# Title Page

SECU424

Ethical Hacking and Penetration Testing

Final Project:

Implementation and Testing of a Defensive

Security Component:

Spring 2023-2024

< Ahmad\_Shaito>

# Executive Summary:

## Brief overview of the project goals:

This project involves the implementation and testing of the Wazuh Security Information and Event Management (SIEM) system, a leading open-source security solution, as part of the Ethical Hacking and Penetration Testing course. The aim was to assess the effectiveness of Wazuh in identifying and responding to various simulated cyber attacks.

Objectives:

Deployment: To deploy the Wazuh SIEM solution on a target machine using container technology to simulate a realistic security environment.

Configuration: To configure Wazuh to accurately monitor and log security events, implementing specific defensive rules tailored to the simulated threat landscape.

Testing: To execute a series of diverse attack scenarios against the system to evaluate the performance of Wazuh in detecting and mitigating threats.

# Table of Contents:

Contents

[Title Page 1](#_Toc165880699)

[Executive Summary: 2](#_Toc165880700)

[Brief overview of the project goals: 2](#_Toc165880701)

[Key findings and conclusions 2](#_Toc165880702)

[Table of Contents: 3](#_Toc165880703)

[Introduction 5](#_Toc165880704)

[Purpose of the project: 5](#_Toc165880705)

[Importance of SIEM solutions in cybersecurity: 5](#_Toc165880706)

[Brief introduction to Wazuh and its capabilities: 6](#_Toc165880707)

[Project Setup and Configuration 7](#_Toc165880708)

[Description of the setup environment 7](#_Toc165880709)

[Step-by-step setup process of Wazuh: 7](#_Toc165880710)

[Rule Configuration 11](#_Toc165880711)

[Detailed explanation of the rules configured in Wazuh: 11](#_Toc165880712)

[Attack Scenarios 13](#_Toc165880713)

[Overview of the attacks planned and executed 13](#_Toc165880714)

[Description of each attack type and the tools used 13](#_Toc165880715)

[Justification for choosing these specific attacks 13](#_Toc165880716)

[Testing and Results 13](#_Toc165880717)

[Description of how each attack was carried out 13](#_Toc165880718)

[Observations on how Wazuh responded to each attack 13](#_Toc165880719)

[Screenshots and detailed explanations of the outcomes 13](#_Toc165880720)

[Analysis of any false positives or negatives encountered 13](#_Toc165880721)

[Discussion 13](#_Toc165880722)

[Analysis of the effectiveness of the Wazuh setup 13](#_Toc165880723)

[Insights gained from the testing process 13](#_Toc165880724)

[Suggestions for improving Wazuh's configuration based on test results 13](#_Toc165880725)

[Conclusion 14](#_Toc165880726)

[Recap of the project's objectives and whether they were met 14](#_Toc165880727)

[Summary of key learnings and findings 14](#_Toc165880728)

[Recommendations for future projects or further research 14](#_Toc165880729)

# Introduction

## Purpose of the project:

This project aims to implement and evaluate the Wazuh Security Information and Event Management (SIEM) system within a controlled environment to understand its capabilities in real-time security monitoring, threat detection, and response. As part of the Ethical Hacking and Penetration Testing course, this initiative seeks to provide hands-on experience with advanced defensive security technologies, allowing students to explore the practical aspects of deploying and optimizing SIEM systems to enhance organizational security posture.

## Importance of SIEM solutions in cybersecurity:

1. Threat Detection and Response:

SIEM systems aggregate and analyze log data from various sources across an organization’s network to identify patterns that may indicate a security threat. By centralizing the security data, SIEMs can detect complex threats that are otherwise difficult to identify. This capability is crucial for early detection of potential breaches, enabling timely response to mitigate damage.

1. Compliance and Forensic Analysis:

Many industries are subject to regulatory requirements that mandate the collection, monitoring, and analysis of security data to ensure compliance with standards such as GDPR, HIPAA, and PCI-DSS. SIEMs help organizations maintain and prove compliance by providing logs of all security-related events and the ability to conduct detailed forensic analyses in the event of a security incident.

1. Automated Security Management:

SIEM systems automate the process of collecting and analyzing security data, which reduces the workload on security teams and increases efficiency. Automation also helps in implementing proactive security measures, such as triggering alerts and automatically responding to certain types of threats based on predefined rules.

1. Enhanced Visibility and Monitoring:

With the increasing complexity of IT environments and the proliferation of cloud services, mobile devices, and IoT, maintaining visibility across all assets becomes challenging. SIEMs provide a holistic view of an organization’s security environment, monitoring all activity and ensuring that no part of the network is left unguarded.

## Brief introduction to Wazuh and its capabilities:

Wazuh is an open-source Security Information and Event Management (SIEM) system that offers a comprehensive solution for threat detection, integrity monitoring, incident response, and compliance. Originating as a fork of the OSSEC project, Wazuh has evolved to provide a broader set of security capabilities and enhanced scalability features suitable for modern IT infrastructures, including cloud, on premise, and hybrid environments.

Key Capabilities of Wazuh:

Threat Detection and Response:

Real-time Detection: Wazuh monitors system and application logs across various platforms to detect anomalies, intrusions, and unauthorized changes in real time.

Active Response: It can automatically respond to certain threats through predefined actions, such as blocking IP addresses or modifying firewall rules, helping to mitigate risks promptly.

Compliance Management:

Wazuh helps organizations comply with various regulatory requirements by ensuring that their security practices align with standards such as PCI-DSS, GDPR, and HIPAA. It provides detailed reporting capabilities that are crucial for audits and maintaining compliance.

Log Data Analysis:

The tool aggregates and analyzes data from multiple sources, providing a centralized view of security alerts generated from network devices, servers, and applications. This centralization aids in comprehensive monitoring and simplified management of security events.

File Integrity Monitoring:

Wazuh continuously checks and verifies the integrity of critical system and configuration files. Any unauthorized changes are reported, allowing security teams to understand and remediate potential security breaches quickly.

Vulnerability Detection:

It regularly scans systems for vulnerabilities, using databases from public repositories and vulnerability feeds to assess and prioritize security weaknesses, facilitating proactive management of security threats.

Cloud Security:

Specifically designed to operate in cloud environments, Wazuh provides security visibility into workloads running on cloud platforms such as AWS, Azure, and Google Cloud, ensuring that cloud-based assets are adequately protected.

# Project Setup and Configuration

## Description of the setup environment

This project utilizes a virtualized environment comprising two main components: a Wazuh server and a Wazuh agent. These components are hosted on separate virtual machines (VMs) to simulate a realistic network environment where the Wazuh server acts as the central monitoring system, and the agent represents a node in the network that is monitored for security events.

Wazuh Server:

Role: Acts as the central monitoring system for security management.

Operating System: Kali / Ubuntu / CentOS

Wazuh Components Installed: Includes the Wazuh manager, API, and Elasticsearch-Logstash-Kibana (ELK) stack for log management and visualization.

Network Configuration: The IP of the server is specified to an agent on creation

Wazuh Agent:

Role: Monitored node that reports security-related events back to the Wazuh server.

Operating System: Windows 10 / Ubuntu / Kali / CentOS

Wazuh Agent Version: 4.x (4.7)

Network Configuration: The agent is given the IP of the server in order to communicate with

## Step-by-step setup process of Wazuh:

The installation of the Wazuh SIEM system was conducted using the Quick start guide available on the official Wazuh website. This guide provides a straightforward approach to setting up Wazuh, ensuring that all necessary components are installed and configured properly.

Virtual Environment Setup:

Configured two virtual machines—one for the Wazuh server and another for the Wazuh agent.

Ensured both machines met the minimum system requirements for Wazuh installation (e.g., CPU, memory, disk space).

Wazuh Server Installation:

Download and Execute the Installation Script:

The installation of the Wazuh server is initiated by downloading and running the Wazuh installation script. This can be done using the following command:

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

This command retrieves the installation script from Wazuh's official package repository and executes it with administrative privileges, ensuring that all necessary components of the Wazuh server are installed automatically.

Upon successful completion of the installation, the terminal displays a summary that includes the URL to access the Wazuh web interface, along with the default username (admin) and an auto-generated password. This information is crucial for accessing the system:

INFO: You can access the web interface at https://<wazuh-dashboard-ip>

User: admin

Password: <ADMIN\_PASSWORD>

INFO: Installation finished.

Accessing the Wazuh Web Interface:

Initial Login to the Dashboard:

To access the Wazuh dashboard, navigate to the provided IP address in a web browser:

