <description>Snyk Critical Alert: Vulnerable Package $(packageName) $(version) D
  </rule>

<!-- This rule detects the high severity vulnerability for the Snyk scan.-->
  <rule id="111162" level="12">
    <if_sid>111160</if_sid>
    <field name="severity">high</field>
    <description>Snyk High Alert: Vulnerable Package $(packageName) $(version) Detec
  </rule>

<!-- This rule detects the medium severity vulnerability for the Snyk scan.-->
  <rule id="111163" level="10">
    <if_sid>111160</if_sid>
    <field name="severity">medium</field>
    <description>Snyk Medium Alert: Vulnerable Package $(packageName) $(version) Det
  </rule>
```

Figure 77 wazuh Local rules

Where:

- Rule ID `111160` detects the JSON output of the Snyk vulnerability scan.
- Rule ID `111161` detects the critical severity vulnerabilities from the Snyk scan.
- Rule ID `111162` detects the high severity vulnerabilities from the Snyk scan.
- Rule ID `111163` detects the medium severity vulnerabilities from the Snyk scan.
- Rule ID `111164` detects the low severity vulnerabilities from the Snyk scan.

3. Save and close the `snyk-vulns_rules.xml` rule file.

Snyk CLI installation

We download the Snyk CLI binary for Linux, change its permissions, and move it to a user-friendly location.
To install the Snyk CLI, follow these steps:
1. Use the filename snyk to save the Linux version of the Snyk CLI binary that you downloaded: curl --compressed https://static.snyk.io/cli/latest/snyk-linux -o snyk
2. Give the snyk binary the executable permission for every class by adding chmod +x./snyk.
3. Use sudo mv./snyk /usr/local/bin/ to move the snyk binary to the /usr/local/bin/ directory so that any user can access it. Authenticating the Snyk CLI with your Snyk account

Container vulnerability scanning results

Below are the images of the alerts generated on the Wazuh dashboard when Snyk discovers a vulnerability on the containers present on an Ubuntu endpoint for vulnerabilities.

Critical severity alerts

1. Navigate to the **Security events** tab and input the Dashboards Query Language (DQL) `rule.id:111161` in the HTML form. Next, click the update/refresh button to view the **`critical severity`** alerts:

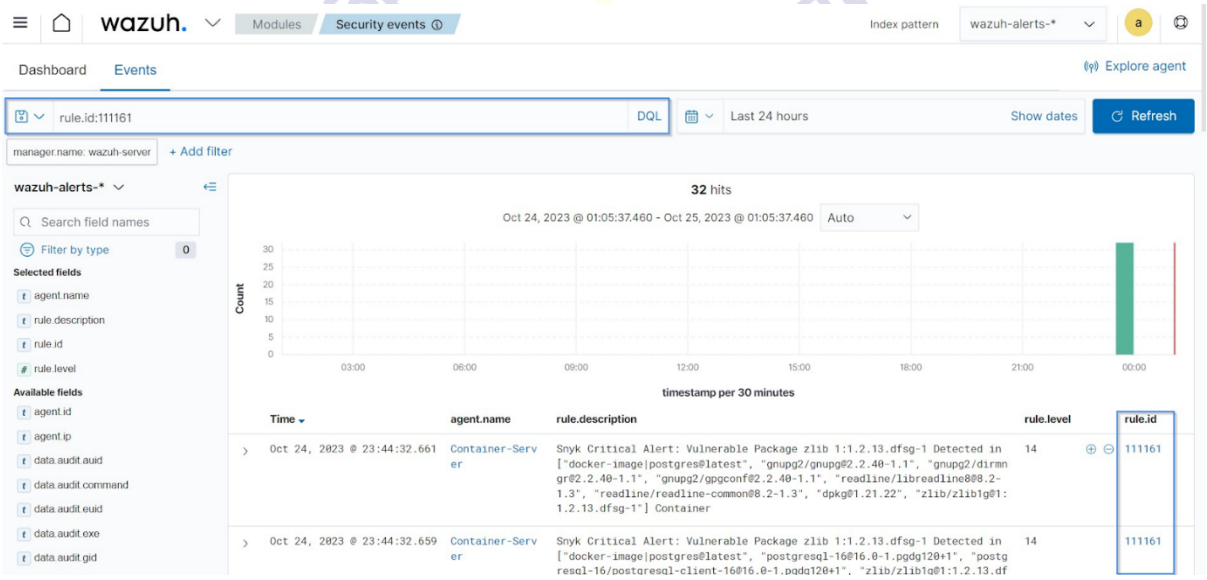


Figure 78 Container vulnerability scanning results (1)

High severity alerts

1. Navigate to the **Security events** tab and input the Dashboards Query Language (DQL) `rule.id:111162` in the HTML form. Next, click the update/refresh button to view the **`High severity`** alerts:

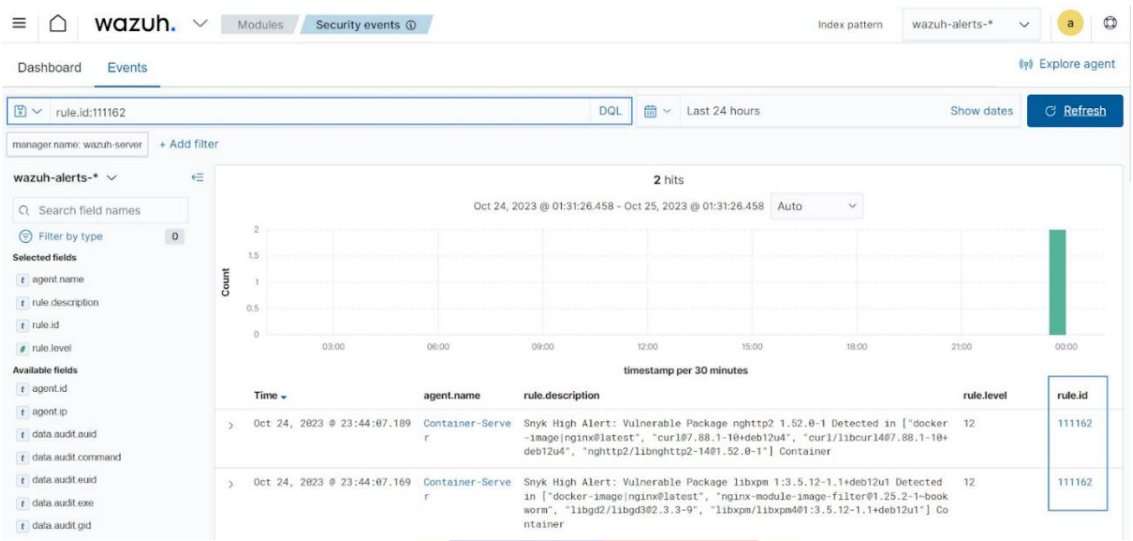


Figure 79 Container vulnerability scanning results (2)

Medium severity scans alerts

1. Navigate to the **Security events** tab and input the Dashboards Query Language (DQL) `rule.id:111163` in the HTML form. Next, click the update/refresh button to view the **`Medium severity`** alerts:

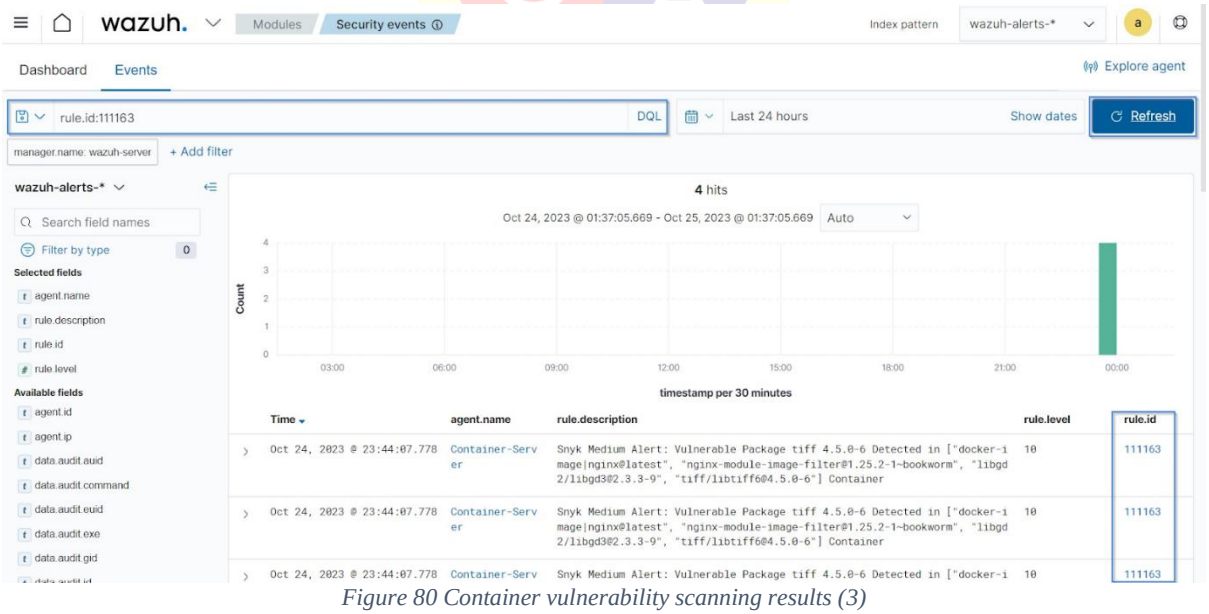


Figure 80 Container vulnerability scanning results (3)

Low severity alerts

1. Navigate to the **Security events** tab and input the Dashboards Query Language (DQL) `rule.id:111161` in the HTML form. Next, click the refresh button to view the **`Low severity`** alerts:

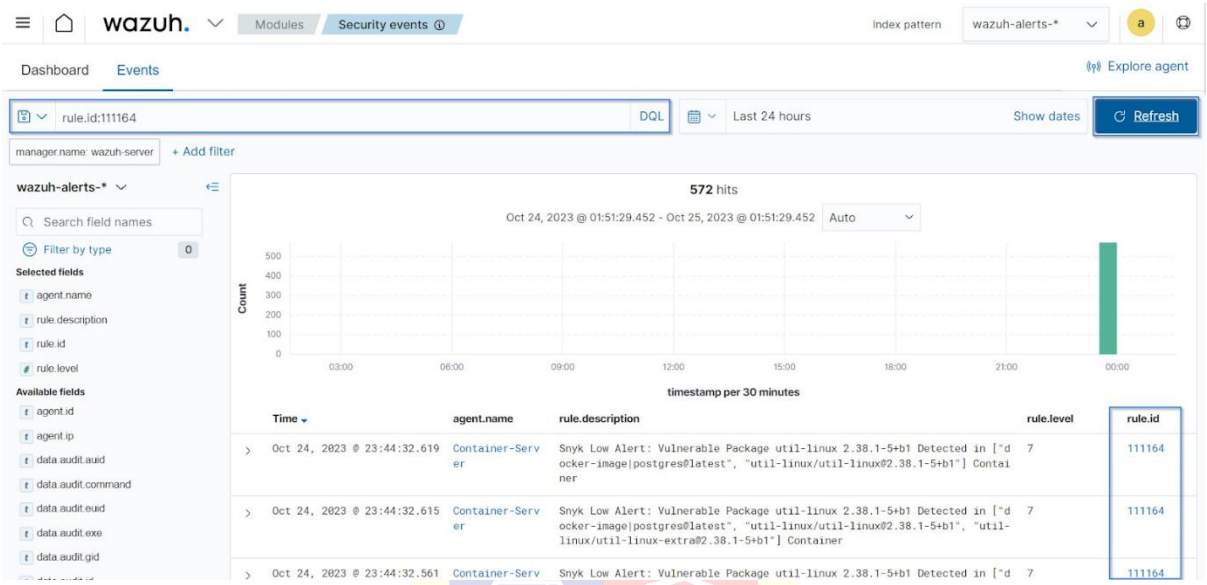


Figure 81 Container vulnerability scanning results (4)

Conclusion

In this blog post, we showcase how integrating Snyk with Wazuh enables organizations to conduct vulnerability scans within their container environments. This integration allows organizations to enhance their container security practices by promptly identifying and remediating vulnerabilities.