https://<wazuh-dashboard-ip>

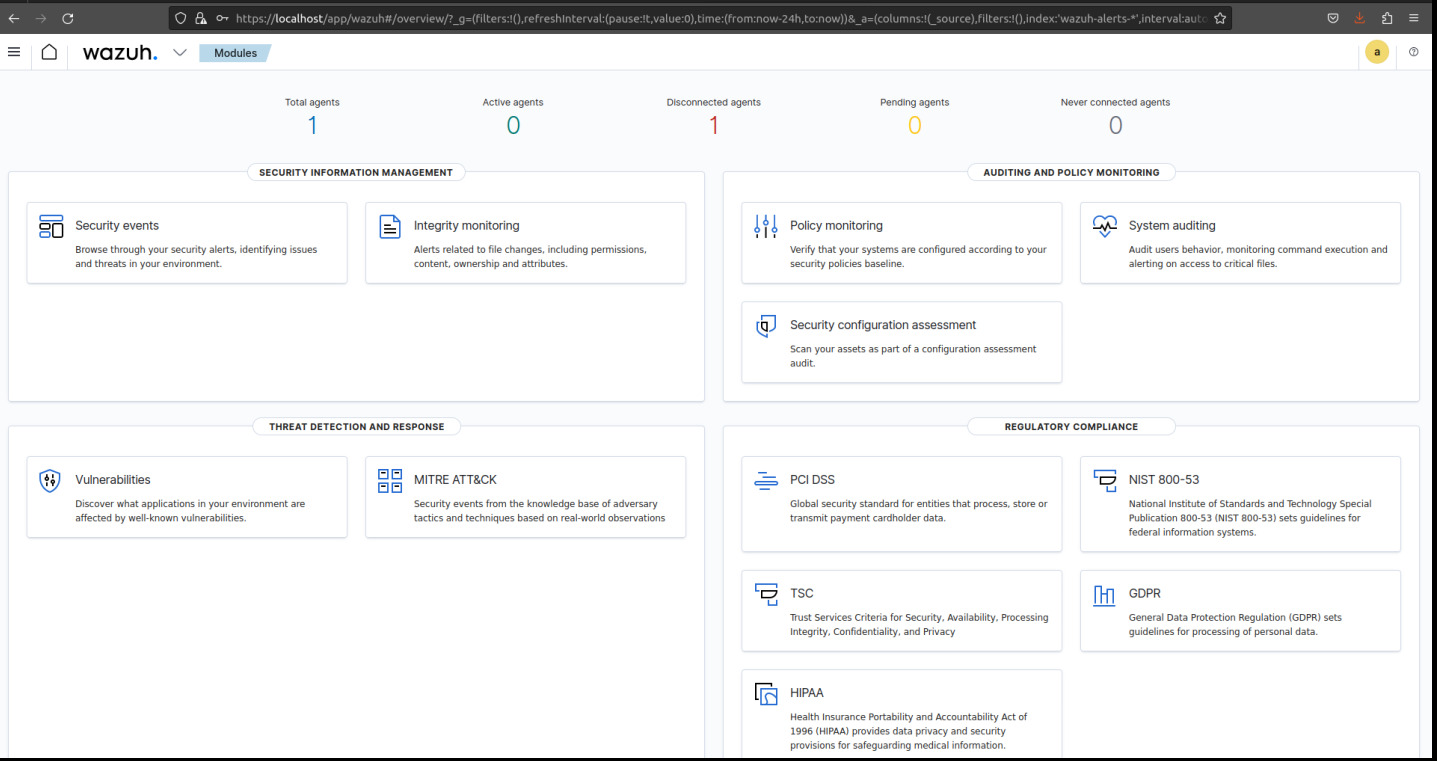


Figure 1 - Server Dashboard

Wazuh Agent:

The agent runs on the host you want to monitor and communicates with the Wazuh server, sending data in near real-time through an encrypted and authenticated channel.

Accessing the Agent Addition Interface:

The process began by logging into the Wazuh dashboard hosted on the Wazuh server. From the main menu, the 'Add Agent' page was accessed, which provides a user-friendly interface for initiating the agent deployment.

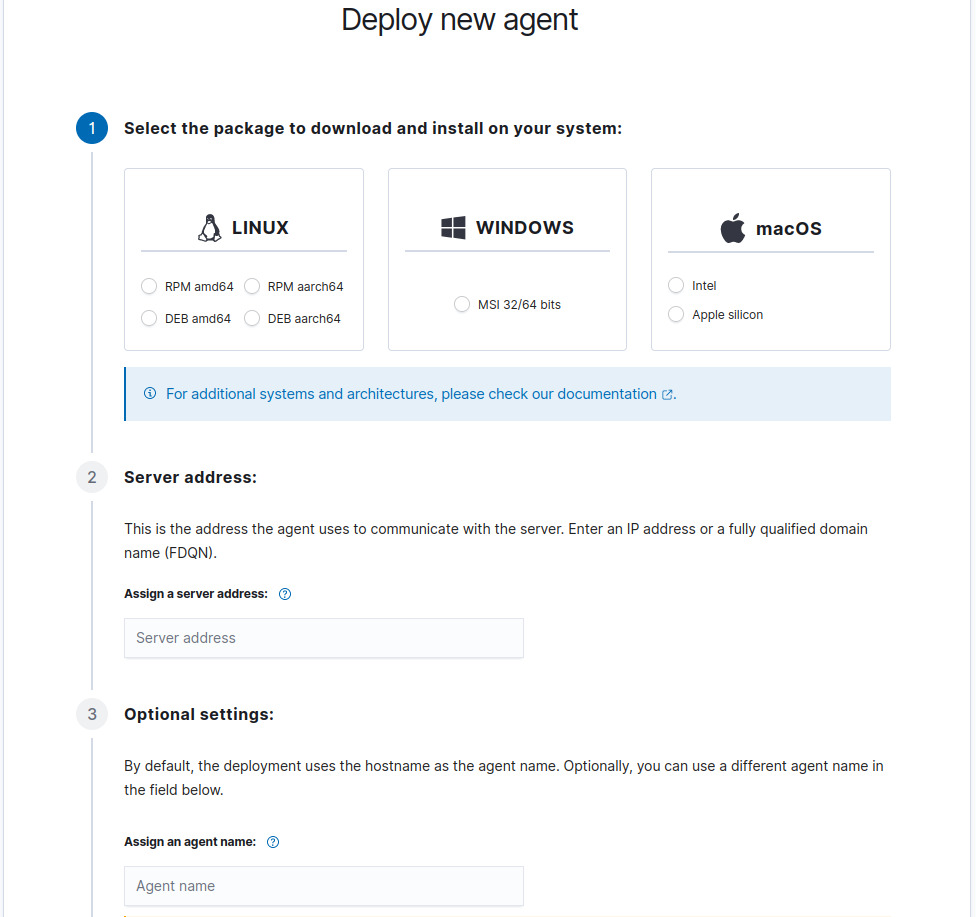


Figure 2 - Adding an Agent

Configuring Agent Details:

On the 'Add Agent' page, the operating system of the target machine (which would host the new agent) was selected. This selection ensures that the instructions and commands provided are specifically tailored to the chosen OS, optimizing compatibility and installation success.

The agent was optionally named for ease of identification within the network. Naming is particularly useful in environments with multiple agents and helps in managing and monitoring specific nodes more effectively.

Additionally, the agent was assigned to a group. Grouping agents allow for the collective application of policies and configurations, which simplifies management when dealing with multiple agents.

Installation Command Retrieval:

The dashboard generated a customized command based on the previous selections. This command is designed to install the Wazuh agent on the target machine and automatically configure it to communicate with the Wazuh server.

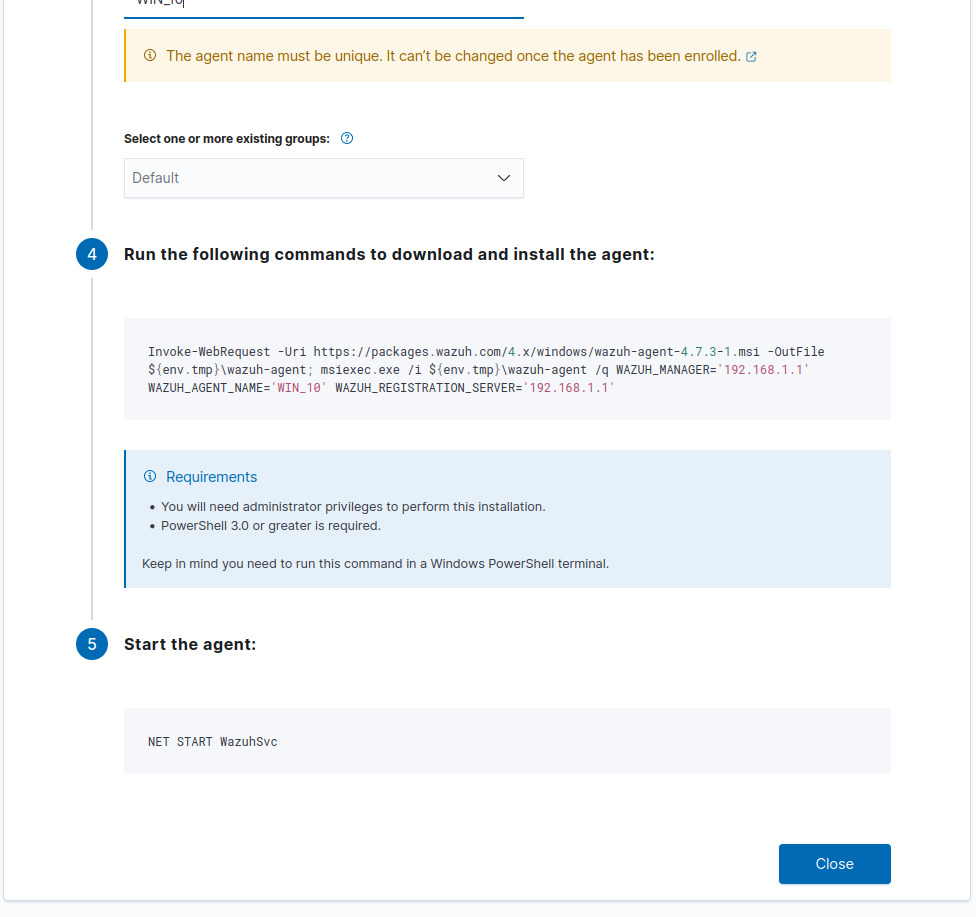


Figure 3 - Adding an Agent - 2

Executing the Installation Command:

The generated command was run on the target machine, where the agent was to be installed. This command handles the installation and initial configuration of the agent, including setting up secure communication channels with the Wazuh server.

Starting the Agent Service:

Following the successful installation, the Wazuh agent service was started on the target machine. This final step establishes the active monitoring and reporting connection to the Wazuh server, enabling the newly deployed agent to begin its security and monitoring functions.