10.10 User Command Activity Monitoring in Linux

Objective:

To implement a robust monitoring system that tracks every command executed by users on a Linux system and logs this activity into Wazuh for comprehensive visibility and auditing.

Purpose:

The purpose of this implementation is to enhance security by closely monitoring user activities on the command prompt. By logging all commands to Wazuh, we can track and analyse user behaviour, detect unauthorized actions, and maintain an audit trail for compliance and forensic investigations.

Configuring:-

To install the auditd utility by the apt installer and it`s up and run

```
ubuntu@ubuntu:~$ sudo apt install auditd
[sudo] password for ubuntu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
auditd is already the newest version (1:3.0.7-1build1).
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
ubuntu@ubuntu:~$ S
```

Figure 82 Auditd installations

To forwarded the log into the Wazuh make the agent config file update by pointing the auditd logs

```
<localfile>
<log_format>audit</log_format>
<location>/var/log/audit/audit.log</location>
</localfile>
```

Figure 83 Endpoint Configuration

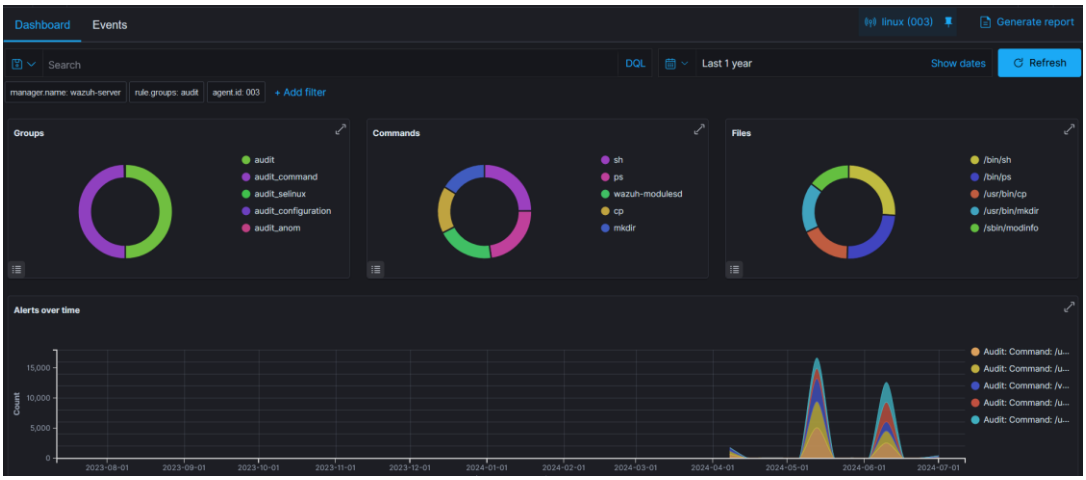


Figure 84Audit dasboard

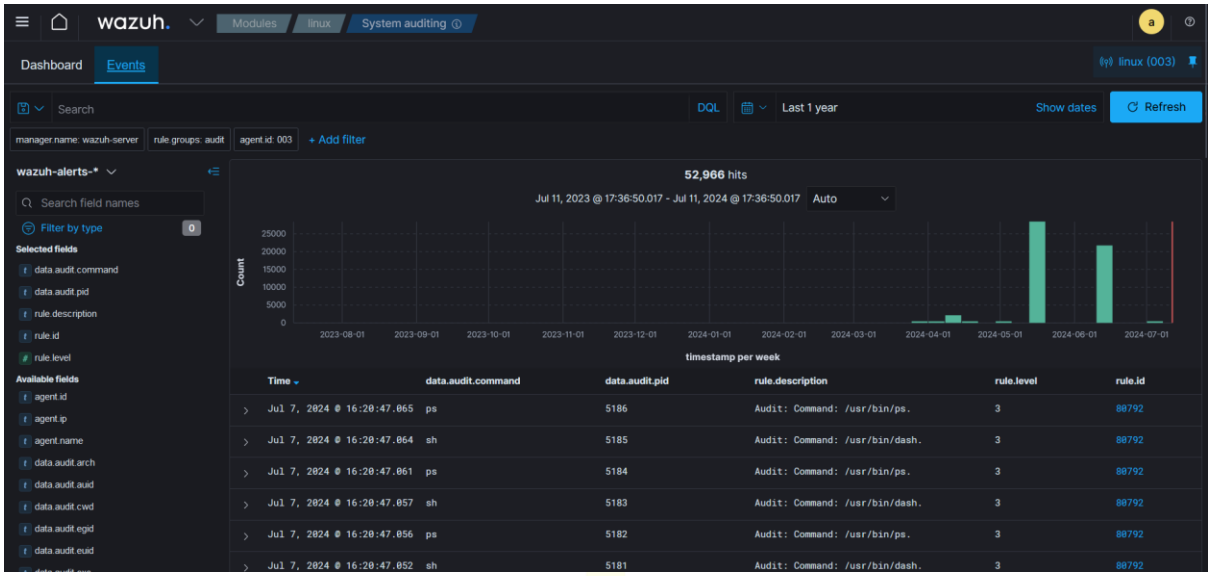


Figure 85 Details audit dashboard

At which location, what time, what command, which user can executed a command we can track down over here

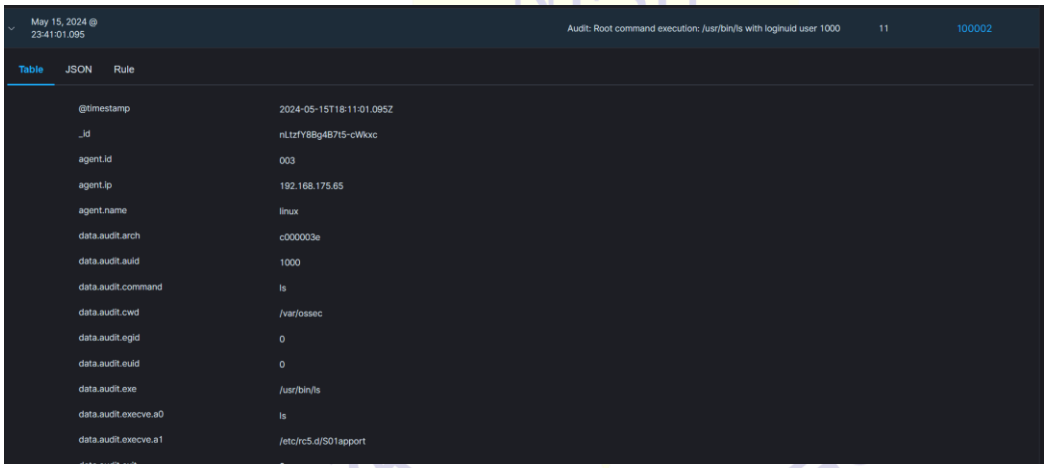


Figure 86 Details Log

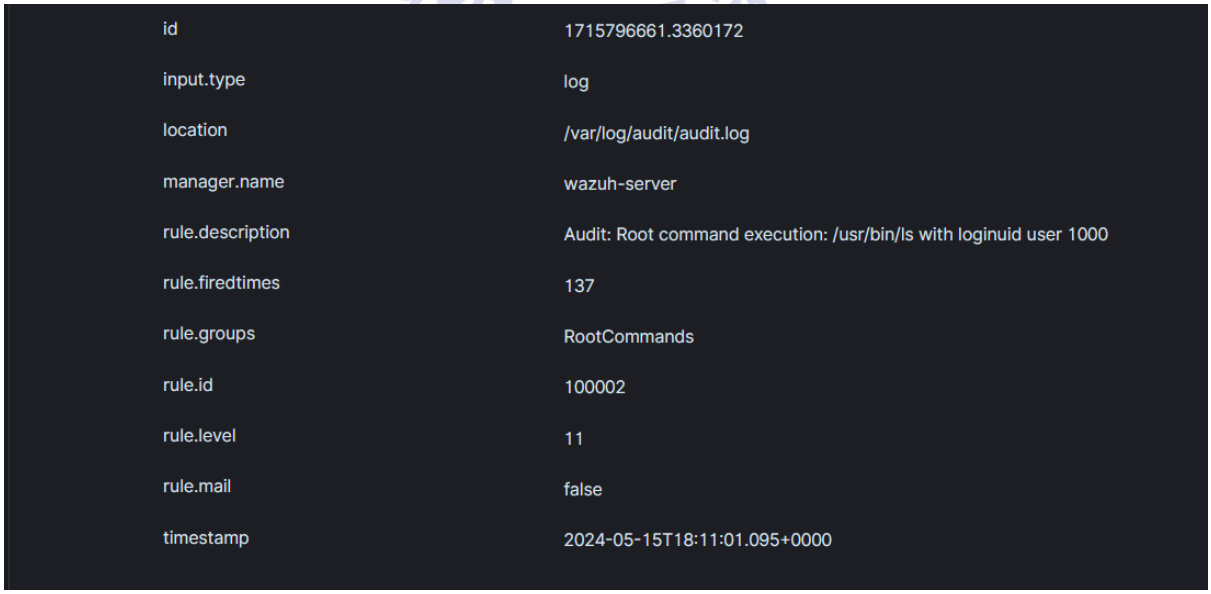


Figure 87 location of command executed

Benefits:

- **Enhanced Security:** Real-time visibility into user actions helps detect unauthorized or suspicious activities promptly.