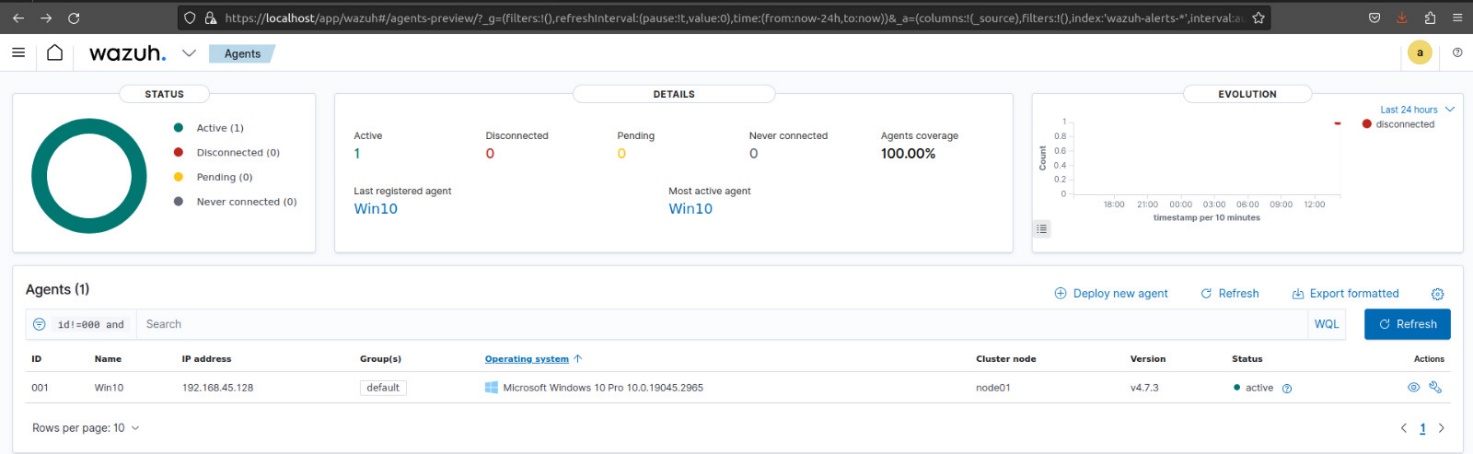


Figure 4 - Agent Dashboard

# Rule Configuration

## Detailed explanation of the rules configured in Wazuh:

After the quick guide installation, the default rules file of wazuh can be found in the directory, which contains a simple ssh connection rule:

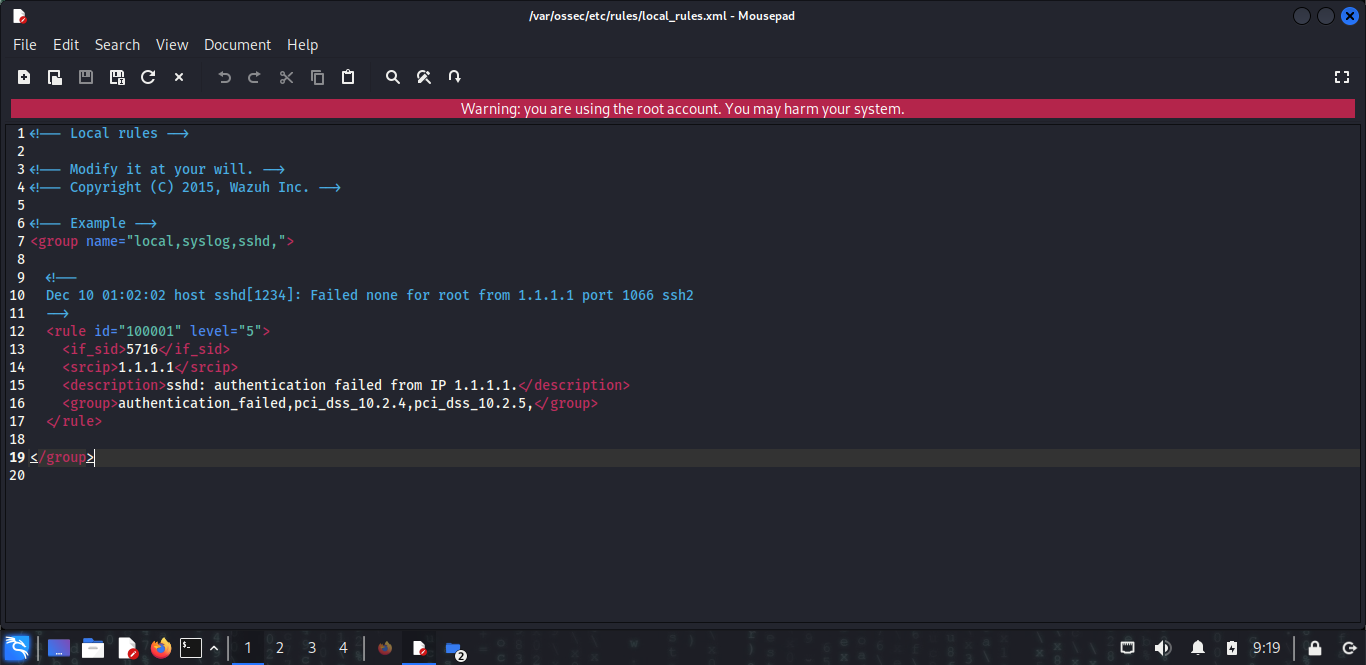


Figure 5 - Default Local Rules File

Rule Structure:

* Rule ID (100001): Each rule must have a unique ID. This ID identifies the rule within the Wazuh system. The rule IDs for local rules generally start from 100000 to avoid conflict with pre-defined Wazuh rules.
* Level (5): This indicates the severity level of the alert that will be generated. Wazuh levels range from 0 to 16, with 16 being the most severe. Level 5 suggests a medium severity.
* If\_sid (5716): This specifies that this rule should only trigger if an alert matching the SID (rule ID) 5716 has already been generated. The SID typically corresponds to a rule that detects a broader category of events, such as any failed authentication attempt via sshd.
* Srcip (1.1.1.1): This condition specifies that the rule only applies if the source IP address of the event is 1.1.1.1. This makes the rule very specific to activity from this IP address.
* Description: Provides a clear, human-readable description of the alert, stating that it is an sshd authentication failure from the specified IP.
* Group Tags: Additional tags (authentication\_failed, pci\_dss\_10.2.4, pci\_dss\_10.2.5) are added to the rule. These tags are useful for filtering and categorizing alerts, especially for compliance with standards like PCI DSS.

Let’s say you want to add an active response to this rule, you add the following to the ossec.config file:

<active-response>

<command>host-deny</command>

<location>local</location>

<rules\_id>100001</rules\_id>

<timeout>600</timeout> <!-- Blocks the IP for 10 minutes -->

</active-response>

Response Structure:

* Command (host-deny): This specifies which command Wazuh should execute as part of the active response. The host-deny command is a built-in Wazuh command that adds the source IP of the alert to a block list on the local firewall, effectively denying access from that IP to the host. This is particularly useful for mitigating brute force attacks or unauthorized access attempts.
* Location (local): This defines where the active response will be executed. The local value means that the response action will be executed on the same agent that generates the alert. Other options include defined-agent to specify a particular agent by ID, or all to execute the command across all agents.
* Rules ID (100001): This is the ID of the rule that triggers this active response. Only alerts generated by the rule with this ID will trigger this active response.
* Timeout (600): This specifies the duration in seconds for which the block will remain effective. In this case, 600 seconds means the IP will be blocked for 10 minutes. After this period, the block is automatically lifted, allowing the IP to attempt connections again. This temporary block can help mitigate the impact of an attack while minimizing disruptions to potentially legitimate users who might be temporarily blocked due to a false positive.

How It Works:

When an alert that matches the specified rule (ID 100001) is generated, the Wazuh manager will execute the host-deny command on the local machine. The command will add firewall rules to block network traffic from the offending IP address (detected in the alert) to the host where the alert was generated. The block will last for the duration specified in the timeout attribute.

Purpose and Effectiveness:

This active response mechanism is a straightforward and effective way to automatically mitigate specific threats, such as repeated unauthorized access attempts from a particular IP address. By integrating active responses like this, Wazuh not only serves as a monitoring tool but also takes proactive measures to enforce security in real-time.

Additional Considerations:

* Precision: Since this response is triggered by a very specific condition (a particular alert from a specific IP), it reduces the risk of false positives affecting legitimate users. However, configuring such responses requires careful consideration to ensure they align with your security policies and do not unintentionally disrupt legitimate activities.
* Management: Over time, you might need to review and adjust your active response strategies based on evolving threat patterns or changes in your network environment.
* Testing: Always test new active responses in a controlled environment before deploying them in production to understand their impact and refine their configuration.

Wazuh also offers vulnerability detection on the agents, to turn that on you should modify these in the ossec.config file:

The first screenshot demonstrates how to configure the Syscollector module in Wazuh. This module collects and forwards system data, which is crucial for vulnerability detection.

The configuration block added to the agent.conf file is shown as follows:

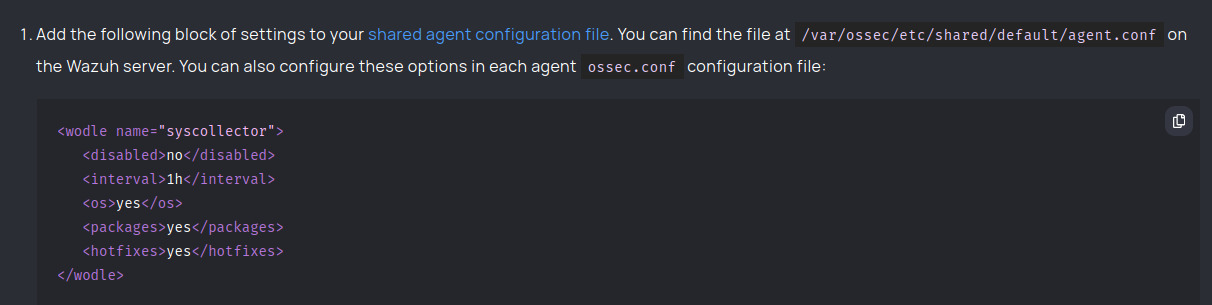


Figure 6 - Vulnerability Detection

The second screenshot outlines the steps to enable and configure the Vulnerability Detector module. This module uses the data collected by Syscollector to check for known vulnerabilities.