- **Audit and Compliance:** Maintain a detailed audit trail of user activities, ensuring compliance with security policies and regulatory requirements.
- **Forensic Analysis:** Easily trace and investigate user actions during security incidents, helping to identify the root cause and mitigate risks.

Conclusion:

Integrating `auditd` with Wazuh for monitoring user activity on Linux provides a powerful solution for enhancing security and visibility. By capturing every command executed by users and analyzing these logs in Wazuh, organizations can proactively detect and respond to potential security threats, maintain compliance, and conduct thorough forensic investigations when needed.

10.11 Detecting web attacks using Wazuh and teler

Setting up the lab

Wazuh is an open-source, free unified XDR and SIEM platform that can be highly customised to meet the demands of any kind of organisation. Wazuh provides use cases such as automatic threat response and security monitoring. This blog post will concentrate on detecting web attacks. We can identify common online attacks with Wazuh. We integrate teler, a lightweight HTTP IDS, to further improve Wazuh's web detection capabilities. Web exploits such as upload assaults, directory traversal, and HTML injection attacks can be found using Wazuh and Teler.
In our configuration, we set up teler and DVWA (Damn Vulnerable Web Application) on an Ubuntu 20.04 endpoint. We then configure Wazuh to ingest and analyze teler logs.

Configuring teler on Ubuntu

This section includes steps on how to configure teler on the Ubuntu endpoint hosting DVWA (Damn Vulnerable Web Application). Download and extract the teler binary on the Ubuntu endpoint:

On the Ubuntu endpoint, download and unpack the teler binary:
To watch the output on the Ubuntu endpoint, add the configuration block below to the /var/ossec/etc/ossec.conf Wazuh agent configuration file.log file that Teler produced:
Turn the Wazuh agent back on:
systemctl restart wazuh-agent

Run the command below on the Ubuntu endpoint to start the teler application:

tail -f /var/log/apache2/access.log | ./teler -c /path/to/teler.yaml

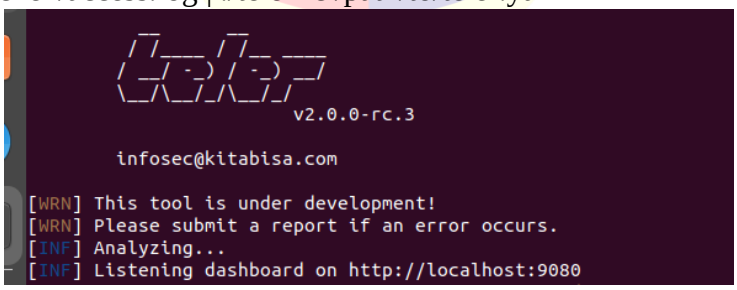


Figure 88 teler start

Configuring rules on Wazuh

1. On the Wazuh server, we add the custom rules below to the `/var/ossec/etc/rules/local_rules.xml` file. The Wazuh rules below work with the default pre-configured rules defined in the `teler.yaml` file:

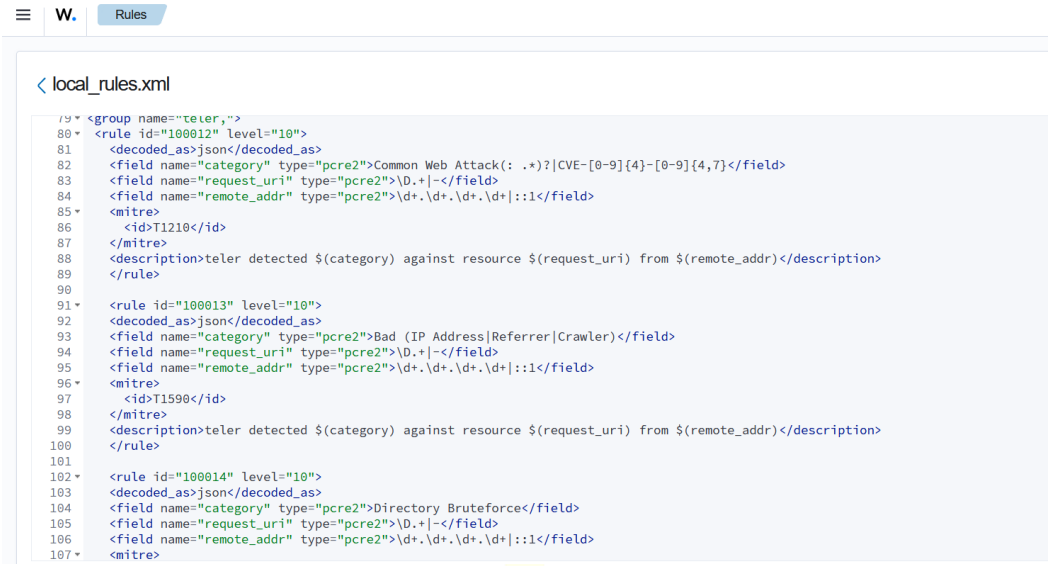


Figure 89 loacl rules for teler

Attack emulation

From the Kali Linux endpoint, use Nikto to launch an attack against the vulnerable web application:

nikto -h http://<UBUNTU_IP>/dvwa/

Where `-h` specifies the target URL.

After completing the steps above, teler will analyze the web server logs based on its default pre-configured rules, generating alerts on the Wazuh dashboard. The screenshot below shows alerts on the Wazuh dashboard after the attack emulation.

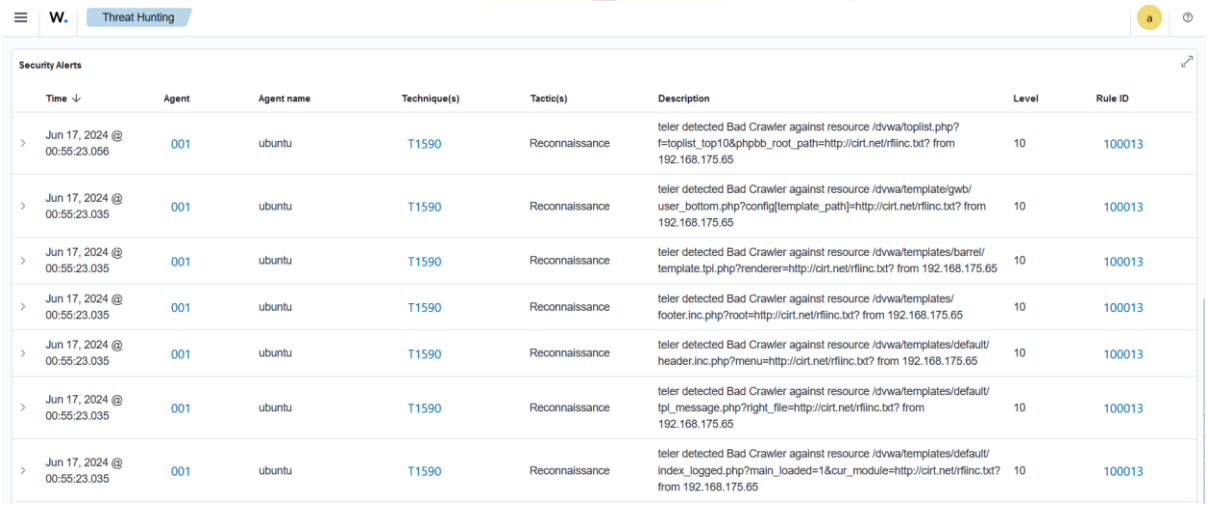


Figure 90 web attack logs

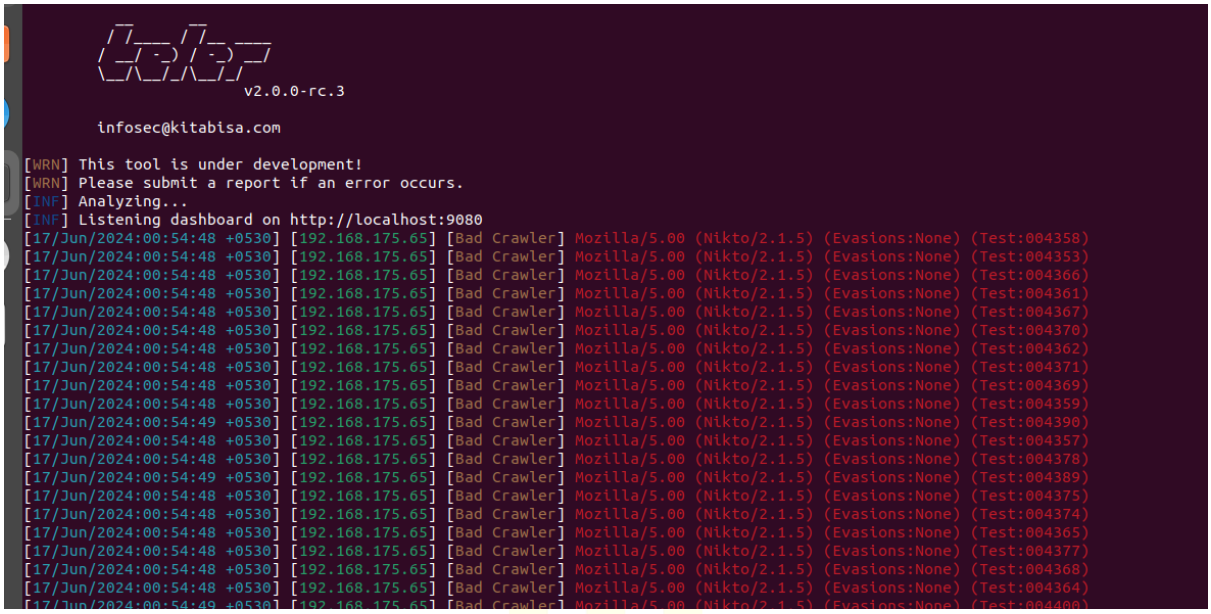


Figure 91 teler logs

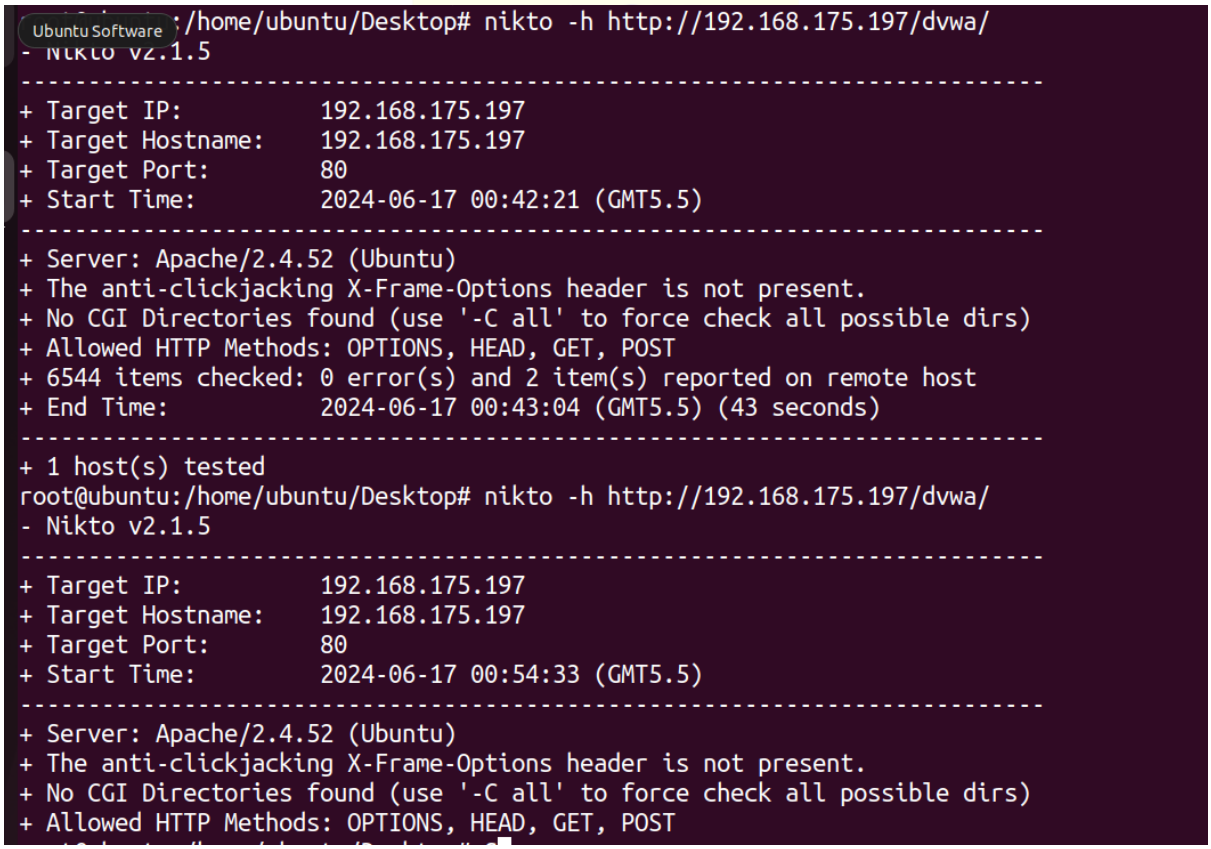


Figure 92 Attacker

Conclusion

We successfully combined Wazuh with Teler in this blog post to improve Wazuh's web assault detection capabilities. Threat analysts and incident responders can utilise the teler HTTP IDS capabilities in conjunction with Wazuh to scan web server logs and retrieve pertinent data for investigations. An organisation can stay ahead of security-related actions by promptly identifying and responding to incidents when using the Wazuh SIEM and XDR solution.

10.12 Integration of TheHive, Cortex, MISP, and Wazuh for SOC Platform

Objective:

To create a comprehensive Security Operations Center (SOC) platform by integrating TheHive, Cortex, MISP, and Wazuh. This integration aims to enhance threat detection, analysis, and response capabilities within the organization.

Purpose:

The integration of these tools is designed to provide a centralized platform for incident management, threat intelligence, and automated response. This setup improves the efficiency and effectiveness of security operations by leveraging the strengths of each tool.

Components and Their Roles:

1. **Wazuh:**
   - **Function:** Provides log analysis, intrusion detection, and vulnerability detection.
   - **Role in Integration:** Collects and analyzes security events from various endpoints and forwards relevant alerts to TheHive for further investigation.
2. **TheHive:**
   - **Function:** Acts as a security incident response platform, managing and investigating security incidents.
   - **Role in Integration:** Receives alerts from Wazuh and creates cases for security analysts to investigate. It also integrates with Cortex for automated analysis and enrichment of security data.
3. **Cortex:**
   - **Function:** Provides automated analysis and response capabilities through analyzers and responders.
   - **Role in Integration:** Works with TheHive to enrich incident data by running analyzers on observables (e.g., IP addresses, files). It can also trigger automated response actions based on analysis results.
4. **MISP (Malware Information Sharing Platform):**
   - **Function:** Acts as a threat intelligence platform, sharing and storing information about threats and indicators of compromise (IOCs).
   - **Role in Integration:** Provides threat intelligence feeds to TheHive and Cortex, enriching the incident data with context from the broader threat landscape.

Integration Workflow:

1. **Data Collection:**
   - Wazuh agents collect security events and logs from endpoints.
   - These events are analyzed and relevant alerts are sent to TheHive.
2. **Incident Management:**
   - TheHive receives alerts from Wazuh and automatically creates cases.
   - Security analysts investigate cases using TheHive's interface.
3. **Threat Intelligence Enrichment:**
   - Observables from cases in TheHive are sent to Cortex for analysis.

- Cortex uses analyzers to gather additional information from various sources, including MISP.

4. **Automated Response:**
   - Based on the analysis, Cortex can trigger automated responses to mitigate threats.
   - These responses can include blocking IP addresses, isolating endpoints, or generating further alerts.
5. **Threat Intelligence Sharing:**
   - MISP provides threat intelligence to TheHive and Cortex.
   - New IOCs discovered during investigations are shared back to MISP, enhancing the overall threat intelligence repository.

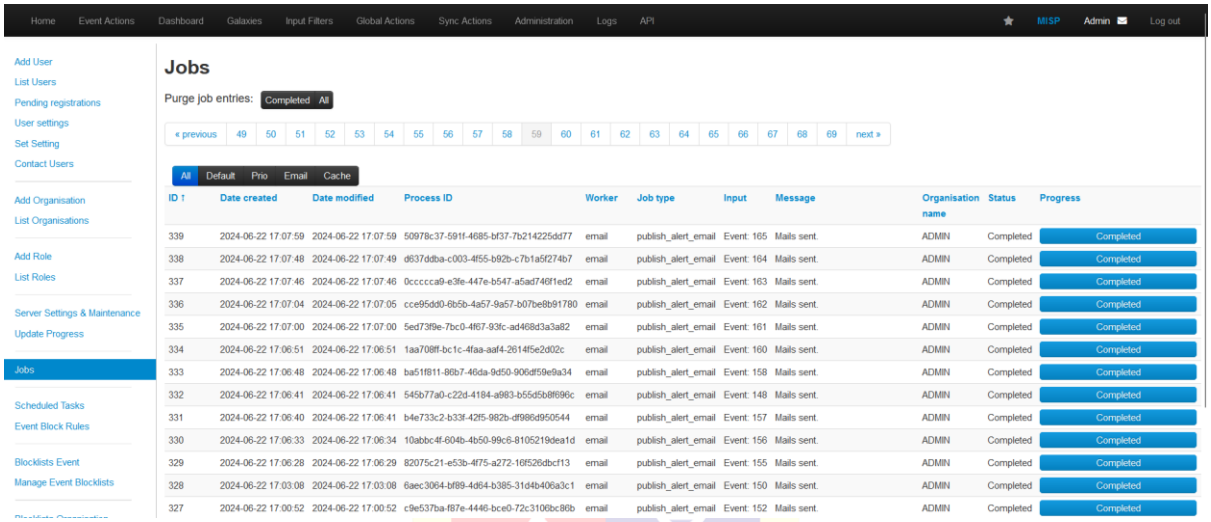


Figure 93 Misp feeds update

Benefits:

- **Centralized Incident Management:** Integration of TheHive centralizes the handling and investigation of security incidents, improving coordination and efficiency.
- **Enhanced Threat Intelligence:** Leveraging MISP enriches incident data with relevant threat intelligence, providing better context for investigations.
- **Automated Analysis and Response:** Cortex's analyzers and responders automate the enrichment and mitigation processes, reducing the manual workload on security teams.
- **Improved Visibility and Detection:** Wazuh's comprehensive monitoring capabilities ensure that potential threats are detected and reported in a timely manner.

Conclusion:

The integration of TheHive, Cortex, MISP, and Wazuh creates a robust SOC platform that enhances the organization's ability to detect, analyze, and respond to security incidents. This comprehensive approach improves incident management, leverages threat intelligence for enriched data, and automates key processes to ensure a proactive and efficient security posture.