An example configuration for scanning Ubuntu and Debian systems is provided:

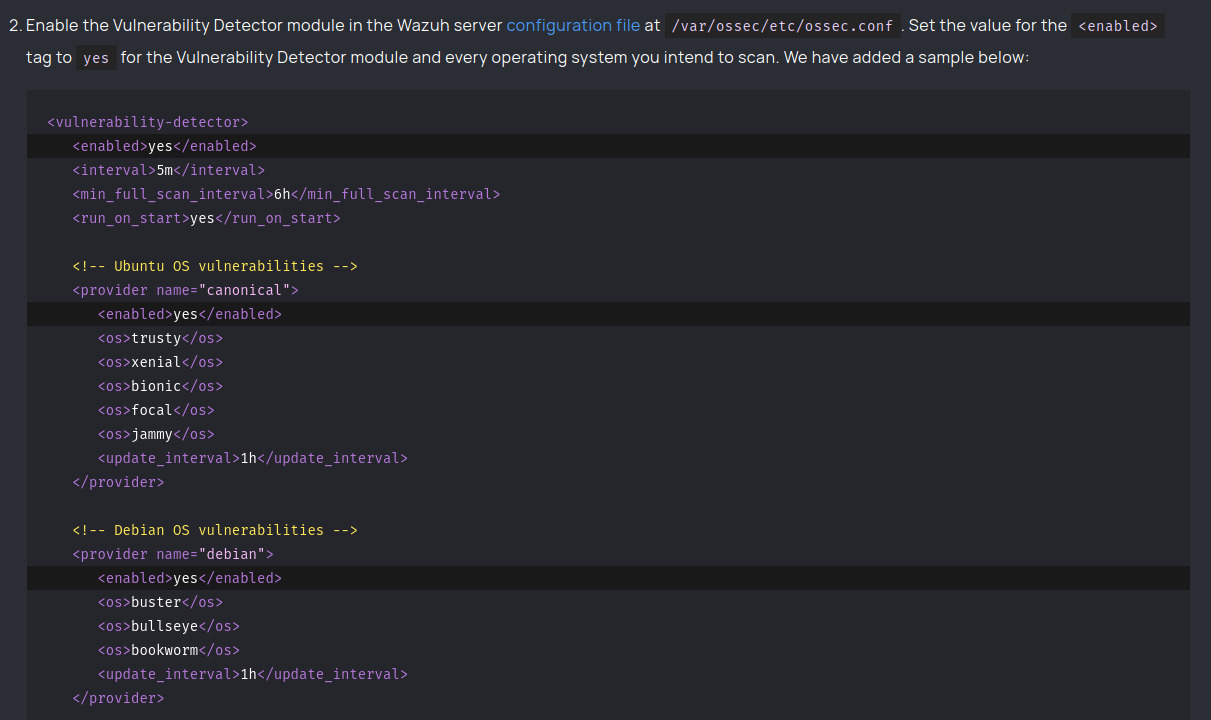


Figure 7 - Vulnerability Detection – 2

This configuration not only helps in maintaining up-to-date security profiles for the systems but also ensures continuous monitoring for any new vulnerabilities that might arise due to system changes or emerging threats. Regular scans and updates are essential for maintaining strong security postures in dynamic IT environments.

File Integrity Monitoring:

The FIM module runs periodic scans on specific paths and monitors specific directories for changes in real time. You can set which paths to monitor in the configuration of the Wazuh agents and manager.

FIM stores the files checksums and other attributes in a local FIM database. Upon a scan, the Wazuh agent reports any changes the FIM module finds in the monitored paths to the Wazuh server. The FIM module looks for file modifications by comparing the checksums of a file to its stored checksums and attribute values. It generates an alert if it finds discrepancies.

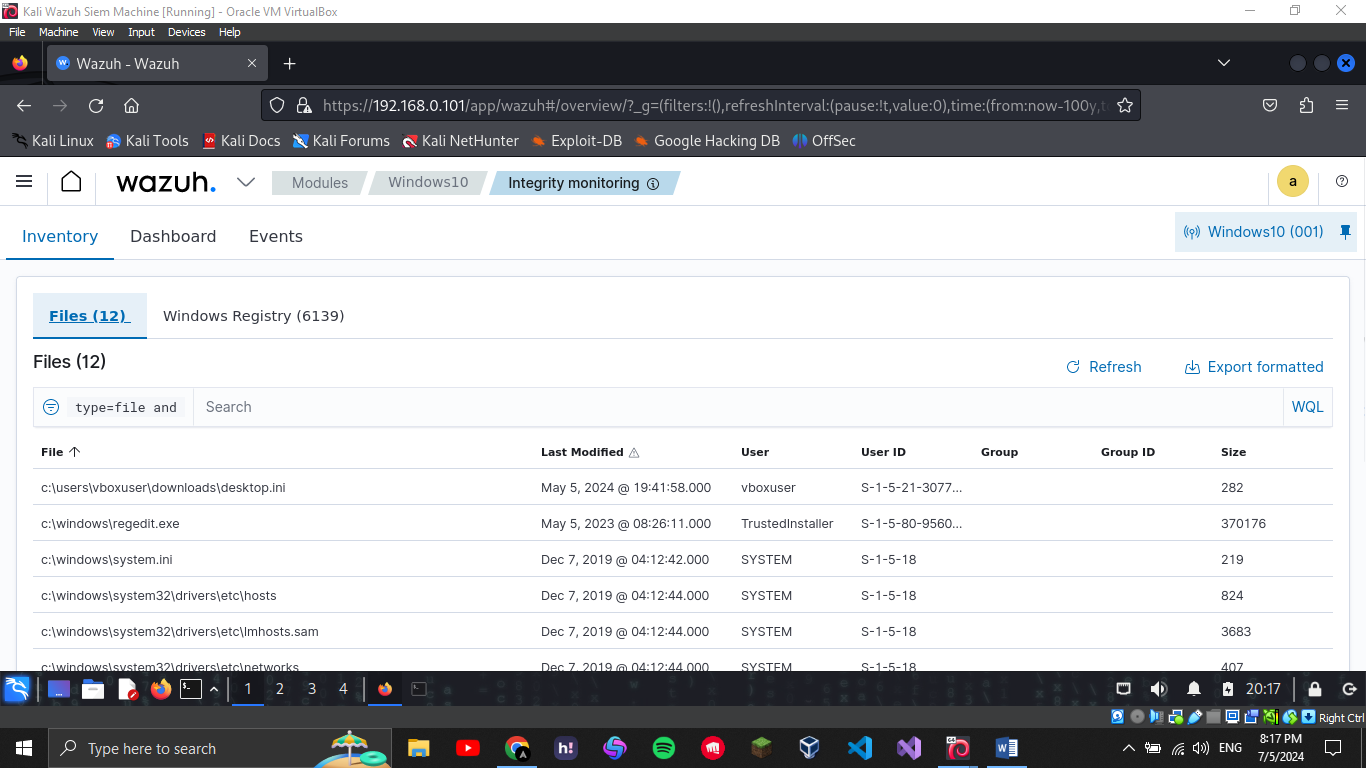


Figure 8 - FIM Inventory

How to configure the FIM module:

Add the following settings to the Wazuh agent configuration file, replacing the directories values with your own filepaths:

*<syscheck>*

*<directories><FILEPATH\_OF\_MONITORED\_FILE></directories>*

*<directories><FILEPATH\_OF\_MONITORED\_DIRECTORY></directories>*

*</syscheck>*

You can view an example of the overview of FIM scan results for all monitored endpoints in the image below:

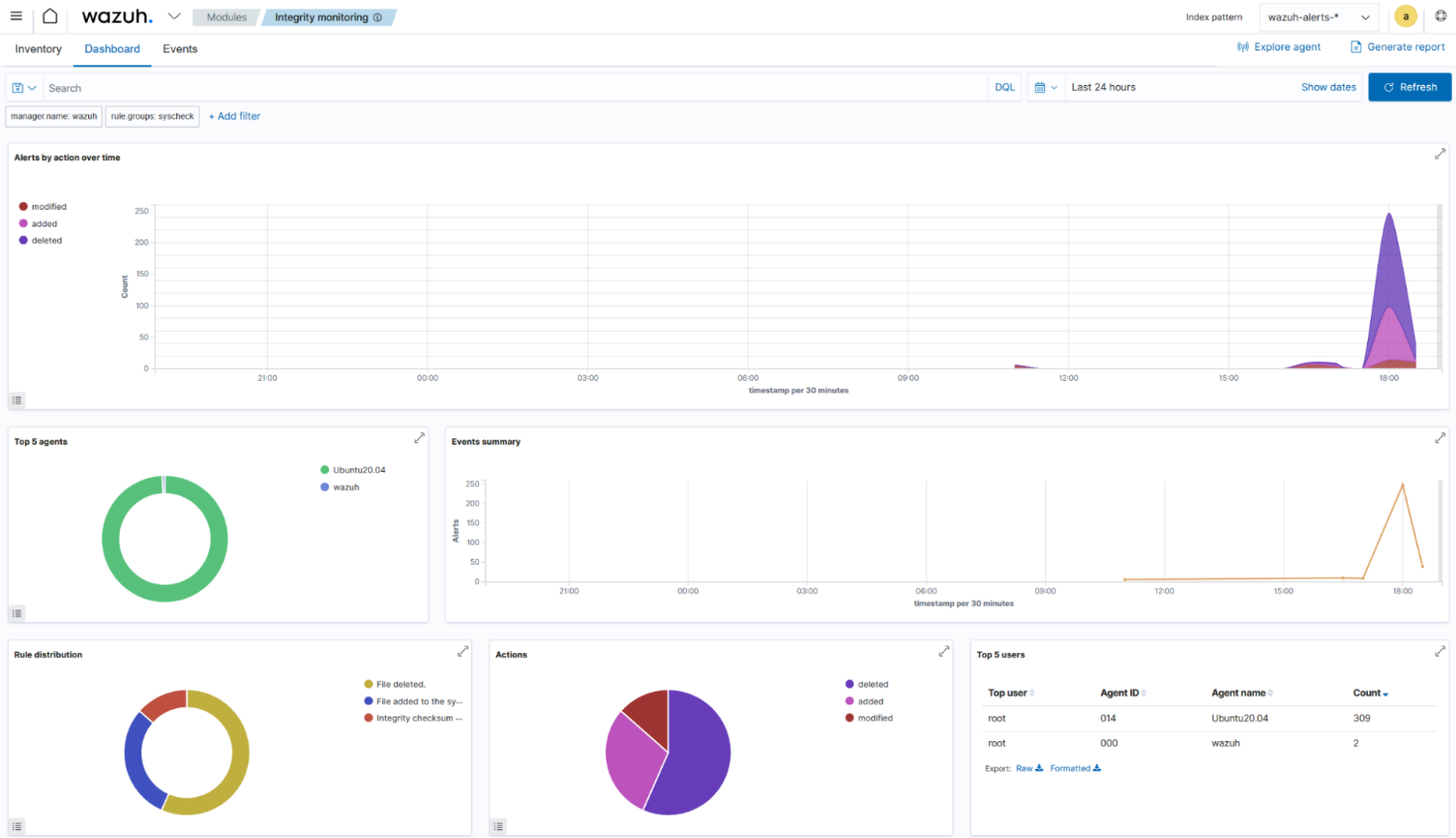


Figure 9 - FIM Dashboard

Malware Detection:

VirusTotal integration:

Wazuh detects malicious files through an integration with VirusTotal, a powerful platform aggregating multiple antivirus products and an online scanning engine. Combining this tool with our FIM module provides an effective way of inspecting monitored files for malicious content.

To integrate with VirusTotal you have to sign up and use the API key in the configuration file as follows:

*<integration>*

*<name>virustotal</name>*

*<api\_key>your-virustotal-api-key-here</api\_key>*

*<group>syscheck</group>*

*<alert\_format>json</alert\_format>*

*<hook\_url>https://www.virustotal.com/vtapi/v2/file/report</hook\_url>*

*<rules\_id>550,554,555,556,557</rules\_id>*

*</integration>*

api\_key: Replace your-virustotal-api-key-here with your actual VirusTotal API key.

group: This specifies which alerts to send to VirusTotal. Common groups are syscheck for file integrity monitoring alerts.

alert\_format: Typically set to json for this type of integration.

hook\_url: The API endpoint for checking file reports on VirusTotal.

rules\_id: Specifies which rules should trigger the VirusTotal integration. Adjust these based on your specific needs.

USB alert:

You can also receive an alert when plugging in a usb into the device, in order to track that you need to configure the server local rules file as follows:

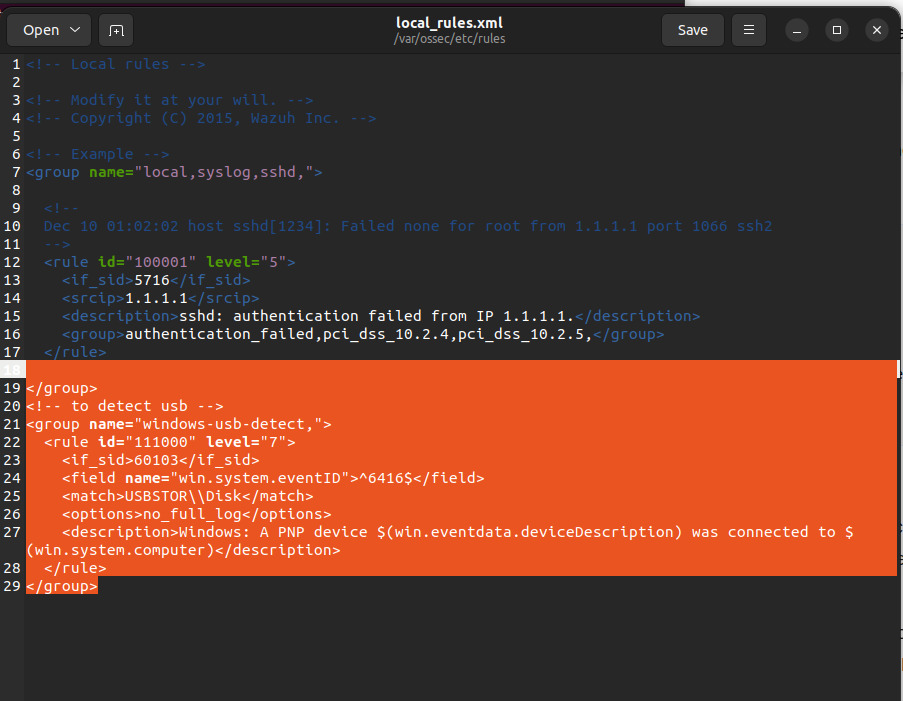


Figure 10 - Wazuh Server USB Detection Configuration

On the Agent machine, Windows10 here for example, you also need to configure the PnP policy so log can be sent to the SIEM:

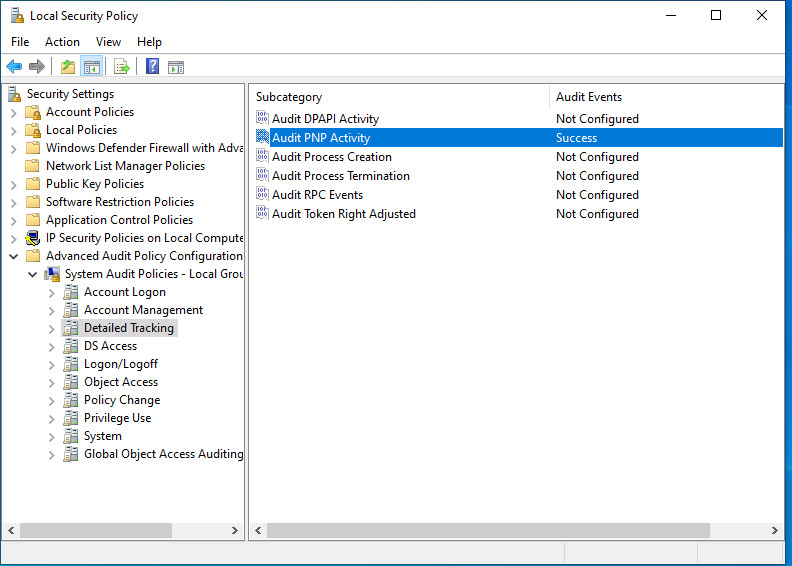


Figure 11 - USB PnP Policy Agent

After these steps the USB plug in alert will show as follows:

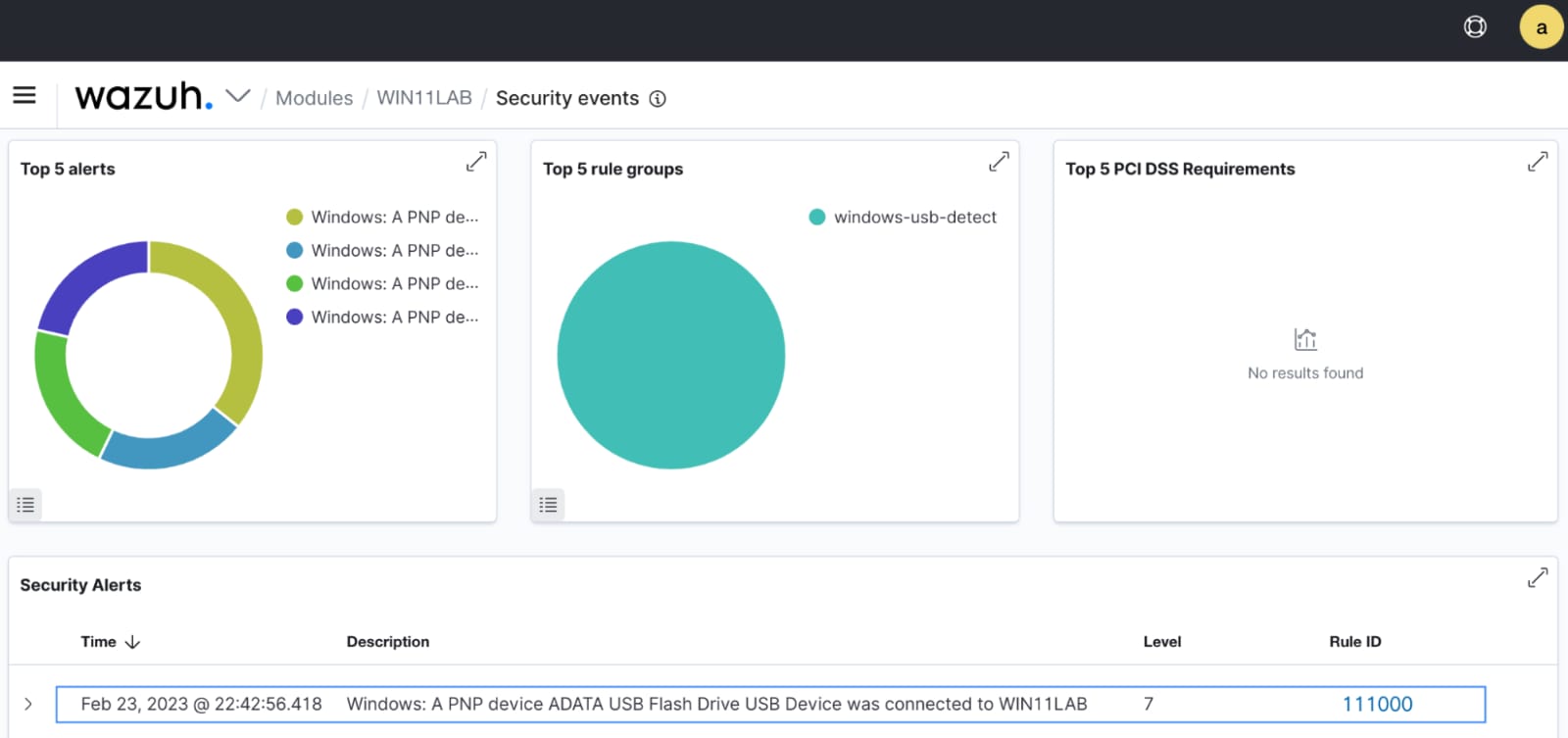


Figure 12 - USB plug alert

# Attack Scenarios

## Overview of the attacks planned and executed

Simulated Failed Login:

To test the Wazuh agent's ability to detect and log unauthorized access attempts using PowerShell to simulate a failed login on a Windows 10 virtual machine.

Using PowerShell, a simulated failed login attempt was performed with the following script:

*$username = 'NonExistentUser'*

*$password = ConvertTo-SecureString 'WrongPassword!' -AsPlainText -Force*

*$credential = New-Object System.Management.Automation.PSCredential ($username, $password)*

*Start-Process -FilePath "cmd.exe" -Credential $credential*

This script attempts to launch cmd.exe using a PSCredential object with a non-existent user and

an incorrect password, which results in a login failure.