MISP and cortex integration

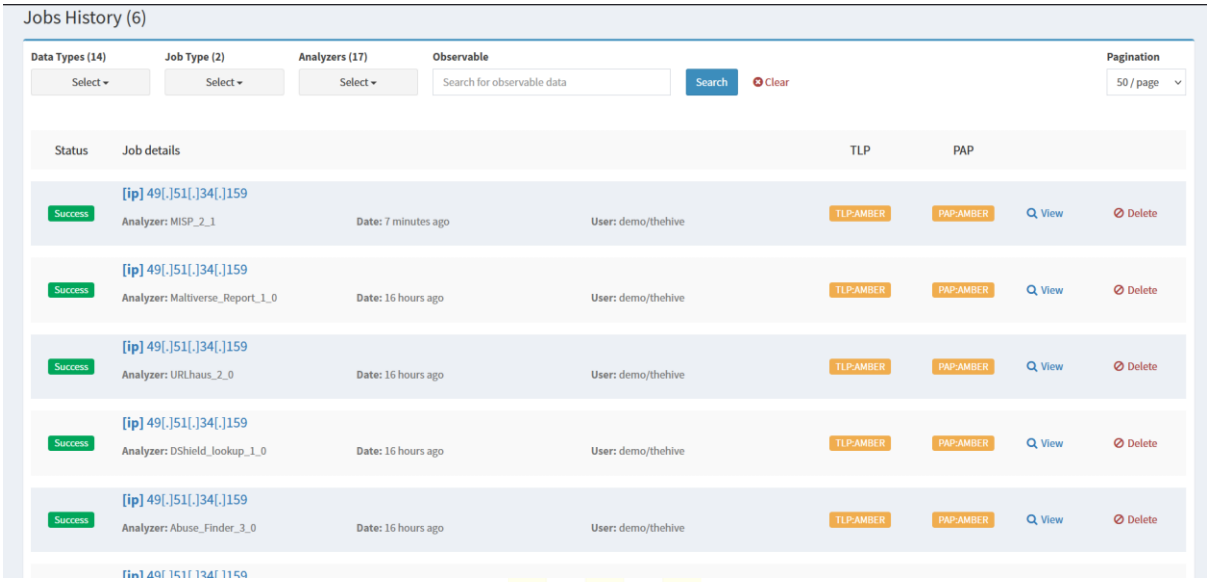


Figure 94 Threat intel misp and cortex integration OUT put

With the help of the cortex gui analyzers, we can now obtain the IOC without having to search for it manually thanks to the integration of the cortex and misp. Instead of wasting time searching for the IOC and finding it manually, we can now obtain the IOC and its details with just one click, making the alert handling process faster.
Integration of Misp with Cortex to get the threat intelligence alert and produce more reliable intelligence.

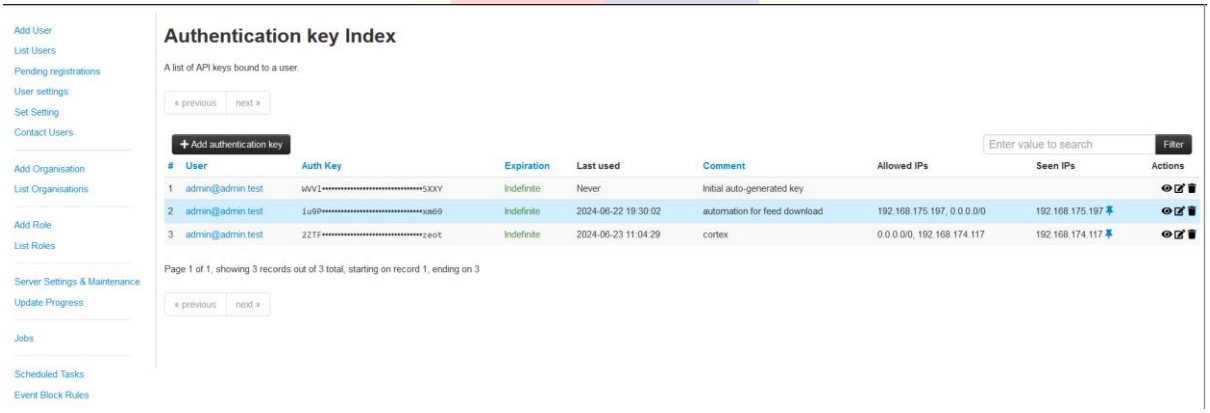


Figure 95 Misp and cortex API Integration

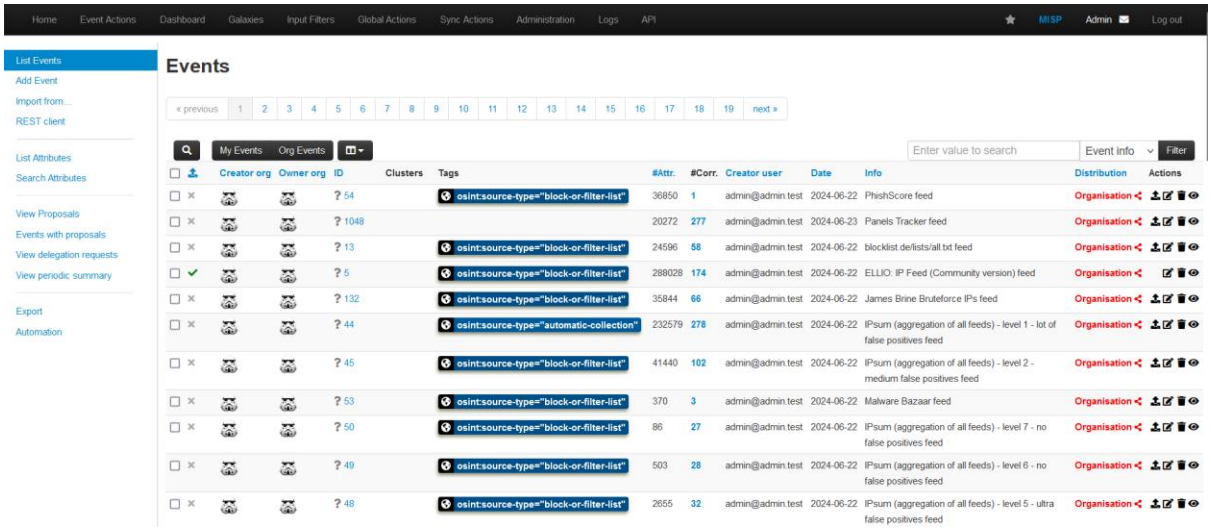


Figure 96 Misp feed

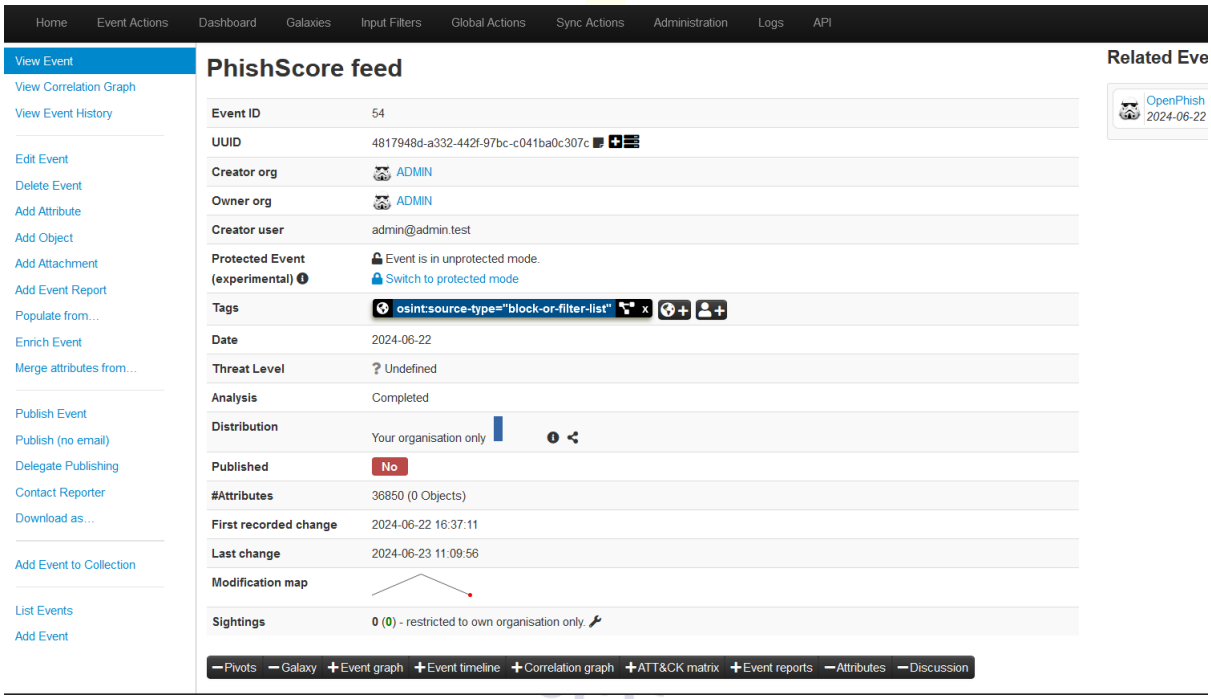


Figure 97 IOC Domain

10.13 Wazuh and misp integration

We can now test the integration on the endpoint and see if we can obtain the IOC from the misp. I will take one IOC domain name and attempt to resolve the case by simply sending a ping request to that IOC domain via the Win endpoint. I will also try to create an alert in Wazuh and an L-1 case in Thehive.

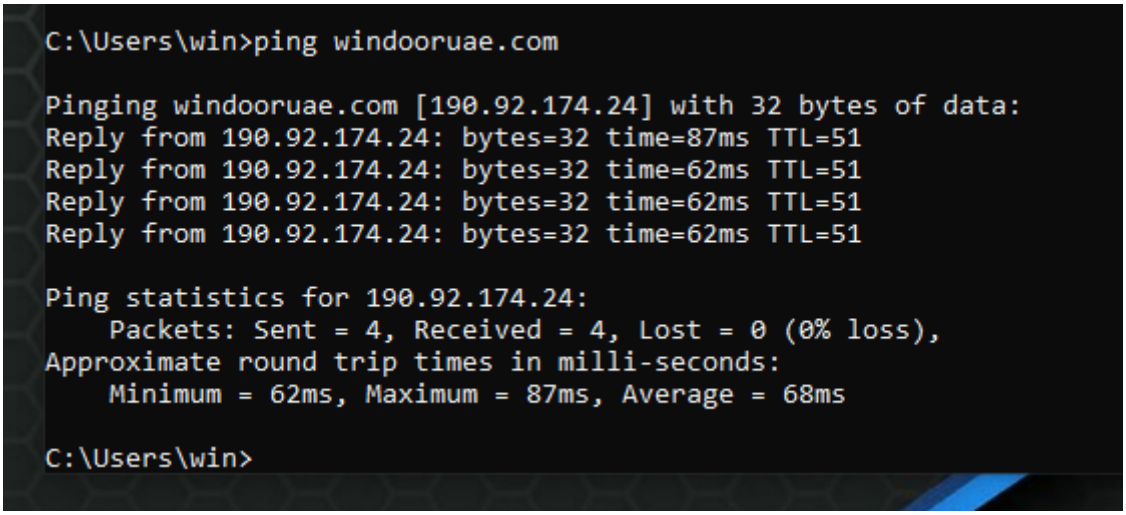
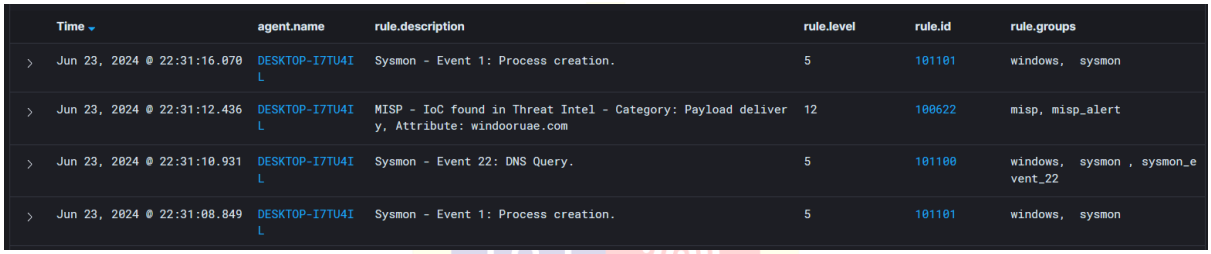


Figure 98 ping the IOC Domain



Time	agent.name	rule.description	rule.level	rule.id	rule.groups
Jun 23, 2024 @ 22:31:16.070	DESKTOP-I7TU4IL	Sysmon - Event 1: Process creation.	5	101101	windows, sysmon
Jun 23, 2024 @ 22:31:12.436	DESKTOP-I7TU4IL	MISP - IoC found in Threat Intel - Category: Payload delivery, Attribute: windooruae.com	12	100622	misp, misp_alert
Jun 23, 2024 @ 22:31:10.931	DESKTOP-I7TU4IL	Sysmon - Event 22: DNS Query.	5	101100	windows, sysmon , sysmon_event_22
Jun 23, 2024 @ 22:31:08.849	DESKTOP-I7TU4IL	Sysmon - Event 1: Process creation.	5	101101	windows, sysmon

Figure 99 recive the alert on wasuh

Figure 100 Details log

We see that we can successfully recivethe alert on the wazuh that at endpoint the MISP IOC has been detected

10.14 Wazuh Cortext, thehive and misp integration

Now that we're here, we may attempt to create the entire SOC platform workflow. As an L-1, we can attempt to determine the IOC by just clicking 1. We can also try to ping the misp IOC on the endpoint, ensure that we can receive the alert on the Wazuh, and establish the case on the Hive.Here, we may use the misp's vyae.top IOC domain to try pinging the destination.

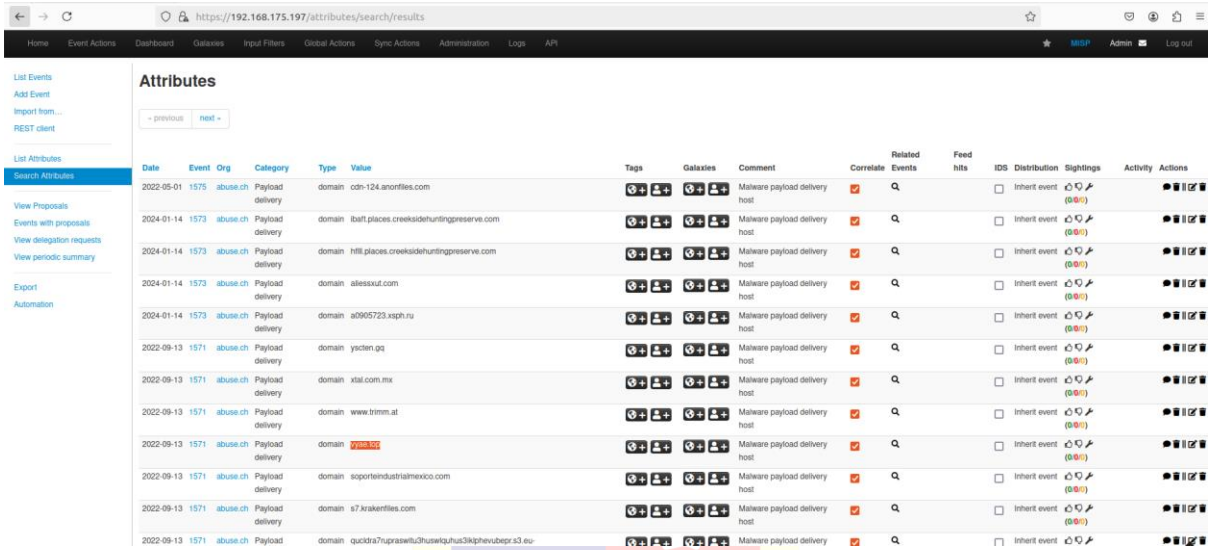