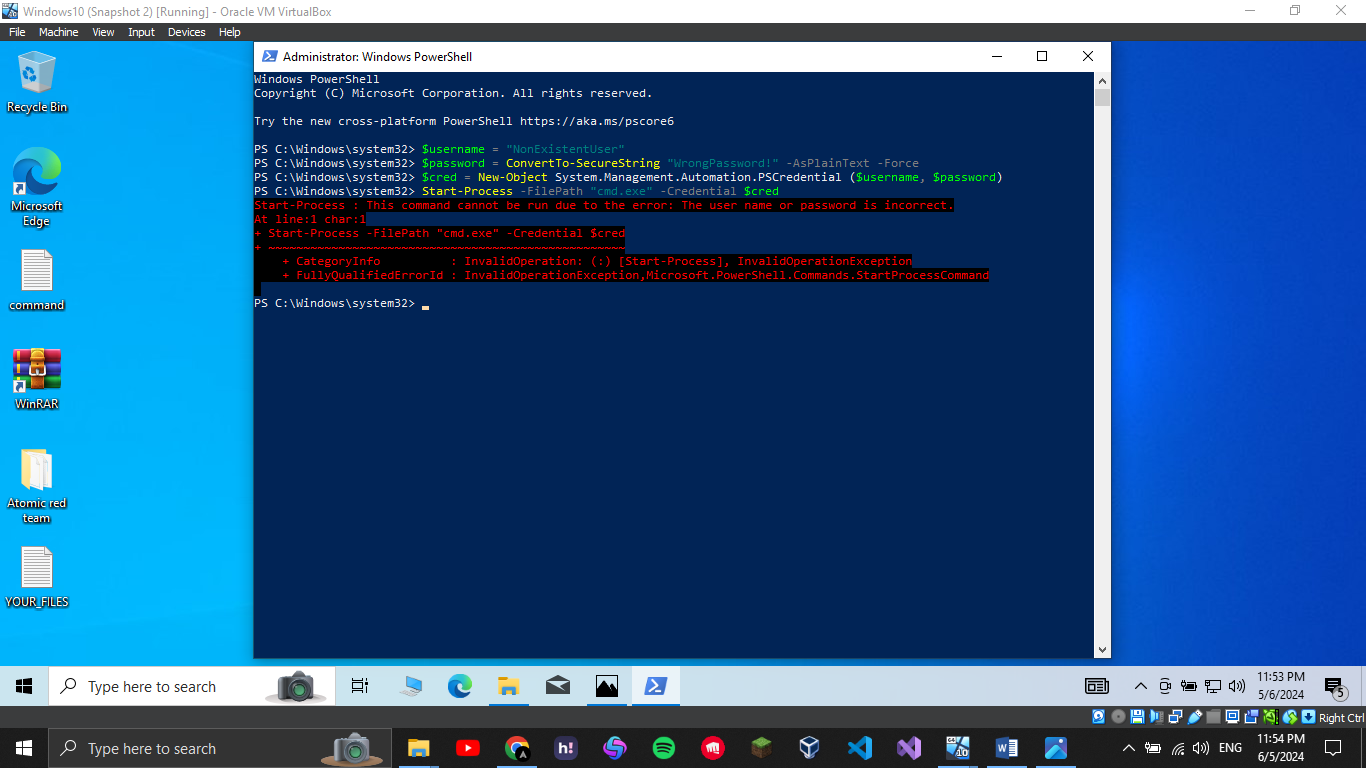


Figure 13 - Invalid Login Attempt

Event Description:

The Wazuh SIEM system successfully detected and logged a failed login attempt on a Windows 10 machine. As seen in the dashboard screenshot, the event is categorized under the alert level 5, indicating a medium severity incident.

Details of the Logged Event:

Event Severity: Level 5 (Medium Severity)

Alert ID: 60122

Event Type: Logon failure - Unknown user or bad password.

MITRE ATT&CK Tactics:

T1078: Valid Accounts - Indicates an attempt to use legitimate credentials, albeit invalid in this scenario.

T1531: Account Access - Reflects the unauthorized access attempt.

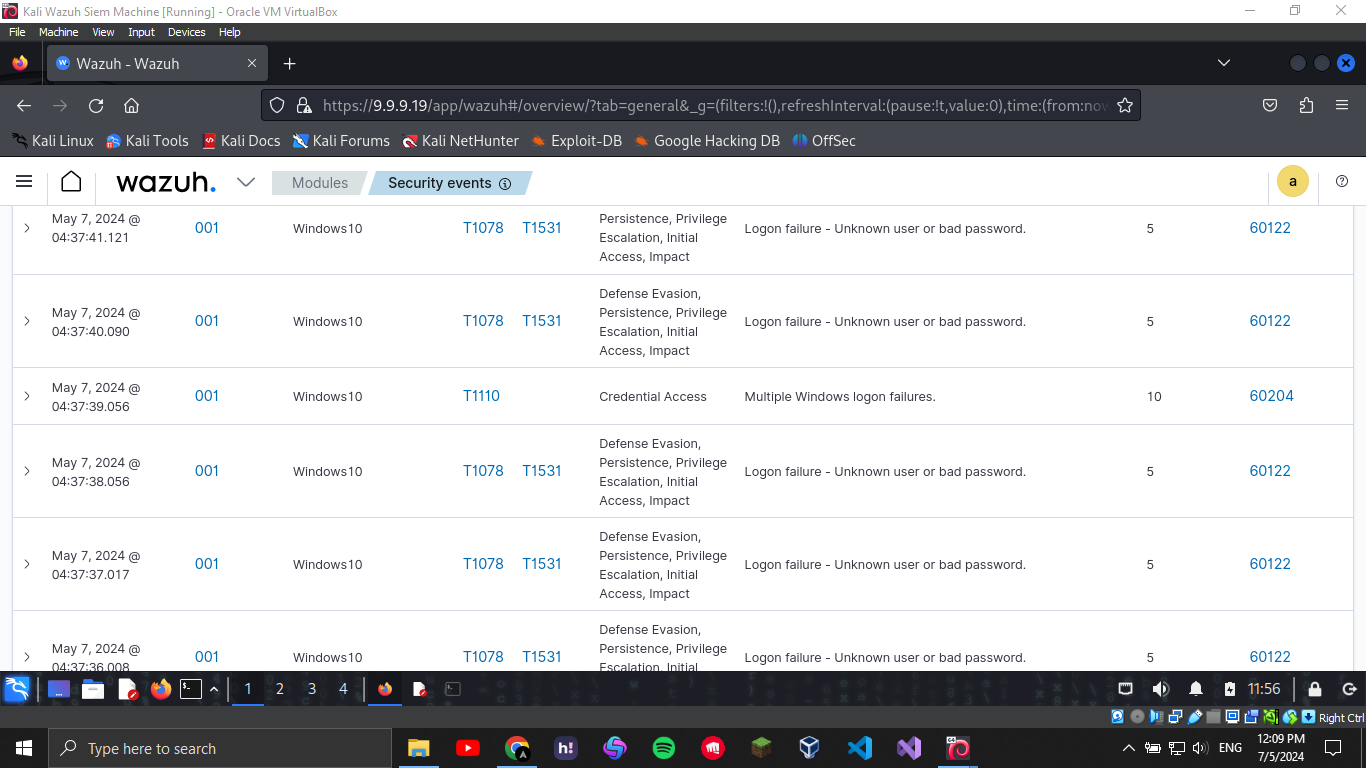


Figure 14 - Login Failure Alert

Significance of Alert Level:

Level 5 severity suggests that while the event is notable, it does not immediately indicate a critical threat to the system. This level is typically assigned to activities that should be monitored to detect patterns or repeated attempts that could escalate in threat potential.

Brute Force Login Attempt:

Script:

*$usernames = @("admin", "user", "test", "guest", "NonExistentUser")*

*$passwords = @("Password1!", "123456", "admin123", "guest!", "WrongPassword!")*

*foreach ($username in $usernames) {*

*foreach ($password in $passwords) {*

*$securePassword = ConvertTo-SecureString $password -AsPlainText -Force*

*$cred = New-Object System.Management.Automation.PSCredential ($username, $securePassword)*

*# Attempt to start a process with the given credentials*

*try {*

*Start-Process -FilePath "cmd.exe" -Credential $cred -NoNewWindow -ErrorAction Stop*

*} catch {*

*# Catch the exception if the login fails*

*Write-Host "Login failed for $username with password $password"*

*}*

*Start-Sleep -Seconds 1*

*}*

*}*

Event Description:

A brute force login attempt was simulated using a PowerShell script on a Windows 10 virtual

machine. This script attempted multiple logins using a combination of commonly used

usernames and passwords, aiming to mimic an attacker trying different credentials to gain

unauthorized access.

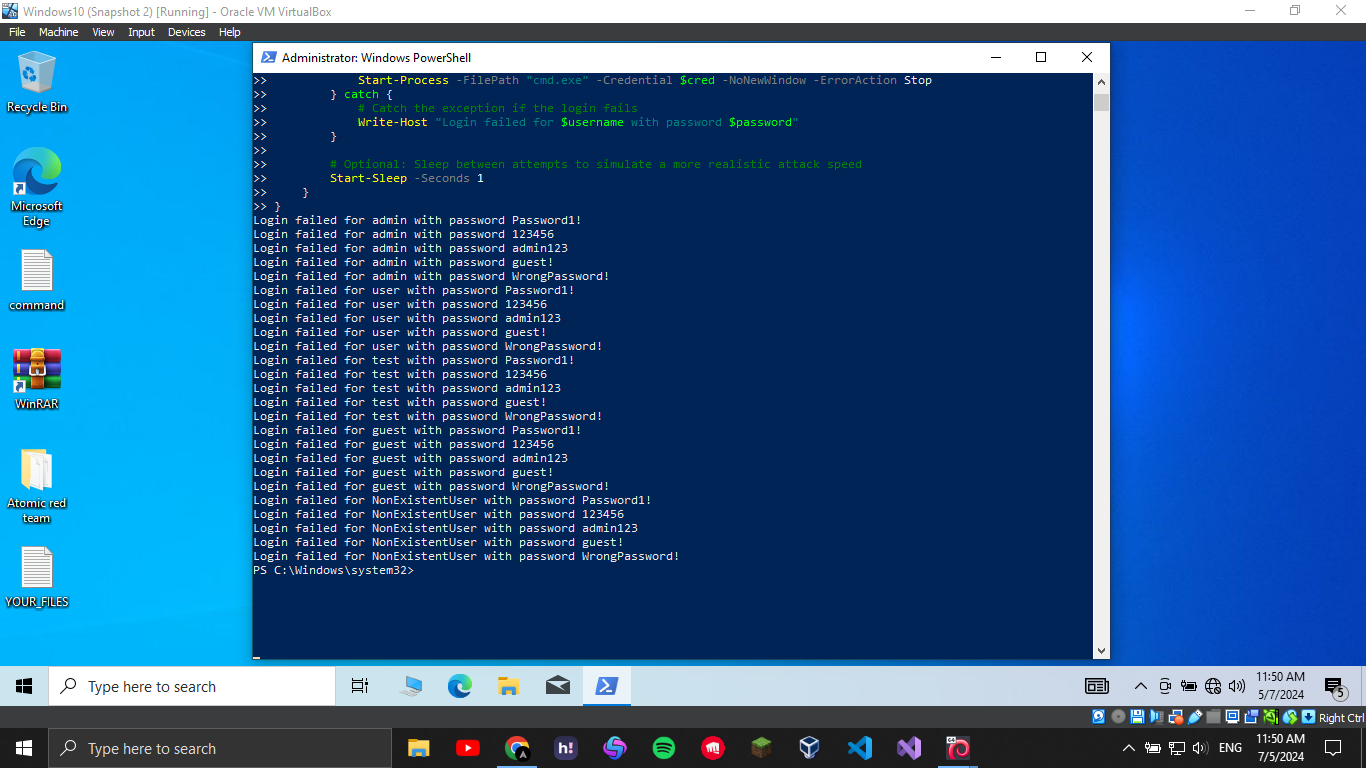


Figure 15 - Brute Force Execution

Wazuh Detection and Response:

Alert Severity: Level 10 (High Severity)

Alert ID: 60204

Event Type: Multiple Windows login failures.