Figure 101 misp IOC table

Figure 102 wazuh alert give misp alert

Here, we observe that an alert has also been generated on Thehive.

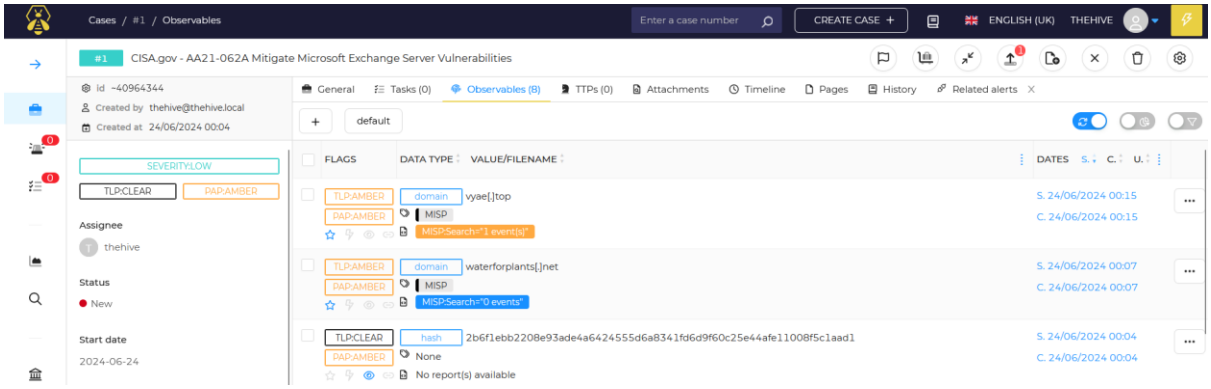


Figure 103 thehive alert created

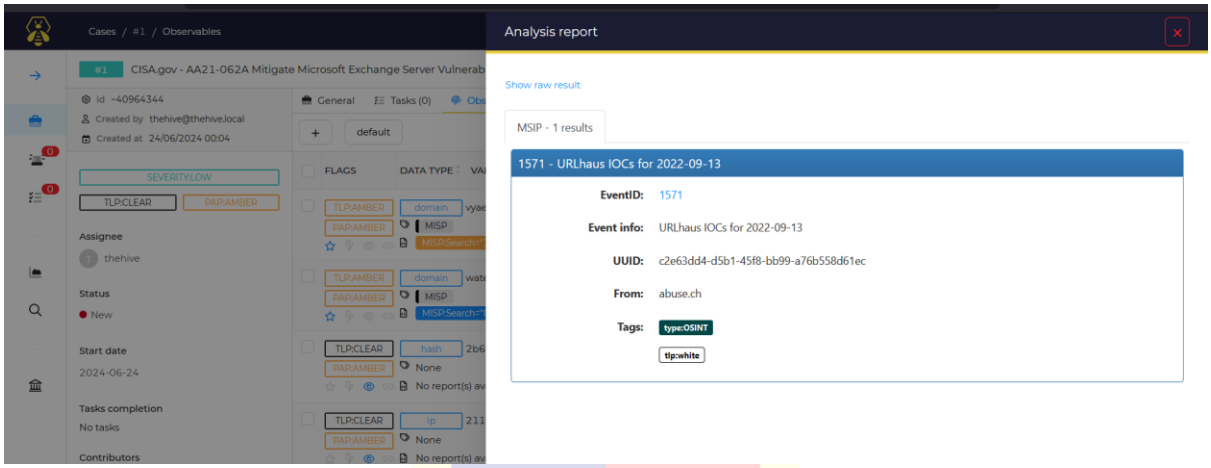


Figure 104 details of IOC

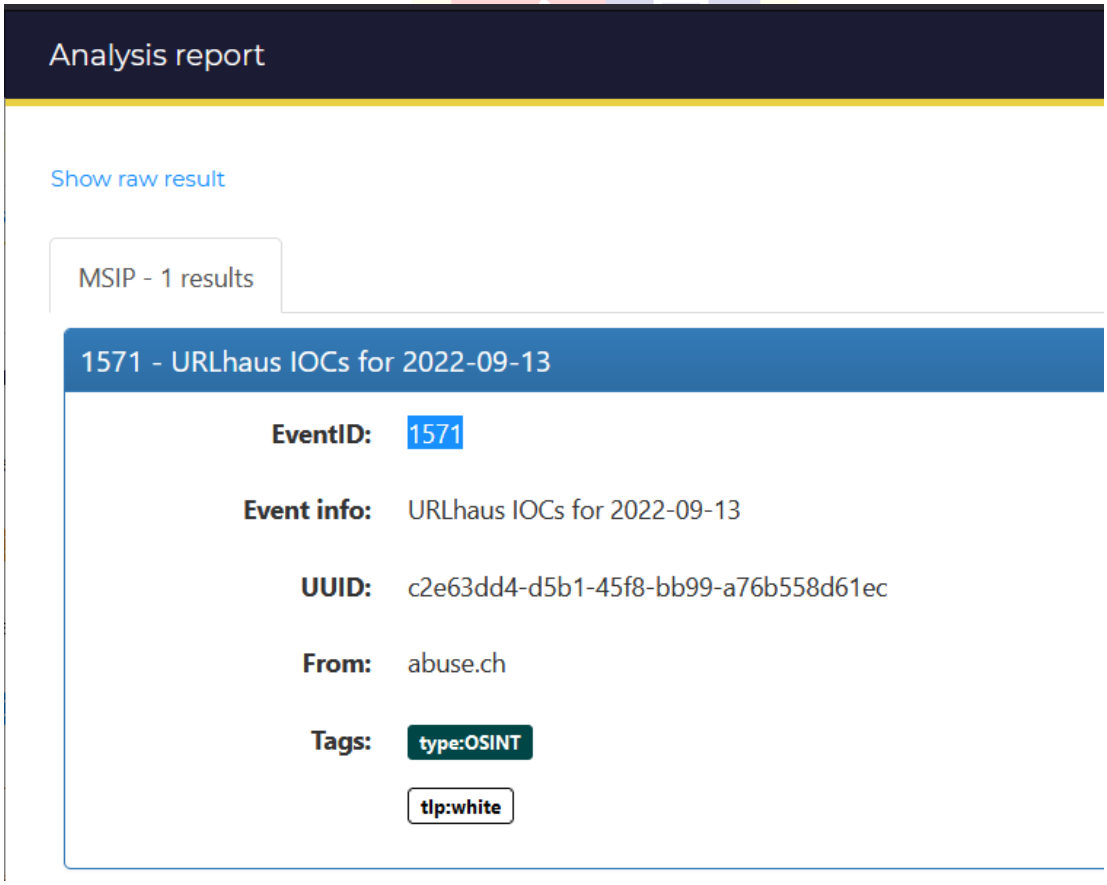
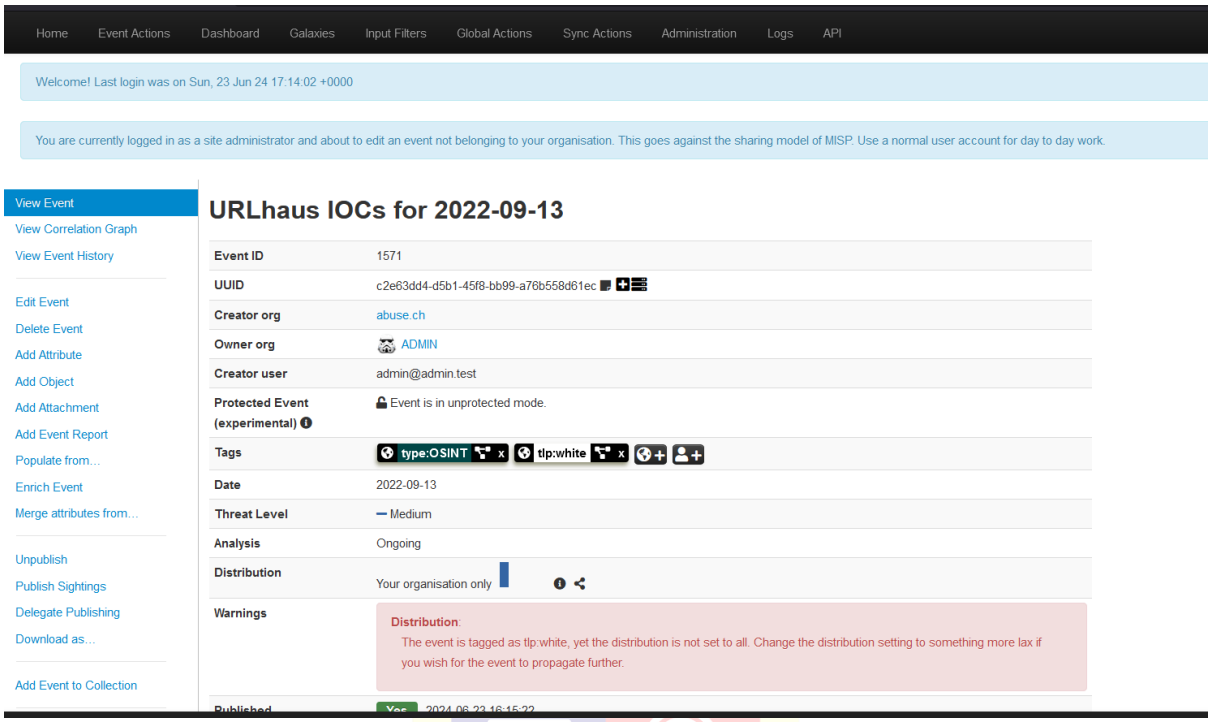
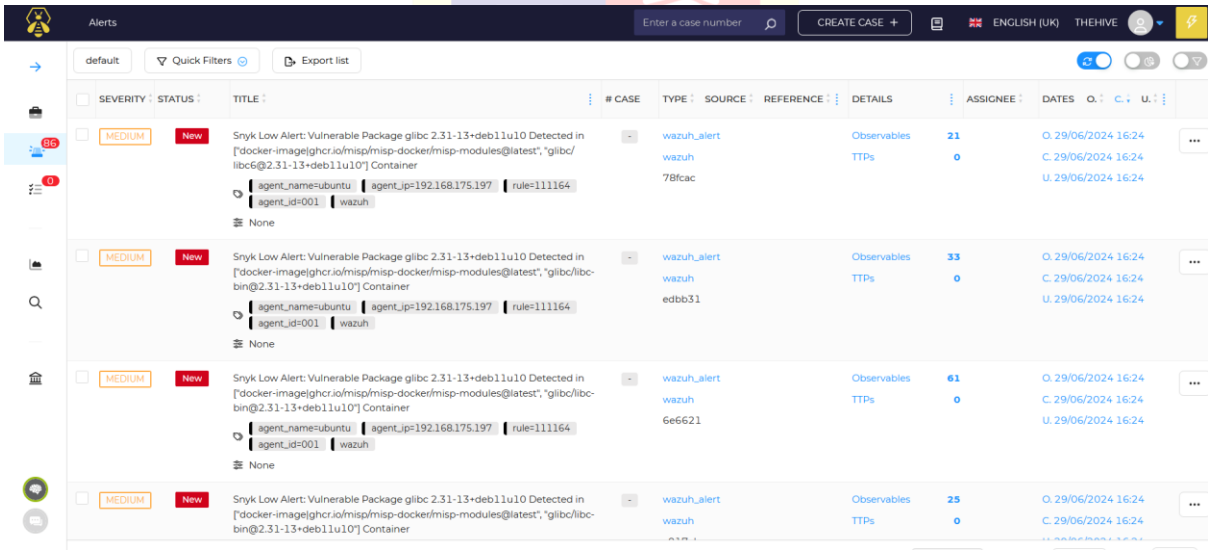


Figure 105 misp IOC event ID

To view more detailed logs, simply click the alert information to be redirected to this misp IOC table.



Integration of Wazuh with Thehive for Magt cases. Our l1 team can easily repair this issue and we can receive all of Wazuh's alerts into the hive.



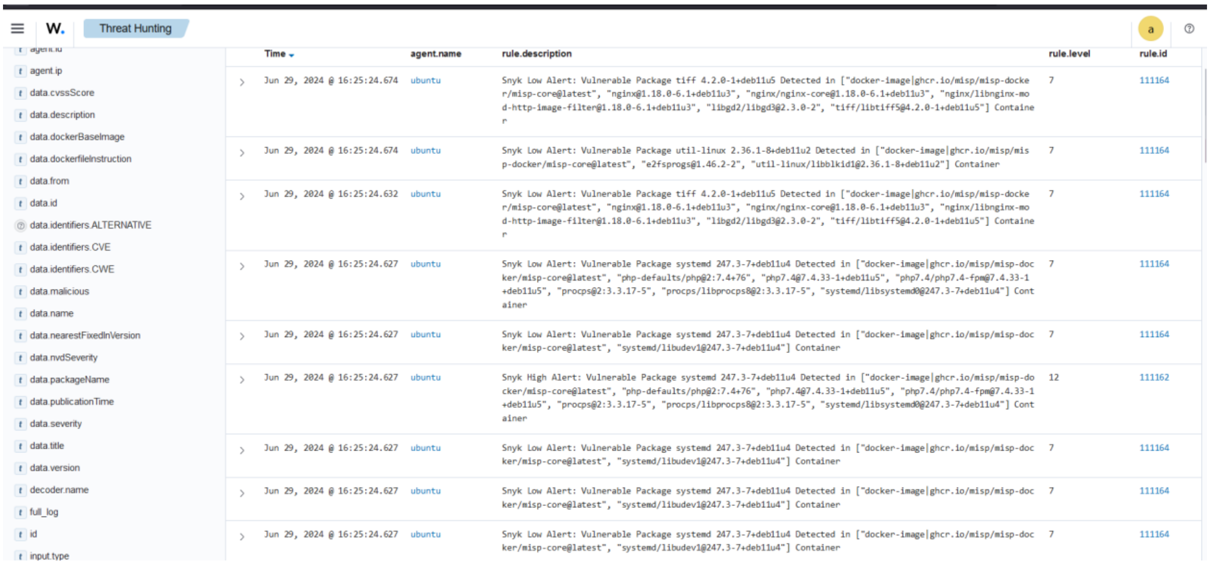
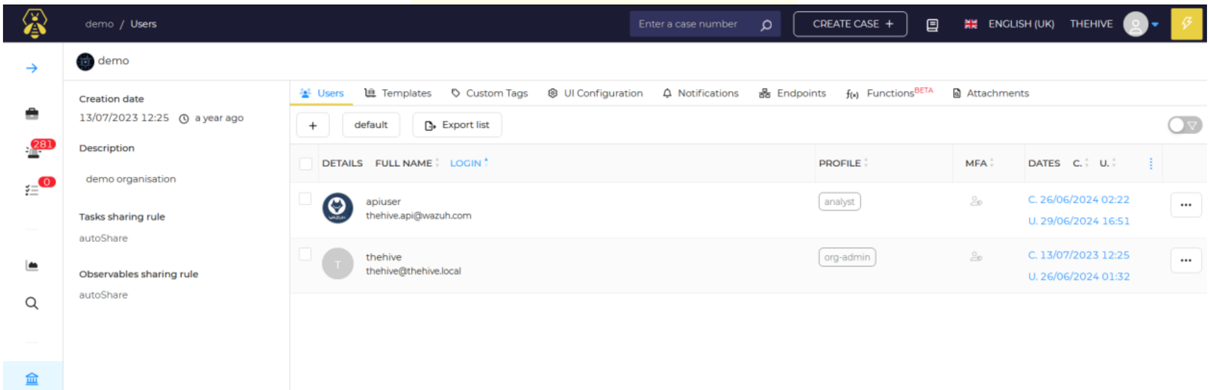


Figure 106 realtime alert show case



Here, we have a bot called apiuser that will gather Wazuh alerts and open a case in the hive. Here, the Misp serves as the publicly accessible IOC holder. This way, in the event that my organisation discovers a new IOC, I can add it to the Misp and allow all other organisations to use it. Additionally, if the same attack occurs again, we can use the predefined IOC and avoid having to repeat the entire process by integrating the Hive and Misp's API.

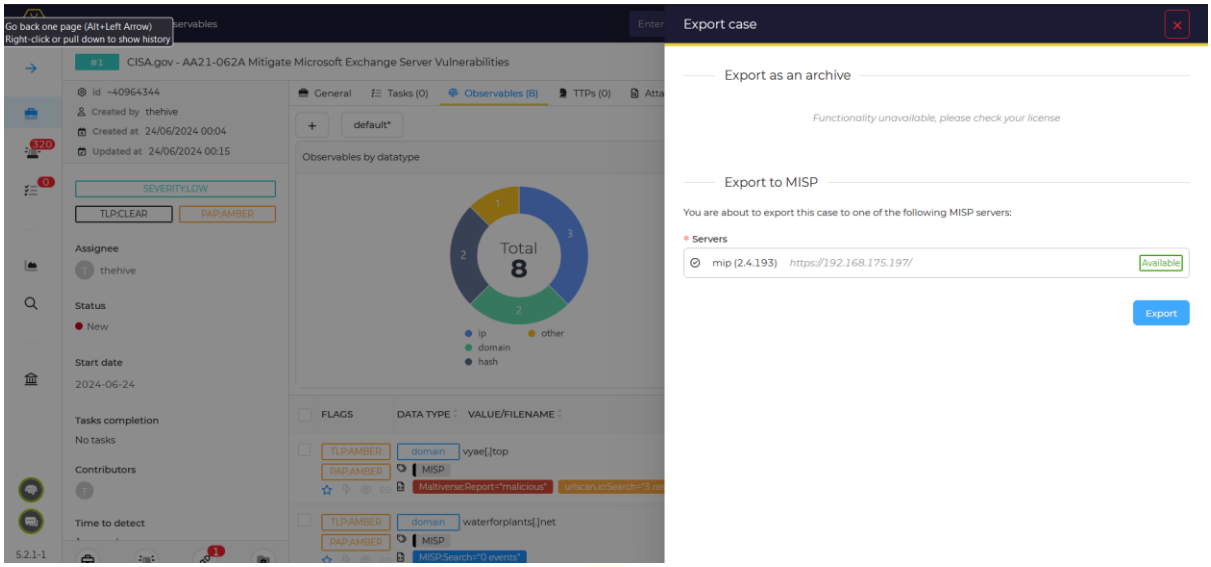


Figure 107 IOC pull

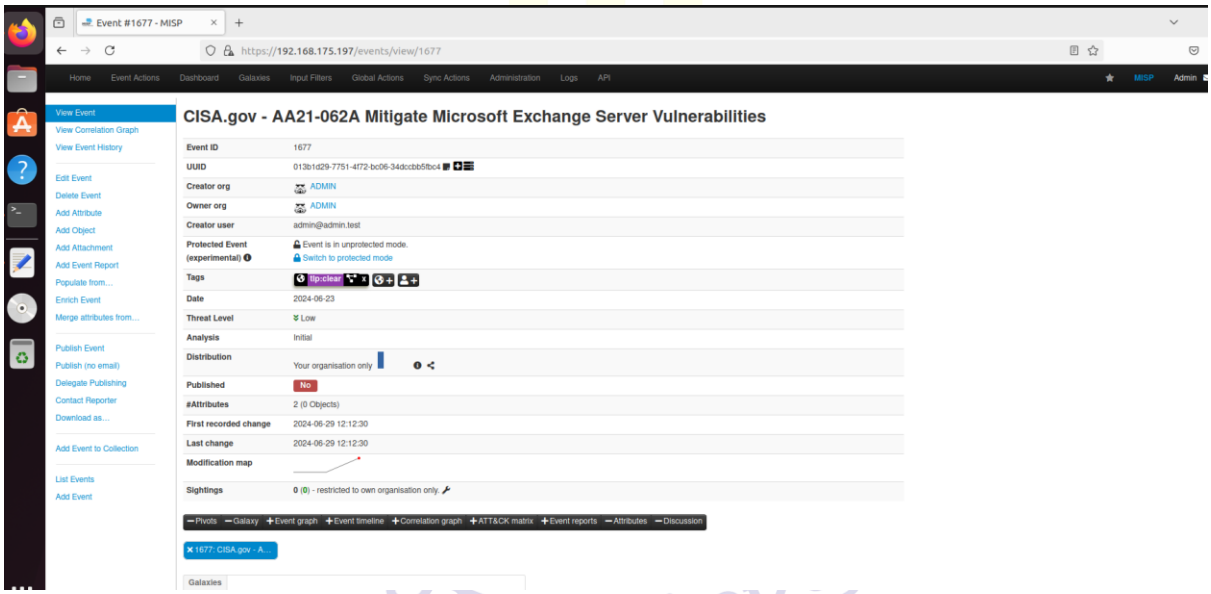


Figure 108 over IOC on AIR

We can cross-check that my IOC is currently in the misp DB over here.

Conclusions

A big step towards democratising access to advanced security operations skills is the open-source SOC initiative. Through the use of strong open-source tools such as Wazuh, TheHive, Cortex, and MISP, this project offers enterprises an affordable yet capable platform for tracking, identifying, and handling cybersecurity risks.

Economical Resolution:

Open-source Advantage: By lowering the cost burden on businesses, using open-source solutions enables a wider range of organisations, particularly small and medium-sized firms (SMEs), to access advanced security operations.

Comprehensive Security:

Integrated Approach: By combining several technologies, a thorough approach to security is ensured, encompassing threat intelligence, vulnerability management, incident response, and threat detection.

Flexibility and Scalability:

Modular Design: Easy scalability and flexibility are made possible by the tools' modular architecture. Businesses can modify the SOC to fit their own requirements and expand it as they expand.

Platform ready for the future:

Innovative Potential: By establishing a solid basis for upcoming advancements, like the use of AI and machine learning for sophisticated threat identification and response, the project equips businesses to counteract changing cyberthreats.

Future Scopes

Enhanced Integration

1. Expanded Toolkit: To offer a more complete security posture, future versions may incorporate more security tools like cloud security platforms, sophisticated SIEM solutions, and endpoint detection and response (EDR) technologies.
2. API Improvements: Create more complex APIs to improve communication between various security technologies, enabling automated processes and smooth data transfer.

AI and machine learning:

1. Anomaly Detection: To improve anomaly detection skills, apply machine learning techniques. This would assist in spotting intricate and nuanced assault patterns that conventional rule-based systems would overlook.
2. Predictive analytics: By using AI to forecast possible security events based on threat intelligence and previous data, proactive threat mitigation is made possible.

IoT and Cloud Security:

Cloud Integration: By tackling the particular security issues associated with cloud computing, expand the SOC's capacity to monitor and safeguard cloud settings.
IoT Security: Provide methods for keeping an eye on and safeguarding Internet of Things (IoT) devices, which are more frequently the focus of cyberattacks.

References

[1] D. Suskalo, Z. Moric, J. Redzepagic, and D. Regvart, “Comparative analysis of IBM Qradar and WaZuh for security information and event management,” 2023 Available: https://www.daaam.info/Downloads/Pdfs/proceedings/proceedings_2023/working_papers/dpn34056_a_3_Moric.pdf

[2] J. Timofte, “Intrusion Detection using Open Source Tools.” Accessed: Jul. 13, 2024. Available: https://www.revistaie.ase.ro/content/46/Timofte.pdf

[3] Stefan Stanković, Slavko Gajin, and Ranko Petrović, Eds., “A Review of Wazuh Tool Capabilities for Detecting Attacks Based on Log Analysis,” Jun. 09, 2022. Available: https://www.etran.rs/2022/zbornik/ICETRAN-22_radovi/068-RTI2.6.pdf

[4] A. Asswad, F. Gringoli, and N. Pasquarè, “Analysis of attacks and prevention methods in cybersecurity,” 2022. Available: https://ans.unibs.it/assets/documents/Thesis_Annas_Asswad.pdf

[5] I. Βούλγαρης and I. Voulgaris, “Information and security event management system,” dione.lib.unipi.gr, Mar. 10, 2020. Available: https://dione.lib.unipi.gr/xmlui/handle/unipi/12691

[6] M. Mildenberger and A. Wachtel, “S H ∞ HPCSA Seminar Report Security Infrastructures and intrusion systems,” 2023. Available: https://hps.vi4io.org/_media/teaching/autumn_term_2022/hpcsa_matthias_mildenberger_security_infrastructure.pdf

[7] Najafi, P., Cheng, F., Meinel, C. (2021). SIEMA: Bringing Advanced Analytics to Legacy Security Information and Event Management. In: Garcia-Alfaro, J., Li, S., Poovendran, R., Debar, H., Yung, M. (eds) Security and Privacy in Communication Networks. SecureComm 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 398. Springer, Cham. https://doi.org/10.1007/978-3-030-90019-9_2

Plagiarism Report

turnitin Similarity Report

PAPER NAME
Himil.docx

WORD COUNT
11920 Words

CHARACTER COUNT
68916 Characters

PAGE COUNT
79 Pages

FILE SIZE
79.7KB

SUBMISSION DATE
Jul 13, 2024 4:52 PM GMT+5:30

REPORT DATE
Jul 13, 2024 4:52 PM GMT+5:30

7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 5% Internet database
- 1% Publications database
- 4% Submitted Works database