MITRE ATT&CK Tactics:

T1110: Credential Access – This technique represents attempts to harvest credentials like

passwords or hash dumping.

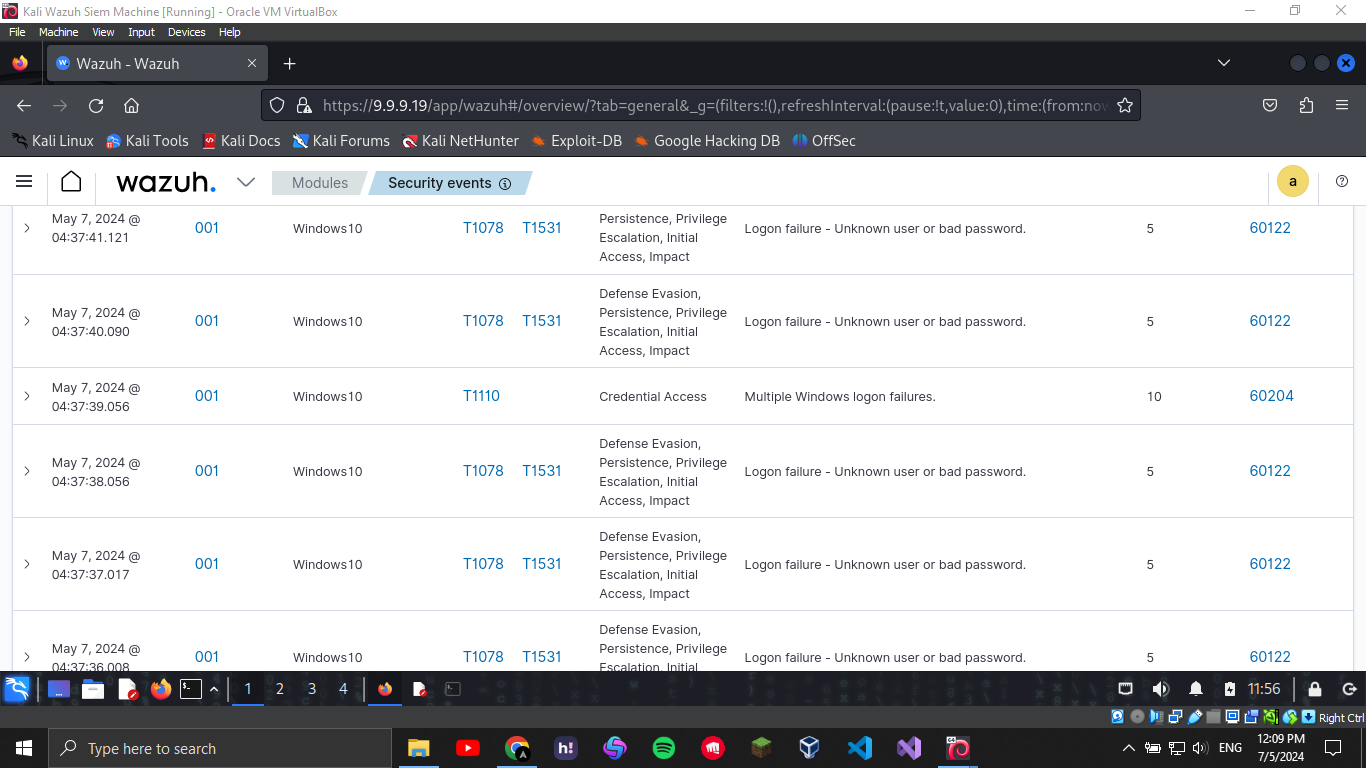


Figure 16 - Brute Force Alert

Significance of Alert Level:

A level 10 alert indicates a higher severity, reflecting the increased risk associated with multiple failed login attempts, which could suggest a more aggressive attack like brute forcing or credential stuffing. The elevated alert level emphasizes the potential for a serious security breach if such attempts were to succeed.

Red Atomic Tests:

the intention is to evaluate the detection capabilities of the Wazuh SIEM system under simulated attack scenarios that mimic sophisticated cyber threats.

Red Atomic tests are a specialized form of security assessment where simulated cyber attacks—often referred to as "red team" activities—are conducted to test the effectiveness of a Security Solution. These tests mimic real-world attacks to see how well a system can detect and respond to threats.

First we install the required files for Red Atomic tests to run on our system:

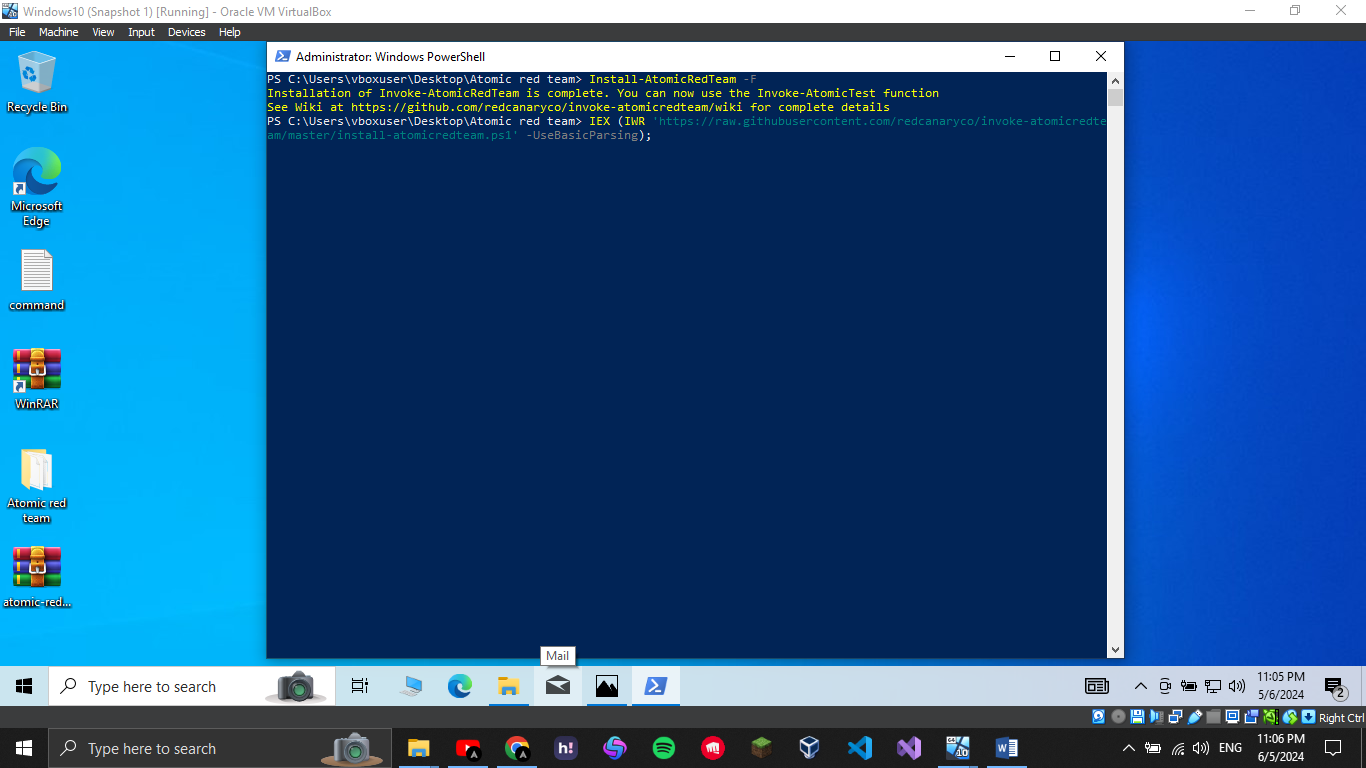


Figure 17 - Installing the Red Atomic Tests

After installation prerequisites are checked to any test we would like to invoke on the machine, after prerequisites are all met we conduct the desired tests:

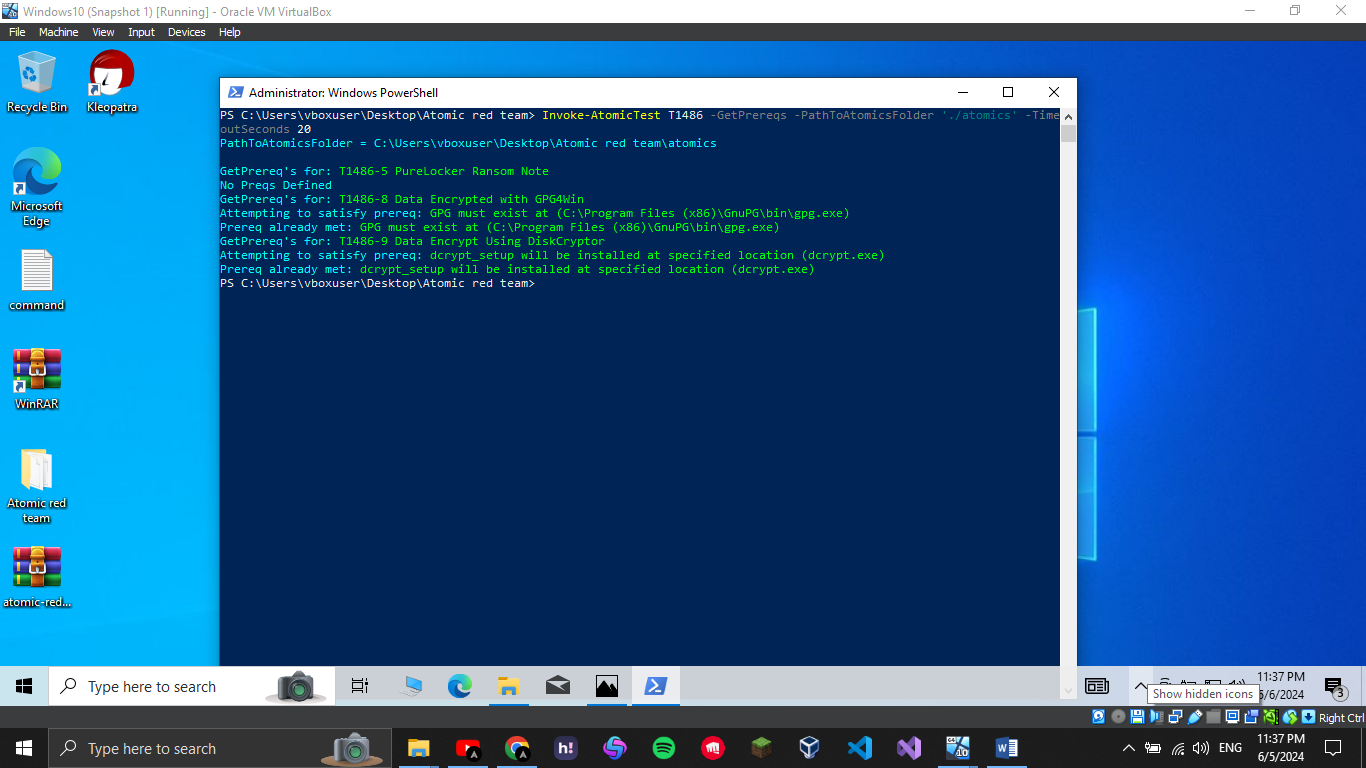


Figure 18 - Checking Atomic Tests Prerequisites

Unfortunately, after conducting the test the wazuh siem did not pick any log and nothing was shown.

Malware file test:

Using the EICAR test file:

The EICAR test file, short for the European Institute for Computer Anti-Virus Research, is a standard tool used to assess the operational capabilities of antivirus software. The file is a harmless string of code that is recognized by all antivirus programs and is used specifically to trigger a response as if it were a real virus. This allows users, administrators, and antivirus vendors to verify that their software is correctly installed and actively scanning for threats without the risk of introducing a harmful virus.

The EICAR test file is widely used because it is a safe and predictable way to ensure that anti-malware software is functioning properly on a system. When the file is scanned or executed, the antivirus program should detect it and respond as it would to a malicious file, typically by alerting the user, quarantining the file, or deleting it. This test helps confirm the software’s detection and response capabilities without any risk of damage to the host system or network.

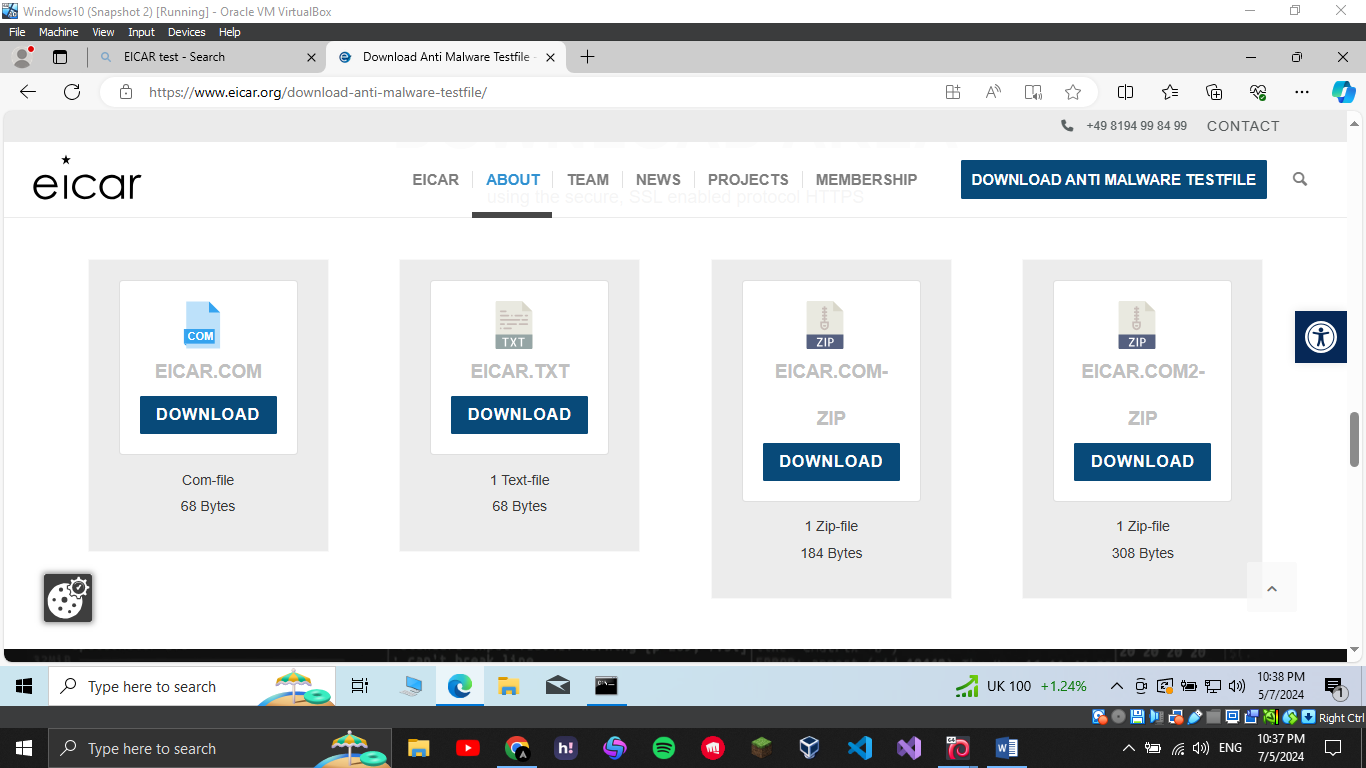


Figure 19 - EICAR test file download

The test file is downloaded to a monitored directory, which was successfully shown on the dashboard, however downloading the file did not show any alert on the siem security event logs.

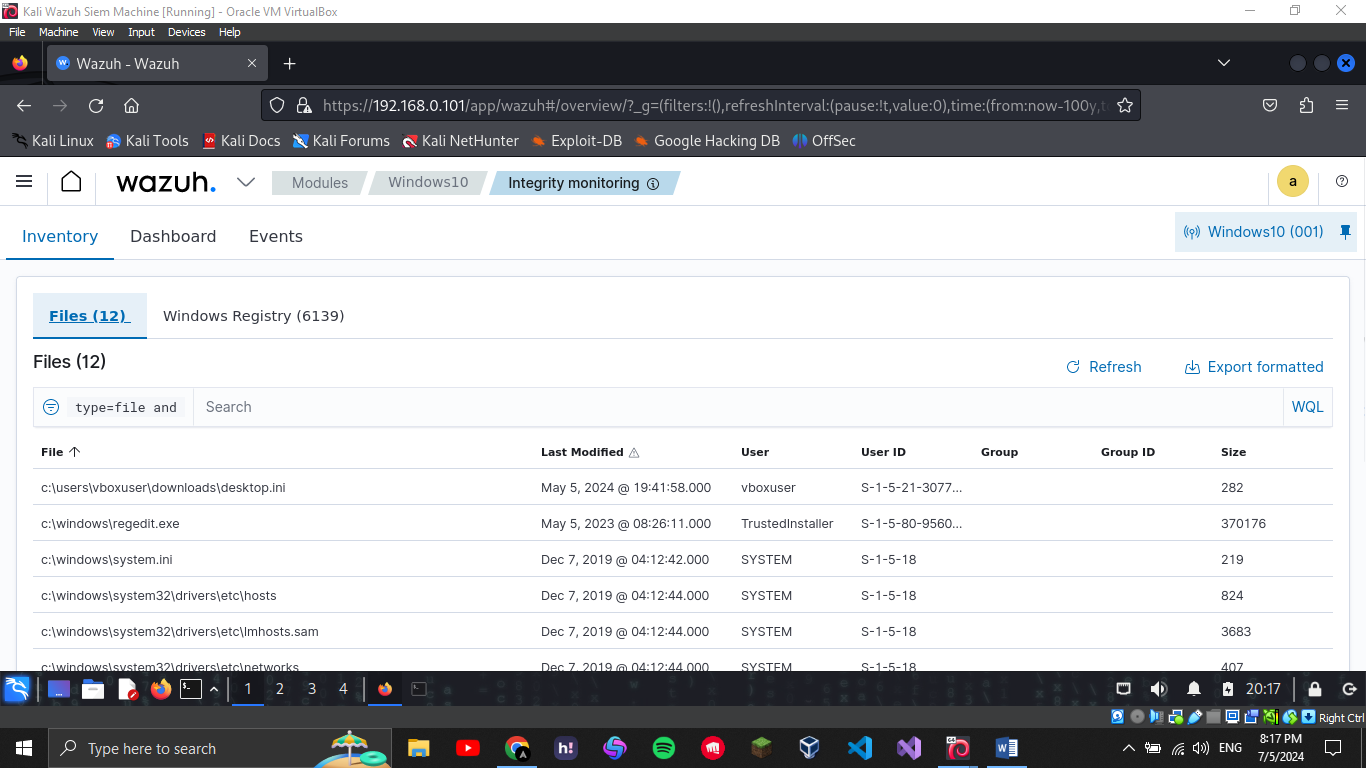


Figure 20 - The Monitored Directory

# Conclusion

This project set out to deploy and critically evaluate the Wazuh SIEM system within a simulated network environment, aiming to assess its effectiveness in real-time monitoring, threat detection, and response. Through the careful configuration of the system and the execution of various simulated attack scenarios, including the use of Red Atomic tests and EICAR test files, we have gained significant insights into the capabilities and areas for improvement of the Wazuh system.